



SPIP

Système de Publication pour l'Internet

VERSION 1.4.2

uZine
3

Avant-propos

SPIP¹ est le système de publication développé par le minirézo pour la gestion du site uZine². Nous le livrons à chacun, sous licence libre (GPL). Vous pouvez donc l'utiliser *librement* pour votre propre site, qu'il soit personnel, associatif ou marchand.

Liens utiles ...

- ▷ Le présent site constitue la *documentation officielle de SPIP* : nous tentons d'y présenter une documentation aussi complète que possible.
- ▷ *L'aide en ligne*, accessible directement ici (section 12 page 229), identique à celle qui est installée dans SPIP, pour une explication de ses différentes fonctionnalités.
- ▷ *L'espace de téléchargement*³ contient la dernière version de SPIP. Vous y trouverez également les versions en cours de développement, des documents de travail (liste des fonctions à installer, des bugs à corriger ...), un système de gestion des bugs, ainsi que *différents squelettes* fournis à titre d'exemple.
- ▷ Les listes de diffusion (section 1.5 page 12) ouvertes aux utilisateurs de SPIP, où vous obtiendrez l'aide nécessaire et le support de la communauté des utilisateurs.

Compilation de la documentation de SPIP.
version 20021122

Philippe Charlier

¹Version actuelle : SPIP 1.4.2

²<http://www.uzine.net>

³<http://rezo.net/spip-dev/>

Table des matières

Avant-propos	i
I Présentation de SPIP et premiers pas	1
1 Présentation	3
1.1 c'est quoi SPIP ?	5
1.1.1 SPIP, un système de publication	5
1.1.2 Pourquoi SPIP ?	5
1.2 Caractéristiques complètes	7
1.2.1 Installation	7
1.2.2 Site public	7
1.2.3 La structure du site	7
1.2.4 Les articles	8
1.2.5 Les brèves	8
1.2.6 Les forums	8
1.2.7 Les pétitions	9
1.2.8 Les statistiques	9
1.2.9 Les rédacteurs/administrateurs	9
1.2.10 Syndication	9
1.2.11 Interface graphique du site public	9
1.2.12 Interface du site privé	10
1.2.13 Paramétrage du site	10
1.2.14 Suivi éditorial	10
1.2.15 Sauvegarde et exportation de la base de données	10
1.3 Conditions d'utilisation	11
1.3.1 Explication de la licence	11
1.3.2 Remarques pratiques	11
1.4 Des sites sous SPIP	12
1.5 Obtenir de l'aide	12

2	Installation	14
2.1	Hébergeurs acceptant SPIP	16
2.1.1	Ca marche chez ...	16
2.1.2	Ca marche plus ou moins chez ...	17
2.1.3	Ca ne marche pas ...	17
2.1.4	Questions / réponses	17
2.2	Installer SPIP	18
2.2.1	Configuration nécessaire	18
2.2.2	Installation automatique (expérimentale)	19
2.3	Tester SPIP chez Free	21
2.3.1	Ouvrir un compte chez Free	21
2.3.2	Activer la base de données	22
2.3.3	Installer SPIP	22
2.4	Tester SPIP chez Multimania	23
2.4.1	Ouvrir un compte chez MultiNian-Nian	23
2.4.2	Activer la base de données	23
2.4.3	Installer SPIP	23
2.4.4	Hop, ça roule ...	24
3	Evolution et mise à jour	25
3.1	Effectuer une mise à jour	27
3.1.1	Sauvegardez votre base de données	27
3.1.2	Installez la nouvelle version de SPIP	28
3.1.3	Déclenchez la mise-à-jour de votre site	28
3.1.4	Videz le cache de votre site public	29
3.1.5	En cas de pépin!	29
3.2	L'histoire minuscule et anecdotique de SPIP	30
3.3	SPIP 1.0.3, 1.0.4, 1.0.5	33
3.4	SPIP 1.2, 1.2.1	34
3.5	SPIP 1.3	35
3.6	SPIP 1.4	38
3.6.1	Mise-à-jour	38
3.6.2	Problèmes éventuels	38
3.6.3	Interface	39
3.6.4	Documents joints et documents multimédia	40
3.6.5	Mots-clés	40

3.6.6	Forums	40
3.6.7	Moteur de recherche	41
3.6.8	Multi-base	41
3.6.9	Sites syndiqués	41
3.6.10	Squelettes	41
3.6.11	Raccourcis	42
3.6.12	Sécurité	42
3.6.13	Statistiques	42
3.6.14	Divers	42
II Guide de l'utilisateur		44
4	Guide de l'utilisateur	46
4.1	FAQ de l'administrateur SPIP	48
4.1.1	Débuter	48
4.1.2	Supprimer	48
4.1.3	Les différents types de navigation	48
4.2	L'aide en ligne	49
4.3	Vous avez ...dist ?	50
4.4	Quels sont les éléments gérés par SPIP ?	51
4.4.1	La hiérarchie des rubriques	51
4.4.2	Les articles	52
4.4.3	Les brèves	53
4.4.4	Les forums de discussion	53
4.4.5	Les auteurs	53
4.4.6	Les sites syndiqués	53
4.4.7	Les pétitions	54
4.4.8	Les mots-clés	54
4.5	Un memento des raccourcis SPIP	54
4.6	Une macro Word > SPIP	54
III Questions et réponses		55
5	FAQ : questions techniques	57
5.1	Après modification de mes <i>squelettes</i> avec Dreamweaver (Macromedia), j'obtiens des messages d'erreur quand je les installe sur le serveur.	59

5.2	Comment fonctionne le moteur de recherche de SPIP ?	60
5.2.1	L'indexation	60
5.2.2	La recherche	64
5.3	Dès ma première connexion à l'espace privé de SPIP, rien ne fonctionne et on m'indique une erreur à la ligne 1 de certains fichiers.	64
5.4	Est-ce qu'il faut réinstaller une sauvegarde de la base de données après avoir fait une mise à jour de SPIP ?	64
5.5	J'ai des messages d'erreur un peu partout sur les pages SPIP, et rien ne marche.	65
5.6	J'obtiens des messages du genre «Maximum execution time exceeded»	65
5.7	Je suis chez Free ou chez Online, et j'obtiens souvent des messages «Maximum execution time exceeded»	66
5.8	Je suis sous Apache/Windows, et après l'installation j'ai droit à une «Internal Server Error» quand je me connecte à l'espace privé	66
5.9	Je suis sous Apache/Windows, l'installation et l'espace privé marchent bien, mais le site public fait régulièrement planter Apache (avec un message d'erreur Windows typiquement incompréhensible)	66
5.10	Je viens d'effectuer une mise à jour de SPIP, et le site public fonctionne mal	67
5.11	Le serveur qui m'héberge tourne sous PHP4, et ne comprend pas les fichiers «.php3»	68
5.12	Pourquoi la création automatique de vignettes ne fonctionne pas avec tous les formats d'images (JPEG, GIF, PNG) ?	68
5.13	SPIP ne fonctionne pas chez mon hébergeur habituel. Y a-t-il un «patch» quelque part ?	69
6	FAQ : premiers pas	70
6.1	C'est quoi cette histoire d'espace privé ?	72
6.2	Comment installer SPIP ?	72
6.3	Comment réinstaller SPIP si je n'arrive plus à accéder à l'espace privé ?	72
6.4	De quoi a besoin SPIP pour fonctionner ?	72
6.5	Est-ce que je peux utiliser SPIP pour réaliser un site marchand, un site avec des pubs, ou un site de droite, ou un site qui débîne le logiciel libre ...	73
6.6	Est-ce que je peux vendre un service utilisant SPIP ? Est-ce que je peux monter une activité professionnelle autour de SPIP ?	73
6.7	J'ai créé une rubrique, mais elle n'apparaît pas dans le site public !	73
6.8	J'ai perdu mon login et/ou mon mot de passe ...	73
6.9	Je voudrais une nouvelle fonctionnalité ...	74
6.10	Mon problème n'est pas mentionné dans cette FAQ. Que faire ?	74
6.11	Mon site est vide, que dois-je faire ?	74
6.12	Mon site sous SPIP est planté, est-ce que je vous fais un procès ?	74

IV	Squelettes : Guide du webmestre	75
7	Squelettes : Guide du webmestre	77
7.1	FAQ webmestre	79
7.1.1	Les bases	79
7.1.2	Créer ses squelettes	79
7.1.3	Partager	80
7.2	Contribuer au développement de SPIP	80
7.2.1	Règles de présentation et d'écriture	80
7.2.2	Règles de programmation	81
7.2.3	Partager vos modifications	82
7.3	La structure de la base de données	82
7.3.1	Contenu rédactionnel	83
7.3.2	Éléments interactifs	86
7.3.3	Les relations entre objets	88
7.3.4	Gestion du site	88
7.3.5	Indexation (moteur de recherche)	88
7.4	Le moteur de recherche	88
7.4.1	Principe	89
7.4.2	L'indexation	89
7.4.3	La recherche	89
7.4.4	Performances	90
7.5	Rapidité du site public	91
7.5.1	Optimiser un site	91
7.5.2	L'influence du cache	91
7.5.3	Sur une machine dédiée	91
7.6	Spip et les feuilles de style	92
7.6.1	Où se trouve la définition de ces feuilles de style ?	92
7.6.2	Les liens hypertextes	92
7.6.3	Les intertitres	93
7.6.4	Code et cadre	93
7.6.5	Les notes de bas de page	94
7.6.6	Les tableaux	94
7.6.7	Ligne de séparation horizontale	95
7.6.8	Gras et italique	95
7.6.9	Les paragraphes	95
7.6.10	Les formulaires	95
7.6.11	Conclusion	96

8	Manuel de référence	97
8.1	Principe général	99
8.1.1	Pour chaque type de document, un couple de fichiers	99
8.1.2	Le principe de fonctionnement du cache	99
8.1.3	Le fichier .PHP3	100
8.1.4	Le fichier .HTML	101
8.1.5	Une interface différente dans le même site	102
8.1.6	Que peut-on mettre dans un fichier .HTML	102
8.2	Des boucles et des balises	103
8.2.1	Des boucles	103
8.2.2	Les balises SPIP	104
8.2.3	Les balises à l'intérieur des boucles	104
8.3	La syntaxe des boucles	105
8.3.1	Syntaxe de base	105
8.3.2	Syntaxe complète	106
8.3.3	Des critères d'environnement en cascade	108
8.3.4	Boucles incluses et boucles successives	109
8.4	La syntaxe des balises SPIP	109
8.4.1	Fonctionnement simplifié	110
8.4.2	Codes optionnels	110
8.4.3	Filtrer les résultats	111
8.4.4	Court-circuiter le traitement par SPIP	112
8.5	La boucle ARTICLES	112
8.5.1	Les critères de sélection	112
8.5.2	Les critères d'affichage	113
8.5.3	Les balises de cette boucle	113
8.6	La boucle RUBRIQUES	114
8.6.1	Les critères de sélection	114
8.6.2	Les critères d'affichage	115
8.6.3	Les balises de cette boucle	115
8.7	La boucle BREVES	116
8.7.1	Les critères de sélection	116
8.7.2	Les critères d'affichage	117
8.7.3	Les balises de cette boucle	117
8.8	La boucle AUTEURS	117

TABLE DES MATIÈRES

8.8.1	Les critères de sélection	118
8.8.2	Les critères d'affichage	118
8.8.3	Les balises de cette boucle	118
8.9	La boucle FORUMS	118
8.9.1	Les critères de sélection	119
8.9.2	Les critères d'affichage	119
8.9.3	Les balises de cette boucle	119
8.10	La boucle MOTS	120
8.10.1	Les critères de sélection	120
8.10.2	Les critères d'affichage	120
8.10.3	Les balises de cette boucle	120
8.11	La boucle DOCUMENTS	121
8.11.1	Les critères de sélection	121
8.11.2	Les critères d'affichage	121
8.11.3	Les balises	122
8.12	La boucle SITES (ou SYNDICATION)	122
8.12.1	Les critères de sélection	123
8.12.2	Les critères d'affichage	123
8.12.3	Les balises de cette boucle	123
8.13	La boucle SYNDIC_ARTICLES	124
8.13.1	Les critères de sélection	124
8.13.2	Les critères d'affichage	124
8.13.3	Les balises de cette boucle	124
8.14	La boucle SIGNATURES	125
8.14.1	Les critères de sélection	125
8.14.2	Les critères d'affichage	125
8.14.3	Les balises de cette boucle	125
8.15	La boucle HIERARCHIE	125
8.15.1	Les critères de sélection	126
8.15.2	Les critères d'affichage	126
8.15.3	Les balises de cette boucle	126
8.16	Les critères communs à toutes les boucles	126
8.16.1	Classer les résultats	126
8.16.2	Comparaisons, égalités	127
8.16.3	Affichage d'une partie des résultats	128

8.16.4	Affichage <i>entre</i> les résultats	129
8.17	Les formulaires	129
8.17.1	Feuilles de style	131
8.18	Les boucles de recherche	131
8.18.1	L'interface de recherche	132
8.18.2	Le squelette des résultats	132
8.19	Les filtres de SPIP	132
8.19.1	Les filtres de mise en page	132
8.19.2	Les filtres des dates	133
8.19.3	Filtres de logos	133
8.19.4	Filtres de texte	133
8.19.5	Filtres techniques	134
8.19.6	Ajouter ses propres fonctions	134
8.20	Les boucles récursives	134
8.21	<INCLUDE> d'autres squelettes	135
8.22	Les variables de personnalisation	136
8.22.1	Où indiquer ces variables ?	136
8.22.2	Les variables du texte	137
8.22.3	Les variables pour les forums publics	138
8.22.4	Exemples	138
8.23	Utiliser des URLs personnalisées	139
8.23.1	Choisir le type d'URLs apparentes	139
8.23.2	Programmer la traduction des adresses apparentes en adresses réelles	139
8.23.3	Générer les URLs apparentes dans les pages SPIP	140
8.24	La «popularité» des articles	140
8.24.1	Comment décompter des visites	141
8.24.2	Balises	141
8.24.3	Critères	141
9	SPIP pas à pas	143
9.1	Mon premier squelette	145
9.2	Un squelette, plusieurs articles	146
9.3	Une rubrique	146
9.4	Boucles en boucles	147
9.5	Gérer le cache	148
9.6	Des filtres	149

10 Trucs et astuces	151
10.1 Présenter les résultats d'une recherche par secteurs	153
10.2 Afficher des éléments par lignes dans un tableau	153
10.3 Afficher le nombre de messages du forum lié à un article	154
10.4 Afficher les derniers articles de vos rédacteurs par rubrique	155
10.4.1 Création du fichier myauteur.php3	155
10.4.2 Création du fichier myauteur.html	155
10.4.3 Et enfin	156
10.5 Liens «Article précédent», «Article suivant»	157
10.6 Ne pas afficher les articles publiés depuis plus d'un an	159
10.7 Réaliser un agenda avec SPIP	159
10.7.1 De quelles informations avons-nous besoin ?	160
10.7.2 La date de publication antérieure ou la vraie date de publication ?	160
10.7.3 Organiser les événements	161
10.7.4 L'affichage de l'agenda, PHP à la rescousse	162
10.7.5 Un peu de personnalisation visuelle	166
10.7.6 Le code complet du squelette	166
10.8 Remplir les meta-tags HTML des pages d'article	167
10.9 Un menu déroulant pour présenter une liste d'articles	167
11 Tutorial : utilisation avancée	169
11.1 Introduction	171
11.1.1 À qui s'adresse ce document ?	171
11.1.2 Quelle version de SPIP ?	171
11.1.3 Quelles autres connaissances techniques sont nécessaires ?	171
11.1.4 Les impératifs	172
11.1.5 Comment utiliser ce tutorial ?	172
11.1.6 La progression de ce tutorial	172
11.2 Le but du jeu : un site consacré aux jeux vidéo	173
11.2.1 Pourquoi les jeux vidéo ?	173
11.2.2 Quelles sont difficultés que présente un tel site ?	173
11.3 La structure du site	174
11.3.1 Les possibilités	175
11.3.2 Les impératifs	175
11.3.3 La structure retenue	176
11.3.4 Que deviennent les machines et les genres d'articles ?	176

11.4	Mise en place de la structure	177
11.4.1	La structure des rubriques	177
11.4.2	Les machines	178
11.4.3	Les types d'articles	179
11.5	Écrire des articles	179
11.5.1	Une rubrique par jeu	179
11.5.2	Le test complet du jeu !	180
11.5.3	Pour quelle(s) machine(s) ? Quel type d'article ?	181
11.6	Première version du squelette des articles	182
11.6.1	La page de base	182
11.6.2	Récupérer les infos sur le jeu	183
11.7	La page des rubriques	184
11.7.1	Les articles contenus dans cette rubrique	185
11.7.2	Les sous-rubriques	185
11.7.3	Les rubriques sans articles	186
11.7.4	Les rubriques de jeu (avec des articles)	187
11.8	Les mots-clés dans les articles	187
11.8.1	Les machines	188
11.8.2	Dans la même rubrique, au sujet du même jeu	188
11.8.3	Les jeux dans la même catégorie	190
11.9	Les mots-clés dans les rubriques	191
11.10	L'interface des news	192
11.10.1	Toutes les news d'un jeu	193
11.11	Et encore d'autres moyens de se compliquer la vie !	194
11.11.1	L'attribution d'une note à un jeu	194
11.11.2	Intégrer la note dans les squelettes	195
11.11.3	La date de sortie officielle des jeux	196
11.11.4	Afficher la date dans les articles	197
11.12	Quelques sommaires alternatifs	199
11.12.1	Le tableau des meilleures notes	199
11.12.2	Le calendrier des sorties	201
11.13	Une première version du sommaire	205
11.14	Un sommaire pour chaque machine	208
11.14.1	Le fichier «sommaire_machine.php3»	209
11.14.2	Le squelette du sommaire par machine	209

11.15	Référencer des articles sur le Web	213
11.16	Un forum pour chaque jeu	215
11.16.1	La page forum_jeu.php3	216
11.16.2	Le squelette forum_jeu.html	217
11.16.3	On recommande ...	219
11.16.4	Afficher chaque message : message.php/message.html	220
11.16.5	Les liens depuis les autres pages	225
11.17	Le site complet	226
V	Aide en ligne	227
12	Aide en ligne	229
12.1	Installation de SPIP	231
12.1.1	Régler les droits d'accès	231
12.1.2	Votre connexion MySQL	231
12.1.3	Choix de votre base	232
12.1.4	Informations personnelles	232
12.1.5	Vérification par FTP	232
12.1.6	Un problème de squelette ?	233
12.2	Les articles	233
12.2.1	Les raccourcis typographiques de SPIP	233
12.2.2	Titre, surtitre, sous-titre	238
12.2.3	Choisir la rubrique	238
12.2.4	Descriptif rapide	239
12.2.5	Chapeau	239
12.2.6	Redirection d'article	239
12.2.7	Texte	239
12.2.8	Date	239
12.2.9	Date de publication antérieure	240
12.2.10	Les auteurs	240
12.2.11	Logo de l'article	240
12.2.12	Le statut de l'article	242
12.2.13	Proposer son article	243
12.2.14	Articles en cours de modification	244
12.3	Les rubriques	244
12.3.1	Une structure hiérarchisée	244

12.3.2 Choisir la rubrique	245
12.3.3 Logo de la rubrique	245
12.4 Les brèves	246
12.4.1 Les brèves	246
12.4.2 Choisir la rubrique	246
12.4.3 Le lien hypertexte	246
12.4.4 Le statut de la brève	246
12.4.5 Le logo de la brève	247
12.5 Images et documents	247
12.5.1 Insérer des images	247
12.5.2 Joindre des documents	249
12.5.3 Installer des fichiers par FTP	252
12.6 Les mots-clés	253
12.6.1 Principe des mots-clés	253
12.6.2 Les mots-clés	253
12.6.3 Les groupes de mots	253
12.7 Les sites référencés	254
12.7.1 Référencer un site	254
12.7.2 Sites syndiqués	254
12.7.3 Articles syndiqués	256
12.7.4 Utiliser un proxy	256
12.8 La messagerie interne	256
12.8.1 Les messages entre utilisateurs	256
12.8.2 Les pense-bête	257
12.8.3 Le calendrier	258
12.8.4 Configuration personnelle de la messagerie	258
12.9 Suivi des forums	258
12.9.1 Suivi des forums	258
12.10 Configuration précise	259
12.10.1 Nom et adresse de votre site	259
12.10.2 Contenu des articles	259
12.10.3 Articles post-datés	260
12.10.4 Fonctionnement des forums	260
12.10.5 Système de brèves	261
12.10.6 Messagerie interne	261

TABLE DES MATIÈRES

12.10.7 Statistiques des visites	262
12.10.8 Envoi automatique de mails	263
12.10.9 Moteur de recherche intégré	264
12.11 Configuration de l'interface personnelle	264
12.11.1 Interface simplifiée / complète	264
12.11.2 Le cookie de correspondance	264
12.11.3 Se déconnecter	265
Historique de la documentation papier	266

Première partie

Présentation de SPIP et premiers pas



Présentation

et premiers pas

Chapitre 1

Présentation



Présentation

et premiers pas

1.1 c'est quoi SPIP ?

1er juillet 2001 par l'équipe de SPIP

Un système de publication pour l'Internet

1.1.1 SPIP, un système de publication

SPIP est un Système de Publication pour l'Internet. Kesako ? Il s'agit d'un ensemble de fichiers, installés sur votre compte Web, qui vous permettent de bénéficier d'un certain nombre d'automatismes : gérer un site à plusieurs, mettre en page vos articles sans avoir à taper de HTML, modifier très facilement la structure de votre site ... Avec le même logiciel qui sert à visiter un site (Netscape, Microsoft Explorer, Mozilla, Opera ...), SPIP permet de fabriquer et de tenir un site à jour, grâce à une interface très simple d'utilisation.



D'autres systèmes de publication existent ; chacun présente ses spécificités. Un des plus connus actuellement est phpNuke¹ ; il impose une structure assez rigide pour le site, sous forme de portail muni de courts articles. SPIP est plus souple, et orienté vers la création d'un site structuré comme un magazine : c'est-à-dire avec des rubriques, sous-rubriques (et ainsi de suite), dans lesquelles sont insérés des articles et des brèves qui peuvent être complétés de forums de discussions.

1.1.2 Pourquoi SPIP ?

SPIP est gratuit, open-source, distribué sous Licence Publique Générale GNU (*GNU General Public License ou GPL*) (section 1.3 page 11). Les besoins logiciels et matériels de SPIP sont raisonnables et se trouvent même chez certains hébergeurs gratuits (voir la Foire Aux Questions (partie III page 56) et le manuel d'installation (section 2.2 page 18) pour plus de détails - ou, pour résumer à l'extrême : PHP+MySQL).

L'intérêt de SPIP est de ...

- ▷ gérer un site Web de type magazine, c'est-à-dire composé principalement d'articles et de brèves insérés dans une arborescence de rubriques imbriquées les unes dans les autres. Voir la liste complète des caractéristiques de SPIP pour plus de détails (section 1.2 page 7).

¹<http://www.phpnuke.org/>

- ▷ séparer entièrement, et distribuer entre différentes personnes, trois types de tâches : la composition graphique, la contribution rédactionnelle via proposition d'articles et de brèves, et la gestion éditoriale du site (tâche qui comprend l'organisation des rubriques, la validation des articles proposés ...).
- ▷ dispenser le webmestre et tous les participants à la vie du site d'un certain nombre d'aspects fastidieux de la publication sur le Web, ainsi que de connaissances techniques trop longues à acquérir. L'installation de SPIP se réalise au moyen d'une interface simple et pas à pas, au terme de laquelle vous pouvez commencer à créer vos rubriques et articles.

SPIP a les caractéristiques suivantes ...

- ▷ *Pour le(s) rédacteur(s) et administrateur(s)*
 - ★ Une interface Web intuitive rend extrêmement simples la proposition d'articles et de brèves ainsi que la gestion éditoriale du site. De plus, des raccourcis typographiques permettent de mettre en forme un texte sans avoir à utiliser le langage HTML, rendant ainsi la contribution rédactionnelle accessible à tous, et aussi simple que l'écriture d'un e-mail.
- ▷ *Pour le webmestre*
 - ★ L'aspect graphique et la navigation sont définis par des squelettes HTML (ou «formats types») définissant chacun une «vue» (par exemple : une vue pour la page d'index, une autre montrant une rubrique et un résumé de son contenu, une troisième pour le détail d'un article, une quatrième pour le détail d'une brève). La façon dont est inséré le contenu rédactionnel du site dans ces pages est défini par un certain nombre de pseudo-tags HTML relativement faciles à maîtriser.
 - ★ SPIP ne restreint pas les possibilités graphiques et navigationnelles du site. Les squelettes HTML étant entièrement définis par le webmestre du site, il est possible de gérer certains éléments du site avec SPIP et le reste à la main ou même avec d'autres systèmes de publication (à condition que ces derniers soient aussi tolérants que SPIP, bien sûr).
- ▷ *Pour les visiteurs*
 - ★ Un système de cache sur la partie publique du site accélère le site en évitant un grand nombre de requêtes à la base de données, et joue en outre un rôle de garde-fou contre les plantages de la dite base (fréquents sur des serveurs «chargés») : dans ce cas, le site reste disponible de façon transparente, même si toute modification des contenus est impossible (y compris la contribution aux forums).
 - ★ Un moteur de recherche et d'indexation intégré à SPIP, s'il est activé par le webmestre, permet d'effectuer des recherches sur l'ensemble du contenu public du site.

Des inconvénients

Pour l'instant, la souplesse de SPIP implique qu'un peu d'efforts d'apprentissage sont nécessaires au webmestre pour modifier la présentation par défaut. Contrairement à des systèmes très contraints comme phpNuke où vous pouvez changer les couleurs et le logo en pressant simplement un bouton (mais c'est tout ce que vous avez le droit de faire), le webmestre sous SPIP doit apprendre les quelques rudiments d'un pseudo-HTML lui permettant ensuite de faire à peu près ce qu'il veut.

SPIP est livré avec un format d'interface de navigation complet ; dès que vous aurez commencé à créer le contenu de votre site, il pourra être immédiatement visité, et adoptera l'interface graphique fournie par défaut. Le webmestre du site peut bien entendu fabriquer sa propre interface graphique s'il le désire.

A l'avenir, il est prévu que plusieurs présentations soient fournies avec SPIP, permettant à la majorité des webmestres d'en réutiliser une qui leur convienne pour minimiser l'effort de personnalisation.

Exemples

L'exemple historique de l'utilisation de SPIP est le webzine *uZine*² (c'est le code, au départ spécifique, de ce site, qui a été repris pour réaliser le SPIP générique). Parmi d'autres sites fonctionnant sous SPIP, citons *Le Monde diplomatique*³ et *Vacarme*⁴.

Poursuivre ...

Pour aller plus loin, et vous lancer sous SPIP, il vous sera utile de lire la documentation, les exemples et les «premiers pas» proposés dans cette rubrique (section 9 page 143). Des listes de diffusion sont également à votre disposition pour échanger questions, trucs et astuces. A bientôt !

1.2 Caractéristiques complètes

1er juin 2001 par l'équipe de SPIP

1.2.1 Installation

L'installation de SPIP est particulièrement simplifiée par rapport à d'autres systèmes :

- ▷ Aucune connaissance technique particulière n'est nécessaire (ni PHP, ni MySQL) pour procéder à son installation.
- ▷ La configuration se fait directement en ligne, au travers d'une interface graphique très simple (il n'est pas nécessaire, en particulier, d'aller modifier un fichier de configuration avec des codes abscons).
- ▷ Nous distribuons une version unique de SPIP ; celle-ci peut évoluer au cours du temps en fonction des améliorations, mais nous faisons très attention à ne pas compliquer en développant des «patches» qu'il faudrait aller chercher à droite ou à gauche (pour adapter SPIP à tel hébergeur par exemple).

1.2.2 Site public

Le site public fabriqué à partir de SPIP offre les caractéristiques suivantes :

- ▷ Interface entièrement adaptable par le webmestre sans connaissances de PHP ni de MySQL ; l'interface de SPIP se programme en HTML, auquel nous avons ajouté un langage relativement simple ; SPIP n'impose donc pas une mise en page rigide (à la manière des «trois colonnes» si caractéristiques de phpNuke).
- ▷ SPIP intègre un système de cache pour chaque page individuelle : les pages sont calculées (à partir des informations de la base de données) individuellement, et stockées dans un fichier de cache ; ainsi le serveur n'est pas ralenti par un trop grand nombre d'appels MySQL, et le site est toujours accessible même en cas de panne du serveur de bases de données.
- ▷ SPIP intègre un petit moteur de recherche basé sur un système d'indexation par mots.

1.2.3 La structure du site

La structure d'un site sous SPIP est construite sur une hiérarchie de *rubriques*. Il n'y a virtuellement pas de limite au nombre de rubriques : une rubrique peut contenir autant de sous-rubriques que nécessaires, qui elles-mêmes contiennent des sous-rubriques, etc. On construit ainsi la structure de son site en imbriquant des rubriques et des sous-rubriques.

²<http://www.uzine.net/>

³<http://www.monde-diplomatique.fr/>

⁴<http://www.vacarme.eu.org>

1.2.4 Les articles

L'objet principal permettant de publier des informations sous SPIP est l'*article*. On peut placer autant d'articles que nécessaire, dans n'importe quelle rubrique.

- ▷ La rédaction des articles est très simple, elle se déroule via une interface graphique sur le Web.
- ▷ Un article est constitué de plusieurs éléments qui permettent de le structurer : titre, surtitre, soustitre, descriptif, chapeau, texte principal, post-scriptum.
- ▷ Les règles de base de la typographie française sont appliquées automatiquement (espaces insécables avant les points d'interrogation, d'exclamation, etc.).
- ▷ Des raccourcis mnémotechniques facilitent l'enrichissement typographique, la création de liens hypertexte, de notes de bas de page . . . mettre en page un article sous SPIP est aussi facile que d'écrire un email.
- ▷ L'interface graphique permet d'inclure simplement des images dans les articles, et chaque article peut être signalé par son propre logo.
- ▷ On peut indiquer, pour chaque article, un ou plusieurs mots-clés.
- ▷ La date de mise en ligne se gère automatiquement (on peut cependant la modifier si nécessaire) ; une seconde date peut être associée à un article, par exemple pour indiquer une date de publication originale (par exemple, un article publié antérieurement dans un magazine papier).
- ▷ Redirections (articles «fantômes») : SPIP permet de fabriquer des articles-fantômes, intégrés dans la structure du site et affichés dans le contenu des rubriques, mais qui en réalité renvoient vers une page dont l'adresse est spécifiée par le rédacteur (sur le même site, ou même sur un autre site). Cette fonction facilite le passage d'un site déjà existant vers SPIP, par l'intégration de contenus statiques préexistants.

1.2.5 Les brèves

En complément des articles, SPIP intègre un système de brèves, qui facilite la publication de courtes notes d'information, telles des revues de presse (ou des revues de Web).

- ▷ Afin de faciliter la structuration et le positionnement des brèves, on ne peut installer des brèves que dans les principales rubriques du site (les rubriques placées à la racine du site).
- ▷ La structure des brèves est simplifiée : un titre et le texte de la brève ; chaque brève peut être complétée très simplement d'un lien hypertexte.
- ▷ La gestion de la publication d'une brève est simplifiée (deux boutons : publier ou refuser).
- ▷ Chaque brève peut être signalée par son propre logo.
- ▷ L'administrateur du site peut décider de désactiver l'usage des brèves sur l'ensemble du site.

1.2.6 Les forums

SPIP intègre un système de forums.

- ▷ Les forums peuvent être associés aux articles (un forum par article), aux rubriques ou aux brèves. Le webmestre pourra programmer son interface pour que chaque article dispose de son propre forum, ou pour que plusieurs articles d'une même rubrique partagent le même forum, etc.
- ▷ SPIP permet de choisir entre plusieurs types de forums : les forums «libres» (modérés à posteriori, les contributions apparaissent immédiatement, les administrateurs peuvent éventuellement supprimer ensuite un message indésirable) ; les forums modérés à priori (les contributions n'apparaissent qu'après avoir été validées par un administrateur du site) ; les forums sur abonnement (chaque intervenant doit, pour pouvoir poster, d'abord indiquer son adresse email pour recevoir un mot de passe lui permettant de poster ses contributions).

SPIP intègre également un système de forums privés, consacré à la discussion entre les différents rédacteurs du site, et cela dans l'espace privé.

1.2.7 Les pétitions

Un article peut être transformé en pétition en ligne en quelques clics.

- ▷ Les pétitions de SPIP sont validées par email automatiquement : un signataire reçoit un message de confirmation qui permet de vérifier la validité des signatures.
- ▷ On peut configurer très simplement le type de pétition : ainsi imposer une seule signature par adresse email, imposer qu'un site Web soit indiqué dans la pétition (dans ce cas, la validité de l'URL est vérifiée automatiquement), accepter ou non des messages accompagnant les signatures.

1.2.8 Les statistiques

SPIP intègre un système très simplifié de statistiques, permettant d'évaluer la popularité des articles et des rubriques.

1.2.9 Les rédacteurs/administrateurs

Un site sous SPIP peut être géré par une seule personne, ou être réalisé par un groupe de rédacteurs.

- ▷ SPIP propose deux niveaux d'accès : les administrateurs, qui gèrent notamment la structure du site et la validation des articles, et les rédacteurs, qui proposent des articles.
- ▷ Le nombre de rédacteurs et d'administrateurs est illimité.
- ▷ On peut décider d'offrir aux utilisateurs du site public de s'inscrire pour devenir rédacteur (la procédure d'inscription est alors gérée automatiquement par SPIP).
- ▷ Chaque auteur peut se voir associer un logo personnel téléchargeable depuis l'interface (par exemple une photo d'identité).

1.2.10 Syndication

Les sites réalisés sous SPIP, phpNuke, ou d'autres systèmes, fournissent un fichier dynamique indiquant leurs dernières publications. SPIP peut analyser de tels fichiers et ainsi indiquer les nouveautés d'autres sites :

- ▷ on peut ajouter autant de sites syndiqués que l'on veut ;
- ▷ les sites syndiqués sont associés aux rubriques de son propre site ; ainsi, on peut associer à une rubrique thématique les liens vers des sites traitant du thème précis de la rubrique.

1.2.11 Interface graphique du site public

L'interface graphique du site public est très souple. Grâce à un langage très simple (mais propre à SPIP), on peut réaliser à peu près n'importe quelle interface graphique.

Il n'est en particulier pas nécessaire de connaître PHP et MySQL pour réaliser une interface graphique originale sous SPIP.

Cependant, le système de cache est totalement compatible avec PHP : le webmestre peut, s'il le désire, intégrer des fonctions PHP dans ses formats-types (squelettes), ou des passerelles CGI. On peut donc enrichir SPIP avec des scripts spécialisés pour compléter ou remplacer des fonctions manquantes (par exemple : compteur, moteur de recherche plus puissant, etc.).

1.2.12 Interface du site privé

La partie privée qui permet de gérer le site dispose d'une interface graphique complète, très simple d'utilisation.

- ▷ Cette interface s'adapte en fonction des activités de chaque rédacteur ou administrateur, et en fonction de l'activité du site. Ainsi chaque auteur a-t-il accès rapidement à ses propres articles, et les articles proposés à la publication sont signalés à tous les utilisateurs. De même l'interface est différente selon que l'on est rédacteur ou administrateur.
- ▷ Chaque utilisateur peut personnaliser son interface. Il peut choisir entre une interface simplifiée, qui n'offre que les fonctions principales, et une interface complète. Il peut également modifier quelque peu l'habillage graphique de l'interface.
- ▷ Lorsqu'un site accueille plusieurs rédacteurs, SPIP devient un outil de travail coopératif : débats autour des articles, système de validation, travail à plusieurs sur un même article . . .

1.2.13 Paramétrage du site

Si l'interface graphique du site public et la gestion du contenu sont, dans SPIP, strictement séparées (par exemple, on ne fixe pas la couleur du fond d'écran du site public dans l'espace privé), il est cependant possible de configurer certains comportements du site dans l'espace privé :

- ▷ accepter ou refuser certains éléments du contenu des articles : ainsi on peut décider d'interdire l'utilisation des surtitre, soustitre, descriptif, chapeau ou post-scriptum, ou la date de publication antérieure et les mots-clés ;
- ▷ configurer (ou désactiver) les forums publics ;
- ▷ indiquer si l'on publie les articles avant la date de publication qu'on leur a fixé (cette option permet par exemple de partir en vacances, le site publiant des articles pendant cette absence) ;
- ▷ désactiver le système de brèves (en effet, certains sites n'en ont pas l'usage ; les désactiver permet de simplifier l'interface pour les rédacteurs) ;
- ▷ activer ou désactiver les statistiques ;
- ▷ activer ou désactiver le moteur de recherche.

1.2.14 Suivi éditorial

Afin de faciliter le suivi éditorial du site, plusieurs options sont offertes :

- ▷ envoi des contributions des forums aux auteurs des articles ; lorsqu'un visiteur du site poste un message sous un article, l'auteur de cet article en est informé par mail, ce qui lui permet de suivre l'activité de son article par mail ;
- ▷ suivi de l'activité éditoriale ; si le site est le fruit d'une équipe de rédacteurs, on peut signaler automatiquement les annonces importantes de l'activité éditoriale à une adresse email (dans l'idéal, une liste de diffusion) ; ainsi, lorsqu'un article est publié ou proposé à la publication, cette liste en est informée ;
- ▷ annonce des nouveautés ; SPIP peut envoyer automatiquement, selon une fréquence fixée par les administrateurs, un courrier électronique recensant les dernières publications sur le site.

1.2.15 Sauvegarde et exportation de la base de données

Le webmestre du site peut réaliser une sauvegarde de sa base de données (un fichier est alors créé) ; si le serveur le permet, cette sauvegarde sera réalisée dans un fichier compressé, facilitant ainsi sa récupération par FTP. SPIP intègre bien entendu la fonction qui permet d'importer un tel fichier.

1.3 Conditions d'utilisation

1er juin 2001 par l'équipe de SPIP

1.3.1 Explication de la licence

SPIP est un *logiciel libre* distribué sous licence GPL, aussi appelée en français *Licence Publique Générale GNU*⁵. Cette licence vous garantit les libertés suivantes :

- ▷ la liberté d'installer et d'utiliser SPIP pour quelque usage que ce soit ;
- ▷ la liberté d'étudier le fonctionnement de SPIP et de l'adapter à vos propres besoins en modifiant le code source, auquel vous avez un accès immédiat puisque SPIP est intégralement programmé en PHP ;
- ▷ la liberté de distribuer des copies à qui que ce soit, tant que vous n'altérez ni ne supprimez la licence ;
- ▷ la liberté d'améliorer SPIP et de diffuser vos améliorations au public, de façon à ce que l'ensemble de la communauté puisse en tirer avantage, tant que vous n'altérez ni ne supprimez la licence.

Il ne faut pas confondre logiciel libre et logiciel en domaine public. L'intérêt de la licence GPL (licence du logiciel libre) est de *garantir la non-confiscation du logiciel*, au contraire d'un logiciel du domaine public qui peut se voir transformé en logiciel propriétaire. Vous bénéficiez des libertés ci-dessus dans le respect de la licence GPL ; en particulier, si vous redistribuez ou si vous modifiez SPIP, vous ne pouvez cependant pas y appliquer une licence qui contredirait la licence GPL (par exemple, qui ne donnerait plus le droit à autrui de modifier le code source ou de redistribuer le code source modifié).

1.3.2 Remarques pratiques

SPIP étant fourni et distribué gratuitement par ses auteurs, ces derniers n'offrent *aucune garantie* d'aucune sorte quant à l'utilisation que vous en ferez.

Vous pouvez utiliser SPIP pour toute activité (y compris personnelle, professionnelle ou commerciale). Vous êtes libre de déterminer la rétribution de vos services le cas échéant, d'ajouter contractuellement à votre prestation une forme de garantie quant au service fourni ; mais, encore une fois, votre propre contrat ne doit pas interférer avec la licence GPL attachée à SPIP (par exemple, vous ne pouvez pas interdire à votre client de modifier le logiciel).

Le texte de la licence GPL (en anglais) est fourni avec SPIP ; il est consultable depuis le bas de chaque page dans l'espace privé.

SPIP, Système de Publication pour l'Internet Copyright © 2001-2002, Arnaud Martin, Antoine Pitrou et Philippe Rivière pour le Minirézo.

Ce programme est un logiciel libre ; vous pouvez le redistribuer et/ou le modifier conformément aux dispositions de la Licence Publique Générale GNU, telle que publiée par la Free Software Foundation ; version 2 de la licence, ou encore (à votre choix) toute version ultérieure.

Ce programme est distribué dans l'espoir qu'il sera utile, mais SANS AUCUNE GARANTIE ; sans même la garantie implicite de COMMERCIALISATION ou D'ADAPTATION A UN OBJET PARTICULIER. Pour plus de détails, voir la Licence Publique Générale GNU .

Un exemplaire de la Licence Publique Générale GNU doit être fourni avec ce programme ; si ce n'est pas le cas, écrivez à la Free Software Foundation Inc., 675 Mass Ave, Cambridge, MA 02139, Etats-Unis.

⁵http://www.april.org/gnu/gpl_french.html

1.4 Des sites sous SPIP

1er juin 2001 par l'équipe de SPIP

La page ci-après recense quelques sites fonctionnant sous SPIP.

<http://www.uzine.net/article884.html>

Si vous êtes webmestre d'un site fonctionnant sous SPIP, merci de remplir le formulaire ci-joint (uniquement des sites réellement en fonctionnement, SVP). Indiquez également si vous publiez vos squelettes ou si vous acceptez de les fournir à ceux qui vous en feraient la demande.

1.5 Obtenir de l'aide

1er juin 2001 par l'équipe de SPIP

Les listes de diffusion. Nous vous proposons 3 listes de discussion par email autour de SPIP.

Pour chacune de ces listes, vous pouvez non seulement vous abonner, mais aussi accéder aux archives et ainsi voir si votre problème n'a pas déjà été abordé.

▷ La liste des utilisateurs : spip@rezo.net

Cette liste est destinée à toutes les questions autour de l'utilisation de SPIP. C'est la liste sur laquelle vous aurez le plus de chances d'obtenir de l'aide (soit de la part des développeurs, soit d'autres utilisateurs). Il s'agit donc de notre liste principale : si vous êtes webmestre d'un site SPIP, rédacteur d'un site SPIP, et que vous voulez discuter de différents problèmes liés à l'*utilisation* de SPIP, c'est là qu'il faut vous rendre... Si vous débutez avec SPIP, nous vous conseillons vivement de vous abonner à cette *liste des utilisateurs* (tout cela est évidemment gratuit). Cette liste est désormais très animée, réactive, et vous y obtiendrez rapidement de nombreux conseils et des réponses à toutes sortes de questions.

Important

Cette liste est très active : de nombreux messages (une trentaine par jour), beaucoup d'abonnés (environ 500), une belle entraide. La liste n'est absolument pas modérée, c'est donc sur la bonne volonté de tous les participants qu'elle fonctionne.

Pour éviter de surcharger la boîte aux lettres des autres participants et de faire perdre du temps à tout le monde, merci de suivre ces quelques recommandations :

- ★ même si vous ne voulez pas lire l'intégralité de la documentation, **consultez au minimum les FAQ** (partie III page 56) avant de poser une question ;
- ★ précisez systématiquement **quelle version de SPIP** vous utilisez ;
- ★ **décrivez aussi précisément que possible votre problème** : à quel endroit cela survient, quels sont les éventuels messages d'erreur... une très grande partie des questions posées sur la liste sont en effet très imprécises, et nécessitent alors plusieurs échanges de messages avant même de comprendre la nature du problème (syndrome : «ça marche pas», «qu'est-ce qui ne marche pas?», «tout, rien ne fonctionne», «tu as réussi l'installation?», «oui, mais ça marche pas», etc.) ;
- ★ si votre question concerne la création de squelettes, **fournissez l'URL du squelette concerné** (inutile de recopier le squelette dans le message) ;
- ★ lors d'un échange au sujet de votre problème, laissez dans les messages suivants la description de votre problème : avec une trentaine de messages par jour, on finit par oublier sur quoi telle discussion portait ;
- ★ si personne ne vous répond, inutile de vous énerver, c'est que personne ne connaît de réponse. Certains utilisateurs postent leur réponse sur la liste *spip*, puis au bout d'une demi-heure ré-expédient le même message sur un ton énervé, puis une heure plus tard posent la même question sur la liste *spip-dev*, puis en fin de soirée écrivent directement sur l'adresse personnelle des développeurs de SPIP ; c'est, à coup sûr, le meilleur moyen de ne jamais obtenir d'aide ;

* envoyez vos messages en mode *texte*, et surtout pas en HTML ; sur une liste très active, les messages HTML ralentissent la consultation.

▷ **La liste des développeurs : spip-dev@rezo.net**

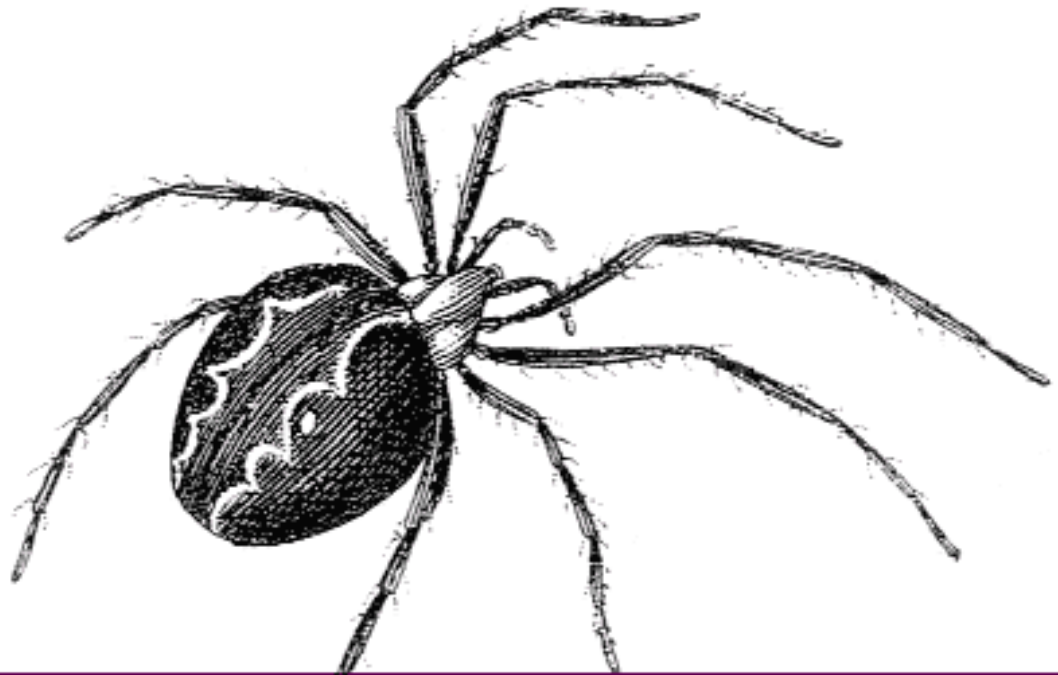
Attention : cette liste est une liste technique ; on y cause PHP et MySQL ... Elle est donc réservée aux web-mestres confirmés. Notez bien : c'est une liste destinée à discuter de la *programmation* de SPIP. Pour toutes les questions liées à l'utilisation, voyez la liste précédente. (L'équipe qui programme SPIP est abonnée aux deux listes !)

▷ **La liste des annonces : spip-ann@rezo.net**

Cette liste est une liste de diffusion, «passive» : il ne s'y déroule pas de discussions. Elle est réservée à l'annonce d'événements importants concernant SPIP (nouvelles versions notamment). Y circulent donc peu de messages.

Chapitre 2

Installation



Installation

2.1 Hébergeurs acceptant SPIP

1er juin 2001 par l'équipe de SPIP

Nous vous proposons des listes d'hébergeurs sur lesquels nous avons connaissance que SPIP fonctionne. Ces listes sont constituées grâce à l'aide des webmasters qui nous informent de leurs essais (échecs et réussites). Elles ne peuvent s'enrichir que par ce biais.

Si vous avez installé SPIP chez un hébergeur qui ne figure pas ci-dessous, merci de nous faire part de votre expérience, en indiquant bien entendu si cela a fonctionné, ainsi que le nom et l'adresse Web de l'hébergeur. Pour cela, vous pouvez utiliser le forum ci-dessous ou écrire directement à la liste de discussion spip@rezo.net.

N.B. Nous sommes plus particulièrement intéressés, pour cette liste, par les hébergeurs gratuits, associatifs ou à bas prix. En effet, le choix d'un hébergeur (semi-)professionnel est plus simple : l'installation de SPIP n'y pose généralement aucun problème.

2.1.1 Ca marche chez ...

Hébergeurs «gratuits» :

- ▷ Multimania (gratuit, fonctionnement correct, publicité envahissante et obligatoire, lire Tester SPIP chez Multimania (section 2.4 page 23) pour plus de détails)
- ▷ Free (fonctionnement bridé, problèmes avec des articles longs, voir Tester SPIP chez Free (section 2.3 page 21) pour des explications plus complètes)
- ▷ Le Village (fonctions d'envoi de mail désactivées)
- ▷ Maxi-Web (syndication désactivée)

Hébergeurs plutôt non commerciaux :

- ▷ APINC (associatif autogéré)
- ▷ Globenet (associatif)
- ▷ Infini (association Internet Finistère)
- ▷ L'Autre Net (associatif autogéré)
- ▷ Ouvaton (société coopérative)
- ▷ R@S (réseau associatif et syndical)

Hébergeurs payants :

- ▷ Abergement
- ▷ All2All
- ▷ Altern (qu'on ne présente plus)
- ▷ Amen
- ▷ Discount-Hosting
- ▷ Infini
- ▷ In-Web
- ▷ JeSurf
- ▷ Nexen (spécialiste du logiciel libre)
- ▷ NFrance
- ▷ OLM
- ▷ Online (fonctionnement bridé, problèmes avec les articles longs)
- ▷ OVH
- ▷ Pair Networks

- ▷ SC3M
- ▷ Sivit
- ▷ VirtuaVia
- ▷ Web-Solutions
- ▷ WebVisio

2.1.2 Ca marche plus ou moins chez ...

- ▷ Hébergement-Discount : voici la marche à suivre selon Ronan Quentel :

«En fait, dans la procédure d'installation de SPIP, le fichier `inc_connect.php3` est créé dans le répertoire «`ecriture`» du site.

Or, via un logiciel FTP, sur Hébergement-Discount, il est impossible de lui donner les autorisations d'accès nécessaires au fonctionnement de SPIP. Il faut donc copier ce fichier sur son disque, le renommer (`inc_connect2.php3` par exemple), le rebasculer sur le FTP pour lui donner les autorisations d'accès nécessaires. Enfin, il faut supprimer le fichier `inc_connect.php3` et renommer le fichier copié avec le nom correspondant.»

2.1.3 Ca ne marche pas ...

Certains hébergeurs restreignent des fonctionnalités du serveur utilisées par SPIP. Les blocages portent en général sur :

- ▷ l'installation automatisée de la base de données ;
- ▷ la création de répertoires et de fichiers en PHP ;
- ▷ la création de dossiers protégés par mot de passe.

2.1.4 Questions / réponses

Mon hébergeur n'est pas dans la liste ci-dessus. Comment je fais ?

S'il propose PHP3 ou 4 avec MySQL, il y a de grandes chances que ça marche. De toute façon, l'ensemble téléchargement plus essai d'installation devrait prendre aux plus maladroits quelques dizaines de minutes *maximum*. Ensuite, merci de nous informer du succès ou de l'échec de la démarche, avec le nom de l'hébergeur et les éventuels problèmes rencontrés, afin que votre expérience puisse enrichir cette documentation et profiter à tous !

Si vous voulez tout de même vraiment savoir à l'avance si ça va marcher, lisez les questions qui suivent.

Avec quelles versions de PHP SPIP est-il compatible ?

Toutes à partir de PHP3 (donc y compris PHP4).

Avec quelles versions de MySQL SPIP est-il compatible ?

Aucun test n'a été fait en ce sens, mais normalement toutes les versions récentes conviennent : notamment, à partir de la 3.22 (à l'exception du tri `{par num titre}`, disponible uniquement à partir de la 3.23).

Et avec quels serveurs Web, sous quels systèmes d'exploitation ?

Encore une fois, nous n'avons pas fait de tests sur le sujet. Bien entendu, le choix recommandé, car le plus répandu, est Apache sous un Unix quelconque (par exemple Linux, FreeBSD, Solaris, MacOS X ...).

Néanmoins, il est fort probable que SPIP marche sous d'autres configurations. Ainsi, il est régulièrement testé avec succès sous Windows, avec PHP4 en module Apache. Merci de nous signaler votre expérience avec des configurations logicielles autres que celles citées ci-dessus.

Où en savoir plus sur l'installation ?

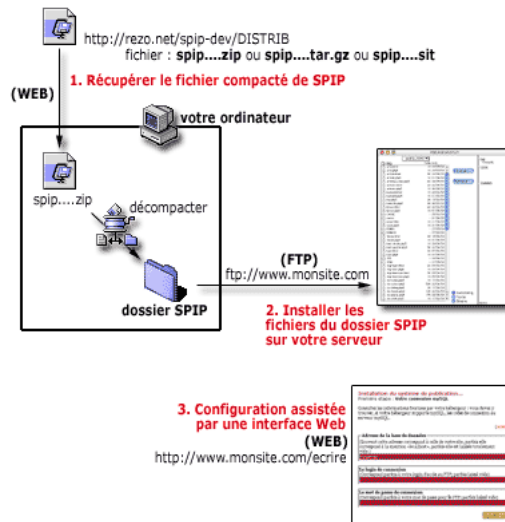
- ▷ Le manuel d'installation (section 2.2 page 18) ;
- ▷ et, en cas de problème, la FAQ technique (section 5 page 57).

2.2 Installer SPIP

1er juin 2001 par l'équipe de SPIP

L'installation de SPIP est très simple : il n'y a pas, en particulier, de fichier à modifier «à la main» avec des variables érotiques. La procédure est très simple (elle est détaillée ci-après) :

1. Récupérer le fichier de SPIP sur notre serveur, et le décompresser sur votre ordinateur personnel. Vous obtenez un dossier «SPIP ...» contenant l'ensemble des fichiers du système SPIP.
2. Installer le contenu de ce dossier sur votre site (par FTP, comme vous le faites habituellement pour installer vos pages sur votre site).
3. Vous connecter avec votre navigateur sur votre site, dans un dossier intitulé «ecrire», où SPIP vous proposera une interface graphique vous permettant de configurer le système. Une fois ces quelques informations de configuration fournies, SPIP sera totalement installé et vous pourrez commencer à travailler sur votre site.



2.2.1 Configuration nécessaire

Vous devez disposer d'un hébergement Web avec :

- ▷ un accès FTP pour l'installation des fichiers ;
- ▷ le support de PHP3 ;
- ▷ un accès à une base de données MySQL.

Avant l'installation, vous devez avoir une base mySQL disponible. Sur de très nombreux hébergements, il faut soit demander l'activation d'une base mySQL à l'administrateur, soit suivre une procédure automatique en ligne (dans tous les cas, l'activation de la base mySQL n'a rien à voir avec SPIP ; si vous avez des difficultés, seul votre hébergeur peut vous fournir les mots de passe nécessaire et vous expliquer comment activer votre compte mySQL).

Vous devez connaître les données de votre connexion mySQL (fournies par l'hébergeur) :

- ▷ l'adresse de la base MySQL : par exemple `sql.free.fr`, ou `localhost`, ou vide ;
- ▷ le nom de la base de données utilisée : souvent le même login que votre compte Web ;
- ▷ votre login MySQL : souvent le même login que votre compte Web ;
- ▷ votre password MySQL : souvent le même que le compte Web.

Exemple : si vous disposez d'un compte nommé «monsie» chez Free (adresse `http://monsie.free.fr`), l'adresse de la base mySQL est «`sql.free.fr`», le nom de la base de données est «monsie», votre login est «monsie» et le mot de passe est celui de votre compte. Il vous suffit d'activer votre base de données pour php¹.

Ces éléments sont indispensables : si vous ne les connaissez pas, contactez votre hébergeur et demandez-lui de vous les rappeler.

Aucune configuration spéciale n'est nécessaire sur votre ordinateur personnel, SPIP se gère entièrement sur le Web. Tout ce dont vous avez besoin, c'est d'un navigateur Web (n'importe lequel), et d'un logiciel de transfert FTP pour installer les fichiers sur votre compte.

2.2.2 Installation automatique (expérimentale)

Il existe pour SPIP une procédure d'installation ultra-simplifiée : un fichier à télécharger sur votre serveur et ça s'installe. *Attention* : cette procédure ne fonctionne pas sur tous les serveurs. Si elle ne fonctionne pas (vous vous en rendez compte immédiatement), passez à l'étape 1 ci-dessous.

Récupérez le fichier `spip_loader.php3` à l'adresse ci-dessous (si le fichier s'affiche dans votre navigateur, faites «Enregistrer sous ...») :

`http://rezo.net/spip-dev/INSTALL`

et téléchargez-le tel quel sur votre serveur (chez votre hébergeur) par FTP. «Visitez» cette page avec votre butineur Web habituel à l'adresse du style :

`http://www.moncompte.com/spip_loader.php3`)

et suivez la procédure indiquée.

Si la procédure fonctionne, ce petit fichier (7 Ko) va récupérer SPIP sur notre propre serveur et l'installer chez votre hébergeur. Ensuite la procédure de configuration démarre automatiquement (étape 3 ci-dessous).

1. Récupérer SPIP

SPIP est disponible en téléchargement par le Web à l'adresse :

`http://rezo.net/spip-dev/DISTRIB`

Ce dossier contient plusieurs fichiers, vous devez récupérer celui qui correspond à votre propre système d'exploitation :

- ▷ `spip.zip` ou `spip.tar.gz` si vous tournez sous Windows ou Linux ;

¹`http://subscribe.free.fr/acces/createbase.html`

▷ *spip.sit* si vous êtes sous Macintosh (vous avez alors besoin de Stuffit Expander² pour décompresser). Décompressez l'archive sur votre ordinateur dans le répertoire de votre choix, puis installez les fichiers chez votre hébergeur via FTP.

2. Installer les fichiers

Installez l'ensemble des fichiers de SPIP sur votre site, à l'endroit où vous voulez que le site géré par le script soit accessible au public : le plus souvent à la racine de votre site, mais ce n'est pas impératif.

La structure est la suivante :

- ▷ **répertoire racine**
 - * squelettes .html
 - * nombreux fichiers .php3
 - * **dossier /CACHE** (vide)
 - * **dossier /IMG** (vide)
 - * **dossier /NAVPICS**
 - * **dossier /ecrire** (le plus important)
 - ◇ nombreux fichiers .php3
 - ◇ *dossier /AIDE*
 - ◇ *dossier /IMG2*
- ▷ *dossier /data* (vide)

3. Procédure de configuration

Désormais tout se déroule en ligne. Il vous suffit d'aller «visiter» votre dossier «/ecrire» par le Web.

Exemple : selon notre exemple précédent, il s'agirait de l'adresse

<http://monsite.free.fr/ecrire>

Installation du système de publication...
Première étape : **Votre connexion MySQL**

Consultez les informations fournies par votre hébergeur : vous devez y trouver, si votre hébergeur supporte MySQL, les codes de connexion au serveur MySQL.

[AIDE]

Adresse de la base de données
(Souvent cette adresse correspond à celle de votre site, parfois elle correspond à la mention «localhost», parfois elle est laissée totalement vide.)
localhost

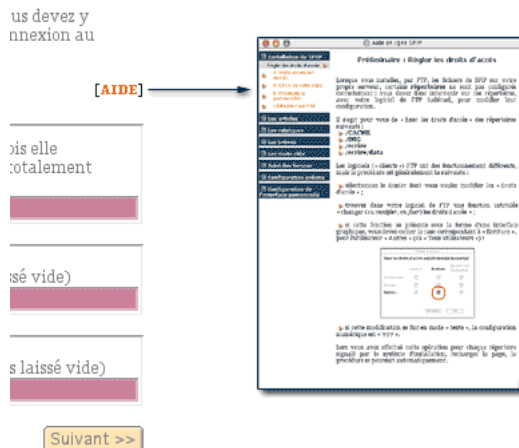
Le login de connexion
(Correspond parfois à votre login d'accès au FTP; parfois laissé vide)

Le mot de passe de connexion
(Correspond parfois à votre mot de passe pour le FTP; parfois laissé vide)

Suivant >>

Lors de la première connexion à cette adresse, une procédure d'installation pas-à-pas démarre. L'interface est très simple, il suffit d'entrer les informations demandées (essentiellement les informations concernant la base de données MySQL indiquées au début). Une fois que c'est terminé, le système vous demande l'identification que vous avez indiquée et vous pouvez commencer à gérer votre site. Par la suite, c'est toujours dans ce dossier «/ecrire» que vous irez travailler, muni de vos codes d'identification.

²<http://www.aladinsys.com/expand/>



À chaque étape de la procédure d'installation, vous trouverez un lien [AIDE], qui provoque l'affichage d'une aide en ligne expliquant chaque détail de l'utilisation de SPIP. (La seule opération un peu complexe apparaît sur certains serveurs : il vous faudra peut-être modifier les «droits d'accès» de certains dossiers ; l'opération n'est pas bien méchante, et l'aide en ligne vous fournit tous les détails nécessaires.)

Si tout s'est bien déroulé jusqu'ici, la procédure d'installation est terminée, et vous pouvez créer et gérer votre site sans aucune autre manipulation exotique . . .

En cas de grosse erreur (du genre : vous avez oublié votre propre accès au site - fréquent au début . . .), pour «relancer» cette procédure d'installation, il faut utiliser votre logiciel FTP et effacer les fichiers suivants :

- ▷ /ecriture/inc-connect.php3
- ▷ /ecriture/.htaccess (s'il existe)

La connexion suivante dans le dossier «ecriture» relancera alors la procédure de configuration (en réalité, c'est l'absence de fichier «inc-connect.php3» qui provoque le lancement de cette procédure).

Votre témoignage

Afin de nous aider à améliorer cette procédure d'installation, merci de faire part de votre expérience dans le forum ci-dessous ou en écrivant à spip@rezo.net. Si vous effectuez l'installation vous-même, veuillez indiquer :

- ▷ le nom de votre hébergeur (important, ça, qu'on ait une idée des différents hébergeurs compatibles, notamment les gratuits) ;
- ▷ les éventuelles difficultés rencontrées (y compris les difficultés d'interface et de compréhension du processus d'installation, histoire qu'on puisse améliorer l'interface ou la documentation) ;
- ▷ même si votre installation s'est déroulée sans aucune difficulté, merci de l'indiquer (c'est une info intéressante).

2.3 Tester SPIP chez Free

11 juin 2001 par l'équipe de SPIP

2.3.1 Ouvrir un compte chez Free

A la différence d'autres hébergeurs, la procédure s'effectue en deux temps. Premièrement : allez remplir le formulaire en ligne³, puis deuxième temps, attendez par retour de courrier la précieuse lettre de M. Free. Ce qui suppose

³<http://inscription.free.fr/formuser.html>

que vous fournissiez une adresse postale à laquelle vous pouvez recevoir du courrier (celle d'une copine, de votre grand-mère, de votre voisin, peu importe).

Si vous ne voulez pas utiliser l'adresse

<http://gerard.leporc.free.fr>,

mais <http://darkWarrior.free.fr>,

vous pouvez créer des comptes supplémentaires⁴.

Vous pouvez héberger chez Free jusqu'à 100 Mo de pages personnelles. Pour cela, il faut tout d'abord les activer avec le lien ci-dessous⁵. Après 4 heures de méditation intensive, vous pourrez placer vos pages sur le serveur, avec un client FTP⁶. Paramètres Valeurs : Host Name : ftpperso.free.fr | User ID : Votre login | Password : Votre mot de passe .

2.3.2 Activer la base de données

Comme de nombreux hébergeurs (y compris payants), il vous faut demander l'activation d'un accès personnel à la base de données. Vous devez donc procéder à cette opération (pas bien compliquée).

Rendez-vous à la page dédiée⁷ : et indiquez vos identifiants (login ou nom d'utilisateur + password ou mot de passe). Les bases de données sont créées automatiquement toutes les nuits. Il faudra donc attendre un certain temps avant de pouvoir accéder à votre base de données. Celle ci porte le nom de votre login. Pour y accéder, vous devez utiliser votre login/mot de passe du compte activé. Par FTP, vous pourrez voir un petit répertoire `/mysql/` apparaître ; c'est le signal pour commencer les choses sérieuses !

Remarque. Il n'est pas possible de créer de nouvelle base de données. Vous devez vous contenter d'utiliser celle qui est créée pour vous et qui porte le nom de votre login.

Remarque. Vous pouvez également l'administrer directement grâce à l'interface web⁸ de Free. Avec SPIP, vous n'en avez pas besoin.

2.3.3 Installer SPIP

L'installation se déroule désormais telle qu'elle est décrite dans notre documentation («Installer SPIP» section 2.2 page 18).

Lors de la phase de configuration de SPIP, la première étape vous demande d'indiquer les éléments de connexion à la base de données. L'adresse de la base de données reste «localhost». Le login de connexion et le mot de passe sont ceux de votre compte activé. Le nom de votre base est celui de votre login.

Et voilà, votre site fonctionne. Rendez vous donc sur <http://moncompte.free.fr/ecrire> pour commencer à créer votre site.

Rappel : Free⁹ a désactivé la fonction ()mail

Enfin, signalons que l'on peut faire tourner spip en php4 chez Free (et non php3) en utilisant le «free pack» proposé par Antoine (télécharger¹⁰). Le calcul et l'affichage des pages de l'espace public sont grandement accélérés.

⁴<http://subscribe.free.fr/acces/newaccount.html>

⁵<http://inscription.free.fr/acces/pagesperso.html>

⁶<http://support.free.fr/web/publier.html>

⁷<http://subscribe.free.fr/acces/createbase.html>

⁸<http://sql.free.fr/phpMyAdmin>

⁹<http://support.free.fr/services/questions.html#7>

¹⁰<http://rezo.net/spip-dev/contrib/Antoine/>

2.4 Tester SPIP chez Multimania

6 juin 2001 par l'équipe de SPIP

Depuis la version 1.0.2, SPIP est compatible avec Multimania. Voici une petite explication pratique pour pouvoir installer SPIP chez eux ...

Remarque. Réglons tout de suite la question : «Multimania c'est caca», parce que je sens que les forums vont être houleux ... Je ne suis pas plus partisan qu'un autre de Multimania (plutôt moins, sachant que s'aventurer à faire du PHP/mysql chez eux est plutôt pénible), seulement voilà : pour l'heure c'est l'un des très rares hébergeurs gratuits proposant PHP et mysql sur lesquels on puisse ouvrir un compte en quelques minutes (pour Free, il faut attendre vos codes de connexion par la poste). C'est donc un moyen pratique pour «essayer» SPIP et se faire un idée rapidement. (Et bien entendu : après les bidouilles d'essais, allez voir ailleurs si la bande passante est plus saine et les fonctionnalités moins bridées.)

N.B. Le service technique de Multimania/Lycos nous informe que, à dater de juillet 2002, les fonctions de gestion des logos et des images de SPIP fonctionnent sur leurs machines.

2.4.1 Ouvrir un compte chez MultiNian-Nian

On ne va pas vous tenir la main : c'est facile. Le tout c'est de trouver le bouton «Devenez membre» en haut à gauche (cartouche orange) de la page d'accueil¹¹.

N'oubliez pas que l'activation de votre compte se déroule par la confirmation d'un email que vous envoie immédiatement l'hébergeur.

2.4.2 Activer la base de données

Comme de nombreux hébergeurs (y compris payants), il vous faut demander l'activation d'un accès personnel à la base de données. Vous devez donc procéder à cette opération (pas bien compliquée).

La seule difficulté, c'est la navigation dans les menus de MultiCaca pour arriver à trouver où ça se passe ...

- ▷ Rendez-vous dans l'espace membre¹² ; indiquez vos identifiants (colonne de gauche).
- ▷ Dans l'espace membre, vous trouverez dans la colonne de gauche une rubrique «Vos pages perso», dans laquelle se trouve : «PHP4/mysql» ; c'est là que ça se passe.
- ▷ Cette nouvelle page contient un bouton (en bas, et c'est écrit en petit) : «Je désire activer ma base mysql». Sélectionnez cette option et validez les modifications. Si ça se passe bien, la base de donnée fonctionne.

Remarque. La même page propose alors un lien «Votre base mysql est activée, vous pouvez à présent l'administrer». Avec SPIP, vous n'en avez pas besoin.

2.4.3 Installer SPIP

L'installation se déroule désormais telle qu'elle est décrite dans notre documentation («Installer SPIP» section 2.2 page 18).

Deux remarques :

- ▷ l'accès par FTP à votre compte MultiMerdia est très mal documenté ; c'est cependant très simple. Le serveur est ftp.multimania.com, votre identifiant est le nom de votre compte (celui que vous avez choisi lors de votre inscription, par exemple «moncompte»), et le mot de passe est également celui de votre inscription (par exemple «monpasse»).

¹¹<http://www.multimania.lycos.fr/>

¹²<http://www.multimania.fr/general/membre/>

- ▷ lors de la phase de configuration de SPIP, la première étape vous demande d'indiquer les éléments de connexion à la base de données. L'adresse de la base de données reste «localhost». Le login de connexion est votre login habituel, **suivi de _db** (par exemple moncompte_db). Le mot de passe de connexion est votre mot de passe habituel sur Niania.
- ▷ l'étape 3 vous demande de sélectionner votre base de données. SPIP a déjà sélectionné l'option nécessaire («moncompte_db»), il suffit de valider.

2.4.4 Hop, ça roule ...

Et voilà, votre site fonctionne.

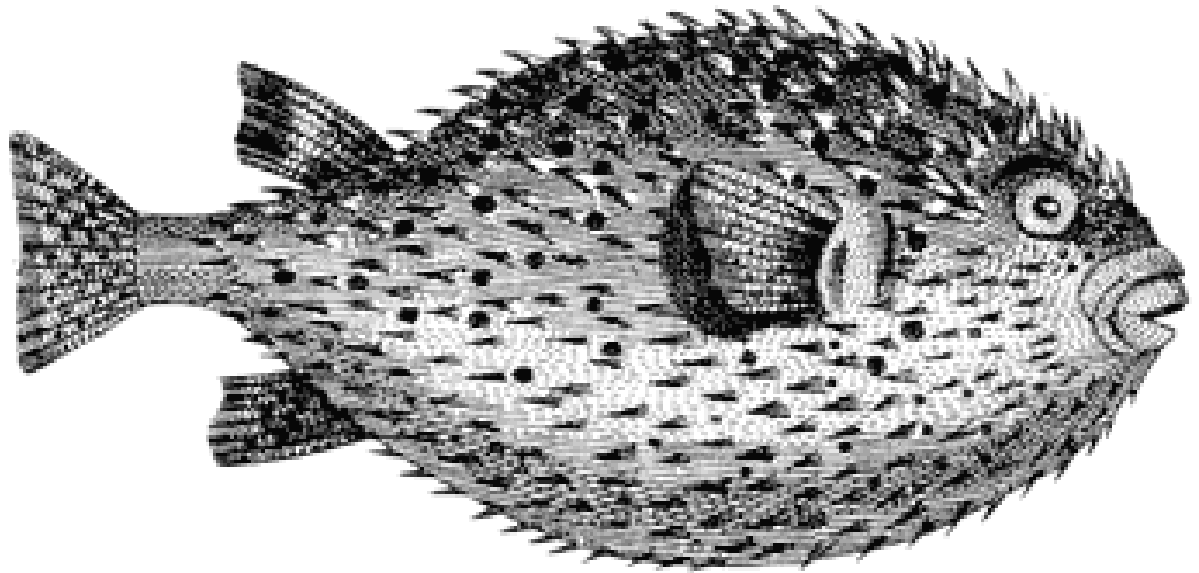
Rendez vous sur

<http://moncompte.multimania.com/ecrire>

pour commencer à créer votre site.

Chapitre 3

Evolution et mise à jour



Évolutions et mises à jour

3.1 Effectuer une mise à jour

15 novembre 2001 par l'équipe de SPIP

Effectuer une mise-à-jour de SPIP est très simple. Cependant, voici une méthode recommandée pour éviter les erreurs.

La procédure, décrite en détail ci-après, peut sembler compliquée. En réalité, nous entrons volontairement dans le détail de chaque opération pour vous éviter certaines erreurs ; mais dans la pratique, la mise-à-jour de SPIP se réalise en quelques minutes et est d'une grande simplicité.

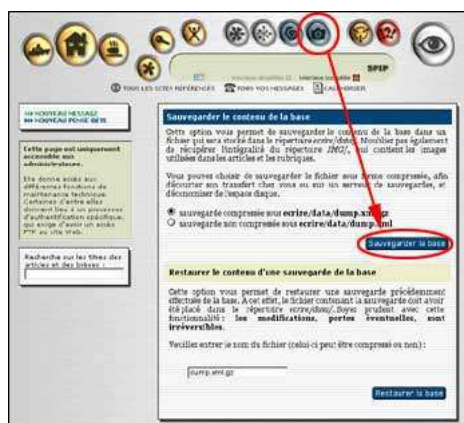
3.1.1 Sauvegardez votre base de données

Avant toute modification importante d'un système informatique, il est toujours conseillé d'effectuer une sauvegarde de précaution.

Notez bien : il s'agit uniquement d'une précaution. Vous n'êtes pas obligé de l'effectuer, et le fichier de cette sauvegarde ne vous servira certainement à rien, puisque la mise à jour de SPIP se déroulera sans problème !

C'est la même logique que lorsque vous modifiez le système d'exploitation de votre ordinateur (installer une nouvelle version de Windows, de MacOS ou de Linux ...) : vous sauvegardez vos documents importants, mais vous savez très bien que, si la mise-à-jour s'est bien déroulée, vous n'aurez pas besoin de réinstaller ces documents.

Ce point est important : **sauvegardez votre base de données avant la mise-à-jour, mais ne la réinstallez pas !** En effet, nous avons constaté que de nombreux utilisateurs sauvegardaient leur base de données, effectuaient la mise-jour, puis *réinstallaient leurs documents à partir de cette sauvegarde* ; c'est une erreur, et leurs sites présentaient alors des dysfonctionnements. La sauvegarde est une simple précaution en cas de gros problème lors de la mise à jour, mais si l'opération se déroule bien (ce qui est presque toujours le cas !), **vous ne devez pas réinstaller cette sauvegarde**. (La sauvegarde est réalisée avec une structure des données correspondant à la version précédente de SPIP ; si vous installez ces données après la mise à jour, vous les réinstallez dans une structure qui a évolué, provoquant ainsi l'apparition de problèmes.)



Pour réaliser la sauvegarde de votre base de données (c'est-à-dire de l'intégralité de vos documents réalisés avec SPIP), rendez-vous dans la page «Sauvegarde/restauration de la base» de l'espace privé, et cliquez sur le bouton «Sauvegarder la base».

La procédure d'authentification par FTP démarre :



La nouvelle page vous indique un nom à recopier, du type «admin_xxxxx». Copiez ce nom, et démarrez votre logiciel-client FTP. Rendez-vous sur votre compte FTP correspondant à votre site, et placez-vous dans le dossier «/data» qui se trouve à l'intérieur du dossier «/ecriture». Dans ce dossier «/data», créez un nouveau dossier auquel vous donnez le nom indiqué ci-dessus (votre nouveau dossier aura donc un nom de la forme «admin_xxxx»).

Une fois ce dossier créé, revenez à la page de votre butineur, et cliquez sur le bouton «recharger cette page». La sauvegarde est alors effectuée.

Si vous le désirez, vous pouvez vérifier dans votre logiciel FTP que le document «dump.xml» (ou «dump.xml.gz») a été créé : ce document est la sauvegarde de votre site. Vous pouvez la laisser sur votre compte FTP, ou la télécharger sur votre propre ordinateur.

3.1.2 Installez la nouvelle version de SPIP

La véritable procédure de mise-à-jour commence ici.

Le principe est très simple : il suffit d'installer les fichiers de SPIP une nouvelle fois, exactement comme vous l'aviez fait lors de la première installation. Soit avec «spip_loader» qui effectue l'installation automatique des fichiers, soit plus traditionnellement en décompactant SPIP sur votre propre disque dur et en envoyant tous les fichiers par FTP chez votre hébergeur.

Notez bien : il n'est pas nécessaire de supprimer les fichiers de la version précédente. Cela n'est ni nécessaire ni conseillé : en effet, si vous supprimez les anciens fichiers, vous devrez certainement procéder à nouveau au paramétrage de SPIP (indiquer les données de connexion à la base de données, etc.), procédure inutile si vous vous contentez d'écraser les anciens fichiers avec les nouveaux. En effaçant les anciens fichiers, vous risquez même d'effacer les images contenues dans vos articles ! Donc : restez simple : inutile d'effacer quoi que ce soit avant d'installer les nouveaux fichiers ; les nouveaux écraseront les anciens automatiquement ...

3.1.3 Déclenchez la mise-à-jour de votre site

Une fois *tous les fichiers* réinstallés (par FTP, ou automatiquement avec «spip_loader»), rendez vous dans l'espace privé de votre site.

Notez bien : attendez que *tous les fichiers* soient bien installés chez votre hébergeur. Inutile d'essayer d'intervenir sur votre site *pendant* le transfert des fichiers, vous obtiendriez des résultats incohérents ...

Dans votre espace privé, vous obtenez invariablement le message : «**Message technique : la procédure de mise à jour doit être lancée afin d'adapter la base de données à la nouvelle version de SPIP.** Si vous êtes administrateur du site, veuillez cliquer sur ce lien.» :



En tant qu'administrateur, suivez le lien pour pouvoir déclencher la mise-à-jour de SPIP. Vous arrivez sur un écran d'*authentification par FTP*, exactement similaire à la procédure décrite plus haut (pour la sauvegarde de la base) :



Une nouvelle fois, copiez le mot indiqué par cet écran et, avec votre logiciel-client FTP, créez un nouveau répertoire dans «/ecriture/data», et donnez-lui le nom que vous venez de copier.

Cliquez sur «recharger cette page», ce qui déclenche la mise à jour de la base de données de SPIP.

3.1.4 Videz le cache de votre site public

Afin d'être certain que les pages de votre site public vont bien exploiter la nouvelle structure de SPIP, il faut vider les pages contenues dans le système de cache.

Dans votre espace privé, rendez-vous dans la page «Sauvegarde/restauration de la base», et cliquez sur le bouton «Vider le cache».

Voilà, l'opération de mise-à-jour est terminée.

Répetons ce conseil, car c'est une erreur très classique : **ne réinstallez pas la base de données à partir de la sauvegarde effectuée précédemment**. Cette sauvegarde correspond désormais à une ancienne structure des données, elle est donc plus ou moins inutilisable. Si l'opération s'est bien déroulée, vous pouvez même détruire cette ancienne sauvegarde.

3.1.5 En cas de pépin !

▷ **Votre espace privé fonctionne correctement, mais certaines pages du site public affichent des messages d'erreur, les formulaires des forums fonctionnent mal, etc.**

Vous avez sans doute oublié de vider le cache après la mise-à-jour.

▷ **À certains endroits de l'espace privé (et parfois sur le site public), vous obtenez des messages d'erreur du type «file not found», «file missing», ou d'autres messages indiquant un problème de fichier incomplet ou manquant.**

Il y a peut-être eu des problèmes lors du téléchargement des fichiers par FTP ; avec votre logiciel-client FTP, vérifiez la taille des fichiers incriminés ; notamment, si un fichier a une taille de 0 ko, réinstallez ce fichier.

▷ **Vous obtenez beaucoup de messages d'erreur.**

Réinstallez à nouveau l'intégralité des fichiers par FTP. On ne sait jamais ...

▷ **Malgré plusieurs transferts des fichiers par FTP, ça ne fonctionne toujours pas.**

Essayez d'envoyer les fichiers selon un autre encodage. Vous pouvez en effet, avec votre client FTP, choisir selon quel encodage vous envoyez les fichiers chez votre hébergeur (texte brut, binaire ...). Sélectionnez un autre format d'encodage pour envoyer vos fichiers.

▷ **Tout fonctionne très bien, mais les dates sont affichées avec des caractères bizarres à la place des accents ; par exemple «février» s'affiche «f]vrier».**

Avez-vous bien récupéré la version compactée de SPIP correspondant à votre système d'exploitation ? Si vous êtes sur Macintosh, la version «.sit» ; sous Windows, «.zip» ; sous Linux, «.tar.gz» ...

- ▷ **Mon site fonctionne à peu près correctement, mais certaines fonctionnalités ne fonctionnent pas bien, les rédacteurs rencontrent des problèmes alors que cela fonctionne bien pour les administrateurs, etc.**

Rendez-vous sur la liste de diffusion des utilisateurs¹ de SPIP. Dans un premier temps, consultez les archives² de cette liste pour voir si votre problème n'a pas été déjà abordé. Sinon, exposez votre question sur la liste en étant le plus précis possible : quel hébergeur, à partir de quelle version avez-vous effectué la mise-à-jour ; n'oubliez pas de signaler si ce problème apparaissait déjà ou non dans la version précédente. Si votre problème est un bug encore inconnu, les développeurs travailleront très rapidement sur la question pour livrer une version corrigée.

3.2 L'histoire minuscule et anecdotique de SPIP

14 février 2002 par ARNO*

Les prémices de SPIP remontent au courant de l'année 1998 : Pierre Lazuly souhaite développer un système de publication pour faciliter la gestion de son site «Les chroniques du Menteur»³. ARNO* a réalisé en Server Side Includes (une technologie très rudimentaire) un petit outil pour gérer les édits du Scarabée⁴ et, de son côté, Erwan a développé un outil pour gérer L'Ornitho⁵.

Erwan est alors le seul à savoir gérer une base de données, Pierre est en train de s'initier à PHP, et ARNO* ne connaît ni PHP ni les bases de données. Mais Pierre passe ses vacances sur un bateau baptisé «SPIP» ; et comme «SPIP» est l'acronyme de «Système de Publication pour l'Internet», cela suffit à lancer le projet : on a le titre, le reste devrait être facile ...

Cependant, malgré quelques essais (un premier système gère un site à base de PHP, mais pas de base de données, les informations étant stockées dans des fichiers selon un format spécifique - une sorte de XML qui s'ignorait -, avec un premier système d'identification des rédacteurs), le projet ne progresse guère. Il faudrait en effet :

- ▷ pouvoir réaliser n'importe quel type de site avec un même outil ;
- ▷ gérer ce site avec une interface accessible à tous.

Lors d'une soirée, Erwan dessine sur un bout de la nappe en papier du restaurant la structure d'une base de données qui permettrait de réaliser n'importe quelle structure de site. L'esthétique de la chose effraie ses deux compères, et SPIP en reste à ce stade de la nappe de restaurant fâchée de café.

Pierre définit l'utilisation de touches rarement utilisées dans un texte (les accolades notamment) pour créer rapidement de l'italique et du gras, ce qu'il nomme les «raccourcis SPIP». Le système de publication des éditoriaux en Server Side Includes est traduit en PHP, mais pas d'utilisation de la base de données MySQL.

• Nouveaux édits				
102	Avec ce coin, mon fils a été percuté et se traîne de se moquer!	Mais que des amnésies, qu'en fait. Mais pour de vrais problèmes de mémoire, c'est encore mieux. Mais pour de vrais problèmes de mémoire, c'est encore mieux.	2000-05-08	supprimer
1	L'été prochain, le capital-étage et la mère	Novel le économie : bientôt son site!	2000-02-14	supprimer
2	Le catéchisme du mille maître	On ne s'opère pas par Fax 2000. Ce sera bien pire...	1999-10-20	supprimer
3	Vae victis!	Le maître et des voisins a entre eux, par de suite de suite le maître.	1999-06-26	supprimer
4	De maître et toujours de maître	L'opérateur et dans l'air de temps et la langue de l'opérateur pour de suite victis.	1999-06-17	supprimer
5	The Thierry Marceau Interne ! Page Page	Un éditeur à l'Homme de l'Année 1998.	1999-07-19	supprimer
6	Des meurtres avec une de France - le SPIP	A la suite de l'article sur le meurtre, le SPIP a été créé et est en ligne. Les meurtres ont été publiés sur le site de France.	1999-07-12	supprimer

¹ <http://listes.rezo.net/mailman/listinfo/spip>
² <http://listes.rezo.net/archives/spip/>
³ <http://www.menteur.com/>
⁴ <http://www.scarabee.com/>
⁵ <http://www.ornitho.org>

L'interface privée du Scarabée

Juste avant l'été 2000, ARNO* réalise le site des éditions Vuibert⁶, dont le principe est un système PHP/mysql, une interface privée où les personnels de l'entreprise enrichissent eux-mêmes le site, et un système de droits qui permet à certaines personnes (les «administrateurs» du site) de valider certains éléments avant leur mise en ligne. À partir des éléments fournis par les éditeurs, le système permet en outre de fabriquer des documents de gestion interne à partir de fichiers HTML contenant des éléments conditionnels, auxquels on peut attribuer des filtres, ce qui deviendra un des principes des squelettes de SPIP.



L'interface de gestion du site Vuibert

Parallèlement, avec Fil, il crée un système de publication simplifié pour gérer les «Cahiers documentaires» du Monde diplomatique⁷ pour lesquels, notamment, le système de raccourcis typographiques est plus développé. La fonction qui gère la typographie française et les raccourcis est diffusée à partir de la fin mai 2000 sous la forme d'un fichier intitulé «sphilib.inc».

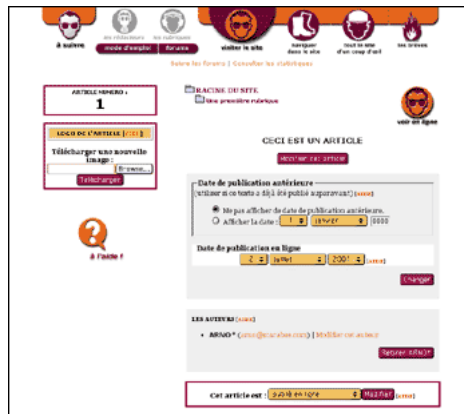
Durant l'été 2000, le Minirézo décide de relancer son site uZine, sous la forme d'un site dynamique auquel n'importe qui pourrait participer. ARNO* développe un outil à base de PHP et de MySQL, dont la particularité est la simplicité de l'interface de gestion du site. L'ensemble est affreusement mal programmé, mais fonctionne ...

En septembre 2000, le lancement d'uZine 2 avec ce système valide l'idée qu'on peut utiliser une interface graphique très simple pour gérer un site relativement complexe, afin que n'importe qui puisse y participer sans connaissances techniques.

L'objectif de pouvoir réaliser n'importe quel type de site est écarté (le système correspond, à la base, aux besoins d'uZine 2), mais il est décidé que le système allait servir de base au système SPIP lui-même.

⁶<http://www.vuibert.com>

⁷<http://www.monde-diplomatique.fr/>



L'interface d'origine de SPIP

Cette première version comprend un correcteur orthographique basé sur le dictionnaire des mots communs de l'ABU⁸. Ce correcteur sera finalement abandonné : impossible à diffuser à cause de la taille du dictionnaire (plusieurs mégaoctets, plus de 300 000 mots) et, appliqué à un texte très long, il mettait n'importe quel serveur sur les genoux ... On trouve également une exportation automatique de l'intégralité d'une rubrique vers un autre site (fonctionnalité elle aussi abandonnée, et remplacée par un système de syndication de contenus); ainsi, les sites Insurgence et Radiophare⁹ proposaient-ils des textes tirés de rubriques d'uZine, récupérés automatiquement et reformatés selon leur propre interface graphique.

Antoine participe à partir de ce moment au développement de SPIP, Fil soutient le projet, et le développement continue dans l'optique d'offrir un système complet sous licence libre.

Une procédure d'installation automatisée est intégrée, le principe des squelettes permettant de réaliser des interfaces graphiques sans utiliser PHP est élaboré, un système de cache est installé, et surtout des modifications importantes du système d'authentification des rédacteurs permettent de faire fonctionner SPIP chez un grand nombre d'hébergeurs.

Quelques semaines avant le lancement officiel de SPIP, les sites uZine, du Monde diplomatique¹⁰ et de Vacarme¹¹ testent le système en conditions réelles d'utilisation, et permettent de valider le système de cache, la gestion d'une grande quantité d'information et le principe de l'interface de gestion d'un site.

Le 1er juillet 2001 : **SPIP 1.0** est lancé officiellement. L'intervalle de temps important entre l'ouverture d'uZine 2 (avec une préversion de SPIP) et le lancement de SPIP est dû à plusieurs éléments :

- ▷ développer de nouvelles fonctionnalités, stabiliser le produit (éliminer des bugs), nettoyer une partie du code (qui était particulièrement sale), assurer une plus grande compatibilité avec de nombreux hébergeurs ;
- ▷ redessiner une interface graphique pour l'espace privé, qui permette d'inclure et hiérarchiser les 200 000 nouvelles fonctionnalités ajoutées au produit initial ;
- ▷ tester les choix du système sur plusieurs sites aux contenus et aux fréquentations très différents ;
- ▷ rédiger une documentation complète du système (une véritable plaie !).

Octobre 2001 : **SPIP 1.2** (il n'y a pas eu de version officielle 1.1). Le processus de développement a évolué : en effet, depuis le lancement officiel, il y a d'autres utilisateurs de SPIP que ses développeurs ! Ainsi, les nouvelles fonctionnalités répondent beaucoup plus aux besoins réels exprimés par les webmasters (et non plus aux besoins d'uZine), plusieurs informaticiens compétents apportent ponctuellement des solutions pour le développement, et le débogage se fait quasiment en temps réel.

⁸<http://abu.cnam.fr>

⁹<http://www.radiophare.net>

¹⁰<http://www.monde-diplomatique.fr/>

¹¹<http://www.vacarme.eu.org>

Janvier 2002 : **SPIP 1.3**. Le développement continue, avec son lot de nouveautés (section 3.5 page 35. Un changement dans le développement : de nombreux utilisateurs sont désormais très compétents avec SPIP. De ce fait :

- ▷ les réponses aux questions des utilisateurs débutants sont largement prises en charge par d'autres utilisateurs, ce qui libère énormément de temps pour le développement du système (le lancement officiel de SPIP avait provoqué une charge de travail énorme pour les développeurs, dans l'explication de l'utilisation du système) ;
- ▷ ces utilisateurs devenant plus compétents, les nouvelles fonctionnalités de SPIP sont moins spectaculaires, et concernent largement une utilisation poussée du système.



L'interface de SPIP 1.3

3.3 SPIP 1.0.3, 1.0.4, 1.0.5

18 juillet 2001 par l'équipe de SPIP

1.0.3. Cette version corrige quelques petits bugs dans le moteur typographique, et accélère notablement l'usage du cache.

1.0.4. Dans la foulée, nous sortons la version 1.0.4, qui corrige un bug de la précédente, qui ne concernait que certains serveurs. Si vous aviez installé la version 1.0.3 et rencontré une erreur du type «Parse error on line 394», installez cette version 1.0.4.

1.0.5. Nouvelle version de SPIP :

- ▷ La gestion des sites syndiqués est améliorée. On peut désormais désactiver la syndication dans la page de configuration (pour alléger l'interface).
- ▷ Les brèves proposent également des forums internes, utilisables pour la validation (comme pour les articles).
- ▷ Une nouvelle page permet de suivre/gérer les forums publics, selon leurs threads, pour chaque article individuel. Accès via la page de chaque article, et via la page générale de suivi des forums.
- ▷ La fonctionnalité «doublons» dans les squelettes est améliorée. Il y a désormais des doublons pour les auteurs, les brèves, les mots-cles, les articles, les rubriques et les forums.
- ▷ Dans le site public, la rapidité est accrue sur certaines grosses pages affichant beaucoup de titres d'articles (notamment plan du site, rubriques, sommaire). En effet, le texte d'un article n'est plus récupéré que quand il est réellement affiché.
- ▷ L'envoi de mail marche désormais sur online.fr.
- ▷ L'envoi automatique des messages de forum aux auteurs des articles marche de nouveau.

3.4 SPIP 1.2, 1.2.1

18 octobre 2001 par l'équipe de SPIP

La version 1.2 de SPIP est disponible.

Parmi les nouvelles fonctionnalités, les plus importantes sont :

- ▷ **Administrateurs à accès restreint** Cette fonction très demandée (mais à l'utilité toute relative . . .) permet de créer des administrateurs aux responsabilités limitées à une ou plusieurs rubriques du site - et ainsi de déléguer une partie de la gestion, sans pour autant donner «tout pouvoir» ; pour les rubriques qui ne lui sont pas attribuées, cet administrateur a les mêmes droits qu'un rédacteur. Certaines pages réservées aux administrateurs, qui concernent la gestion globale du site (telles que «Configuration précise», ou la gestion des rédacteurs), ne sont pas accessibles aux *administrateurs à accès restreint*.
- ▷ **Messagerie interne** Un système de messagerie interne complète les outils destinés à faciliter le travail coopératif sur un site SPIP. Les rédacteurs peuvent échanger des messages (à un ou plusieurs destinataires), chaque message ouvrant un forum privé entre ses destinataires. Il est également possible de noter des «pense-bête». Un calendrier affiche les rendez-vous importants et récapitule l'activité éditoriale du site selon une nouvelle présentation graphique.
La messagerie interne (que l'on peut activer ou désactiver pour l'ensemble du site, mais à laquelle chaque rédacteur peut décider individuellement de ne pas participer) est complétée d'une liste des rédacteurs connectés à l'espace privé. Chaque rédacteur connecté peut être ainsi contacté par l'envoi d'un message, simplement en cliquant sur le logo associé à son nom. Chaque rédacteur peut décider, individuellement, de ne pas apparaître dans la liste des rédacteurs connectés.
- ▷ **Nouveau système de cache** L'espace public (toujours calculé à partir des squelettes) bénéficie d'un nouveau moteur (nom de code : «Pantagruel») et d'un nouveau système de cache. Le système de cache est désormais décomposé en deux opérations : l'analyse des squelettes (création d'un fichier PHP pré-interprété), puis l'intégration des données de la base de données pour chaque page et sauvegarde d'un fichier cache indépendant pour chaque page.
Les gains de vitesse (et de charge sur le serveur) apportés par ce nouveau moteur *sont très importants*.
- ▷ **Flux compressé** SPIP utilise désormais, lorsque le serveur l'autorise et lorsque le navigateur du visiteur est compatible, la compression des données de PHP4 : les données échangées entre le client et le serveur sont compactées, réduisant ainsi de manière très importante la bande passante utilisée et les temps de chargement.
- ▷ **Feuilles de style et variables PHP** pour modifier un peu plus les squelettes Certaines informations générées à partir de la base de données et des squelettes sont désormais complétées d'indications de feuilles de style. Le webmestre a donc la possibilité, s'il le souhaite, de pousser la personnalisation de sa mise en page. Par exemple : des classes CSS différentes sont attribuées aux liens hypertextes à l'intérieur du site et vers l'extérieur, ce qui permet de les différencier graphiquement.
- ▷ **Mots-clés sur les brèves** Des mots-clés peuvent être associés aux brèves.
- ▷ **Moteur de recherche interne** Un moteur de recherche interne permet d'effectuer des recherches sur les titres des articles et des brèves.
- ▷ **Syndication RSS1.0** La syndication des sites (récupération de fichiers *backend*) par SPIP est désormais compatible avec les fichiers RSS 1.0 (jusque là, seuls les RSS 0.9x étaient compatibles).
- ▷ **Critères négatifs dans les boucles des squelettes** Il est possible d'utiliser des critères d'exclusion dans les critères de sélection des boucles. Par exemple de choisir les articles dont le surtitre n'est pas égal à «Edito», les rubriques autres que la rubrique 6 . . .

◇ ◇ ◇

Les utilisateurs trouveront encore une foule de petites modifications graphiques (la plupart discrètes), il y évidemment de nombreuses corrections de bugs, et de nombreuses fonctions ont reçu des optimisations permettant de gagner plus ou moins de temps (très variables selon les sites, les textes . . .).



Une version de SPIP, 1.2.1, est disponible en ligne.

Elle corrige un bug apparu chez *certain*s hébergeurs (notamment Altern), pour des sites ayant effectué la mise-à-jour de la version 1.0.6 à la 1.2. Ce bug se manifeste par la disparition des messages des forums lorsque la messagerie interne de SPIP est activée. Ce problème ne concerne donc que quelques sites.

Si vous avez rencontré ce problème lors d'une mise à jour de votre site, cette version 1.2.1 corrige le problème.

Si l'installation de la version 1.2 n'a pas provoqué de difficultés avec les messages des forums, vous n'avez pas besoin d'effectuer cette mise-à-jour.

3.5 SPIP 1.3

12 janvier 2002 par l'équipe de SPIP

La version 1.3 de SPIP contient de très nombreuses modifications et nouvelles fonctionnalités. Les nouveautés peuvent sembler peu spectaculaires (la version 1.2, notamment, proposait des nouveautés beaucoup plus visibles), mais elles sont très importantes. La plus visible est sans doute le nouveau système de référencement de sites. Les nouvelles possibilités offertes par le langage de boucles concernent essentiellement les webmasters qui savent créer leur propre interface graphique avec les squelettes de SPIP ; fonctionnalités discrètes, mais qui permettent de réaliser des sites à la navigation beaucoup plus complète qu'auparavant.

On trouvera le détail de ces nouvelles modifications dans l'«Aide en ligne» de l'espace privé de son site, dans la documentation de SPIP (sur uZine¹²), et dans le nouveau tutorial sur l'utilisation avancée des boucles (section 11 page 169).

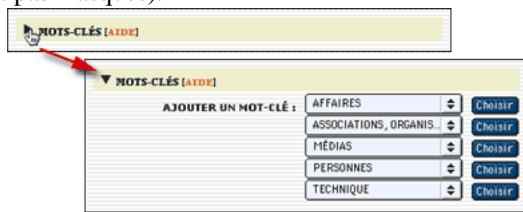
▷ **Système de référencement de sites.** Le précédent système de *syndication* de sites Web est entièrement refondu. Il est désormais possible de référencer n'importe quel site Web ; pour chaque site référencé, on peut indiquer de manière optionnelle une syndication de contenu (récupérer automatiquement la liste des derniers articles publiés sur un site). De plus, pour chaque site référencé, on peut installer un logo, et lui attribuer des mots-clés.

Les sites référencés peuvent être proposés par les administrateurs, les rédacteurs ou les visiteurs du site public (selon réglage dans la «Configuration précise» du site). Un forum est attribué à chaque site pour discuter dans l'espace privé du référencement (ou non) d'un site ; il est également possible d'attribuer un forum public à chaque site référencé.

Le fonctionnement de la syndication de contenu est affiné : les auteurs et le descriptif de l'article sont récupérés (si le site syndiqué les indique) ; il est possible de bloquer un article syndiqué précis sans bloquer l'intégralité de la syndication. Lorsqu'un site syndiqué n'est plus accessible, ou son fichier backend inutilisable, la syndication est désactivée et les administrateurs se voient signaler le problème (ce qui évite les blocages d'une rubrique contenant un site syndiqué défaillant).

Cette nouvelle fonctionnalité est sans doute la modification la plus visible de cette version 1.3.

▷ **Éléments dépliables.** Dans l'espace privé, afin de limiter l'encombrement de certaines pages, et de privilégier la présentation de certaines informations, de nombreux éléments apparaissent masqués, mais affichages d'un simple clic sur un triangle noir. Cette fonctionnalité n'est disponible qu'avec Mozilla et MSIE (pour les autres navigateurs, les éléments ne sont pas masqués).



¹²<http://www.uzine.net/rubrique91.html>

- ▷ La présentation des mots-clés se fait désormais selon plusieurs menus déroulants, par *groupes* de mots-clés.
- ▷ **Date des brèves.** Il est désormais possible de modifier manuellement la date d'une brève (auparavant, la date était fixée automatiquement lors de la validation et il n'était pas possible de la modifier).
- ▷ **Dates imprécises.** Pour les articles et les brèves, il est possible d'indiquer une date imprécise. C'est-à-dire une date sans jour, ou même sans mois. On peut ainsi indiquer une date du genre «5 mai 2001» ou «mai 2001» ou «2001».
- ▷ **Upload d'images par FTP.** Pour contourner la limitation imposées par certains hébergeurs qui interdisent l'installation d'images par l'intermédiaire d'un formulaire Web, il est possible d'installer les fichiers des images par FTP dans un dossier «/ecriture/upload». Ces fichiers seront alors proposés, dans l'espace privé, dans un menu déroulant remplaçant l'interface de téléchargement habituelle.
- ▷ **Forum interne des administrateurs.** En plus de l'habituel forum interne accessible à tous les rédacteurs, apparition d'un forum interne réservé aux administrateurs.
- ▷ **Nouveau raccourci <cadre>.** Pour afficher un morceau de code dans un article, un nouveau raccourci : <cadre>...</cadre>. Le texte à l'intérieur de ces balises sera affiché dans une fenêtre de formulaire, ce qui facilite le copier-coller par le lecteur.
Cela donne par exemple :

```
<html>
  <head>
    <title>Le titre</title>
  </head>
  <body>

    <h1>Ma page</h1>

    Blah blah blah...

  </body>
</html>
```

- ▷ Les déplacements de rubriques gèrent de manière plus cohérente le déplacement des brèves qui s'y trouvent.
- ▷ Lorsque l'on peut supprimer une rubrique (parce que cette rubrique est vide - on ne peut pas effacer une rubrique contenant des éléments publiés, proposés ou en cours de rédaction), un bouton «Supprimer cette rubrique» apparaît sur la page de cette rubrique (auparavant, il fallait passer par la page «Afficher tout le site»).
- ▷ Une rubrique qui ne contient pas d'articles publiés, mais seulement des brèves ou des sites référencés, est désormais accessible sur le site public (auparavant, il fallait au moins un article publié).

◇ ◇ ◇

Les modifications qui suivent concernent le site public. Elles concernent donc, pour l'essentiel, le système de boucles qui permet de créer les squelettes de l'interface du site public.

- ▷ **Modification backend.** Le fichier de backend d'un site sous SPIP peut désormais n'afficher que les articles d'un secteur. «backend.php3» affichera les derniers articles de l'ensemble du site ; «backend.php3?id_rubrique=3» n'affichera que les derniers articles d'un secteur (où «id_rubrique=3» indique le numéro du secteur).
- ▷ **Backend pour les brèves.** On peut désormais appeler un fichier backend contenant les dernières brèves publiées sur le site : «backend-brevs.php3». Possibilité également de restreindre l'affichage à un secteur.
- ▷ **Sélection d'éléments selon un mot-clé ou un groupe de mots.** Il est désormais possible de sélectionner des articles, des brèves ou des sites référencés en fonction d'un nom de mot-clé, ou d'un nom de groupe de mot-clé. Par exemple :

```
<BOUCLE_importants (ARTICLES) {id_rubrique} {titre_mot=importants}>
```

sélectionne les articles de la rubrique courante, liés au mot-clé «importants».

```
<BOUCLE_importantes(ARTICLES){id_rubrique}{type_mot=note}>
```

sélectionne les articles de la rubrique courante, liés à des mots-clés du groupe de mots «Note».

Limitation : il n'est pas possible de sélectionner selon plusieurs critères «titre_mot» dans une même boucle. Par exemple, on ne peut pas récupérer en une seule boucle les articles associés aux mots «importants» et «résumé». Pour cela, il faut imbriquer deux boucles successives.

- ▷ **Date des rubriques.** Il est désormais possible de trier les rubriques {par date}. La «date» des rubriques est calculée automatiquement : il s'agit de la date du dernier article publié dans cette rubrique ou ses sous-rubriques. Cela permet ainsi d'afficher les rubriques en fonction des derniers articles publiés dans ces rubriques.
- ▷ **Nouveau critère d'âge.** Il était possible d'afficher les rubriques en fonction de leur «âge» par rapport à la date actuelle. Par exemple, les articles sélectionnés selon le critère {age < 30} étaient les articles publiés depuis moins de 30 jours. Un nouveau critère apparaît : «age_relatif», qui calcule l'âge par rapport à une date «courante» (par exemple la date d'un article, ou même une date passée dans l'URL de la page. Ce nouveau critère permet par exemple :
 - ★ d'afficher les articles publiés avant ou après un autre article ;
 - ★ de créer des affichages sous forme de calendrier (tous les articles publiés en mai 2002, par exemple).
- ▷ **Nouveaux squelettes.** Les squelettes fournis en standard avec SPIP sont nommés «article-dist.html», «rubrique-dist.html» ... Lorsque l'on réalise ses squelettes personnels, on les nomme «article.html», «rubrique.html» ... ; de cette façon, lors d'une mise à jour de SPIP, les squelettes personnels ne sont pas écrasés.
- ▷ **Squelettes pour une seule rubrique.** On peut désormais créer un squelette qui ne s'applique qu'à une seule et unique rubrique (et non à ses sous-rubriques). Par exemple : «article=60.html» s'applique à la rubrique 60, mais pas à ses sous-rubriques. Pour résumer la nouvelle façon de nommer les squelettes :
 - ★ «article=60.html» s'applique aux articles de la rubrique 60, mais pas aux articles de ses sous-rubriques ;
 - ★ «article-60.html» s'applique à tous les articles de la rubrique 60 et aux sous-rubriques de la rubrique 60 ;
 - ★ «article.html» est le squelette personnalisé qui s'applique à l'intégralité du site (si le fichier «article.html» existe, «article-dist.html» n'est plus du tout utilisé) ;
 - ★ «article-dist.html» est le squelette fourni par défaut avec SPIP.
- ▷ **Classement numéroté.** Pour forcer l'ordre d'affichage d'éléments tels que les rubriques ou les articles, il est très simple de faire précéder leur titre d'un numéro d'ordre (par exemple : «1. Mon premier article», «2. Mon deuxième article» ...).

Pour forcer l'affichage selon le numéro qui précède le titre, on peut utiliser le critère {par num titre}. Si l'on se contente d'utiliser {par titre}, on obtient un classement du type : 1, 10, 11, 2, 3 ...

De plus, pour ne pas afficher ce numéro, on utilise le filtre «supprimer_numero». Dans les squelettes concernés, il suffit d'afficher le titre ainsi :

```
[ (#TITRE|supprimer_numero) ]
```

◇ ◇ ◇

Une bonne partie de ces nouvelles fonctionnalités concerne les utilisateurs confirmés de SPIP, c'est-à-dire les webmasters qui modifient eux-mêmes les squelettes de leur site.

Afin de montrer comment utiliser ces nouvelles fonctions, mais aussi pour réaliser des sites à la navigation plus complète (et complexe) que les sites réalisés avec les squelettes standards, un nouveau tutorial explique l'utilisation avancée des boucles et des mots-clés (section 11 page 169). Ce tutorial exploite notamment plusieurs des nouvelles fonctionnalités de la version 1.3 (par exemple l'appel d'articles en fonction d'un mot-clé).

◇ ◇ ◇

Enfin il y a une multitude de petites modifications d'interface, souvent très discrètes, l'optimisation de certaines

parties du code, et la correction de nombreux bugs.

3.6 SPIP 1.4

2 septembre 2002 par l'équipe de SPIP

La version 1.4 de SPIP est la version officielle.

Cette nouvelle version de SPIP, en développement depuis six mois, propose des changements très importants. Certains concernent tous les utilisateurs (notamment les rédacteurs), d'autres sont destinés à faciliter le travail des webmasters qui créent leurs propres squelettes, et certains sont très techniques (sécurité, développement ...).

Pour télécharger cette version, rendez-vous à l'adresse <http://rezo.net/spip-dev/DISTRIB/>

3.6.1 Mise-à-jour

La mise à jour peut se faire depuis n'importe quelle version antérieure de SPIP.

Si vous utilisez habituellement l'installation automatique, il vous suffit d'appeler le fichier `spip_loader.php3` depuis votre navigateur comme pour les versions précédentes, puis de suivre les instructions affichées pour la mise à niveau de la base de données.

Si vous préférez l'installation manuelle à partir d'une archive téléchargée depuis l'URL ci-dessus, il vous faut :

1. décompresser l'archive que vous aurez choisie (il y a trois formats : zip, sit et tgz, mais ce sont les mêmes fichiers) ;
2. envoyer les fichiers par FTP sur votre site (en écrasant éventuellement les fichiers précédents : pensez à faire une sauvegarde de vos squelettes auparavant, en cas de fausse manipulation).
3. vous rendre dans `ecrire/` et suivre les instructions (il vous faudra créer un fichier ou répertoire particulier dans `ecrire/data/` puis laisser SPIP faire la mise à niveau de votre base de données).

Et c'est tout !

Remarques importantes :

- ▷ Vous pouvez auparavant faire une sauvegarde de votre base de données ; sachez toutefois que vous n'aurez normalement pas besoin de cette sauvegarde, SPIP se chargeant tout seul de la mise à niveau de votre base de données. D'autre part, cette sauvegarde ne pourra pas être restaurée sans dommage sur une version plus récente que votre version actuelle ; elle ne doit donc être utilisée qu'en dernier ressort !
- ▷ Surtout, **NE VIDEZ PAS VOTRE BASE DE DONNÉES AVANT DE FAIRE LA MISE À JOUR**. (L'erreur est classique, certains imaginant, à tort, qu'il faut vider la base de données pour ensuite restaurer la sauvegarde).

3.6.2 Problèmes éventuels

- ▷ En cas de soucis d'affichage sur le site public, essayez de vider le cache de votre site pour voir si le problème persiste.
- ▷ Si votre site affiche des messages d'erreur étranges (erreurs PHP, etc.) vérifiez que tous les fichiers ont été correctement transférés via FTP, et qu'aucun des fichiers n'a une taille zéro sur le serveur. Eventuellement, réessayez le transfert en changeant le mode de transfert dans votre logiciel FTP (le mode binaire est normalement préférable).
- ▷ Si vos problèmes persistent, ou pour toute autre question, n'hésitez pas, après avoir cherché dans la documentation, à écrire à spip@rezo.net. N'oubliez pas de mentionner la version de SPIP utilisée et de décrire précisément le problème (URL de la page incriminée, etc.).

3.6.3 Interface

L'interface privée propose une interface graphique radicalement différente des versions précédentes. Elle permet en particulier :

- ▷ de mieux distinguer et hiérarchiser les différentes fonctionnalités de SPIP ;
- ▷ une intégration facilitée des nouvelles fonctionnalités tout en conservant la cohérence de la navigation ;
- ▷ de créer (pour les prochaines versions) des versions non françaises de SPIP.



Impossible de détailler toutes les nouvelles caractéristiques de l'interface, celle-ci étant entièrement nouvelle. Signalons tout de même :

- ▷ le choix entre une interface «normale» (logos et textes), une interface réduite n'affichant que les icônes, et une interface allégée entièrement en mode texte (pour les connexions lentes) ;
- ▷ une plus grande différenciation entre l'interface simplifiée et l'interface complète, facilitant le travail des débutants ;
- ▷ un mode «grand écran» pour les utilisateurs disposant d'écrans à la largeur supérieure ou égale à 1024 pixels ;
- ▷ des «raccourcis» dans l'interface proposant les fonctions les plus utilisées en fonction de la page où l'on se trouve.

3.6.4 Documents joints et documents multimédia

Il s'agit sans doute de la fonction la plus attendue de cette nouvelle version : SPIP permet d'associer des documents de formats multimédia (audio, vidéo, PDF ...) à des articles ou de les installer dans des rubriques.

- ▷ Ces documents peuvent être présentés en tant que documents joints, ou présentés à l'intérieur d'un article (sous la forme d'une vignette dotée d'un lien hypertexte). Pour cela, SPIP propose un nouveau raccourci : `<docxxx|center>`.

Il est possible également d'insérer directement certains documents (vidéo, animations flash ...) à l'intérieur des articles, grâce au nouveau raccourci :

`<embxxx|center>`. Il est possible, pour ceux qui désirent un contrôle plus précis du comportement de ces documents, de compléter ce raccourci des paramètres propres à ces formats, par exemple :

```
<embxxx|center|autostart=true|quality=high>
```

- ▷ Pour gérer les documents qui ne sont pas directement insérés dans le texte des articles, un nouveau format de boucles apparaît : `(DOCUMENTS)`.
- ▷ Au passage, grâce à l'introduction de ces documents, les images profitent de certaines améliorations : possibilité de leur donner un titre et d'indiquer un descriptif. Ces informations seront affichées dans les articles grâce au raccourci : `<docxxx|center>`.

De plus, on peut désormais insérer des images dans les brèves.

- ▷ Sur *certaines serveurs*, SPIP facilite grandement la création automatisée de portfolio (collections d'images présentées sous forme de vignettes cliquables), avec création automatique de vignettes de prévisualisation.

3.6.5 Mots-clés

Le système de mots-clés évolue largement, afin d'offrir une plus grande précision des affichages ; cette nouveauté est en particulier conçue pour les webmasters qui gèrent beaucoup de mots-clés sur leur site.

- ▷ Mots-clés sur les rubriques.
- ▷ Les mots-clés appartiennent forcément à un groupe de mots.
- ▷ Chaque groupe peut être «lié» aux articles, et/ou brèves, et/ou rubriques, et/ou sites syndiqués. De plus, on peut décider que certains groupes sont réservés aux admins et/ou aux rédacteurs. On peut également décider que certains groupes de mots sont accessibles à partir des forums publics, et même avec des icônes.
- ▷ Possibilité d'entrer plusieurs mots-clés d'un seul coup dans les cases de formulaire, séparés par des virgules ou des points-virgule. À partir de 4 mots-clés associés à un article, un bouton «retirer tous les mots» apparaît.

3.6.6 Forums

- ▷ Prévisualisation des messages des forums publics avant de poster.
- ▷ La modération des forums se décide désormais article par article, avec une option par défaut (qui s'applique également aux forums de rubrique, de brèves, etc.. de manière indifférenciée pour le coup).
- ▷ Prévisualisation des messages des forums privés avant de poster.

3.6.7 Moteur de recherche

- ▷ Quand on demande un article depuis une page recherche, coloration des mots de la recherche dans le texte de l'article.
- ▷ La recherche dans l'espace privé utilise désormais, en plus des «titres et numéros», la recherche en texte intégral si elle est disponible.

3.6.8 Multi-base

Il est possible de gérer plusieurs sites sous SPIP dans une même base MySQL : configuration manuelle dans `ecrire/inc_version.php3`, tout au début (mettre un préfixe différent pour chaque installation).

Attention : cette fonctionnalité est réservée aux utilisateurs confirmés.

3.6.9 Sites syndiqués

- ▷ Possibilité de passer certains sites syndiqués en «modération a priori», de manière à valider les articles syndiqués un par un. Évidemment, cela enlève beaucoup de charme à la syndication de sites, censée faire vivre votre propre site en l'absence du webmaster
- ▷ Possibilité d'utilisation d'un proxy HTTP pour syndiquer les sites (réservé aux utilisateurs confirmés).

3.6.10 Squelettes

- ▷ Inclusion de squelettes à l'intérieur d'un autre squelette.
Pour inclure un squelette `machin.php3` en lui passant le numéro de rubrique, faire par exemple :

```
<INCLUDE(machin.php3){id_rubrique}>
```

- ▷ Modification du comportement de `#INTRODUCTION` des articles :
- ▷ s'il y a un descriptif, c'est ce descriptif qui est directement utilisé (tel quel, avec propre) ;
- ▷ s'il n'y a pas de descriptif, comportement habituel (resume chapeau + texte).
- ▷ Critère `{branche}`, qui permet de récupérer toutes les sous-rubriques d'une rubrique (expérimental).
- ▷ Nouvelles balises `#LOGO_RUBRIQUE_SURVOL` et `#LOGO_RUBRIQUE_NORMAL`, utile pour une maquette ou le logo de la rubrique courante est toujours affichée en survol.
(Principe similaire aux `#LOGO_ARTICLE_NORMAL` et `#LOGO_ARTICLE_SURVOL` qui existent déjà dans les versions précédentes.)
- ▷ Nouvelle balise `#LOGO_BREVE_RUBRIQUE`, qui affiche le logo de la brève ou, à défaut, celui de la rubrique contenant la brève.
(Principe similaire à `#LOGO_ARTICLE_RUBRIQUE`.)
- ▷ Nouvelle balise `#FORMULAIRE_ECRIRE_AUTEUR` qui affiche un formulaire permettant d'écrire à un auteur, sans jamais faire apparaître son adresse email sur le site public.
- ▷ Gestion des filtres sur `LOGO_xxx` et sur `FORMULAIRE_RECHERCHE`, la syntaxe `[({#TOTO|filtre})]` assure que `|filtre` est un filtre.
- ▷ Aide au débogage des squelettes en cas d'erreur MySQL
- ▷ Variables de présentation du type
`$debut_intertitre`... réglables soit de manière globale dans `mes_fonctions.php3`, soit de manière plus fine dans `article.php3`, `rubrique.php3`, etc.

3.6.11 Raccourcis

De nouveaux raccourcis complètent la gestion des listes :

Raccourci	Fonction
- (tiret espace)	puce spip standard
_ (underscore espace)	 saut de ligne sans puce
-* , -** Listes hiérarchiques
-# , -## Listes numériques

3.6.12 Sécurité

- ▷ Nouveau système d'authentification des visiteurs (dans l'espace privé, mais aussi dans l'espace public), à base de cookies. Le système est conçu pour assurer un plus grand niveau de sécurité, mais aussi pour offrir une plus grande compatibilité avec les différents hébergeurs.
Lors de la mise à jour de spip, il est conseillé de vérifier qu'on n'utilise pas la méthode « .htaccess » (supprimer le fichier écrire/.htaccess s'il existe). Si on ne veut pas de cookies (ou si le navigateur ne les aime pas), le système propose de basculer sur une authentification http à l'ancienne.
- ▷ Le système propose deux niveaux de sécurité : l'un est plus adopté aux utilisateurs qui «bidouillent» avec plusieurs navigateurs en même temps, ou plus ordinateurs simultanément, et un système nettement plus strict, qui interdit toutes connexions simultanées et offre un niveau de sécurité plus élevé.
- ▷ Un bouton «Se déconnecter» est proposé en permanence, son utilisation est notamment conseillée aux utilisateurs «mobiles» (connexion depuis un ordinateur qu'ils ne sont pas seuls à utiliser).
- ▷ Pour les utilisateurs qui ont oublié leur mot de passe, le système gère désormais la possibilité de récupérer un nouveau mot de passe, grâce à un échange d'email.
- ▷ ATTENTION : On ne crée plus les fichiers .htpasswd et .htpasswd-admin s'ils n'existent pas déjà (sécurité).

3.6.13 Statistiques

Nouveau système de statistiques, nettement plus fiable que la version précédente (la version précédente était destinée à fournir une «indication» des visites, et non une information précise).

Le système se compose de deux parties : visites (relativement léger) et referers (plus lourd). La connaissance des referers permet de plus la mise à jour quotidienne d'un «pourcentage de popularité» par article. Ainsi l'article le plus «populaire» est à 100%. (Pour plus de détails lire «La popularité des articles» (section 8.24 page 140.)

La nouveauté la plus visible est la présence de graphiques affichant l'évolution des visites jour après jour pour l'ensemble du site et pour chaque article publié.

3.6.14 Divers

- ▷ Exportation de la base en plusieurs étapes si le serveur interrompt la sauvegarde avant la fin de l'exportation complète.
- ▷ Amélioration de la compatibilité avec les différentes configurations de PHP : les tags PHP passent en <?php (compatibilité avec l'option PHP «short_open_tags»).
- ▷ Ajout d'une page écrire/admin_repair.php3 permettant de mettre en œuvre le système d'auto-réparation de MySQL suite à un plantage (crash disque, etc.) [Versions de MySQL à partir de 3.23.14]

◇ ◇ ◇

Quelques incompatibilités et petits bugs ont été corrigés depuis la sortie de la version 1.4 - après les avoir corrigés, nous avons publié une 1.4.2 dont les caractéristiques principales sont identiques à celles mentionnées dans cet article. Pour plus de détails voir les annonces récentes¹³.

¹³<http://listes.rezo.net/archives/spip-ann/>

Deuxième partie

Guide de l'utilisateur



Guide

de l'utilisateur

Chapitre 4

Guide de l'utilisateur



Guide

de l'utilisateur

4.1 FAQ de l'administrateur SPIP

18 juillet 2001 par l'équipe de SPIP

4.1.1 Débuter

1. **J'ai l'impression que je n'ai pas accès à toutes les fonctionnalités depuis l'interface. Pourtant, je suis bien administrateur.**

Vérifiez que vous êtes bien en interface complète. C'est affiché dans le tableau de bord en haut de chaque page de l'espace privé. Pour rester en interface complète, votre navigateur doit accepter les cookies. En interface simplifiée, beaucoup de fonctionnalités sont masquées afin de rendre l'utilisation plus simple pour les néophytes.

4.1.2 Supprimer

1. **Comment supprimer un article ?**

On ne peut pas supprimer directement un article, mais on peut le mettre «à la poubelle» dans le menu de sélection du statut de l'article. Les articles à la poubelle sont automatiquement effacés au bout de 24 heures ; cela vous laisse un temps de répit en cas de mauvaise manipulation.

2. **Comment supprimer une brève ?**

De la même manière que pour les articles (cf. ci-dessus), on ne peut supprimer directement une brève ; mais les brèves refusées sont effacées automatiquement au bout du même délai (24 heures).

3. **Comment supprimer une rubrique ?**

Pour pouvoir être supprimée, une rubrique doit être vide (i.e. ne contenir ni article - sauf à la poubelle - ni sous-rubrique). Si cette condition est vérifiée, la rubrique peut être supprimée dans «afficher tout le site», en dépliant l'arborescence jusqu'à rendre visible la rubrique, et en cliquant sur le lien «supprimer» à côté de celle-ci.

4.1.3 Les différents types de navigation

1. **Quel est l'intérêt des mots-clés ?**

Dans la plupart des sites, la navigation la plus évidente sera celle imposée par les rubriques : on navigue dans le site en se repérant dans la classification arborescente mise en place grâce aux rubriques. Les mots-clés permettent d'avoir un autre niveau de navigation, transversal et indépendant. Chaque article peut se voir associer plusieurs mots-clés. Ainsi dans le site public, on peut afficher la liste des mots-clés associés à un article ; puis la liste des autres articles associés à chacun de ces mots. La navigation définie ne décrit pas un arbre, elle est beaucoup plus lâche, horizontale et permet de se déplacer de proche en proche.

Pour résumer les différences fonctionnelles :

- ▷ tout article est dans une rubrique et une seule ;
- ▷ les rubriques peuvent être imbriquées à l'infini (sous-rubriques, etc.) ;
- ▷ un nombre arbitraire (zéro, un, plusieurs) de mots-clés peut être associé à chaque article, et de même chaque mot-clé peut être associé à un nombre arbitraire (zéro, une, plusieurs) de rubriques ;
- ▷ les mots-clés ne peuvent pas être imbriqués les uns dans les autres.

Pour un exemple opérationnel d'utilisation des mots-clés, on pourra consulter le site du *Monde diplomatique*¹. Les rubriques y définissent la classification rigide du site (dossiers, cahier, cartes, archives classées par date ...). Les mots-clés permettent de lier les articles traitant d'un même thème ; ils y sont classés en deux groupes, «sujets» et «pays».

¹<http://www.monde-diplomatique.fr>

2. Je ne comprends pas la différence entre mots-clés et moteur de recherche. Est-ce que c'est la même chose ?

Les mots-clés et le moteur de recherche sont deux choses fondamentalement différentes dans SPIP (d'ailleurs, on peut désactiver le moteur de recherche tout en conservant les mots-clés, et vice-versa).

Avec les mots-clés, ce sont les administrateurs du site qui définissent les relations entre les articles en liant des mots-clés à ces articles. Ces mots-clés peuvent ensuite être affichés explicitement dans le site public, ainsi que la liste des articles associés à chacun d'entre eux ; sans cet affichage, ils servent à très peu de choses. Cela donne une navigation de proche en proche, indépendante des rubriques, mais toujours fixée par les administrateurs.

Le moteur de recherche effectue des recherches à la demande du visiteur sur n'importe quel terme ou groupe de termes. Les mots-clés sont certes inclus dans les champs utilisés par la recherche, mais au même titre que les différents champs des articles (chapo, texte, etc.). Le moteur de recherche sert ainsi à trouver des informations sans avoir à passer par les navigations (rubriques, mots-clés) définies par les administrateurs, qui ne sauraient penser à tous les termes susceptibles d'être recherchés dans un site.

Pour des informations plus techniques sur le moteur de recherche, voir l'article (section 7.4 page 88) y consacré dans le guide du webmestre SPIP.

3. Les mots-clés peuvent-ils ralentir mon site ?

Non.

4. Le moteur de recherche peut-il ralentir mon site ?

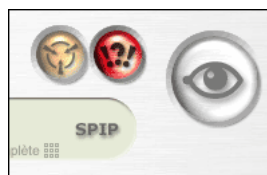
C'est possible (section 7.4 page 88).

4.2 L'aide en ligne

1er juin 2001 par l'équipe de SPIP

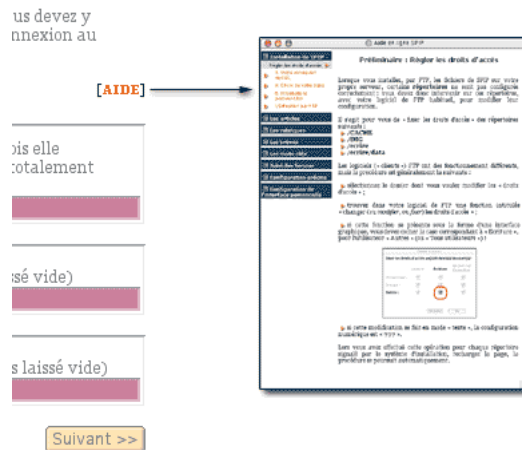
SPIP intègre une *aide en ligne* complète, que vous pouvez retrouver, dans l'espace privé, en cliquant sur le mot [AIDE] disponible un peu partout (*javascript obligatoire*).

Le bouton principal d'ouverture de l'aide en ligne se situe dans la barre de navigation du site privé, en haut et à droite : il s'agit du bouton rouge marqué d'un point d'interrogation (vous ne pouvez pas le manquer) :



Nous vous invitons en particulier à lire la page «Articles/Raccourcis typographiques», qui vous indique comment mettre en page vos articles sous SPIP.

Tout l'intérêt de cette aide en ligne réside dans les innombrables petits liens [AIDE] installés dans l'interface de la partie privée : chacun ouvre l'aide en ligne à la rubrique correspondant exactement à l'action en cours. N'hésitez pas à en abuser :



Pour le reste, vous constaterez que l'interface de SPIP est très simple. Et en tant que rédacteur, n'hésitez pas à expérimenter : vous ne pouvez pas réellement faire de dégâts.

4.3 Vous avez ...dist ?

5 février 2002 par Ecran de Bureau

Qu'est-ce que les fichiers «*-dist.html» ?

Que sont les fichiers sommaire-dist, rubrique-dist ou encore article-dist ? en fait ce sont les fichier de la distribution de SPIP..

Comme nous le savons déjà, le système de squelettes est basé sur des fichiers ".html" contenant la présentation graphique du site. Par exemple, «article.html» présente les articles, «rubrique.html» présente les rubriques ...

Or, les développeurs de SPIP ont remarqué que fréquemment, les utilisateurs qui manipulaient leur interface graphique en modifiant ces fichiers «.html» fournis avec SPIP rencontraient des problèmes lors des mises à jour, si ils n'avaient pas pris leur précaution en sauvegardant leur fichiers modifiés ;-).

En effet, en réinstallant tous les nouveaux fichiers, ils écrasaient purement et simplement leurs fichiers modifiés (oubliant de sauvegarder leur modification).

Donc, depuis la version 1.3, vous pouvez constater que les fichiers de squelettes fournis dans la distribution de SPIP sont nommés «article-dist.html», «rubrique-dist.html» ... Si on veut personnaliser ces fichiers, il suffit de les renommer «article.html», «rubrique.html» (sans le dist).

Ainsi, à la prochaine mise-à-jour, on réinstalle des fichiers «rubrique-dist.html», et ses propres fichiers «rubrique.html» ne sont pas écrasés. Une petite amélioration mais qui évite bien des déboires (et je sais de quoi je parle).

A SAVOIR

Pour rappel, voici l'ordre dans lequel sont utilisés les fichiers de squelettes selon leurs noms :

▷ **Un ordre bien établi !!**

rubrique=10.html, si ce fichier existe, il ne s'applique qu'à la rubrique numéro 10, si ce fichier n'existe pas, SPIP regarde si il n'y a pas un fichier **rubrique-10.html**, si ce fichier existe, la rubrique 10 ainsi que ses sous-rubriques l'utilisent, c'est donc récursif. Si ce fichier n'existe pas, SPIP regarde si il n'y a pas un fichier **rubrique.html**², qui s'applique à toutes les rubriques du site, à l'exception de celles concernées éventuellement

²ceci est donc le nom que l'on donne à son fichier de rubrique général, en renommant rubrique-dist.html fourni en standard

par des fichiers indiqués ci-dessus. Si ce fichier n'existe pas, SPIP utilise alors le fichier `rubrique-dist.html` qui est le fichier fourni par défaut. Si vous voulez modifier ce fichier, renommez-le en «rubrique.html», de façon à ne pas écraser vos modifications à la prochaine mise à jour de SPIP.

4.4 Quels sont les éléments gérés par SPIP ?

1er juin 2001 par l'équipe de SPIP

Développé, au départ, pour gérer le site uZine 2, SPIP est naturellement destiné à gérer un site de type *webzine* : à la base, une hiérarchie de rubriques, et des articles installés dans ces rubriques. Le système gère également les forums et des brèves (par exemple : revue de presse ...).

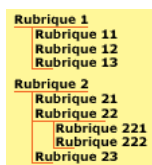
Voyons plus en détail quels sont ces différents éléments pris en charge par SPIP, ce qui nous permettra par ailleurs de clarifier le vocabulaire utilisé par la suite.

4.4.1 La hiérarchie des rubriques

Une *rubrique* est un espace destiné à accueillir des articles, des brèves ... Rien de plus simple : on peut aussi dire *dossier* (comme sur votre ordinateur : vos documents sont rangés dans des dossiers).

Les rubriques peuvent être installées les unes dans les autres, formant ainsi une *hiérarchie*. Une rubrique est soit à l'intérieur d'une autre rubrique, soit elle n'est rattachée à aucune autre et constitue alors un point d'entrée dans le site (nous parlons alors de *tête de rubrique*, ou de *secteur*). L'emboîtement des rubriques les unes dans les autres constitue l'ossature de votre site, puisque c'est autour de cette structure que viendront se greffer les différents éléments de votre site (articles, brèves, sites syndiqués ...)

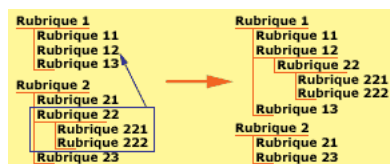
Ci-dessous, les rubriques 1 et 2 sont des secteurs (logiquement, ces rubriques définissent les grands «secteurs» thématiques du site).



Rien de plus simple. Ci-dessus, les rubriques 11, 12 et 13 sont dans la rubrique 1. Les rubriques 221 et 222 sont dans la rubrique 22, elle-même dans la rubrique 2.

On nomme *hiérarchie* le chemin logique qui mène à une rubrique. Ainsi, la hiérarchie de la rubrique 221 est : rubrique 2, puis rubrique 22.

La gestion de la structure hiérarchique est très simple : il suffit d'indiquer dans quelle rubrique se situe chacune des rubriques (cela se règle par un simple menu déroulant).



Le schéma ci-dessus montre comment on déplace une rubrique : lorsqu'une rubrique est déplacée, toutes les sous-rubriques qu'elle contient la 'suivent' vers son nouvel emplacement. Par exemple, si nous déplaçons la rubrique 22 à l'intérieur de la rubrique 12, les rubriques 221 et 222 la suivent (la rubrique 22 aurait tout aussi bien pu être placée comme tête de rubrique, ou à l'intérieur de la rubrique 23 par exemple. En revanche, l'interface graphique vous interdit de placer la rubrique 22 à l'intérieur de la rubrique 221 : sinon on obtiendrait une boucle que le système ne saurait pas gérer.

Signalons ici la première grosse limitation de SPIP : SPIP ne gère qu'une seule structure, et c'est la structure hiérarchique que nous venons de décrire. En particulier :

- ▷ il n'est pas possible qu'une rubrique appartienne à deux rubriques différentes (par exemple, pour un site de cinéma, on ne pourrait pas créer une rubrique «Orson Welles» qui dépendrait à la fois d'une rubrique «Réalisateurs» et en même temps d'une rubrique «Acteurs»); cela interdit également de réaliser plusieurs hiérarchies entrecroisées ;
- ▷ SPIP ne gère pas les structures en boucle (ou récursives).

Ces limitations ne sont pas dues à des difficultés techniques : l'impératif, ici, a été de conserver la simplicité d'utilisation, et notamment la simplicité de l'interface (créer une interface pour une telle hiérarchie est aisé, car c'est d'un emploi fréquent ; en revanche, gérer simultanément plusieurs niveaux de hiérarchie ou des structures en boucle pose de gros problèmes d'ergonomie).

On peut attacher un forum individuel à chaque rubrique (voir plus loin).

Terminons cette partie sur les rubriques en expliquant le principe des *rubriques actives*. Il arrive fréquemment, lorsqu'on travaille sur le site, que des rubriques soient vides, ou qu'elles ne contiennent que des articles qui ne sont pas encore publiés (ils sont en préparation et donc pas encore diffusés publiquement). Imaginons par exemple que la rubrique 221 ne contienne aucun article publié ; il est évident que, si un visiteur du site arrivait sur cette rubrique, il serait dans une impasse, une rubrique qui ne lui proposerait rigoureusement aucune information. C'est pourquoi nous parlons de rubriques *actives* : sur le site visité par le public, seules les rubriques contenant des articles publiés (ou des sous-rubriques contenant des articles publiés) sont considérées comme actives, et donc affichées sur le site public. Cette gestion des rubriques actives/non actives est automatique ; cependant le webmestre doit être conscient que toutes les rubriques créées dans la partie privée de SPIP n'apparaissent pas forcément sur le site public.

4.4.2 Les articles

Les articles, c'est encore plus simple : un article se trouve dans une rubrique. Point. Cela se gère très simplement par un menu déroulant.

Notez qu'une rubrique contenant elle-même des sous-rubriques peut parfaitement recevoir des articles.

La seule subtilité des articles, c'est leur *statut*. Un article peut être :

- ▷ *en cours de rédaction* : son (ou ses) auteur(s) sont en train d'y travailler, il n'apparaît donc pas sur le site public, et son accès est limité sur le site privé ;
- ▷ *proposé à la publication* : lorsque l'auteur décide que son article est terminé, il le propose au comité de rédaction (les administrateurs et les autres rédacteurs) afin de décider s'il doit être publié ou non. L'article n'est toujours pas visible publiquement, mais tous les participants à l'espace privé peuvent le voir et son invité à le commenter dans un forum lié à cet article ;
- ▷ *publié* : l'article est publié sur le site public ;
- ▷ *refusé* : l'article n'est pas publié.

C'est la seule chose à comprendre pour les articles ; pour le reste, c'est très simple, et tout se gère par une interface Web.

Limitation : un article ne peut se trouver que dans une seule rubrique à la fois (même problème de conception d'interface que précédemment).

On peut attacher un forum à chaque article (voir plus loin).

4.4.3 Les brèves

La description la plus simple pour les *brèves*, c'est l'anglicisme *news*. Ce sont des «articles» de moindre importance que les véritables articles, et ils ne sont pas signés. En revanche, il est très simple de leur adjoindre un lien vers un article ou un site Web. Les brèves sont donc idéales pour constituer une revue de presse en ligne (mais rien n'interdit de les détourner de leur usage).

Les brèves ont une gestion plus sommaire que les articles :

- ▷ les brèves ne peuvent être attachées qu'à des *secteurs*, des têtes de chapitre (dans notre exemple, les brèves correspondraient aux rubrique 1 et 2) ;
- ▷ les brèves ne sont pas signées, et leur mise en place est très simple : interface réduite, validation d'un clic.

On peut attacher un forum à chaque brève (ci-dessous).

4.4.4 Les forums de discussion

Les forums de discussion sont gérés automatiquement par SPIP. Les forums de discussion sont ici directement liés au contenu rédactionnel du site : on peut ouvrir un forum indépendant pour chaque article, pour chaque rubrique et pour chaque brève.

Par défaut, les forums de SPIP sont modérés à postériori. Cela signifie que chaque message envoyé par un utilisateur du site est immédiatement publié. En revanche, les administrateurs du site bénéficient d'une interface qui leur permet de lire les derniers messages postés depuis une semaine et, le cas échéant, de les supprimer.

L'administrateur du site pourra décider de modifier le comportement des forums. Il pourra choisir :

- ▷ l'absence totale de forums sur son site ;
- ▷ des forums modérés à priori : les contributions n'apparaissent qu'une fois validées par un administrateur ;
- ▷ des forums sur abonnement : les participants doivent auparavant s'inscrire et recevoir (automatiquement) par mail un code leur permettant de participer.

Les messages supprimés ne sont pas détruits de la base : ils sont mis de côté, et affichent l'adresse IP de l'expéditeur ainsi que la date et l'heure de l'envoi. En cas de problème juridique (ou de spammeur fou), c'est un recours indispensable.

Lorsque les forums sont actifs, il est possible, pour chaque article, d'y interdire localement l'usage d'un forum.

4.4.5 Les auteurs

SPIP gère les auteurs du site de deux façons : à la fois pour la signature des articles (pseudo, gestion des adresses email, biographie ...), et pour la gestion des accès au site privé. Ces deux aspects se gèrent via la même interface (réservée aux administrateurs).

4.4.6 Les sites syndiqués

Les systèmes de publication automatique (SPIP, phpNuke ...) fabriquent automatiquement un fichier standardisé (en XML) indiquant leurs dernières publications.

SPIP permet d'aller récupérer de tels fichiers sur le réseau, et de les inclure dans sa propre navigation. On peut ainsi indiquer sur son propre site des listes des dernières publications d'autres sites. Lorsque ces sites sont mis à jour, les nouveautés apparaissent automatiquement sur votre propre site.

Dans SPIP, les sites syndiqués sont indiqués dans les rubriques (de façon à afficher, à côté de ses propres articles, des articles tirés d'autres sites ayant une thématique similaire).

4.4.7 Les pétitions

Il est possible d'attacher à n'importe quel article une pétition validée par email. Quelques clics permettent de configurer une telle pétition (invitant les utilisateurs à «signer» tel texte).

Le processus de signature effectue automatiquement la validation par email (un mail est envoyé au signataire, lui indiquant une URL sur laquelle il «validera» sa signature). Ainsi on obtient des pétitions plus «fiables», puisque chaque signature correspond bien à une adresse email existante.

4.4.8 Les mots-clés

Il est possible de créer des mots-clés liés aux articles. Par exemple, un article pourra être lié aux mots clés «France», «Politique» ... L'usage des mots-clés permet de proposer une navigation entre différents articles portant sur les mêmes thèmes ; en particulier, cela permet de contourner la limitation de SPIP selon laquelle un article ne peut appartenir qu'à une seule rubrique.

4.5 Un mémento des raccourcis SPIP

2 mai 2002

Un fichier PDF qui regroupe sur une seule page les raccourcis de mise en page de SPIP.

Philippe Allart a réalisé ce mémento des raccourcis de mise en page SPIP.

MEMENTO-SPIP (PDF - 29.7 ko)³

4.6 Une macro Word > SPIP

31 mai 2001 par l'équipe de SPIP

Une macro qui permet de passer facilement, dans Word, au format SPIP pour faire un copier-coller qui respecte : italiques, guillemets, gras, etc.

Voici une macro qui, exécutée dans Word, commence à nettoyer un texte pour qu'il soit plus facile à transférer. Olivier Demoly a proposé le début du code. Alexis Dupont-Roc a ensuite donné une version étendue de cette macro. Beaucoup plus complète, elle gère notamment les notes automatiques !

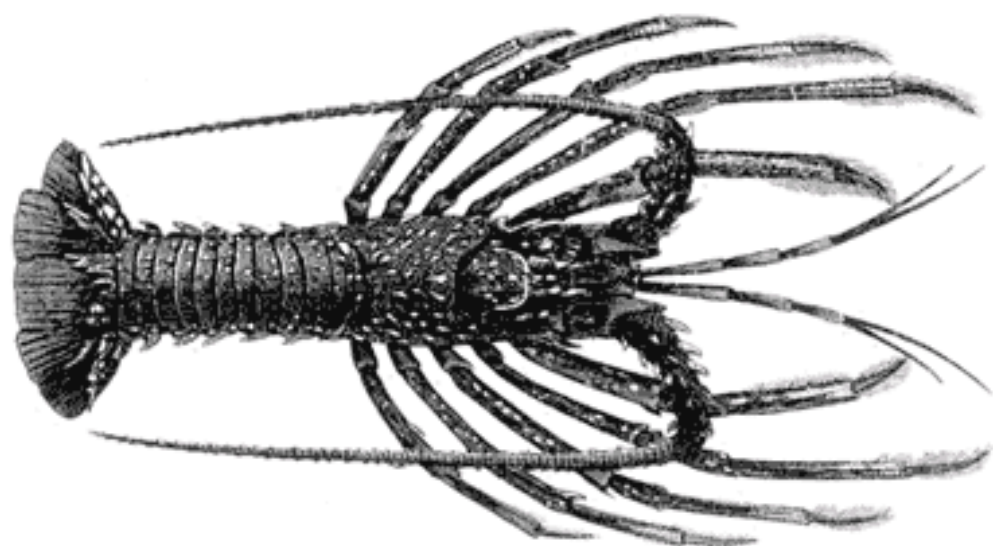
▷ **à télécharger** sur

<http://rezo.net/spip-dev/contrib/Alexis/>

³<http://www.uzine.net/IMG/pdf/doc-273.pdf>

Troisième partie

Questions et réponses



Questions

et réponses

Chapitre 5

FAQ : questions techniques



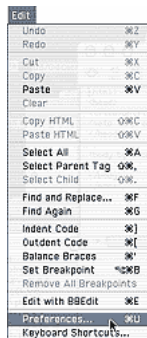
FAQ

Questions de technique

5.1 Après modification de mes *squelettes* avec Dreamweaver (Macromedia), j'obtiens des messages d'erreur quand je les installe sur le serveur.

8 avril 2002

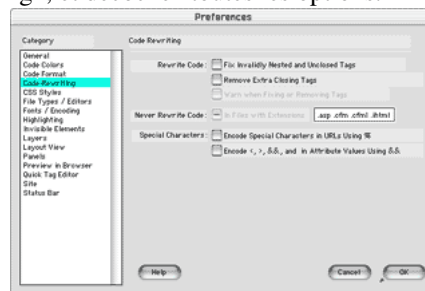
De nombreux utilisateurs de SPIP utilisent Dreamweaver (de Macromedia) pour modifier les squelettes de mise en page de leur site. Ce choix peut en effet se justifier par le fait que Dreamweaver permet à la fois de travailler en mode graphique (WYSIWYG) et d'accéder immédiatement au code source (ce qui est indispensable pour travailler proprement sur les squelettes).



Cependant, Dreamweaver a une fâcheuse tendance à *modifier* le code des squelettes et à perturber le fonctionnement des boucles. Les codes spécifiques à SPIP ne sont en effet pas reconnus comme du code HTML, et Dreamweaver essaie de les corriger, ce qui interdit ensuite le bon fonctionnement de vos pages (on obtient un message d'erreur : «tag fermant manquant»).

La solution consiste tout simplement à interdire à Dreamweaver de corriger «tout seul» le code source :

- ▷ dans le menu «Edit», sélectionnez «Preferences» ;
- ▷ sélectionnez l'onglet «Code rewriting», et décochez toutes les options.



Cela fait, Dreamweaver ne devrait plus modifier votre code source à votre insu.

◇ ◇ ◇

Et voici un autre «truc» pour que l'affichage des codes spip soit correct :

Coller dans `...../dreamweaver/configuration/ThirdPartyTags`

- ▷ deux images nommées SPIP.gif (par exemple `redacteurs-off.gif` et `redacteurs-on.gif`)
- ▷ un fichier `spip.xml` contenant :

```
<tagspec tag_name="Boucle_1" start_string="<BOUCLE" end_string=">"
detect_in_attribute="true" icon="SPIP.gif" icon_width="17"
icon_height="15"></tagspec>

<tagspec tag_name="Boucle_0" start_string="</BOUCLE_" end_string=">"
detect_in_attribute="true" icon="SPIP.gif" icon_width="17"
icon_height="15"></tagspec>

<tagspec tag_name="B_0" start_string="<B_" end_string=">"
detect_in_attribute="true" icon="SPIP2.gif" icon_width="17"
icon_height="15"></tagspec>

<tagspec tag_name="B_1" start_string="</B_" end_string=">"
detect_in_attribute="true" icon="SPIP2.gif" icon_width="17"
icon_height="15"></tagspec>

<tagspec tag_name="B_2" start_string="</B_" end_string=">"
detect_in_attribute="true" icon="SPIP2.gif" icon_width="17"
icon_height="15"></tagspec>
```

Les balises SPIP seront alors remplacées par l'image, et éditables dans la fenêtre «propriété de l'objet».

5.2 Comment fonctionne le moteur de recherche de SPIP ?

1er mai 2002

Le moteur de recherche intégré à SPIP est très simple d'utilisation, et cependant relativement puissant. Même si la plupart des utilisateurs n'ont aucune raison de se demander «comment ça marche?», nombreux sont les courriers qui demandent des précisions sur son fonctionnement ...

Voici les principes sur lesquels repose le moteur de SPIP.

Afin d'être rapide et efficace (c'est-à-dire pour qu'il donne des réponses pertinentes), le moteur de SPIP utilise un système d'**indexation** des contenus. L'indexation consiste à analyser tous les textes contenus dans SPIP, à en extraire tous les mots, et à sauvegarder pour chaque mot l'endroit où il apparaît.

Comme l'index dans un livre vous présente les mots importants du livre, suivis des numéros des pages où les retrouver, l'index du moteur de recherche fait correspondre à chaque mot utilisé sur le site le numéro de l'article, de la brève ... où le retrouver.

Ensuite, lorsqu'on utilise le moteur pour effectuer une recherche, SPIP n'a plus besoin de consulter l'ensemble des textes du site, mais simplement d'aller consulter l'index pour savoir où se trouvent ces mots. On gagne ainsi énormément de temps (et plus le site est gros, plus on gagne de temps).

5.2.1 L'indexation

Le principe est donc le suivant : prendre un texte (plus ou moins long), en extraire tous les mots, et noter chacun de ces mots dans une base de données, en faisant correspondre ce mot à l'endroit où il se trouve.

Par exemple, notre site a trois articles, dont les textes (très courts) sont :

- ▷ *article 1* : «Le petit chat est mort de froid et de faim.»
- ▷ *article 2* : «Le gros chat est rentré à la maison.»

▷ *article 3* : «La maison résiste au froid.»

Nous allons extraire les mots de chaque article, et enregistrer pour chaque mot à quel article il correspond (nous ne prendrons que les mots de plus de trois lettres, nous expliquerons plus loin pourquoi) :

- ▷ *petit* : 1
- ▷ *chat* : 1, 2
- ▷ *mort* : 1
- ▷ *froid* : 1, 3
- ▷ *faim* : 1
- ▷ *gros* : 1
- ▷ *rentré* : 2
- ▷ *maison* : 2, 3
- ▷ *résiste* : 3

Et ainsi de suite, en considérons que notre site est certainement beaucoup plus gros, et les articles beaucoup plus long.

Si l'on recherche le mot *chat* :

- ▷ une solution sans indexation consisterait à relire tous les articles, pour y trouver le mot *chat* ; sur un gros site, même pour un ordinateur, cela prend beaucoup de temps ;
- ▷ puisque nous avons un index, il suffit de consulter l'entrée *chat* : on sait immédiatement qu'il se trouve dans les articles 1 et 2.

La pondération

À l'indexation s'ajoute un deuxième principe : la pondération. Il s'agit d'essayer de rendre le moteur plus pertinent. Par exemple, si un mot apparaît dans le titre d'un article, et dans le corps du texte d'un autre article, on considère que si l'on recherche ce mot, il faut en premier indiquer celui où il apparaît dans le titre. De plus, si un mot apparaît 25 fois dans un article, et seulement deux fois dans un autre, on veut afficher en premier l'article où le mot est le plus fréquent.

On voit que l'indexation simple ne suffit pas. Si on recherche *chat*, on trouvera les articles où il apparaît, mais sans pouvoir ensuite classer ces articles entre eux (selon que le mot *chat* apparaît une fois ou 20 fois, ou s'il se trouve dans le titre ou seulement dans le texte ...).

SPIP va donc calculer une pondération pour chaque mot dans chaque article. C'est-à-dire donner des points à ce mot en fonction de l'endroit où il se trouve, et du nombre de fois où il apparaît :

dans le titre	8 points
dans le soustitre	5 points
dans le surtitre	5 points
dans le descriptif	4 points
dans le chapo	3 points
dans le texte	1 point
dans le post-scriptum	1 point

Si le mot apparaît plusieurs fois, on additionne les occurrences.

Par exemple, si dans un article, le mot *chat* apparaît :

- ▷ une fois dans le titre : 8 points
- ▷ deux fois dans le descriptif : 2 fois 4 = 8 points
- ▷ six fois dans le texte : 6 fois 1 = 6 points
- ▷ *total* : 8 + 8 + 6 = 22 points.

Le mot *chat*, dans l'index, est donc ainsi enregistré :

- ▷ *chat*, dans l'article numéro X, 22 points ;
- ▷ *chat*, dans l'article numéro Y, 15 points ;
- ▷ ...

Si l'on recherche le mot *chat*, grâce à l'index, on saura donc qu'il se trouve dans les articles X et Y, et de plus on peut classer ces articles entre eux : 22 points dans X, 15 points dans Y, donc on considère que l'article X répond mieux à la recherche.

Les mots-clés : beaucoup d'utilisateurs confondent les mots-clés et l'indexation. Les mots-clés n'ont, par nature, aucun rapport avec l'indexation : quand SPIP effectue une recherche, il ne recherche pas dans les mots-clés associés à des articles (ce serait très limité), il recherche dans l'index qu'il a fabriqué à partir du texte exact des articles. Donc, dans le principe même de l'indexation, les mots-clés n'interviennent pas du tout.

En revanche, les mots-clés sont tout de même utilisés pour fabriquer la pondération des articles. Si un mot-clé est associé à un article, il entre alors dans l'indexation de l'article lui-même, avec une forte pondération (12 points pour le nom du mot-clé, 3 points pour son descriptif). Ainsi, si notre article Y (15 points en prenant simplement compte son contenu propre) est associé au mot-clé *chat* (parce qu'on indique par ce mot que c'est le sujet de cet article), il faut ajouter à l'indexation de cet article, pour le mot *chat*, les 12 points du mot-clé : total 27 points. Dans notre recherche sur *chat*, l'article Y devance désormais l'article X.

Notons enfin que tous les éléments de contenu de SPIP font ainsi l'objet d'une indexation : les articles, les brèves, les rubriques, les auteurs, les mots-clés, les sites référencés (si le site est une syndication, les titres des articles récupérés sur ce site entrent également dans l'indexation).

Où est-ce stocké ?

Les données de l'indexation sont directement stockées dans la base de données. Cette partie est un peu technique, aussi je ne m'étendrai pas longtemps sur le sujet.

Sachez simplement qu'il y a plusieurs index (listes des mots), correspondant chacun à un type de contenu (un index pour les articles, un index pour les rubriques, un index pour les brèves ...). De plus, contrairement à l'explication ci-dessus, où les entrées de l'index sont des mots, dans SPIP les entrées des index sont des nombres calculés à partir de ces mots (des *hachages*) ; une autre table de la base de données stockant par ailleurs les correspondances entre les véritables mots et ces nombres ; cette méthode permet d'accélérer les recherches dans les index (il est plus rapide pour le logiciel de rechercher des nombres plutôt que des mots).

Notez surtout que la taille des index est très importante. Sur uZine, par exemple, la partie de la base de données consacrée au stockage des articles pèse 9,7 Mo. La partie qui stocke l'index correspond aux articles pèse 10,5 Mo. Et la partie qui stocke la correspondance entre les mots et leur traduction en nombres 4,1 Mo. Ainsi, pour un site dont les articles occupent 9,7 Mo, l'indexation de ces articles représente, à elle seule, près de 14,6 Mo. L'espace nécessaire à ces articles et à la recherche a donc plus que doublé la taille occupée par le site. C'est l'une des raisons, notamment, pour lesquelles on peut préférer désactiver le moteur de recherche, si l'on a d'importantes limitations en matière d'hébergement.

Quels mots sont indexés ?

Nous l'avons vu, tous les mots de tous les éléments de contenu du site sont extraits, puis analysés (pour pondération). Cependant, SPIP ne va pas retenir tous les mots.

- ▷ Les codes HTML sont exclus de l'indexation. Cela est évidemment nécessaire pour obtenir des recherches «propres» ...

- ▷ Les mots de moins de 4 lettres ne sont pas retenus (en réalité, de moins de 3 lettres, mais les mots de 3 lettres ne sont pour l'instant pas exploités réellement lors des recherches). Ce point soulève beaucoup de questions de la part des utilisateurs ...

Le problème est d'obtenir des résultats aussi pertinents que possible. Il faut donc privilégier, dans les recherches, les mots réellement importants dans le sens de la recherche. Par exemple, si l'on recherche les mots *le chat*, le mot important est *chat*, et non le ...

Reprenons notre premier exemple (avec trois articles constitués chacun d'une phrase). Si l'on avait indexé tous les mots, nous aurions notamment les mots :

- ▷ *le* : 1, 2
- ▷ *est* : 1, 2
- ▷ ...

En recherchant les mot *le froid est dangereux*, nous trouverions les entrées :

- ▷ *le* : 1, 2
- ▷ *froid* : 1,3
- ▷ *est* : 1, 2
- ▷ *dangereux* : nulle part.

En ajoutant les résultats de ces mots pour chaque article (en réalité, la pondération de chaque article, mais considérons pour notre exemple que chaque mot a une pondération d'un seul point), on obtiendrait :

- ▷ article 1 : 3 mots
- ▷ article 2 : 2 mots
- ▷ article 3 : 1 mot.

Le classement mettrait donc en tête l'article 1, puis l'article 2, puis l'article 3. Or, l'article 2 ne parle pas de froid, contrairement à l'article 3. À cause de l'utilisation des mots sans importance pour le sens (*le, est*), le classement est faussé.

D'où le besoin, dans l'indexation, de ne pas tenir compte des mots qui n'entrent pas dans le sens de la recherche. La méthode la plus propre consiste à fournir au système une liste de mots à ne pas indexer ; cependant, cela nécessite un travail énorme, c'est-à-dire la constitution de dictionnaires d'exclusion (et cela dans toutes les langues) ... Donc, plus simplement, dans SPIP nous choisissons de considérer que les mots de trois lettres et moins ne sont pas « importants » ; évidemment, il y a de la casse, puisque des mots comme *ONU, Val, PHP, mer, sud* ... ne sont plus pris en compte ; c'est donc un compromis, qui se juge sur l'efficacité des recherches (qui sont globalement de bonne qualité).

Quand a lieu l'indexation ?

L'indexation a lieu à trois moments différents :

- ▷ lorsque vous validez un article, celui-ci est immédiatement indexé ;
- ▷ lorsque vous modifiez un article déjà publié, il est à nouveau indexé ;
- ▷ lors de la visite du site public, si un élément accessible au public n'est pas indexé (par exemple, lorsque vous venez d'effacer les données d'indexation depuis l'espace privé, ou lorsque vous venez de rétablir une sauvegarde de votre site - les index ne sont pas sauvegardés -, ou si vous avez activé le moteur de recherche après avoir déjà publié des articles), alors il est indexé (en tâche de fond).

Retenez que l'opération d'indexation est relativement lourde : elle demande de nombreux calculs (calculs peu complexes, mais effectués sur tous les mots de l'article), et provoque de très nombreux appels à la base de données. Là aussi, chez un hébergeur très lent (vraiment très lent !), il peut être préférable de désactiver le moteur de recherche.

Retenez également que, si vous activez le moteur après avoir déjà publié des articles, ceux-ci ne sont pas immédiatement indexés : ce seront les visites sur le site public qui provoqueront leur indexation. Sur un gros site, cela peut prendre un certain temps.

5.2.2 La recherche

Puisque tous les documents sont indexés, on peut désormais effectuer des recherches.

Si vous recherchez un seul mot ...

SPIP va consulter l'index, et trouver l'entrée correspond à ce mot. Pour le mot *chat*, nous avons trouvé les articles X et Y. SPIP va de plus récupérer le nombre de points attribués à ce mot pour chaque article (22 points dans X, et 27 points pour Y). On peut donc classer nos résultats : l'article Y, puis l'article X.

Si vous recherchez plusieurs mots ...

SPIP n'autorise pas les constructions du type «ET», «OU», il ne fonctionne pas de cette manière.

Lorsque vous recherchez plusieurs mots, SPIP va effectuer l'opération de recherche pour chaque mot, récupérer les points de chaque article (ou brève, ou rubrique, etc.), et ajouter ces points.

Recherchons par exemple les mots *chat*, *gros*, *maison*. On obtient les résultats suivants pour chaque mot :

- ▷ *chat* : article X (22 points), article Y (27 points),
- ▷ *gros* : article X (12 points), article Y (2 points), article Z (5 points),
- ▷ *maison* : article Y (3 points), article Z (17 points).

SPIP fait l'addition de tous ces points pour chaque article :

- ▷ article X : $22 + 12 = 34$ points,
- ▷ article Y : $27 + 2 + 3 = 32$ points,
- ▷ article Z : $5 + 17 = 22$ points.

Le classement des articles, pour la recherche *chat*, *gros*, *maison*, est : article X, puis article Y, puis article Z. (Dans les squelettes, on peut d'ailleurs demander l'affichage des points de chaque résultat grâce à la base #POINTS.)

Ca n'est donc pas une recherche de type «ET» ni «OU», c'est une addition de points. Et à l'usage, cela se montre plutôt efficace (on trouve ce que l'on cherche, ce qui est bien le but d'un moteur ...).

5.3 Dès ma première connexion à l'espace privé de SPIP, rien ne fonctionne et on m'indique une erreur à la ligne 1 de certains fichiers.

8 avril 2002

Lorsque vous avez transféré vos fichiers vers votre hébergeur, par FTP, vous avez sélectionné un mauvais format de fichier. Essayez en indiquant à votre client FTP un autre format (format binaire, texte ...).

De temps en temps, c'est nous-même qui faisons l'erreur lors de l'installation de SPIP sur notre serveur public, et les fichiers ne sont pas utilisables ! Dans ce cas, nous postons rapidement une nouvelle version corrigée.

5.4 Est-ce qu'il faut réinstaller une sauvegarde de la base de données après avoir fait une mise à jour de SPIP ?

8 avril 2002

NON. Surtout pas. Jamais. À aucun prix. Never ever. Nichts.

Lisez-bien ce qui suit : «Il ne faut surtout pas installer une vieille sauvegarde dans une nouvelle version de SPIP. Il ne faut surtout pas installer une vieille sauvegarde dans une nouvelle version de SPIP. Il ne faut surtout pas installer une vieille sauvegarde dans une nouvelle version de SPIP. Il ne faut surtout pas installer une vieille sauvegarde dans

une nouvelle version de SPIP. Il ne faut surtout pas installer une vieille sauvegarde dans une nouvelle version de SPIP. Il ne faut surtout pas installer une vieille sauvegarde dans une nouvelle version de SPIP. Il ne faut surtout pas installer une vieille sauvegarde dans une nouvelle version de SPIP. Il ne faut surtout pas installer une vieille sauvegarde dans une nouvelle version de SPIP. Il ne faut surtout pas installer une vieille sauvegarde dans une nouvelle version de SPIP.»

Allez savoir pourquoi, un jour un individu a annoncé sur la liste de SPIP qu'il fallait faire comme cela et, malgré toutes nos explications contraires, nombre d'utilisateurs restent persuadés qu'il faut réinstaller sa vieille sauvegarde quand on change de version de SPIP.

C'est une erreur *fatale*, pire qu'un virus dans votre Outlook. Votre ordinateur prend feu, le disque dur fond, votre hébergeur fait faillite, France Telecom se trompe dans votre facture, et vous êtes interdits à vie de jamais plus utiliser le net.

La procédure correcte pour effectuer une mise à jour, c'est expliqué là (section 3.1 page 27).

5.5 J'ai des messages d'erreur un peu partout sur les pages SPIP, et rien ne marche.

8 avril 2002

Les messages sont tous du type :

```
Warning : Undefined variable :  
      HTTP_X_HOST in /repertoires/ecriture/inc_version.php3 on line 18
```

Ce ne sont pas des messages d'erreur mais de simples avertissements («warning») bénins, qui sont affichés par PHP quand le niveau de sensibilité aux erreurs et avertissements est mal réglé. Pour corriger le problème, il faut modifier le fichier de configuration de PHP (`php.ini`), que vous trouverez dans un répertoire dépendant de votre serveur (`C:/Windows/php.ini` sous Windows, un répertoire du type `/etc/php/apache/php.ini` sous Unix ...). Le paramètre à modifier est `error_reporting`; la ligne suivante (qui signifie : afficher toutes les erreurs sauf les simples avertissements) devrait fonctionner :

```
error_reporting = E_ALL & ~E_NOTICE
```

5.6 J'obtiens des messages du genre «Maximum execution time exceeded»

8 avril 2002

Cela signifie que le temps de calcul d'une page (soit sur le site public, soit sur la partie privée) a dépassé la limite fixée par votre hébergeur.

Cette limite est souvent fixée à 30 secondes, ce qui est normalement suffisant pour SPIP, et ce message ne devrait pas apparaître. En revanche, certains hébergeurs (généralement les gratuits) fixent une limite très faible. Par exemple, sur Free.fr, cette limite est de 5 secondes. Sur le calcul de pages longues ou compliquées sur un serveur plutôt lent, vous pouvez donc voir apparaître cette erreur.

Si vous avez accès à la configuration de php (fichier `/etc/php/apache/php.ini` sur Linux), il faut modifier le paramètre `max_execution_time`.

Enfin, si le problème perdure, il vous reste à enquiquiner votre hébergeur pour qu'il fixe une limite qui permette de travailler sérieusement.

5.7 Je suis chez Free ou chez Online, et j'obtiens souvent des messages «Maximum execution time exceeded»

13 mai 2002

Pour éviter qu'une erreur de programmation dans les scripts PHP ne bloque le fonctionnement du serveur, les hébergeurs fixent une durée maximale d'exécution de chaque page PHP. Cette limite est généralement fixée à 30 secondes : si une page «part» dans un calcul qui prend plus de 30 secondes (un énorme calcul, ou une boucle sans fin), le serveur interrompt le calcul et l'indique par un message d'erreur. Pour SPIP, cette limite de 30 secondes est largement suffisante pour effectuer tous ses calculs.

Certains hébergeurs fixent cependant des durées maximales très courtes. C'est notamment le cas chez Free (5 secondes) et chez Online (8 secondes). De plus, chez ces hébergeurs, la mémoire allouée au fonctionnement de PHP est plutôt faible, ce qui en ralentit l'exécution. Ainsi, les fonctions de SPIP, chez ces hébergeurs, prennent plus de temps à s'exécuter, et la durée maximale pour qu'elles s'exécutent est ridiculement courte. Imaginez un train qui ne peut fonctionner que pendant une heure, et qui doit parcourir une distance de 120 kilomètres sur des lignes où la vitesse est limitée à 60 km/h : à mi-chemin le train s'arrêtera en rase campagne.

Ce problème se rencontre systématiquement avec des articles longs : chez ces hébergeurs, le calcul de telles pages est interrompu par le serveur avant la fin des calculs, ce qui interdit leur affichage et provoque le message d'erreur «Maximum execution time exceeded».

Il n'y a aucune solution technique. SPIP a déjà été largement optimisé pour tourner plus vite, le problème est désormais du côté des hébergeurs.

- ▷ Seule solution immédiate : faites-vous héberger ailleurs, chez des hébergeurs qui fournissent un service utilisable, et non pas bridé d'une manière extrême.
- ▷ Ecrivez au responsable de l'hébergement pour lui demander d'augmenter la durée maximale d'exécution des scripts (30 secondes comme tout le monde, par exemple). On ne sait jamais, peut-être qu'à la longue... Mais pour l'instant, l'absence de réactions semble plus indiquer une volonté délibérée de brider l'hébergement gratuit pour pousser les webmasters à passer à l'hébergement payant qu'un réel impératif technique.

5.8 Je suis sous Apache/Windows, et après l'installation j'ai droit à une «Internal Server Error» quand je me connecte à l'espace privé

8 avril 2002

SPIP ne tourne pas sous *Apache Windows avec PHP en CGI* (!). Sur cette plateforme, il faut obligatoirement installer PHP sous forme de module Apache (ou éventuellement un autre serveur). Les instructions d'installation de PHP sous forme de module sont livrées avec PHP.

La raison de ce problème est un peu compliquée... Par défaut, PHP crée un fichier «.htpasswd» pour restreindre l'accès au répertoire `ecriture/`, et nécessite pour cela la fonction `crypt()`. Or la fonction `crypt()` de PHP sous Windows, soit n'existe pas, soit n'est pas compatible avec l'authentification d'Apache. Donc, à la place, on utilise l'authentification PHP en envoyant à la main un en-tête HTTP «401 Unauthorized». Chose qui est impossible avec PHP en CGI, uniquement en module.

5.9 Je suis sous Apache/Windows, l'installation et l'espace privé marchent bien, mais le site public fait régulièrement planter Apache (avec un message d'erreur Windows typiquement incompréhensible)

8 avril 2002

(NB : ce problème est avéré avec Apache 1.3.20 et php 4.0.5)

Ce message d'erreur peut signifier que c'est Apache ou PHP qui est légèrement buggé (il y a plus de chances que ce soit PHP, d'ailleurs). Apache et PHP sont tous deux conçus à l'origine pour Unix, et les versions Windows ne sont donc pas toujours d'une très grande robustesse (Apache le reconnaît d'ailleurs sur sa page Web). Comme l'affichage du site public de SPIP peut donner lieu à des calculs nombreux et complexes, il y a d'autant plus de chances qu'une faiblesse de PHP y soit mise en évidence. Il n'y a pas de solution à part prendre une autre version de PHP et/ou d'Apache (pour indication, Apache 1.3.19 avec PHP 4.0.4p11 ne posent aucun problème).

5.10 Je viens d'effectuer une mise à jour de SPIP, et le site public fonctionne mal

8 avril 2002

Il y a trois raisons principales pour qu'une mise à jour provoque des messages d'erreur.

1. L'espace privé fonctionne parfaitement, sans erreurs. Seul le site public déconne. C'est normal ... En effet, certaines «fonctions» de SPIP ont été modifiées (sinon il n'y aurait pas mise à jour) ; dans le même temps, les pages du site public sont stockées pendant un certain temps en cache. Au moment de la mise à jour, il peut arriver que les pages du site public (calculées avant la mise à jour) utilisent des fonctions dont le fonctionnement a changé.

La solution la plus simple : **vider le cache**. Rendez vous sur la page «Sauvegarde/restauration de la base», et cliquez sur le bouton «Vider le cache».

2. Vous avez récupéré la dernière version beta-x24 planquée au fin fond de notre serveur, pour voir à quoi elle ressemble.

Si vous n'avez aucune connaissance en PHP/mysql, c'est une très mauvaise idée. Les versions beta sont des versions en cours de développement, non terminées. Certains jours et avec beaucoup de chance, elles sont relativement stables ; d'autres jours, elles contiennent des bouts de fonctions pas terminées, et surtout, pas déboguées.

Deux cas de figure :

- ▷ vous avez effectué une sauvegarde de votre base de données (page «Sauvegarde / Restauration de la base») avant la mise à jour malencontreuse ; dans ce cas, toujours sur cette page, effacez totalement la base de données (bouton «Effacer tout»). Réinstallez la version précédente de SPIP (important : la version avec laquelle vous avez effectué la sauvegarde), et réinstallez le site à partir de votre sauvegarde ;
- ▷ vous n'avez pas de sauvegarde de votre base de données ... grave erreur. Si vous n'êtes pas bricoleur, le mieux à faire est encore d'attendre quelques jours la prochaine version beta, en espérant qu'elle soit plus stable (généralement, les versions beta ne restent pas totalement instables très longtemps ; il y a toujours un moment où, grosso modo, elles fonctionnent : mais souvent l'interface n'est pas terrible ...).

3. Après la mise à jour de SPIP, vous avez voulu réinstaller la sauvegarde de votre base effectuée avant la mise à jour. On se demande pourquoi tant de monde a cette idée saugrenue ...

Nous consacrons une page spécifique (section 3.1 page 27) expliquant comment «bien» effectuer une mise à jour.

Lorsque SPIP change de version, il est fréquent que, pour installer les nouvelles fonctionnalités et améliorer le fonctionnement du système, nous modifions la structure de la base de données, c'est-à-dire la façon dont les informations de votre site sont stockées. Vouloir réinstaller une ancienne sauvegarde dans une nouvelle version de SPIP, c'est un peu comme faire passer des carrés dans des trous triangulaires.

La seule solution «propre» pour rattraper cela : rendez-vous dans la page «Sauvegarde / restauration de la base», et effacez la base de données («Effacer tout»). Installez l'ancienne version de SPIP (toutes les versions sont accessibles dans les archives), restaurez la base à partir de votre sauvegarde. Cela fait, vous pourrez installer la nouvelle version de SPIP, le système effectuera de lui-même les modifications nécessaires sur la structure de votre site.

5.11 Le serveur qui m'héberge tourne sous PHP4, et ne comprend pas les fichiers «.php3»

8 avril 2002

Dans la plupart des cas, les serveurs utilisant PHP4 sont configurés pour reconnaître les fichiers utilisant la terminaison «.php3», afin d'assurer la compatibilité. Si tel n'était pas le cas (vous voyez apparaître le texte source à l'écran, ou bien votre butineur vous dit qu'il ne reconnaît pas le format du fichier), contactez votre hébergeur pour qu'il effectue la manipulation.

Karl signale, à l'attention de ceux qui contrôlent leur propre serveur : Si vous avez accès à votre `httpd.conf` et que vous avez compilé PHP4 modifiez ainsi votre `httpd.conf` :

```
# # And for PHP 4.x, use : #
AddType application/x-httpd-php .php .php3
AddType application/x-httpd-php-source .phps
```

5.12 Pourquoi la création automatique de vignettes ne fonctionne pas avec tous les formats d'images (JPEG, GIF, PNG) ?

12 mai 2002

[SPIP 1.4] La création automatique de vignettes de prévisualisation utilise une fonctionnalité associée à PHP, nommée «GD» (on appelle cela une «bibliothèque»). Cette bibliothèque ne fait pas partie des fonctions intégrées à PHP, c'est une option développée à part (par Thomas Boutell¹), que les responsables des serveurs peuvent ou non installer avec PHP.

1. Première conséquence : certains hébergeurs n'installent pas GD avec PHP. Chez ces hébergeurs, les fonctions graphiques sont donc inutilisables. Notamment SPIP ne pourra pas créer automatiquement les vignettes.
Dans ce cas, il faut demander à l'hébergeur d'installer la bibliothèque GD.
2. Le format GIF est un format propriétaire, qui utilise des fonctions qui ne sont pas des logiciels libres (notamment, leur propriétaire réclame une redevance aux programmes graphiques qui manipulent le format GIF). On ne peut donc pas l'installer dans un logiciel libre. Cependant, ignorant ce problème de copyright, les toutes premières versions de GD manipulaient le format GIF ; les versions récentes ne l'exploitent plus. Chez certains hébergeurs, avec des versions anciennes de GD, le GIF est donc utilisable ; mais chez la plupart, il n'est pas possible de l'exploiter.
Si le GIF n'est pas exploitable, il semble inutile de le réclamer à votre hébergeur : il faudrait en effet qu'il installe une ancienne version de GD plutôt qu'une version récente (autant demander à remplacer Windows 2000 par Windows 95 sur une machine de bureau).
3. Pour fonctionner, la bibliothèque GD utilise elle-même d'autres logiciels et bibliothèques pour traiter les formats PNG et JPEG. Si l'hébergeur n'a pas installé auparavant ces logiciels, GD fonctionnera, mais n'exploitera pas l'un ou l'autre de ces formats. Ainsi, même avec une version récente de GD, il est fréquent que seul le format PNG soit exploité, parce que les logiciels permettant de travailler en JPEG ne sont pas installés sur la machine ...
Dans ce cas, demandez à votre hébergeur de compléter son installation pour que le format JPEG soit utilisable.

◇ ◇ ◇

¹<http://www.boutell.com/gd/>

Dans la configuration précise de SPIP, lorsque vous demandez l'activation de la création automatique des vignettes, SPIP va tester les trois formats d'images, pour déterminer lesquels peuvent être exploités chez votre hébergeur. Si vous avez compris l'explication qui précède, vous savez que vous n'obtenez jamais l'utilisation des trois formats d'image ; fréquemment vous n'obtiendrez que le PNG et le JPEG ; parfois seulement le format PNG ; et le GIF encore plus rarement.

5.13 SPIP ne fonctionne pas chez mon hébergeur habituel. Y a-t-il un «patch» quelque part ?

8 avril 2002

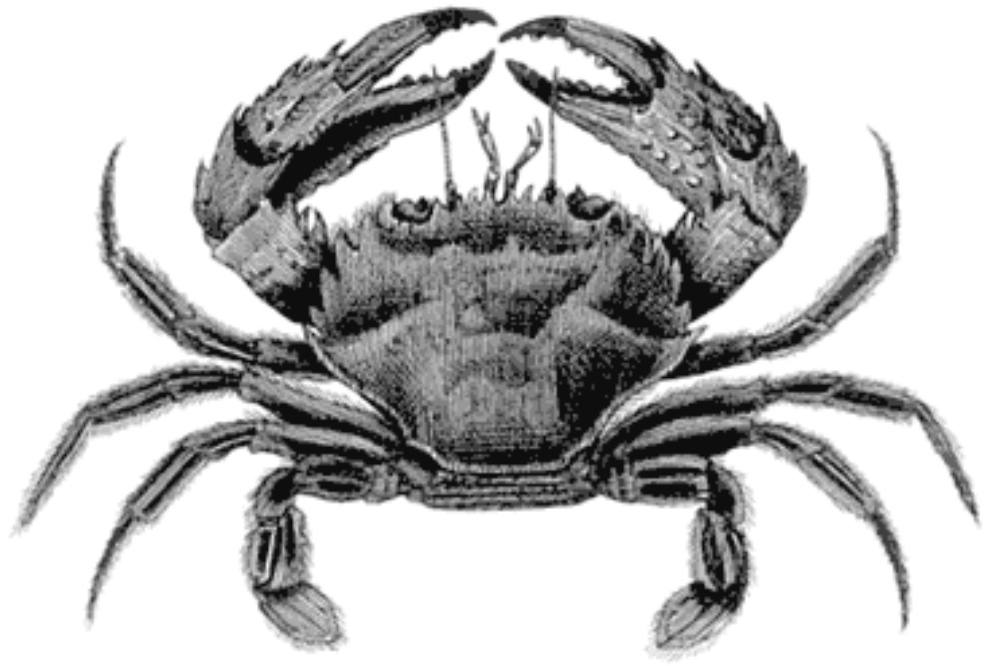
Avant tout, vérifiez que le problème vient bien de votre hébergeur. Utilisez notamment la liste de diffusion de SPIP (section 1.5 page 12) pour demander à d'autres utilisateurs s'ils ont eu des difficultés similaires avec cet hébergeur.

Dans un second temps, si le problème que présente cet hébergeur est identifié, et dépend d'un choix de configuration de sa part, n'hésitez pas à contacter le responsable pour lui demander de modifier sa configuration pour qu'elle soit compatible avec SPIP ; certains responsables de serveurs d'hébergement ont déjà fait des efforts pour aider leurs utilisateurs sous SPIP.

Finalement, en dernier recours, votre hébergeur ne veut ou ne peut pas modifier sa configuration pour qu'elle ne soit plus incompatible avec SPIP. Est-ce qu'il existe des «patch» spécifiques pour certains hébergeurs, c'est-à-dire des petits bouts de programmes, des modifications spécifiques dans SPIP ? Non. Le but de l'équipe de développement est d'éviter la multiplication des «patches» et des bidouilles dans le code, afin de livrer une unique version utilisable par tous (la facilité d'installation de SPIP est l'une de nos priorités). Si votre hébergeur présente un problème spécifique, nous verrons s'il y a moyen de le résoudre, et nous incorporerons la solution dans la version suivante de SPIP.

Chapitre 6

FAQ : premiers pas



FAQ

Premiers pas

6.1 C'est quoi cette histoire d'espace privé ?

8 avril 2002

Un site créé avec SPIP est divisé entre deux parties distinctes :

- ▷ *le site public* est l'ensemble des pages accessibles à tout visiteur. Il s'agit de l'affichage des articles, brèves, rubriques que vous avez publiés, ainsi que des interventions dans les forums qui y sont associés. Il s'agit de tous les fichiers se trouvant à la racine de l'installation SPIP, ainsi que dans les répertoires `IMG/` et `NAVPICS/`.
- ▷ *l'espace privé* est l'espace à accès restreint dans lequel les administrateurs et rédacteurs peuvent écrire des articles, des brèves, modifier la structure du site, suivre les contributions postées dans les forums, etc. Cet espace nécessite d'être *enregistré*, c'est-à-dire d'avoir un accès login/mot de passe. Il est accessible depuis le répertoire `ecrire/` de l'installation SPIP (il s'agit en fait de la même adresse que celle utilisée pour installer SPIP).

6.2 Comment installer SPIP ?

8 avril 2002

La page «Installer SPIP» (section 2.2 page 18) explique tout ce que vous devez savoir.

6.3 Comment réinstaller SPIP si je n'arrive plus à accéder à l'espace privé ?

8 avril 2002

Allez dans le répertoire `ecrire/` avec votre client FTP, effacez le fichier `inc-connect.php3` et, s'il existe, le fichier `.htaccess`. Puis suivez la procédure d'installation standard (cf. question précédente).

Important : la procédure de réinstallation ne détruit pas la base de données (articles, rubriques, etc.). Elle sert simplement à recréer un accès administrateur au site, et à générer des fichiers d'authentification corrects.

Lors de l'installation, la vérification des droits d'accès échoue à cause des répertoires `CACHE` et `IMG`. Pourtant, je suis sûr d'avoir réglé les droits correctement.

Il s'agit probablement d'un problème de majuscules. Vérifiez la façon dont ces deux répertoires ont été créés chez votre hébergeur. S'ils sont en minuscules, renommez-les en majuscules (avec votre logiciel de transfert de fichiers).

6.4 De quoi a besoin SPIP pour fonctionner ?

8 avril 2002 par l'équipe de SPIP

SPIP a besoin (outre, bien sûr, un compte Web) de PHP3 ou 4, et d'une base de données MySQL. Voir la page «Installer SPIP» (section 2.2 page 18) pour plus de détails.

Est-ce que cela marche sous tous les hébergeurs proposant les logiciels ci-dessus ? Quels sont les hébergeurs testés ?

Cela ne marchera pas forcément sous tous les hébergeurs car certains offrent une configuration extrêmement bridée (par exemple, obligation de créer les tables MySQL à la main).

Nous établissons progressivement une liste des hébergeurs acceptant SPIP. Consultez cette page pour plus d'informations (section 2.1 page 16).

6.5 Est-ce que je peux utiliser SPIP pour réaliser un site marchand, un site avec des pubs, ou un site de droite, ou un site qui débîne le logiciel libre ...

8 avril 2002

Vous faites bien ce que vous voulez ...

6.6 Est-ce que je peux vendre un service utilisant SPIP ? Est-ce que je peux monter une activité professionnelle autour de SPIP ?

8 avril 2002

Vous faites bien ce que vous voulez ...

Plus sérieusement, plusieurs utilisateurs nous ont déjà demandé si «ça ne nous embêtait pas» qu'ils utilisent SPIP dans le cadre de leur activité professionnelle (vendre des sites réalisés sous SPIP, vendre des formations à SPIP ...). La réponse est claire : «ça ne nous embête pas», bien au contraire. Si nous avons choisi de diffuser SPIP sous GPL, c'est bien pour cela. Si SPIP peut vous aider économiquement, nous en serons plus que flattés.

6.7 J'ai créé une rubrique, mais elle n'apparaît pas dans le site public !

8 avril 2002

Seules apparaissent les rubriques contenant au moins un article *publié* (nous soulignons), ainsi que les rubriques *englobant* celles-ci. Il faut donc au minimum écrire un article et le publier. Vous pouvez bien sûr publier un article fantoche afin de tester le système.

J'ai fait tout cela, mais rien n'apparaît !

Le site public utilise un système de cache, c'est-à-dire que les fichiers sont recalculés à intervalles réguliers (toutes les heures ou toutes les deux heures, en fonction des réglages à l'intérieur des scripts PHP). Ce système permet d'économiser les accès à la base MySQL, et de continuer à fonctionner au cas où MySQL est planté (ce qui est fréquent chez certains hébergeurs de piètre qualité).

Il faut donc forcer la régénération de la page que vous pensez erronée. Pour cela, posez un cookie depuis l'espace privé, ce qui affichera un bouton «recalculer cette page» sur chaque page de l'espace public. Puis appuyez sur le dit bouton sur la page désirée (par exemple, sur la page d'index si vous désirez voir apparaître vos premières rubriques).

6.8 J'ai perdu mon login et/ou mon mot de passe ...

8 avril 2002

Les mots de passe ne sont pas récupérables pour des raisons de sécurité. S'il y a d'autres administrateurs que vous, vous pouvez leur écrire afin qu'ils vous changent votre mot de passe en prenant soin de vous communiquer les nouveaux codes (il leur suffit d'aller dans la section «rédacteurs» de l'espace privé et de cliquer sur «modifier» à côté de votre accès, puis de choisir un nouveau mot de passe).

Si vous étiez le seul administrateur, il va falloir réinstaller SPIP pour créer un accès (voir plus haut : «Comment réinstaller SPIP ?»). Ne vous inquiétez pas, nul besoin de télécharger des fichiers : il s'agit juste de recréer un accès administrateur, sans toucher aux données enregistrées.

6.9 Je voudrais une nouvelle fonctionnalité ...

8 avril 2002

Vous pouvez écrire à spip@rezo.net pour la proposer, en donnant moult détails et en précisant les raisons qui vous poussent à vouloir une telle fonctionnalité.

N'oubliez pas cependant que SPIP est un logiciel libre, et que vous pouvez implémenter cette fonctionnalité vous-même si vous avez les compétences requises - auquel cas vous feriez des heureux en nous communiquant la version ainsi modifiée !

6.10 Mon problème n'est pas mentionné dans cette FAQ. Que faire ?

8 avril 2002

Parcourez la documentation (chapitre [page i](#)) correspondant au type de problème que vous rencontrez. Par exemple, si c'est un problème d'installation, consultez le manuel d'installation ou la FAQ technique.

N'oubliez pas également de vous familiariser avec l'interface de l'espace privé (où vous créez des rubriques, publiez des articles ...). Rien de tel que de «jouer» un peu avec les différentes pages disponibles pour appréhender au mieux le fonctionnement de SPIP !

Enfin, éditez les squelettes de mise en page du site public (les fichiers `.html`) avec un éditeur de texte afin de comprendre la façon dont les données sont structurées et affichées, grâce aux balises et aux champs spécifiques à SPIP.

6.11 Mon site est vide, que dois-je faire ?

8 avril 2002

La première chose à faire est de commencer à créer l'arborescence du site (au minimum une rubrique), que vous pourrez bien sûr compléter par la suite. Pour cela, cliquer sur «naviguer dans le site» sur le bandeau du haut de l'espace privé, puis utiliser le bouton *ad hoc* pour créer une nouvelle (sous-)rubrique.

6.12 Mon site sous SPIP est planté, est-ce que je vous fais un procès ?

8 avril 2002

Nous diffusons SPIP gratuitement. Pour cette raison, nous insistons sur le point suivant : nous n'offrons aucune garantie d'aucune sorte, ni service après-vente (vu que SPIP n'est pas vendu ...), ni garantie de résultat, ni obligation d'aide.

L'aide pour SPIP est donc basée sur le bénévolat et sur l'entraide des utilisateurs.

Quatrième partie

Squelettes : Guide du webmestre

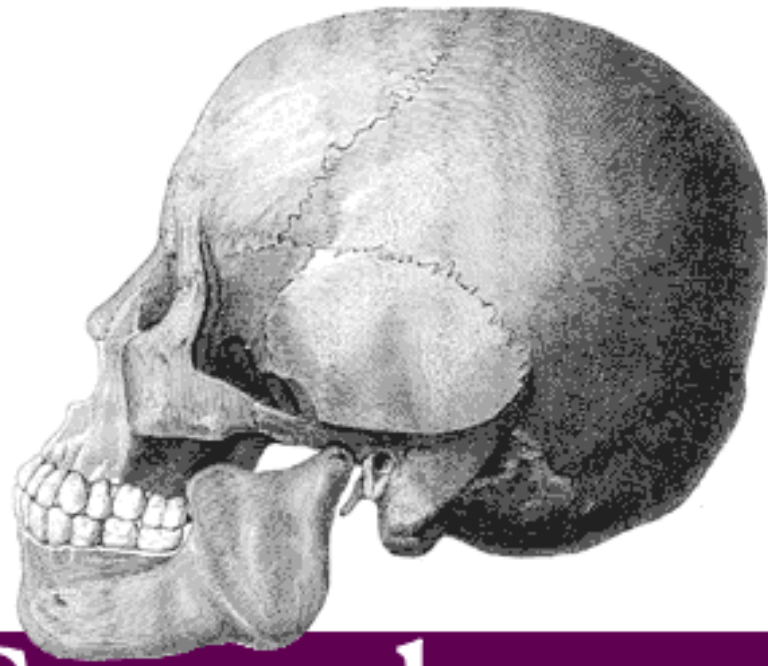


Squelettes

Guide du webmestre

Chapitre 7

Squelettes : Guide du webmestre



Squelettes

Guide du webmestre

Lien utile : nous fournissons sur notre site FTP de nombreux exemples de squelettes¹.

7.1 FAQ webmestre

2 juillet 2001 par l'équipe de SPIP

7.1.1 Les bases

1. Comment fais-je pour modifier la mise en page du site public ?

La gestion de la mise en page s'appuie sur des fichiers à l'extension `.html` appelés *squelettes* de mise en page. Leur rôle correspond grosso modo à ce que d'autres logiciels nomment «modèles» (ou en anglais, «templates»).

Chaque fichier est associé à un type de page différent : ainsi un squelette pour le sommaire, un pour l'affichage des articles, un pour l'affichage des rubriques, etc. Un squelette contient du HTML standard définissant l'habillage de la page, dans lequel on insère des tags (ou balises) spécifiques à SPIP afin de définir quelles informations vont venir «habiter» cet habillage.

Le langage des squelettes de SPIP est très souple et permet de réaliser des mises en page très variées : un simple coup d'oeil à uZine², Vacarme³, Hactivist News Service⁴ ainsi que les sites enregistrés par leurs créateurs sur cette page (section 1.4 page 12) saura vous en convaincre. Il est donc dommage de garder la mise en page d'origine.

2. Est-il possible d'écrire ces squelettes soi-même ?

Oui. Pour cela allez voir :

- ▷ le tutorial (section 9 page 143), pour comprendre les bases de la programmation des squelettes.
- ▷ le manuel de référence (section 8 page 97), qui liste toutes les possibilités de programmation.

Si vous souhaitez un exemple didactique, vous pouvez aussi partir du jeu de squelettes «simples»⁵, qui utilise un HTML très basique afin de bien laisser comprendre l'utilisation des tags spécifiques à SPIP.

3. Je ne sais pas / ne veux pas apprendre à programmer. Peut-on utiliser des mises en pages déjà existantes ?

Oui. En dehors de la mise en page par défaut, d'autres jeux de squelettes sont disponibles sur le serveur de téléchargement⁶, dans le répertoire «SQUELETTES».

Il suffit en général de récupérer l'archive voulue (le fichier au format `.zip` ou `.tar.gz`, au choix), de la décompresser chez vous, et de transférer son contenu par FTP à la racine de votre site SPIP. Vous pouvez faire une sauvegarde de vos fichiers `.html` actuels, au cas où vous voulez revenir en arrière.

4. Il n'y a pas beaucoup de jeux de squelettes disponibles. Pourquoi ?

Ces jeux de squelettes sont alimentés par les webmestres SPIP qui nous fournissent leurs créations. Nous comptons donc sur les webmestres pour compléter cette base de squelettes afin d'encourager l'entraide et la richesse des sites SPIP. (cf. section «Partager» plus bas dans cette FAQ)

7.1.2 Créer ses squelettes

1. Peut-on utiliser un éditeur textuel pour créer et modifier ses squelettes ?

Oui, comme on le ferait pour du HTML classique.

¹<http://rezo.net/spip-dev/SQUELETTES/>

²<http://www.uzine.net/>

³<http://www.vacarme.eu.org>

⁴<http://hns.samizdat.net/>

⁵<http://rezo.net/spip-dev/SQUELETTES/simples/>

⁶<http://rezo.net/spip-dev/>

2. **Peut-on utiliser un éditeur graphique (WYSIWYG) pour créer et modifier ses squelettes ?**

Oui, comme on le ferait pour du HTML classique. Voir cependant la question suivante.

3. **J'essaie d'utiliser un éditeur graphique pour créer mes pages, mais il modifie les tags SPIP. Peut-on résoudre ce problème ?**

Certains éditeurs graphiques «corrige» automatiquement les tags qu'ils ne comprennent pas. La plupart ont toutefois une option permettant de désactiver cette fonctionnalité. Nous avons consacré un article spécifique à DreamWeaver, mais la démarche est équivalente pour les autres éditeurs (GoLive. . .).

7.1.3 Partager

1. **J'ai écrit des squelettes pour mon site. Comment fais-je pour qu'ils soient disponibles à tous ?**

N'hésitez pas à les envoyer à l'adresse `spip-dev@rezo.net`, qui est la mailing-list des développeurs. Pour cela, créez simplement un fichier compressé (.zip, .tar.gz, etc.) contenant les différents fichiers .html. N'oubliez pas d'inclure les fichiers graphiques (images) le cas échéant.

7.2 Contribuer au développement de SPIP

10 juin 2001 par l'équipe de SPIP

Quelques règles

Si vous voulez contribuer à la programmation de SPIP, l'idée la plus importante à retenir est la suivante : vous arrivez sur un projet qui est déjà fonctionnel. Ce projet est muni d'un ensemble de règles qui, toutes arbitraires qu'elles peuvent paraître, assurent sa cohérence. Ces règles n'ont pas besoin d'être énoncées explicitement pour exister : certaines sont clairement visibles après un examen plus ou moins détaillé du code, et les règles tacites doivent être respectées au même titre que les autres.

Il est formellement conseillé de bien suivre ces règles. Ce respect n'a pas à être lié ou non à vos goûts personnels : il permet de garder sa cohérence et son unité au projet, et de le conserver aussi lisible qu'il l'était auparavant. N'oubliez pas que d'autres personnes que vous sont amenées à lire, comprendre, voire modifier votre code.

Par exemple, il est évident que les fonctions SPIP sont écrites sous la forme `ma_fonction()`. Dans le cadre de ce projet, il serait donc parfaitement déplacé d'ajouter des fonctions en les écrivant `MaFonction()` - même si dans l'absolu cette forme n'est pas plus critiquable que l'autre.

Tout ceci n'empêche pas évidemment de critiquer une règle et d'en proposer une meilleure, le cas échéant. N'hésitez pas à le faire ; mais il doit y avoir de véritables raisons à cela.

Enfin, toute règle souffre des exceptions. Mais celles-ci doivent avoir une véritable justification, non uniquement la paresse du programmeur ; elles doivent être le plus rares possible. Notamment, garder à l'esprit que le «provisoire» a souvent tendance à devenir définitif quand personne n'a envie de le corriger ; or il est logique et juste que tout programmeur soit responsable de la finition de son propre code, mais non de celui des autres.

7.2.1 Règles de présentation et d'écriture

Les règles qui suivent sont communes à un nombre plus ou moins grand de langages de programmation : au minimum tous les langages présentant une syntaxe similaire à PHP (c'est-à-dire, outre PHP lui-même, C, C++, Java. . .).

Ces règles sont communément acceptées, de façon aussi naturelle que les règles de présentation et de typographie d'un texte en langage naturel ; d'ailleurs elles sont fréquemment similaires.

Présentation

- ▷ Le code doit être espacé et indenté de manière à mettre en valeur sa structure et la séparation entre les différents blocs logiques (fonctions notamment). L'espacement et l'indentation doivent être suffisants pour rendre la structure compréhensible dès le premier regard ; ils ne doivent pas être excessifs. On doit y apporter le même soin qu'à la division en paragraphes d'un texte en langage naturel.
- ▷ L'indentation sera faite de préférence avec le caractère de tabulation. Cela permet de choisir librement la profondeur d'indentation dans les options de son éditeur de texte, tout en n'imposant pas ce choix aux autres développeurs.
- ▷ Tout bloc contenu à l'intérieur d'une paire d'accollades sera indenté d'une et une seule tabulation. De même, récursivement, pour les sous-blocs : ajout d'une et une seule tabulation supplémentaire à chaque entrée dans un niveau de profondeur supplémentaire. Cette règle vaut bien aussi pour la déclaration de fonctions.
- ▷ Le code qui ne fait pas partie d'une fonction ne doit pas être indenté.
- ▷ L'utilisation des transitions PHP-HTML (`<?php` et `?>`) doit être minimisée. L'éviter lors qu'il s'agit d'afficher uniquement de petits morceaux de HTML. Ne pas oublier qu'un petit morceau de code PHP inséré au milieu d'un océan de HTML est invisible sans un examen très attentionné.

Typographie

- ▷ Lors de l'utilisation de parenthèses ou de crochets, il ne faut pas laisser d'espace après la parenthèse ouvrante ni avant la parenthèse fermante.
- ▷ Lors de l'utilisation d'opérateurs binaires (+, =, *, AND, ...), il faut laisser un espace de part et d'autre de l'opérateur. Exception manifeste dans cette phrase, où les opérateurs sont mentionnés et non utilisés en tant que tels.
- ▷ Les opérateurs unaires (!, ...) doivent être collés au paramètre auquel ils s'appliquent.
- ▷ Par convention, lors d'un appel de fonction, il n'y a pas d'espace devant la parenthèse ouvrante : `<f($x)>` et non `<f ($x)>`. A contrario, et pour bien distinguer, on laisse un espace devant la parenthèse quand il s'agit d'une structure de contrôle intégrée au langage : `<if (!$x)>` et non `<if(!$x)>`.
- ▷ Les virgules et points-virgules sont suivis mais non précédés d'un espace.

7.2.2 Règles de programmation

Réfléchir

Avant de programmer une nouvelle fonctionnalité, réfléchir...

- ▷ méthodes et algorithmes utilisés pour l'implémentation : légèreté, performance, robustesse (ne pas hésiter à faire quelques calculs grossiers pour valider les choix) ;
- ▷ adéquation au projet : portabilité, sécurité, souplesse ;
- ▷ implications sur les autres fonctionnalités : modifications et ajouts à faire sur les fonctionnalités existantes ;
- ▷ place «naturelle» pour cette fonctionnalité dans le projet : en matière d'interface, de fichiers...

Ne pas négliger la factorisation ou mise en commun du code (par des fonctions, notamment dans des fichiers à inclure). Éviter par contre le plus possible les fichiers inclus contenant du code hors fonctions (sauf lorsque c'est «naturel» et voulu).

Nommer

- ▷ *Variables et fonctions :*

Quel que soit le projet, le nommage doit rester homogène pour que le code soit facile à lire. Ainsi, sous SPIP, les noms de variables et de fonctions seront en minuscules ; les noms composés, de la forme `variable_composee`.

D'une manière générale, les noms seront ni trop brefs, ni trop longs ; ils seront suffisamment explicites. Cette règle est particulièrement importante pour les variables globales, qui peuvent être partagées entre plusieurs fichiers et de nombreuses fonctions. Pour les variables locales (i.e. à une fonction), la règle est plus souple. Notamment, on peut employer des variables d'une lettre, par exemple pour obtenir des expressions plus compactes. Remarquer que dans tous les langages de programmation, un certain nombre de lettres sont par tradition associées à certains usages (exemples : `$i`, `$j` pour des compteurs de boucles, `$n` pour un dénombrement, `$t` pour un instant ou une durée en secondes...). Ne pas détourner ces conventions permet à vos lecteurs d'être plus vite dans le bain.

▷ *Fichiers* :

Pour des raisons historiques, les fichiers à inclure dans l'espace public seront appelés

`inc-fichier.php3`.

Dans l'espace privé, ce sera

`ecrire/inc_fichier.php3` (noter le tiret bas à la place du tiret normal).

Les fichiers de l'espace public appelés par redirection HTTP depuis l'espace privé sont appelés

`spip_fichier.php3`.

Tous les autres fichiers ont un nom qui ne commence

ni par "inc", ni par "spip".

▷ *Tester* :

Une fois une modification importante apportée, il est bon de la tester soi-même, sans attendre que quelqu'un d'autre le fasse à sa place. Dans le cadre de SPIP, cela veut dire vérifier que le programme marche de manière correcte sur un certain nombre d'hébergeurs (par exemple : Altern, Free...) et de configurations (par exemple : différentes versions de PHP, de MySQL, restriction plus ou moins grande des droits d'accès aux répertoires...); mais aussi qu'un certain nombre de situations parmi les plus courantes (dans le cas d'une interface graphique notamment) sont traitées correctement.

7.2.3 Partager vos modifications

Une fois que vous êtes satisfait de votre modification du code, il est grand temps d'en parler avec les autres développeurs de SPIP, et de voir s'il mérite d'être intégré à la distribution officielle de SPIP... Rendez-vous sur la liste de diffusion `spip-dev`⁷.

A bientôt !

7.3 La structure de la base de données

12 juin 2001 par l'équipe de SPIP

La structure de la base de données est assez simple. Certaines conventions ont été utilisées, que vous repérerez assez facilement au cours de ce document. Par exemple, la plupart des objets sont indexés par un entier autoincrémenté dont le nom est du type `id_objet`, et qui est déclaré comme clé primaire dans la table appropriée.

⁷<http://listes.rezo.net/mailman/listinfo/spip-dev>

7.3.1 Contenu rédactionnel

Les rubriques : spip_rubriques

id_rubrique	bigint(21)
id_parent	bigint(21)
titre	text
descriptif	text
texte	longblob
id_secteur	bigint(21)
maj	timestamp(14)
export	varchar(10)
id_import	bigint(20)

- ▷ Chaque rubrique est identifiée par son **id_rubrique**.
- ▷ **id_parent** est l'*id_rubrique* de la rubrique qui contient cette rubrique (zéro si la rubrique se trouve à la racine du site).
- ▷ **titre**, **descriptif**, **texte** parlent d'eux-mêmes.
- ▷ **id_secteur** est l'*id_rubrique* de la rubrique en tête de la hiérarchie contenant cette rubrique. Une rubrique dépend d'une rubrique qui dépend d'une rubrique. . . jusqu'à une rubrique placée à la racine du site ; c'est cette dernière rubrique qui détermine l'*id_secteur*. Cette valeur précalculée permet d'accélérer certains calculs de l'espace public (en effet, les brèves sont classées par secteur uniquement, et non selon toute la hiérarchie).
- ▷ **maj** est un champ technique mis à jour automatiquement par MySQL, qui contient la date de la dernière modification de l'entrée dans la table.
- ▷ **export**, **id_import** sont des champs réservés pour des fonctionnalités futures.

Les articles : spip_articles

id_article	bigint(21)
surtitre	text
titre	text
soustitre	text
id_rubrique	bigint(21)
descriptif	text
chapo	mediumtext
texte	longblob
ps	mediumtext
date	datetime
statut	varchar(10)
id_secteur	bigint(21)
maj	timestamp(14)
export	varchar(10)
images	text
date_redac	datetime
visites	int(11)
referers	blob
accepter_forum	char(3)

- ▷ Chaque article est identifié par son **id_article**.
- ▷ **id_rubrique** indique dans quelle rubrique est rangé l'article.
- ▷ **id_secteur** indique le secteur correspondant à la rubrique susmentionnée (voir le paragraphe précédent pour l'explication de la différence entre les deux).
- ▷ **titre, surtitre, soustitre, descriptif, chapo, texte, ps** parlent d'eux-mêmes.
- ▷ **date** est la date de publication de l'article (si l'article n'a pas encore été publié, c'est la date de création).
- ▷ **date_redac** est la date de publication antérieure si vous réglez cette valeur, sinon elle est égale à «0000-00-00».
- ▷ **statut** est le statut actuel de l'article : `prepa` (en cours de rédaction), `prop` (proposé à la publication), `publie` (publié), `refuse` (refusé), `poubelle` (à la poubelle).
- ▷ **accepter_forum** : permet de régler manuellement si l'article accepte des forums (par défaut, oui).
- ▷ **maj** : même signification que dans la table des rubriques.
- ▷ **export** est un champ réservé pour des fonctionnalités futures.
- ▷ **images** est un champ contenant la liste des images utilisées par l'article, dans un format particulier. Ce champ est généré par `spip_image.php3`.
- ▷ **visites** et **referers** sont utilisés pour les statistiques sur les articles. Le premier est le nombre de chargements de l'article dans l'espace public ; le deuxième contient un extrait de hash des différents referers, afin de connaître le nombre de referers distincts. Voir `inc-stats.php3`.

Les auteurs : `spip_auteurs`

<code>id_auteur</code>	<code>bigint(21)</code>
<code>nom</code>	<code>text</code>
<code>bio</code>	<code>text</code>
<code>email</code>	<code>tinytext</code>
<code>nom_site</code>	<code>tinytext</code>
<code>url_site</code>	<code>text</code>
<code>login</code>	<code>tinytext</code>
<code>pass</code>	<code>tinyblob</code>
<code>statut</code>	<code>tinytext</code>
<code>maj</code>	<code>timestamp(14)</code>
<code>pgp</code>	<code>blob</code>
<code>htpasswd</code>	<code>tinyblob</code>

- ▷ Chaque auteur est identifié par son **id_auteur**.
- ▷ **nom, bio, nom_site, url_site, pgp** sont respectivement le nom de l'auteur, sa courte biographie, son adresse e-mail, le nom et l'URL de son site Web, sa clé PGP. Informations modifiables librement par l'auteur.
- ▷ **email, login** sont son e-mail d'inscription et son login. Ils ne sont modifiables que par un administrateur.
- ▷ **pass** est le hash MD5 du mot de passe.
- ▷ **htpasswd** est la valeur cryptée (i.e. générée par `crypt()`) du mot de passe pour le `.htpasswd`.
- ▷ **statut** est le statut de l'auteur : `0minirezo` (administrateur), `1comite` (rédacteur), `5poubelle` (à la poubelle), `6forum` (abonné aux forums, lorsque ceux-ci sont réglés en mode «par abonnement»).
- ▷ **maj** a la même signification que dans les autres tables.

Les brèves : `spip_breves`

<code>id_breve</code>	<code>bigint(21)</code>
<code>date_heure</code>	<code>datetime</code>
<code>titre</code>	<code>text</code>
<code>texte</code>	<code>longblob</code>
<code>lien_titre</code>	<code>text</code>
<code>lien_url</code>	<code>text</code>
<code>statut</code>	<code>varchar(6)</code>
<code>id_rubrique</code>	<code>bigint(21)</code>
<code>maj</code>	<code>timestamp(14)</code>

- ▷ Chaque brève est identifiée par son **id_breve**.
- ▷ **id_rubrique** est la rubrique (en fait, le secteur) dans laquelle est classée la brève.
- ▷ **titre**, **texte**, **lien_titre**, **lien_url** sont le titre, le texte, le nom et l'adresse du lien associé à la brève.
- ▷ **date_heure** est la date de la brève.
- ▷ **statut** est le statut de la brève : `prop` (proposée à la publication), `publie` (publiée), `refuse` (refusée).
- ▷ **maj** : idem que dans les autres tables.

Les mots-clés : `spip_mots`

<code>id_mot</code>	<code>bigint(21)</code>
<code>type</code>	<code>varchar(100)</code>
<code>titre</code>	<code>text</code>
<code>descriptif</code>	<code>text</code>
<code>texte</code>	<code>longblob</code>
<code>maj</code>	<code>timestamp(14)</code>

- ▷ Chaque mot-clé est identifié par son **id_mot**.
- ▷ Le **type** du mot-clé est le type, ou groupe, choisi pour le mot-clé. En définissant plusieurs types, on définit plusieurs classifications indépendantes (par exemple «sujet», «époque», «pays»...).
- ▷ **titre**, **descriptif**, **texte** parlent d'eux-mêmes.
- ▷ **maj** : idem que dans les autres tables.

Les sites syndiqués : `spip_syndic`

<code>id_syndic</code>	<code>bigint(20)</code>
<code>id_rubrique</code>	<code>bigint(20)</code>
<code>id_secteur</code>	<code>bigint(20)</code>
<code>nom_site</code>	<code>blob</code>
<code>url_site</code>	<code>blob</code>
<code>url_syndic</code>	<code>blob</code>
<code>descriptif</code>	<code>blob</code>

- ▷ Chaque site syndiqué est identifié par son **id_syndic**.
- ▷ **id_rubrique** et **id_secteur** définissent l'endroit dans la hiérarchie du site où viennent s'insérer les contenus syndiqués.

- ▷ **nom_site**, **url_site**, **descriptif** sont le nom, l'adresse et le descriptif du site syndiqué.
- ▷ **url_syndic** est l'adresse du fichier dynamique utilisé pour récupérer les contenus syndiqués (souvent il s'agit de *url_site* suivi de `backend.php3`).

Les articles syndiqués : `spip_syndic_articles`

<code>id_syndic_article</code>	<code>bigint(20)</code>
<code>id_syndic</code>	<code>bigint(20)</code>
<code>titre</code>	<code>text</code>
<code>url</code>	<code>text</code>
<code>date</code>	<code>datetime</code>
<code>lesauteurs</code>	<code>text</code>

- ▷ Chaque article syndiqué est identifié par son **id_syndic_article**.
- ▷ **id_syndic** réfère au site syndiqué d'où est tiré l'article.
- ▷ **titre**, **url**, **date**, **lesauteurs** parlent d'eux-mêmes.

7.3.2 Éléments interactifs

Les messages de forums : `spip_forum`

<code>id_forum</code>	<code>bigint(21)</code>
<code>id_parent</code>	<code>bigint(21)</code>
<code>id_rubrique</code>	<code>bigint(21)</code>
<code>id_article</code>	<code>bigint(21)</code>
<code>id_breve</code>	<code>bigint(21)</code>
<code>date_heure</code>	<code>datetime</code>
<code>titre</code>	<code>text</code>
<code>texte</code>	<code>mediumtext</code>
<code>auteur</code>	<code>text</code>
<code>email_auteur</code>	<code>text</code>
<code>nom_site</code>	<code>text</code>
<code>url_site</code>	<code>text</code>
<code>statut</code>	<code>varchar(8)</code>
<code>ip</code>	<code>varchar(16)</code>
<code>maj</code>	<code>timestamp(14)</code>
<code>id_auteur</code>	<code>bigint(20)</code>
<code>id_message</code>	<code>bigint(21)</code>

- ▷ Chaque message de forum est identifié par son **id_forum**.
- ▷ L'objet auquel est attaché le forum est identifié par son **id_rubrique**, **id_article** ou **id_breve**. Par défaut, ces valeurs sont égales à zéro.
- ▷ Le message parent (c'est-à-dire le message auquel répond ce message) est identifié par **id_parent**. Si le message ne répond à aucun autre message, cette valeur est égale à zéro.
- ▷ **titre**, **texte**, **nom_site**, **url_site** sont le titre et le texte du message, le nom et l'adresse du lien y attaché.

- ▷ **auteur** et **email_auteur** sont le nom et l'e-mail déclarés par l'auteur. Dans le cas des forums par abonnement, ils ne sont pas forcément identiques aux données enregistrées dans la fiche de l'auteur (i.e. dans la table `spip_auteurs`).
- ▷ **id_auteur** identifie l'auteur du message dans le cas de forums par abonnement.
- ▷ **statut** est le statut du message : `publie` (lisible dans l'espace public), `prive` (écrit en réaction à un article dans l'espace privé), `privrac` (écrit dans le forum interne dans l'espace privé), `off` (supprimé ou à valider, selon la modération des forums - a priori ou a posteriori).
- ▷ **ip** est l'adresse IP de l'auteur, dans les forums publics.
- ▷ **maj** a la même signification que dans les autres tables.

Les pétitions : `spip_petitions`

<code>id_article</code>	<code>bigint(21)</code>
<code>email_unique</code>	<code>char(3)</code>
<code>site_obli</code>	<code>char(3)</code>
<code>site_unique</code>	<code>char(3)</code>
<code>message</code>	<code>char(3)</code>
<code>texte</code>	<code>longblob</code>
<code>maj</code>	<code>timestamp(14)</code>

- ▷ **id_article** identifie l'article auquel est associée la pétition (une seule pétition par article).
- ▷ **email_unique**, **site_obli**, **site_unique**, **message** définissent la configuration de la pétition : l'adresse e-mail des signataires doit-elle être unique dans les signatures, l'adresse Web est-elle obligatoire, est-elle unique, un message attendant aux signatures est-il autorisé (`oui` ou `non`).
- ▷ **texte** est le texte de la pétition.
- ▷ **maj** : pareil que dans les autres tables.

Les signatures de pétitions : `spip_signatures`

<code>id_signature</code>	<code>bigint(21)</code>
<code>id_article</code>	<code>bigint(21)</code>
<code>date_time</code>	<code>datetime</code>
<code>nom_email</code>	<code>text</code>
<code>ad_email</code>	<code>text</code>
<code>nom_site</code>	<code>text</code>
<code>url_site</code>	<code>text</code>
<code>message</code>	<code>mediumtext</code>
<code>statut</code>	<code>varchar(10)</code>
<code>maj</code>	<code>timestamp(14)</code>

- ▷ Chaque signature est identifiée par son **id_signature**.
- ▷ **id_article** identifie l'article, donc la pétition sur laquelle est apposée la signature.
- ▷ **nom_email**, **ad_email**, **nom_site**, **url_site** sont le nom, l'adresse e-mail, ainsi que le site Web déclarés par le signataire.
- ▷ **message** est le message éventuellement entré par le signataire.
- ▷ **statut** est le statut de la signature : `publie` (acceptée), `poubelle` (supprimée) ; toute autre valeur donne la valeur de la clé de validation utilisée pour la confirmation par e-mail.
- ▷ **maj** a la même signification que dans les autres tables.

7.3.3 Les relations entre objets

Ces tables ne gèrent aucun contenu, simplement une relation entre les objets présents dans d'autres tables. Ainsi :

- ▷ **spip_auteurs_articles** spécifie la relation entre auteurs et articles. Si un *id_auteur* y est associé à un *id_article*, cela veut dire que l'auteur en question a écrit ou co-écrit l'article (il peut y avoir plusieurs auteurs par article, et vice-versa bien sûr).
- ▷ **spip_mots_articles** définit de même la relation de référencement des articles par des mots-clés.

7.3.4 Gestion du site

La table **spip_meta** est primordiale. Elle contient des couples (*nom*, *valeur*) indexés par le nom (clé primaire) ; ces couples permettent de stocker différentes informations telles que la configuration du site, ou la version installée de SPIP.

La table **spip_forum_cache** est utilisée afin d'adapter le système de cache à l'instantanéité des forums. Pour chaque fichier du cache ayant donné lieu à une requête sur la table *spip_forum*, la table *spip_forum_cache* stocke les paramètres de la requête (article, rubrique, brève et éventuel message parent du forum). Lorsqu'un message est posté, les paramètres du message sont comparés avec ceux présents dans *spip_forum_cache*, et pour chaque correspondance trouvée le fichier cache indiqué dans la table est effacé. Ainsi les messages n'attendent pas le recalcul régulier de la page dans laquelle ils s'insèrent pour apparaître dans l'espace public.

7.3.5 Indexation (moteur de recherche)

Six tables sont utilisées par le moteur de recherche. Elles se divisent en deux catégories.

Le dictionnaire d'indexation : **spip_index_dico**

Chaque mot rencontré au cours de l'indexation est stocké dans cette table, ainsi que les 64 premiers bits de son hash MD5. C'est le mot qui sert de clé primaire, permettant ainsi d'effectuer très rapidement des requêtes sur un début de mot ; on récupère alors le(s) hash satisfaisant à la requête, afin d'effectuer la recherche proprement dite dans les tables d'indexation.

Les tables d'indexation : **spip_index_***

Ces tables, au nombre de cinq, gèrent chacune l'indexation d'un type d'objet : articles, rubriques, brèves, auteurs, mots-clés. Une entrée par mot et par objet est stockée. Chaque entrée contient le hash du mot (en fait les 64 bits de poids fort du hash MD5, cf. ci-dessus), l'identifiant de l'objet indexé (par exemple l'*id_article* pour un article), et le nombre de points associé à l'indexation du mot dans l'objet. Ce nombre de points est calculé en fonction du nombre d'occurrences du mot, pondéré par le champ où ont lieu les occurrences : une occurrence dans le titre d'un article génère plus de points qu'une occurrence dans le corps de l'article.

Le mécanisme d'indexation est expliqué plus en détail ici (section 7.4 page 88).

7.4 Le moteur de recherche

15 juin 2001 par l'équipe de SPIP

SPIP intègre un moteur de recherche désactivé par défaut. Ce moteur, lorsqu'il est activé par un administrateur dans la page de configuration, permet d'effectuer des recherches sur différents types d'informations présentes dans la base de données : les articles, les rubriques, les brèves, les mots-clés et les auteurs.

7.4.1 Principe

Il y a deux grandes façons de faire un moteur de recherche. La première est de chercher tout bêtement dans le type de stockage existant (fichiers HTML, base de données...selon le type de site). Cette méthode est très lente car le type de stockage n'est pas prévu à cet effet.

La seconde méthode, qui a été choisie pour SPIP (et qui est aussi celle de tous les moteurs professionnels), est d'établir un mode de stockage spécifique aux besoins de la recherche. Par exemple, le score de chaque mots d'un article peut être stocké directement afin d'être facilement retrouvé, et d'avoir le score total d'une recherche par une simple addition. L'avantage est que la recherche est très rapide : presque aussi rapide que n'importe quel calcul de page. L'inconvénient est qu'il faut une phase de construction du dit stockage des informations : cela s'appelle l'indexation. L'indexation a un coût en termes de ressources (temps de calcul et espace disque), et elle introduit également un léger décalage temporel entre l'ajout ou la modification d'un contenu, et la répercussion de cet ajout ou de cette modification sur les résultats de recherche.

D'autre part, dans le cas de SPIP, nous sommes obligés d'utiliser PHP et MySQL comme pour le reste du logiciel, ce qui ne permet pas de réaliser un moteur très performant. C'est pourquoi le moteur est désactivé par défaut : il risque de ralentir quelque peu votre site (voir plus loin au chapitre performances pour des indications plus précises), et il sera avantageusement remplacé par un vrai moteur si vous avez la possibilité d'en installer un (si oui, vous pouvez par exemple utiliser ht://Dig⁸, un moteur en logiciel libre).

7.4.2 L'indexation

L'indexation est réalisée selon deux modalités différentes :

1. Si un contenu (article, brève...) est publié ou modifié depuis l'espace privé, alors il est réindexé immédiatement afin de tenir les données à jour.
2. Lors de la visite d'un contenu dans le site public, ce contenu est réindexé uniquement s'il n'a pas déjà été indexé. Pour être plus précis (mais ça ne change rien au principe général), SPIP attend qu'une page soit affichée en utilisant le cache, afin d'éviter que le cumul d'une indexation et d'un calcul de page ne mène à un *timeout* sur les serveurs particulièrement lents.

Le deuxième cas est là pour assurer que, si vous activez le moteur de recherche sur un site déjà rempli, les anciens contenus seront également pris en compte par le moteur ; de même si vous restaurez une sauvegarde (les données d'indexation ne sont pas incluses dans les sauvegardes pour des raisons d'espace disque).

L'indexation traite un à un les différentes données textuelles d'un contenu donné : par exemple, pour un article, le chapo, le descriptif, le titre, le texte... Pour chaque donnée textuelle, le score de chaque mot est calculé en comptant simplement le nombre d'occurrences. A cet effet, les mots de trois caractères ou moins sont ignorés (ils sont, pour la plupart, non significatifs, et alourdiraient la base de données) ; d'autre part, les caractères accentués sont convertis en leurs équivalents non-accentués, pour éviter les problèmes de jeux de caractères et aussi pour permettre d'effectuer des recherches en version non accentuée (en cas de clavier inadapté...). Ensuite, les scores de chaque mot sont cumulés, de façon pondérée, entre les différentes données textuelles du contenu indexé. La pondération permet, par exemple, de donner plus de poids aux mots présents dans le titre d'un article que dans le corps du texte ou le post-scriptum...

Les fonctions d'indexation peuvent être étudiées au sein du fichier `ecrire/inc_index.php3`.

7.4.3 La recherche

La recherche s'effectue simplement en séparant le texte de recherche en ses différents mots ; le même filtre est appliqué que lors de l'indexation : suppression des mots de trois lettres ou moins, et conversion des caractères

⁸<http://htdig.org/>

accentués. Pour chaque contenu recherché, le score des différents mots est ensuite récupéré puis additionné afin d'obtenir le score total. Enfin, les résultats sont en général affichés par ordre décroissant de score, c'est-à-dire de pertinence (mais cela est laissé à la volonté de la personne qui écrit les squelettes de mise en page (section 8.18 page 131)).

7.4.4 Performances

Rapidité

Sur un serveur récent et pas trop chargé, l'indexation d'un texte long (plusieurs dizaines de milliers de caractères) prendra entre une et deux secondes : l'attente est presque imperceptible, comparée aux délais de chargement via le réseau. Les contenus courts sont indexés de façon quasi-instantanée. Bien sûr, ces affirmations doivent être modulées selon la taille du site. Un site vraiment très gros risque de voir les temps d'indexation s'allonger légèrement ; pour relativiser, signalons qu'uZine⁹ comporte, à la date d'écriture de ce texte, environ 400 articles et 400 brèves publiés, et que le moteur de recherche de SPIP ne donne aucun signe de faiblesse.

Par ailleurs, statistiquement, on peut considérer de façon approximative que chaque contenu ne sera indexé qu'une seule fois : compte tenu qu'il y a en général beaucoup plus de visites sur un site que de mises à jour de contenus, le surcroît de charge du serveur apparaît négligeable (sauf en cas de machine très lente).

Qualité

La qualité de l'indexation est plus faible que sous des moteurs de recherche professionnels. PHP étant un langage plutôt lent, la phase d'extraction des mots a dû être simplifiée au maximum pour que les temps d'indexation restent minimales. Par conséquent, les données d'indexation comportent quelques «déchets», c'est-à-dire des morceaux de texte qui ne correspondent pas à de «vrais» mots, mais ont été indexés comme tels (il s'agit souvent de contenus techniques comme des noms de fichiers, ou de passages à la ponctuation malmenée). L'exemple d'uZine (où l'on constate environ 2 % de tels «déchets») nous laisse cependant penser que ces données sont quantité négligeable, d'autant qu'il y a peu de chance qu'elles déclenchent un résultat positif lors d'une recherche.

Quant à la recherche, elle est simplement additive : il n'y a pas d'opérateurs booléens, l'opérateur implicite étant grosso modo un «OU» logique. D'autre part, le fait que les mots de trois lettres ou moins soient ignorés peut vous gêner si vous avez des besoins spécifiques (beaucoup de sigles, d'abréviations. . .).

Espace disque

MySQL n'étant pas spécialement conçu pour le stockage de données d'indexation, l'utilisation du moteur de recherche a tendance à faire beaucoup grossir l'espace disque utilisé par la base de données. Pour donner quelque précision, disons qu'un contenu génère des données d'indexation de taille comprise entre la taille du contenu et le double de celle-ci. Donc, si l'on fait abstraction des données ne donnant pas lieu à indexation (les forums par exemple), l'indexation fait entre doubler et tripler la place prise par la base de données. Cela peut être gênant si la place vous est très comptée.

Si jamais vous désactivez le moteur de recherche afin d'économiser de l'espace disque, n'oubliez pas ensuite d'effacer les données d'indexation (dans la page de sauvegarde/restauration de la base de données) afin de réellement libérer l'espace disque occupé par ces données.

⁹<http://uzine.net/>

7.5 Rapidité du site public

16 juin 2001 par l'équipe de SPIP

Contrairement à la plupart des systèmes de publication gratuits, SPIP intègre un système de cache permettant d'accélérer l'affichage du site public. Quelques pistes pour comprendre ce qui influe sur la rapidité de votre site...

7.5.1 Optimiser un site

Si vous vous inquiétez pour la rapidité de votre site, il est bon de vous intéresser aux pistes suivantes :

- ▷ Votre hébergement Web offre-t-il des performances de bonne qualité ? Evidemment, c'est subjectif. L'expression «mauvaise qualité» recouvre à coup sûr la plupart des hébergeurs gratuits (notamment Free). «Bonne qualité» inclut forcément une machine dédiée (i.e. qui ne sert qu'à votre site) de fabrication récente, mais aussi des hébergeurs commerciaux pas trop au rabais. Entre les deux, ça devient très subjectif, en fonction de vos exigences, de la taille de votre site. ...
- ▷ Si la qualité de votre hébergement laisse à désirer, vous aurez intérêt à ne pas créer de squelettes trop complexes, i.e. qui demandent à SPIP d'afficher trop d'informations différentes. Cela vaut pour tout type d'informations : tout ce qui, dans les squelettes, est susceptible d'être transformé par SPIP en données affichables. Notez, en particulier, que les squelettes fournis par défaut démontrent au maximum les possibilités de SPIP, et par conséquent génèrent des pages assez lourdes.
- ▷ N'oubliez pas non plus de régler les délais d'expiration des différents types de pages. Ainsi, si votre site contient un grand nombre d'articles en archives, vous avez peut-être intérêt à augmenter la durée d'expiration des articles, sinon les articles consultés peu souvent ne bénéficieraient pas du système de cache.

7.5.2 L'influence du cache

La présence du cache change quelque peu la donne en matière de rapidité. Ce n'est pas tant le nombre de visites de votre site qui sera le point critique, que la capacité de votre serveur à recalculer les pages dans le temps imparti au script PHP (en effet, sur la plupart des serveurs, une limite de durée d'exécution par appel de script est fixée afin d'éviter les abus et les erreurs de programmation). Par contre, si la page demandée est dans le cache et n'a pas expiré, la réponse du serveur devrait être quasi-instantanée (dans le cas contraire, votre serveur est vraiment très chargé).

La qualité des performances devient ainsi objectivement mesurable si, lors du recalcul d'une page du site, on obtient un «*timeout*», c'est-à-dire que le serveur a dépassé le temps maximal d'exécution d'un script PHP. Alors il faut soit changer d'hébergement, soit se résoudre à afficher des pages plus simples : pour cela, modifier les squelettes pour afficher moins d'informations sur une même page.

7.5.3 Sur une machine dédiée

Si vous utilisez votre propre machine, il faut vous assurer qu'elle pourra tenir la charge. N'importe quelle machine récente devrait en être capable ; pour donner des chiffres très approximatifs, mettons qu'il faut au minimum un processeur à 200 MHz, et 128 Mo de mémoire.

Par contre, l'utilisation de SPIP, par rapport à d'autres systèmes de publication, permet de mutualiser les ressources techniques entre plusieurs sites. En effet, tant que le cache est utilisé, la machine est peu sollicitée, donc plusieurs sites peuvent cohabiter sans problème (sauf s'il y a vraiment un très grand nombre de visites).

Version 1.2 - Le moteur de spip a été entièrement refondu dans la version 1.2. Sous le nom de code «Pantagruel» il comporte désormais deux phases de «cache» : la première lors de la lecture initiale des squelettes (elle crée

les fichiers `CACHE/skel_nomdusquelette.php3`); la seconde lors de la lecture des données dans la base `mySQL`. Chacune des étapes du calcul des pages a été revue et optimisée. Le gain total permet d'obtenir une génération de page en moins de la moitié du temps nécessaire pour la version 1.0.

7.6 Spip et les feuilles de style

23 septembre 2001 par l'équipe de SPIP

SPIP 1.2 Lorsque l'on utilise les raccourcis typographiques dans les articles dans SPIP (permettant par exemple de mettre en gras, en italique, créer des liens hypertextes, des intertitres, etc.), SPIP produit les balises HTML nécessaires à ces effets, chacune de ces balises étant alors associée à une classe de style CSS.

Par exemple,

```
Ceci est un [lien->http://www.uzine.net]
```

est transformé en code HTML ainsi :

```
Ceci est un <a href="http://www.uzine.net" class="spip_out">lien</a>
```

Le code HTML est ainsi complété par l'appel à un style CSS intitulé «`spip_out`». L'utilisateur peut donc pousser la personnalisation de son interface graphique en définissant ce style «`spip_out`» (couleur différente, fond coloré, police utilisée...).

La plupart des raccourcis typographiques de SPIP peuvent ainsi être paramétrés avec des feuilles de style ; certains sont très utiles, d'autres seront réservés aux webmestres qui souhaitent obtenir des effets exotiques...

7.6.1 Où se trouve la définition de ces feuilles de style ?

Lors de l'installation de SPIP, avec les squelettes fournis en standard, la définition des feuilles de style se trouve dans le fichier :

▷ `spip_style.css`

Vous pouvez le modifier (c'est même conseillé), définir votre propre fichier de feuilles de style, ou intégrer directement des définitions dans vos squelettes.

*Notez bien : la notion de feuille de style, ou *cascading style sheets*, n'est pas une norme propre à SPIP, il s'agit d'un standard du Web. De très nombreuses documentations existent sur ce sujet par ailleurs ; consulter par exemple la page du W3C¹⁰ à ce sujet.*

Afin de suivre la suite de la présente explication, il est vivement conseillé d'ouvrir le fichier «`spip_style.css`» dans un éditeur de textes.

7.6.2 Les liens hypertextes

Les deux premières définitions permettent de modifier le comportement de «`a`» et «`a:hover`» ; très classiques, elles concernent tous les liens affichés sur votre page Web (afficher les liens sans soulignement, et régler le «`survol`» des liens hypertextes).

Viennent ensuite trois définitions propres aux raccourcis typographiques de SPIP : «`a.spip_in`», «`a.spip_out`», «`a.spip_url`».

¹⁰<http://www.w3.org/Style/CSS/>

▷ **a.spip_in** concerne les liens à l'intérieur de votre propre site. Par exemple :

```
<HTML>Ceci est un [lien interne->article1177]</HTML>
```

▷ **a.spip_out** concerne les liens vers l'extérieur de votre site. Par exemple :

```
<HTML>Ceci est un [lien externe->http://www.uzine.net]</HTML>
```

▷ **a.spip_url** traite les adresses URL transformées en lien hypertexte. Par exemple :

```
<HTML>[->http://www.uzine.net]</HTML>
```

(ce raccourci affiche directement l'URL, avec un lien hypertexte vers cette adresse, ainsi : <http://www.uzine.net>).

Le principal intérêt de ces trois styles différents est de permettre de différencier graphiquement les liens internes au site et les liens vers un autre site.

7.6.3 Les intertitres

Les intertitres, créés par le raccourci suivant :

```
<HTML>{{{Un intertitre}}}</HTML>
```

peuvent être définis par le style `h3.spip`. Ce style est sans doute l'un des plus utiles, car il permet de définir la taille, la police et le positionnement des intertitres dans les articles.

Par défaut, la définition en est :

```
h3.spip {
  font-family: Verdana,Arial,Helvetica,sans-serif;
  font-weight: bold;
  font-size: 120%;
  text-align: center;
}
```

7.6.4 Code et cadre

Les éléments de code, définis par le raccourci :

```
<HTML>&lt;code>Du code dans le texte&lt;/code></HTML>
```

sont paramétrés par le style `.spip_code`. Peu utilisé.

Introduit dans **SPIP 1.3**, la balise `<cadre>...</cadre>` permet de présenter du code source dans un tableau (élément de formulaire) dans lequel il est facile copier-coller le texte. La feuille de style associée est : `.spip_cadre`, définie ainsi par défaut :

```
.spip_cadre {
  width : 100%;
  background-color: #FFFFFF;
  padding: 5px;
}
```

7.6.5 Les notes de bas de page

Les notes de bas de page, définies par le raccourci :

```
<HTML>Le texte[[Une note de bas de page]]</HTML>
```

sont paramétrés par le style **p.spip_note**. Souvent inutile, puisque les notes peuvent être modifiées directement en HTML lors de l'emploi de la balise #NOTES dans vos squelettes.

7.6.6 Les tableaux

Les tableaux sont définis dans SPIP de la façon suivante :

```
| {{Nom}} | {{Date de naissance}} | {{Ville}} |  
| Jacques | 5/10/1970 | Paris |  
| Claire | 12/2/1975 | Belfort |  
| Martin | 1/31/1957 | Nice |  
| Marie | 23/12/1948 | Perpignan |
```

ce qui donne :

Nom	Date de naissance	Ville
Jacques	5/10/1970	Paris
Claire	12/2/1975	Belfort
Martin	1/31/1957	Nice
Marie	23/12/1948	Perpignan

Les feuilles de style permettent de paramétrer finement l'affichage de tels tableaux :

```
table.spip {  
}  
  
table.spip tr.row_first {  
    background-color: #FCF4D0;  
}  
  
table.spip tr.row_odd {  
    background-color: #C0C0C0;  
}  
  
table.spip tr.row_even {  
    background-color: #F0F0F0;  
}  
  
table.spip td {  
    padding: 1px;  
    text-align: left;  
    vertical-align: center;  
}
```

▷ **table.spip** permet de modifier le comportement général du tableau (notamment sa position, à gauche, centré...);

- ▷ **table.spip tr.row_first** définit le comportement de la «première ligne» du tableau (ici en jaune) (pour que la «première ligne» soit prise en compte, il faut que les éléments qu'elle contient soient placés en gras) ;
- ▷ **table.spip tr.row_odd** pour les lignes paires ;
- ▷ **table.spip re.row_even** pour les lignes impaires ;
- ▷ **table.spip td** permet de modifier le comportement des cases du tableau.

Un des intérêts repose sur le choix de couleurs différentes pour «row_odd» et «row_even», permettant de faire une présentation de couleurs alternées pour les lignes d'un tableau (ici, gris clair et gris foncé).

7.6.7 Ligne de séparation horizontale

Une ligne de séparation horizontale, définie par :

```
<HTML>----</HTML>
```

peut être modifiée par : **hr.spip**.

7.6.8 Gras et italique

Le gras et l'italique sont définis par les raccourcis :

```
<HTML>Du texte {{en gras}}, du texte {en italique}</HTML>
```

Ils peuvent être modifiés par les styles : **b.spip** et **i.spip**. Styles peu utiles.

7.6.9 Les paragraphes

Les paragraphes créés par SPIP (en laissant des lignes vides entre les paragraphes) peuvent être modifiés par le style : **p.spip**.

A priori, peu utile, car on peut directement paramétrer le comportement des éléments de texte en HTML.

7.6.10 Les formulaires

Dans l'espace public, différents formulaires sont utilisés pour le moteur de recherche interne, l'interface de rédaction des messages des forums, les inscriptions à l'espace privé...

Les feuilles de style sont : **.forml**, **.spip_encadrer**, **spip_bouton**, **.formrecherche**.

Par défaut, ils sont définis ainsi :

```
.forml { width: 100% ; background-color: #FFDDAA;}
.spip_encadrer { background-color: #EEEEEE; }
.spip_bouton { background-color: #FFCC00;}
.formrecherche { width: 100% ; background-color: #FFDDAA;}
```

- ▷ **.forml** définit les «cases» de texte des formulaires ; utile pour définir la largeur de ces cases, et la couleur du fond ;
- ▷ **.spip_encadrer** ; lorsqu'un formulaire propose différentes «parties», la séparation entre ces différentes parties peut être paramétrées avec ce style (par exemple, encadrer chaque partie, créer un espace avant ou après...);
- ▷ **.spip_bouton** modifie l'aspect du bouton de validation du formulaire ;
- ▷ **.formrecherche** modifie l'aspect de la case «Rechercher» du moteur de recherche.

7.6.11 Conclusion

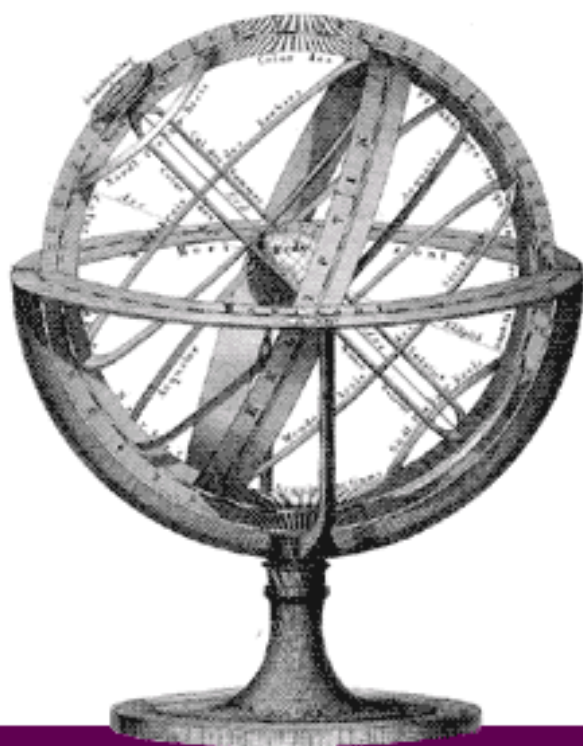
Vous remarquerez que, par défaut, certaines feuille de style ne sont pas définies. Elles peuvent être considérées comme très accessoires (réservées aux webmestres voulant obtenir des effets graphiques très spécifiques).

En règle générale, les styles qui provoquent des modifications graphiques spectaculaires sur un site, par ailleurs simples à paramétrer, sont celles qui concernent :

- ▷ les liens de l'ensemble de la page, **a** et **a :hover**,
- ▷ le comportement des intertitres, **h3.spip**,
- ▷ les formulaires.

Chapitre 8

Manuel de référence



Le manuel

de référence

8.1 Principe général

2 juin 2001 par l'équipe de SPIP

Tout le contenu d'un site géré sous SPIP est installé dans une base de données MySQL (dans la partie protégée par mot de passe */ecriture*). Pour présenter ces informations aux visiteurs du site, il faut donc réaliser l'opération qui consiste à récupérer les informations, à les organiser et à les mettre en page, afin de délivrer une page HTML.

Cette opération est traditionnellement assez pénible :

- ▷ il faut connaître PHP et MySQL, et effectuer des routines relativement complexes ;
- ▷ l'intégration de telles routines dans une mise en page HTML élaborée est assez pénible ;
- ▷ le recours systématique à des requêtes MySQL à chaque affichage d'une page est gourmand en ressources, ralentit la visite et, dans des cas extrêmes, provoque des plantages du serveur.

SPIP propose ici une solution complète pour contourner ces difficultés :

- ▷ la mise en page du site est effectuée au moyen de pages HTML nommées *squelettes*, contenant des instructions simplifiées permettant d'indiquer où et comment se placent les informations tirées de la base de données dans la page ;
- ▷ un système de cache permet de stocker chaque page et ainsi d'éviter de provoquer des appels à la base de données à chaque visite. Non seulement la charge sur le serveur est réduite, la vitesse très largement accélérée, de plus un site sous SPIP reste consultable même lorsque la base MySQL est plantée.

8.1.1 Pour chaque type de document, un couple de fichiers

L'intérêt (et la limite) d'un système de publication automatisé, c'est que l'on ne va pas redéfinir une interface différente en HTML pour chaque page isolée. Par exemple, toutes les articles bénéficieront de la même interface, simplement le système placera des informations différentes dans ce graphisme (on verra plus loin que SPIP autorise cependant une certaine souplesse).

L'avantage de cette manière de procéder est évident : on définit un format-type (squelette) pour, par exemple, tous les articles, et le système fabriquera chaque page individuelle en plaçant automatiquement le titre, le texte, les liens de navigation ... de chaque article.

Pour chaque type de document, SPIP vous demande deux fichiers : un fichier `.php3` et un fichier `.html`. Lors de l'installation de SPIP, vous trouverez ainsi les couples :

«`article.php3 / article.html`», «`rubrique.php3 / rubrique.html`», etc.

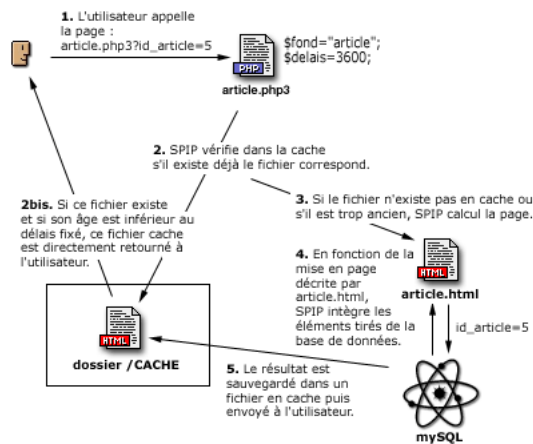
Vous pouvez naturellement modifier ces couples, et en créer d'autres.

8.1.2 Le principe de fonctionnement du cache

L'appel d'une page spécifique se fait par l'intermédiaire du fichier `.php3`.

Par exemple, pour appeler l'article 5, l'URL correspond est :

```
article.php3?id_article=5
```



- 1 Le fichier appelé est donc `article.php3`, avec en variable `id_article=5`.
- 2 Le fichier `article.php3` est un fichier PHP ; sa première tâche consiste à vérifier dans le dossier `/CACHE` sur votre serveur, s'il existe déjà un fichier correspondant à cet article.
- 2bis Si un tel fichier existe dans `/CACHE`, `article.php3` vérifie sa date de création. Si ce fichier est suffisamment récent, il le retourne directement à l'utilisateur. Le processus de consultation est alors terminé.
- 3 S'il n'existe pas un tel fichier dans `/CACHE` (première visite sur cet article, par exemple), ou si son âge est trop ancien, SPIP démarre le calcul de cette page.
- 4 C'est alors la page `article.html` qui est chargée et analysée. Cette page contient la mise en page correspondant à ce type de document. Il s'agit de HTML complété d'indications permettant de placer les éléments tirés de la base de données. En fonction des éléments requis par `article.html`, SPIP va chercher les informations nécessaires tirées de la base de données MySQL et les insérer aux endroits prévus.
- 5 Un fichier est ainsi fabriqué, à partir de la description contenue dans `article.html`, avec les éléments tirés de la base de données. Ce fichier est alors sauvegardé dans le dossier `/CACHE` et renvoyé au visiteur.

Lors d'une visite suivante, si le délais entre les deux visites est suffisamment court, c'est donc ce nouveau fichier stocké dans `/CACHE` qui est retourné, sans avoir à faire un nouveau calcul à partir de la base de données. En cas de plantage de la base de données, c'est forcément le fichier en cache qui est retourné, même s'il est «trop âgé».

Remarque. On voit ici que chaque page du site est mise en cache individuellement, et chaque recalcul est provoqué par les visites du site. Il n'y a pas, en particulier, un recalcul de toutes les pages du site d'un seul coup à échéance régulière (ce genre de «grosse manoeuvre» ayant le bon goût de surcharger le serveur et de le faire parfois planter).

8.1.3 Le fichier .PHP3

Le fichier `.php3` est très simple. Par exemple, `article.php3` contient uniquement :

```
<?
$fond = "article";
$delaix = 24 * 3600;

include ("inc-public.php3");
?>
```

Son seul but est donc de fixer deux variables (`$fond` et `$delais`) et d'appeler le fichier qui déclenche le fonctionnement de SPIP (`inc-public.php3`).

La variable `$fond` est le nom du fichier qui contient la description de la mise en page (*le squelette*). Ici, puisque `$fond="article"`, le fichier de description sera contenu dans `article.html`. (Notez bien que, dans la variable `$fond`, on n'indique pas la terminaison `.html`.)

Remarque.

L'intérêt de choisir soi-même le nom du fichier de squelette (que l'on aurait pu déduire automatiquement du nom du fichier `.php3`) est, si nécessaire, d'utiliser un autre nom. Cela pour ne pas écraser, éventuellement, des fichiers HTML qui subsisteraient d'une ancienne version du site que l'on ne souhaite pas supprimer. S'il existe, d'une ancienne version du site, un fichier `article.html` que l'on ne souhaite pas effacer, on utilisera par exemple un fichier squelette pour SPIP intitulé `spip-article.html`, et on fixera dans `article.php3` : `$fond="spip-article"`.

La variable `$delais` est l'âge maximum pour l'utilisation du fichier stocké en `/CACHE`. Ce délai est fixé en secondes. Un délai de 3600 correspond donc à une heure ; un délai de `24*3600` est donc de 24 heures.

On jouera sur cette valeur en fonction de la fréquence des ajouts de contenu du site (nouveaux articles, nouvelles brèves ...). Un site actualisé plusieurs fois par jour pourra adopter un délai d'une heure ; un site publiant quelques articles par semaine pourra adopter un délai nettement plus long. De même, le contenu des pages est important : si vous insérez la syndication du contenu de sites fréquemment mis à jour, vous souhaiterez sans doute adapter votre propre délais à celui des sites référencés.

Remarque.

Certains webmestre succombent à la tentation de fixer des délais dérisoires (quelques secondes), pour que le site corresponde très exactement, à chaque instant, aux dernières modifications de la base de données. Dans ce cas, vous perdez tous les avantages du système de cache : les visites sont nettement ralenties et, à l'extrême, sur des sites très fréquentés, vous pouvez provoquer des plantages de la base de données (ou vous faire virer par votre hébergeur parce que vous monopolisez la puissance de sa machine ...).

Remarque.

Une autre raison qui semble pousser les webmestres à fixer des délais très courts est la présence des forums sur leur site. En effet, pour que les contributions s'affichent dès qu'elles sont postées, ils pensent nécessaire de réduire le délais de recalcul. C'est inutile : SPIP gère cet aspect automatiquement ; lorsqu'une contribution à un forum est postée, la page correspondante est effacée du cache, et recalculée immédiatement, quel que soit le délai fixé pour cette page.

8.1.4 Le fichier .HTML

Dans SPIP, nous appelons les fichiers `.html` les **squelettes**. Ce sont eux qui décrivent l'interface graphique de vos pages.

Ces fichiers sont rédigés directement en HTML, auquel on ajoute des instructions permettant d'indiquer à SPIP où il devra placer les éléments tirés de la base de données (du genre : «placer le titre ici», «indiquer à cet endroit la liste des articles portant sur le même thème» ...).

Les instructions de placement des éléments sont rédigées dans un langage spécifique, qui fait l'objet du présent manuel d'utilisation. Ce langage constitue par ailleurs la seule difficulté de SPIP.

«**Encore un langage ?**» Hé oui, il va vous falloir apprendre un nouveau langage. Il n'est cependant pas très compliqué, et il permet de créer des interfaces complexes très rapidement. Par rapport au couple PHP/mysql, vous verrez qu'il vous fait gagner un temps fou (surtout : il est beaucoup plus simple). C'est un *markup language*, c'est-à-dire un langage utilisant des balises similaires à celles du HTML.

Remarque.

De la même façon que l'on apprend le HTML en s'inspirant du code source des sites que l'on visite, vous pouvez vous inspirer des squelettes utilisés sur d'autres sites fonctionnant sous SPIP. Il suffit d'aller chercher le fichier «.html» correspondant. Par exemple, vous pouvez voir le squelette des articles d'uZine¹ (visualisez le code source pour obtenir le texte du squelette).

8.1.5 Une interface différente dans le même site

Tout d'abord, notez qu'il est possible de créer des couples de fichiers pour le même élément logique (articles, rubriques, ...). Par exemple, vous pouvez créer des fichiers pour visiter un même article avec des interfaces différentes : `article.php3/html` pour le format normal, `imprimer.php3/html` pour le même article dans un format adapté à l'impression, `article-texte.php3/html` pour l'article dans un format texte (adapté aux mal-voyants par exemple), `article-heavy.php/html` avec une interface lourdingue adaptée au haut-débit, etc.

Une interface différente selon les rubriques.

Vous pouvez, pour un même type de document, créer des squelettes différents selon les rubriques du site. Il s'agit de créer simplement de nouveaux fichiers `.html` en fonction des rubriques (inutile, ici, de modifier le fichier `.php3`, on se contente de jouer sur les noms des fichiers squelettes).

Il suffit de compléter le nom du fichier squelette de «-numéro» (tiret suivi d'un numéro de rubrique). Par exemple, si vous créez un fichier : `article-60.html`, tous articles contenus dans la rubrique 60 utiliseront ce squelette (et non plus le squelette par défaut `article.html`). Notez bien : le numéro indiqué est celui d'une *rubrique*. Si cette rubrique 60 contient des sous-rubriques, les articles contenus dans ces sous-rubriques utiliseront également ce nouveau squelette.

Remarque.

Dans notre exemple, on aura certainement également intérêt à créer :
un `rubrique-60.html`, un `breve-60.html`, etc.
pour accompagner le changement de mise en page de cette rubrique.

Une interface pour une seule rubrique.

(SPIP 1.3) On peut créer une interface qui s'applique à une rubrique, *mais pas à ses sous-rubriques*. Pour cela, il faut créer un fichier : `article=60.html`, qui s'appliquera uniquement aux articles de la rubrique 60, mais pas à ses sous-rubriques.

8.1.6 Que peut-on mettre dans un fichier .HTML

Les fichiers `.html` sont essentiellement des fichiers «texte», complétés d'instructions de placement des éléments de la base de données.

SPIP analyse uniquement les instructions de placement des éléments de la base de données (codées selon le langage spécifique de SPIP) ; il se contrefiche de ce qui est placé dans ce fichier et qui ne correspond pas à ces instructions.

Leur contenu essentiel est donc du HTML. Vous déterminez la mise en page, la version du HTML désiré, etc. Vous pouvez évidemment y inclure des feuilles de style (CSS), mais également du JavaScript, du Flash ... en gros : tout ce qu'on place habituellement dans une page Web.

¹<http://www.uzine.net/article.html>

Mais vous pouvez également (tout cela n'est jamais que du texte) créer du XML (par exemple, «backend.php3/html» génère du XML).

Plus original : toutes les pages retournées au visiteur sont tirées du /CACHE par une page écrite en PHP. Vous pouvez donc inclure dans vos squelette des instructions en PHP, elles seront exécutées lors de la visite. Utilisée de manière assez fine, cette possibilité permet une grande souplesse à SPIP, que vous pouvez ainsi compléter (par exemple ajouter un compteur, etc.), ou même faire évoluer certains éléments de mise en page en fonction des informations tirées de la base de données.

8.2 Des boucles et des balises

1er juin 2001 par l'équipe de SPIP

Tout ce qui suit concerne désormais le langage de description de la mise en page des *squelettes* dans SPIP ; si vous avez bien compris l'explication précédente (section 8.1 page 99), vous savez que nous travaillons donc dans les fichiers «.html».

La présente documentation est volontairement technique (il s'agit ici de références techniques) ; vous pouvez préférer commencer par notre guide Pas à pas (chapitre 9 page 143), plus didactique, et revenir ensuite ici pour une documentation plus précise.

8.2.1 Des boucles

La notion de base du langage de SPIP est la *boucle*.

La logique de la boucle

Une base de données, classiquement, c'est une liste d'éléments : ici, une liste des articles, une liste des rubriques, une liste des auteurs, etc. Pour «fabriquer» le site, on va donc extraire de cette liste certains de ses éléments :

- ▷ à la base, on veut extraire *un seul élément* d'une liste ; par exemple, afficher l'article désiré ;
- ▷ mais il est fréquent d'extraire *plusieurs éléments* d'une liste ; par exemple, dans la page d'une rubrique, on veut afficher tous les articles contenus dans cette rubrique, ainsi que toutes les sous-rubriques contenues dans cette rubrique ;
- ▷ plus subtile : il arrive fréquemment qu'il n'y ait pas d'éléments satisfaisants à tel ou tel endroit ; SPIP doit alors pouvoir gérer l'éventualité de l'absence de ces éléments ; par exemple, le squelette de l'affichage des rubriques demande l'affichage de toutes les sous-rubriques contenues une rubrique ; que faire, alors, s'il n'y a pas sous-rubriques dans cette rubrique spécifique ?

Ces trois situations sont traitées par la notion unique de *boucle*, qui permet à la fois de gérer l'affichage d'un seul élément, de plusieurs éléments successifs, ou l'absence d'éléments.

Le système de boucle permet, dans un code unique :

- ▷ d'indiquer à quel endroit du code HTML on a besoin de quel type d'élément (à tel endroit on veut récupérer la liste des articles, à tel endroit on veut inclure la liste des sous-rubriques ...) ;
- ▷ de prévoir l'affichage d'un élément unique ;
- ▷ d'indiquer comment est affichée une liste de plusieurs éléments ;
- ▷ de déterminer ce qu'on affiche lorsqu'il n'y a aucun élément correspondant.

Analogie avec la programmation en PHP/mysql

Ceux qui ont déjà programmer des requêtes mySQL en PHP savent que le traitement se déroule en deux temps :

- ▷ la construction de la syntaxe de la requête (qui consiste à dire «je veux récupérer la liste des articles contenus dans telle rubrique . . .»);
- ▷ l'analyse et l'affichage des résultats au travers d'une boucle.

Ce sont ces deux événements qui sont gérés, dans SPIP, au travers des boucles.

8.2.2 Les balises SPIP

Grâce aux boucles, on a donc récupéré des éléments uniques ou des listes d'éléments : par exemple une liste d'articles ou une liste de rubriques . . .

Cependant, chaque élément de telles listes est composé de plusieurs éléments précis : par exemple un article se compose d'un titre, d'un surtitre, d'un sous-titre, d'un texte d'introduction (chapeau), d'un texte principal, d'un post-scriptum, etc. Il existe ainsi des balises spécifiques à SPIP, permettant d'indiquer précisément à quel endroit on affiche des éléments : «placer le titre ici», «placer le texte ici» . . .

8.2.3 Les balises à l'intérieur des boucles

Voici, au travers d'un cas classique, le principe de fonctionnement général d'une boucle accompagnée de ses balises (attention, ça n'est pas du langage SPIP, c'est une description logique) :

BOUCLE : afficher la liste des articles de cette rubrique

- ▷ afficher ici le titre de l'article
- ▷ afficher le sous-titre
- ▷ afficher le texte

Fin de la BOUCLE

Cette boucle, analysée par SPIP, peut donner trois résultats différents.

Il n'y a aucun article dans cette rubrique.

Dans ce cas, bien évidemment, aucun des éléments «afficher ici . . . (titre, sous-titre . . .)» n'est utilisé. En revanche, si on l'a prévu, on peut afficher un message du genre «Il n'y a pas d'article».

Il y a un seul article dans cette rubrique.

Dans ce cas, très simplement, la page HTML est construite sur le modèle de la boucle :

- ▷ Titre de l'article
- ▷ Sous-titre
- ▷ Texte de l'article

Il y a plusieurs articles dans cette rubrique.

La description de la mise en page («placer ici . . .») va alors être calculée successivement pour chacun des articles. Ce qui donne simplement :

- ▷ Titre de l'article 1
- ▷ Sous-titre de l'article 1
- ▷ Texte de l'article 1
- ▷ Titre de l'article 2
- ▷ Sous-titre de l'article 2
- ▷ Texte de l'article 2
- ▷ ...
- ▷ Titre du dernier article
- ▷ Sous-titre du dernier article
- ▷ Texte du dernier article

La suite de ce guide de référence se construira donc de la manière suivante :

- ▷ syntaxe générale des boucles (section 8.3 page 105) ;
- ▷ syntaxe générale des balises de SPIP (section 8.4 page 109) ;
- ▷ et, ensuite, une page spécifique à chaque type de boucles, indiquant quelles balises on peut y utiliser.

8.3 La syntaxe des boucles

28 mai 2001 par l'équipe de SPIP

8.3.1 Syntaxe de base

La syntaxe simplifiée d'une boucle est la suivante :

```
<BOUCLEn(TYPE){critère1}{critère2}...{critèrex}>
Code HTML + balises SPIP
</BOUCLEn>
```

On a vu, dans l'explication sur les boucles et les balises (section 8.2 page 103), que le *Code HTML + balises SPIP* se répétait autant de fois que la boucle obtenait d'éléments tirés de la base de données (c'est-à-dire une fois, plusieurs fois, ou zéro fois).

La ligne importante, ici, est :

```
<BOUCLEn(TYPE){critère1}{critère2}...{critèrex}>
```

- ▷ L'élément `BOUCLE` est l'ordre indiquant qu'il s'agit d'une boucle SPIP ; on ne peut donc pas le modifier ; dit autrement, toutes les boucles de SPIP commencent par l'instruction `BOUCLE`.
- ▷ L'élément `n` est, au choix, le nom de la boucle, ou le numéro de la boucle. Cet élément est choisi par le webmestre, pour chaque boucle qu'il utilise. On verra plus loin qu'il est possible (c'est même tout l'intérêt de la manoeuvre) d'utiliser plusieurs boucles dans un même squelette : leur donner un nom est donc indispensable pour les identifier.

Si vous décidez de numéroter vos boucles, la syntaxe devient par exemple (pour la boucle 5) :


```
<BOUCLE5...>
...
</BOUCLE5>
```

Si vous décidez de donner un nom à vos boucles (c'est généralement plus pratique, votre code est plus lisible), il faut impérativement faire précéder ce nom par le symbole «_» (que l'on appelle habituellement *underscore*). Par exemple :

```
<BOUCLE_sousrubriques...>
...
</BOUCLE_sousrubriques>
```

- ▷ L'élément (TYPE). Cet élément est primordial : il indique quel type d'éléments on veut récupérer. La syntaxe est importante : le TYPE est indiqué entre parenthèses (sans espaces), en majuscules, et ce TYPE doit correspondre obligatoirement à l'un des types prévus dans SPIP (qu'on trouvera dans la présente documentation) : ARTICLES, RUBRIQUES, AUTEURS, BREVES, etc.

Pour l'exemple précédent, on aurait donc :

```
<BOUCLE_sousrubriques(RUBRIQUES)...>
...
</BOUCLE_sousrubriques>
```

- ▷ Les critères {critère1}{critère2}... Ils indiquent à la fois selon quels critères on veut sélectionner les éléments de la base de données (afficher les sous-rubriques incluses dans cette rubrique, afficher les autres rubriques installées au même niveau hiérarchique que la présente rubrique...), et la façon dont on va classer ou sélectionner les éléments (classer les articles selon leur date, selon leur titre... afficher uniquement les 3 premiers articles, afficher la moitié des articles...). Comme on peut combiner les critères, on peut très aisément fabriquer des requêtes très puissantes, du genre «afficher la liste des 5 articles les plus récents écrits par cet auteur».

```
<BOUCLE_meme_auteur(ARTICLES){id_auteur}{par date}{inverse}{0,5}>
...
</BOUCLE_meme_auteur>
```

Les différents critères et leur syntaxe seront explicités dans la suite, pour chaque type de boucle (certains critères fonctionnent pour tous les types de boucles (section 8.16 page 126), certains sont spécifiques à certaines boucles).

8.3.2 Syntaxe complète

Le syntaxe indiquée précédemment peut être complétée par des éléments conditionnels. En effet, la boucle précédente affiche successivement les éléments contenus à l'intérieur de la boucle. SPIP permet de plus d'indiquer ce qu'on affiche avant et après la boucle au cas où elle contient un ou plusieurs résultats, et ce qu'on affiche s'il n'y a aucun élément.

Cela donne :

```
<Bn>

Code HTML optionnel avant

<BOUCLEn(TYPE){critère1}{critère2}...{critèrex}>
```

```
Code HTML + balises SPIP
</BOUCLEn>

Code HTML optionnel après
</Bn>

Code HTML alternatif
</Bn>
```

Le *code optionnel* avant (précédé de `<Bn>`) n'est affiché que si la boucle contient au moins une réponse. Il est affiché avant les résultats de la boucle.

Le *code optionnel* après (terminé par `</Bn>`) n'est affiché que si la boucle contient au moins une réponse. Il est affiché après les résultats de la boucle.

Le *code alternatif* (terminé par `</Bn>`) est affiché à la place de la boucle (et donc également à la place des codes optionnels avant et après) si la boucle n'a trouvé aucune réponse.

Par exemple, le code :

```
<B1>

  Cette rubrique contient les éléments suivants:
  <UL>

<BOUCLE1(ARTICLES){id_rubrique}>

  <LI>#TITRE
</BOUCLE1>

  </UL>

</B1>

  Cette rubrique ne contient pas d'article.

</B1>
```

donne les résultats suivants :

▷ **Il y a un seul article :**

```
Cette rubrique contient les éléments suivants:
<UL>
  <LI> Titre de l'article
</UL>
```

▷ **Il y a plusieurs articles :**

```
Cette rubrique contient les éléments suivants:
<UL>
```

```
<LI> Titre de l'article 1
<LI> Titre de l'article 2
...
<LI> Titre du dernier article
</UL>
```

▷ **Il n'y a aucun article :**

```
Cette rubrique ne contient pas d'article.
```

8.3.3 Des critères d'environnement en cascade

Chaque boucle effectue la sélection des éléments tirés de la base de données en fonction de critères. Ces critères correspondent à l'environnement dans lequel se trouve la boucle.

Par exemple : si on prévoit une boucle du genre «Afficher les articles inclus dans cette rubrique», il faut savoir de quelle rubrique il s'agit. C'est ce que l'on nomme l'environnement.

L'environnement fourni par l'URL

Lorsque l'on visite une page d'un site SPIP, son adresse contient généralement une variable. Par exemple :

```
rubrique.php3 ?id_rubrique=15
```

Cette variable définit donc un premier environnement : la boucle «Afficher les articles inclus dans cette rubrique» doit alors être comprise comme «Afficher les articles de la rubrique 15».

Clairement, avec le même code de squelette, si on appelle l'adresse :

```
rubrique.php3 ?id_rubrique=7
```

l'interprétation de cette boucle deviendra «Afficher les articles de la rubrique 7».

L'environnement fourni par les autres boucles

À l'intérieur d'une boucle, l'environnement est modifié par chaque élément de la boucle. En plaçant des boucles les unes à l'intérieur des autres, on hérite ainsi d'environnements imbriqués les uns dans les autres.

Ainsi, dans la structure suivante :

```
<BOUCLE_articles: afficher les articles de cette rubrique>

  <LI>Afficher le titre de l'article
  <BOUCLE_auteurs: afficher les auteurs de cet article>
  Nom de l'auteur
  </BOUCLE_auteurs>

</BOUCLE_articles>
```

On doit comprendre que :

- ▷ la première boucle (BOUCLE_articles) affiche les articles en fonction de la rubrique, selon l'environnement fourni par l'URL (id_rubrique=15 par exemple) ;

- ▷ dans cette boucle, on obtient un ou plusieurs articles ;
- ▷ «à l'intérieur» de chacun de ces articles, on a un environnement différent (celui de l'article, c'est-à-dire, par exemple, `id_article=199`) ;
- ▷ la seconde boucle (`BOUCLE_auteurs`), qui est installée à l'intérieur de la première boucle, dépend pour chacun de ses exécutions successives (elle est exécutée pour chaque article de la première boucle) : «afficher les auteurs de cet article» devient successivement «afficher les auteurs du premier article», «du deuxième article» et ainsi de suite.

On voit que, par l'imbrication de boucles successives, on obtient différentes boucles, incluses les unes dans les autres, qui dépendent du résultat des boucles dans lesquelles elles sont situées. Et finalement, la toute première boucle (celle qui contient toutes les autres) dépend d'un paramètre fixé dans l'adresse de la page.

8.3.4 Boucles incluses et boucles successives

Si l'on peut inclure des boucles les unes à l'intérieur des autres (chaque boucle incluse dépendant alors du résultat de la boucle à l'intérieure de laquelle elle est installée), on peut tout aussi bien installer des boucles les unes à la suite des autres ; des boucles successives n'influent pas les unes les autres.

Par exemple, la page d'une rubrique est typiquement constituée des éléments suivants :

```
<BOUCLE_rubrique (RUBRIQUES) {id_rubrique}>

  Titre de la rubrique
  <BOUCLE_articles (ARTICLES) {id_rubrique}>
  <LI> Titre de l'article
  </BOUCLE_articles>

  <BOUCLE_sous_rubriques (RUBRIQUES) {id_rubrique}>
  <LI> Titre de la sous-rubrique
</BOUCLE_sous_rubriques>

</BOUCLE_rubrique>

  Il n'y a pas de rubrique à cette adresse.

</B_rubrique>
```

La première boucle (`BOUCLE_rubrique`) dépend de la variable passée dans l'URL de la page (`id_rubrique=15` par exemple).

Les boucles suivantes (`BOUCLE_articles` et `BOUCLE_sous_rubriques`) sont installées à l'intérieur de la première boucle. Ainsi, s'il n'existe pas de rubrique 15, la première boucle ne donne aucun résultat (le code alternatif «Il n'y a pas de rubrique ...» est affiché), et donc les deux rubriques incluses sont totalement ignorées. Mais s'il existe une rubrique 15, ces deux sous-boucles seront analysées.

On constate également que ces deux boucles se présentent *l'une après l'autre*.

Ainsi, elles fonctionnent en fonction de la première boucle, mais indépendamment l'une de l'autre.

S'il n'y a pas d'articles dans la rubrique 15 (`BOUCLE_articles`), on affichera tout de même la liste des sous-rubriques de la rubrique 15 (`BOUCLE_sous_rubriques`) ; et inversement.

8.4 La syntaxe des balises SPIP

Chaque type de boucle permet de sélectionner des éléments de la base de données de SPIP : des articles, des rubriques, des brèves, etc. Chacun de ces éléments est lui-même constitué d'éléments précis : un titre, une date, un texte, etc. A l'intérieur d'une boucle, il faut donc pouvoir indiquer à quel endroit du code HTML on place tel ou tel de ces éléments précis.

Pour cela, on va utiliser des balises SPIP.

8.4.1 Fonctionnement simplifié

Une balise SPIP se place à l'intérieur d'une boucle (puisque'il faut savoir si l'on veut récupérer un élément d'un article, d'une rubrique, etc.). Le nom de ces balises est généralement simple, et nous fournirons, pour chaque type de boucle, la liste complète des balises que l'on peut utiliser.

Une balise est toujours précédée du signe dièse (#).

Par exemple, affichons une liste de noms d'articles :

```
<BOUCLE_articles (ARTICLES) {id_rubrique}>
<LI> #TITRE
</BOUCLE_articles>
```

Lorsque la boucle sera exécutée, la balise SPIP #TITRE sera à chaque fois remplacée par le titre de l'article en question :

```
<LI> Titre de l'article 1
<LI> Titre de l'article 2
...
<LI> Titre du dernier article
```

Rien de bien compliqué : on se contente d'indiquer à l'intérieur du code HTML le nom de l'élément désiré, et celui-ci est remplacé par le contenu tiré de la base de données.

8.4.2 Codes optionnels

Dans la pratique, un élément de contenu est souvent accompagné de code HTML qui ne doit s'afficher que si cet élément existe, faute de quoi la mise en page devient imprécise.

Par exemple : il existe une balise SPIP pour indiquer le surtitre d'un article. Or de nombreux articles n'ont pas de surtitre.

Complétons l'exemple précédent :

```
<BOUCLE_articles (ARTICLES) {id_rubrique}>
<LI> #SURTITRE<BR>
#TITRE
</BOUCLE_articles>
```

qui, classiquement, nous donne une liste d'articles, avec désormais l'indication du titre et du surtitre de chaque article. Mais que se passe-t-il si l'article n'a pas de surtitre ? On obtient le code : «
», c'est-à-dire une petite puce suivie d'une ligne blanche.

Ce que nous devons faire : n'afficher le code «
» que si un surtitre existe pour l'article.

La syntaxe de la balise SPIP devient alors :

```
[ texte optionnel avant (#BALISE) texte optionnel après ]
```

La balise qui détermine l'option est placée entre parenthèses, et l'ensemble du texte conditionnel entre crochets. Le *texte optionnel avant* et le *texte optionnel après* ne s'affichent que s'il existe, dans la base de données, un élément correspondant à cette balise.

Notre exemple devient :

```
<BOUCLE_articles (ARTICLES) {id_rubrique}>
<LI> [ (#SURTITRE) <BR>]
#TITRE
</BOUCLE_articles>
```

On obtient alors le résultat recherché : s'il existe un surtitre pour cet article, il est affiché et suivi du
 ; s'il n'existe pas de surtitre, même le
 est occulté.

8.4.3 Filtrer les résultats

Il est fréquent de vouloir modifier un élément tiré de la base de données, soit pour obtenir un affichage différent (par exemple, afficher le titre entièrement en majuscules), ou pour récupérer une valeur découlant de cet élément (par exemple, afficher le jour de la semaine correspondant à une date).

Dans SPIP, on peut directement appliquer des *filtres* aux éléments récupérés de la base de données, en les indiquant dans la syntaxe des balises SPIP, qui devient :

```
[ option avant (#BALISE|filtre1|filtre2|...|filtren) option après ]
```

La syntaxe est donc de faire suivre le nom de la balise, entre les parenthèses, par les filtres succesifs, séparés par une barre verticale (nommée habituellement *pipe*).

Remarque. En réalité, les *filtres* sont des fonctions PHP. Vous pouvez donc utiliser directement les fonctions de PHP (à condition que ces fonctions fonctionnent avec une seule variable), en plus des fonctions livrées avec SPIP.

Voici quelques filtres fournis par SPIP :

▷ *majuscules*,

 passe le texte en majuscules (plus puissant que la fonction de PHP correspondante, qui ne fonctionne pas correctement avec les caractères accentués) ; par exemple :

```
[ (#TITRE|majuscules) ]
```

▷ *justifier*,

 affiche le texte en justification totale (c'est-à-dire <P align=justify>); par exemple :

```
[ (#TEXTE|justifier) ]
```

La présente documentation consacre un article (section 8.19 page 132) aux différents filtres livrés avec SPIP.

8.4.4 Court-circuiter le traitement par SPIP

SPIP applique un traitement typographique à tous les textes tirés de la base de données. En particulier, il place des espaces insécables avant certains symboles (point-virgule, point d’interrogation, etc.), et analyse des raccourcis de mise en page.

Dans certains cas, vous pouvez avoir besoin de court-circuiter ce traitement, afin de récupérer directement le texte brut tel qu’il est placé dans la base de données. Pour cela, il suffit d’ajouter une astérisque (*) à la suite de la balise SPIP. Ce qui donne :

```
[ option avant (#BALISE*|filtre1|filtre2|...|filtren) option après ]
```

8.5 La boucle ARTICLES

24 mai 2001 par l’équipe de SPIP

Une boucle d’articles se code en plaçant ARTICLES (avec un « s ») entre parenthèses :

```
<BOUCLEn(ARTICLES){critères...}>
```

Les éléments contenus dans une telle boucle sont des articles.

Remarque. Une boucle ARTICLES ne retourne que des articles *publiés*. (Il n’existe aucun moyen d’afficher les articles «en cours de rédaction», «proposés à la publication» ou «refusés».)

8.5.1 Les critères de sélection

On utilisera l’un ou autre des critères suivants pour indiquer comment on sélectionne les éléments.

- ▷ {tout} les articles sont sélectionnés dans l’intégralité du site (dans toutes les rubriques). Utile notamment pour afficher les articles les plus récents (dans l’intégralité du site) sur la page d’accueil. [En réalité, le critère «tout» n’est pas traité de manière informatique : c’est un aide-mémoire pour le webmestre ; on obtient le même résultat en n’indiquant aucun des critères suivants.]
- ▷ {id_article} retourne l’article dont l’identifiant est id_article. Comme l’identifiant de chaque article est unique, ce critère ne retourne qu’une ou zéro réponse.
- ▷ {id_rubrique} retourne la liste des articles contenus dans la rubrique id_rubrique.
- ▷ {id_secteur} retourne les articles dans ce secteur (un secteur est une rubrique qui ne dépend d’aucune autre rubrique, c’est-à-dire située à la racine du site).
- ▷ {id_auteur} retourne les articles correspondant à cet identifiant d’auteur (utile pour indiquer la liste des articles écrits par un auteur).
- ▷ {id_mot} retourne les articles correspondant à cet identifiant de mot-clé (utile pour indiquer la liste des articles traitant d’un sujet donné).
- ▷ [SPIP 1.3] {titre_mot=xxxx}, ou {type_mot=yyyy} retourne les articles liés au mot-clé dont le nom est «xxxx», ou liés à des mots-clés du groupe de mots-clés «yyyy». Attention, on ne peut pas utiliser plusieurs critères {titre_mot=xxxx} ou {type_mot=yyyy} dans une même boucle.
- ▷ [SPIP 1.4] {id_groupe=zzzz} permet de sélectionner les articles liés à un groupe de mots-clés ; principe identique au {type_mot} précédent, mais puisque l’on travaille avec un identifiant (numéro du groupe), la syntaxe sera plus «propre».
- ▷ {recherche} retourne les articles correspondant aux mots indiqués dans l’interface de recherche (moteur de recherche incorporé à SPIP). Voir la page consacrée au moteur de recherche (section 8.18 page 131).

8.5.2 Les critères d’affichage

Une fois fixé l’un des critères ci-dessus, on pourra ajouter les critères suivants pour restreindre le nombre d’éléments affichés.

- ▷ Les critères communs à toutes les boucles (section 8.16 page 126) s’appliquent évidemment.
- ▷ `{exclus}` permet d’exclure du résultat l’article dans lequel on se trouve déjà (par exemple, lorsque l’on affiche les articles contenus dans la même rubrique, on ne veut pas afficher un lien vers l’article dans lequel on se trouve déjà).
- ▷ `{doublons}` ou `{unique}` (ces deux critères sont rigoureusement identiques) permettent d’interdire l’affichage d’articles déjà affichés dans d’autres boucles elles-mêmes marquées `{doublons}`.

8.5.3 Les balises de cette boucle

Les balises tirées de la base de données

Les balises suivantes correspondent aux éléments directement tirés de la base de données. Vous pouvez les utiliser également en tant que critère de classement (par exemple : `{par date}` ou `{par titre}`).

- ▷ `#ID_ARTICLE` affiche l’identifiant unique de l’article. Utile pour fabriquer des liens hypertextes non prévus (par exemple vers une page «Afficher au format impression».
- ▷ `#SURTITRE` retourne le surtitre.
- ▷ `#TITRE` retourne le titre de l’article.
- ▷ `#SOUSTITRE` retourne le soustitre.
- ▷ `#DESCRIPTIF` retourne le descriptif.
- ▷ `#CHAPO` retourne le texte d’introduction (chapeau).
- ▷ `#TEXTE` retourne le texte principal de l’article.
- ▷ `#PS` retourne le post-scriptum.
- ▷ `#DATE` est la date de mise en ligne.
- ▷ `#DATE_REDAC` est la date de première publication.
- ▷ `#ID_RUBRIQUE` est l’identifiant de la rubrique dont dépend l’article.
- ▷ `#ID_SECTEUR` est l’identifiant du secteur dont dépend l’article (le secteur étant la rubrique située à la racine du site).
- ▷ `#VISITES` est le nombre de visites sur cet article.
- ▷ `#POPULARITE` donne le pourcentage de popularité de cet article, voir la documentation 23.La «popularité» des articles (section 8.24 page 140).

Les balises calculées par SPIP

Les éléments suivants sont calculés par SPIP. (Ils ne peuvent pas être utilisés comme critère de classement.)

- ▷ `#NOTES` les notes de bas de page (calculées à partir de l’analyse du texte.
- ▷ `#INTRODUCTION` : [SPIP 1.4] si l’article contient un descriptif, c’est celui-ci qui est utilisé ici ; sinon, SPIP affiche les 600 premiers caractères du début de l’article (chapeau puis texte). [SPIP 1.3] Dans les versions précédentes de SPIP, ce sont systématiquement les premiers caractères de l’article (chapeau puis texte) qui sont pris en compte (le descriptif n’est pas utilisé).
- ▷ `#LESAUTEURS` les auteurs de cet article. Cela permet d’éviter de créer une boucle `AUTEURS` pour obtenir le même résultat.
- ▷ `#URL_ARTICLE` est l’URL de la page de l’article.
- ▷ `#FORMULAIRE_FORUM` fabrique l’interface permettant de poster un message répondant à cet article.
- ▷ `#FORMULAIRE_SIGNATURE` fabrique l’interface permettant de signer la pétition associée à cet article.

- ▷ #PARAMETRES_FORUM fabrique la liste des variables exploitées par l'interface du formulaire permettant de répondre à cet article. Par exemple :

```
[<A HREF="forum.php3?(#PARAMETRES_FORUM)">Répondre à cet article</A>]
```

Les logos

- ▷ #LOGO_ARTICLE le logo de l'article, éventuellement avec la gestion du survol.
- ▷ #LOGO_ARTICLE_RUBRIQUE le logo de l'article, éventuellement remplacé par le logo de la rubrique s'il n'existe pas de logo spécifique à l'article.
- ▷ #LOGO_RUBRIQUE le logo de la rubrique de l'article.

Les logos s'installent de la manière suivante :

```
[ (#LOGO_ARTICLE|alignement|adresse) ]
```

L'alignement peut être `left` ou `right`.

L'adresse est l'URL de destination du lien de ce logo (par exemple #URL_ARTICLE). Si l'on n'indique pas d'adresse, le bouton n'est pas cliquable.

Si l'on veut récupérer directement le nom du fichier du logo (alors que les balises précédentes fabriquent le code HTML complet pour insérer l'image dans la page), par exemple pour afficher une image en fond de tableau, on utilisera le filtre |fichier comme suit : [(#LOGO_ARTICLE|fichier)]

Par ailleurs deux balises permettent de récupérer un seul des deux logos :

- ▷ #LOGO_ARTICLE_NORMAL est le logo sans survol ;
- ▷ #LOGO_ARTICLE_SURVOL est le logo de survol.

8.6 La boucle RUBRIQUES

22 mai 2001 par l'équipe de SPIP

La boucle RUBRIQUES retourne une liste de ... rubriques (étonnant, non ?)

```
<BOUCLEn(RUBRIQUES) {critères...}>
```

Remarque. Une boucle RUBRIQUES n'affiche que des rubriques «actives», c'est-à-dire contenant des articles publiés, des documents joints (à partir de [SPIP 1.4]), des sites publiés - ou des sous-rubriques elles-mêmes actives. De cette façon, on évite de se trouver dans des rubriques «culs de sac» n'offrant aucun élément de navigation.

8.6.1 Les critères de sélection

On utilisera l'un ou autre des critères suivants pour indiquer comment on sélectionne les éléments.

- ▷ {tout} les rubriques sont sélectionnées dans l'intégralité du site.
- ▷ {id_rubrique} retourne la rubrique dont l'identifiant est id_rubrique. Comme l'identifiant de chaque rubrique est unique, ce critère retourne une ou zéro réponse.
- ▷ {id_secteur} retourne les rubriques de ce secteur.
- ▷ {id_parent} retourne la liste des rubriques contenues dans une rubrique.

- ▷ `{id_enfant}` retourne la rubrique qui contient la rubrique (une seule réponse ; ou zéro réponse si la présente rubrique est située à la racine du site).
- ▷ `{meme_parent}` retourne la liste des rubriques dépendant de la même rubrique que la rubrique en cours. Permet d'afficher les rubriques «sœurs» qui se trouvent au même niveau dans la hiérarchie.
- ▷ `{recherche}` retourne les rubriques correspondant aux mots indiqués dans l'interface de recherche (moteur de recherche incorporé à SPIP). Voir la page consacrée au moteur de recherche (section 8.18 page 131).
- ▷ À partir de la version **SPIP 1.4**, les rubriques peuvent être liées à des mots-clés. Les critères de mots-clés peuvent donc être désormais utilisés dans les boucles (RUBRIQUES) :
 - ★ `{id_mot}, {titre_mot=xxx}` récupèrent les rubriques liées au mot dont le numéro est `id_mot` ou dont le titre est `titre_mot` ;
 - ★ `{id_groupe}, {type_mot=yyyy}` récupèrent les rubriques liées à des mots du groupe `id_groupe`, ou du groupe dont le titre est `type_mot`.

8.6.2 Les critères d'affichage

Une fois fixé l'un des critères ci-dessus, on pourra ajouter les critères suivants pour restreindre le nombre d'éléments affichés.

- ▷ Les critères communs à toutes les boucles s'appliquent évidemment.
- ▷ `{exclus}` permet d'exclure du résultat la rubrique dans lequel on se trouve déjà (utile avec `meme_parent`).
- ▷ `{doublons}` ou `{unique}` (ces deux critères sont rigoureusement identiques) permettent d'interdire l'affichage de rubriques déjà affichées dans d'autres boucles.

8.6.3 Les balises de cette boucle

Les balises tirées de la base de données

Les balises suivantes correspondent aux éléments directement tirés de la base de données. Vous pouvez les utiliser également en tant que critère de classement (généralement : `{par titre}`).

- ▷ `#ID_RUBRIQUE` affiche l'identifiant unique de la rubrique.
- ▷ `#TITRE` retourne le titre de la rubrique.
- ▷ `#DESCRIPTIF` retourne le descriptif.
- ▷ `#TEXTE` retourne le texte principal de la rubrique.
- ▷ `#ID_SECTEUR` est l'identifiant du secteur dont dépend la rubrique (le secteur étant la rubrique située à la racine du site).

Les balises calculées par SPIP

Les éléments suivants sont calculés par SPIP. (Ils ne peuvent pas être utilisés comme critère de classement.)

- ▷ `#NOTES` les notes de bas de page (calculées à partir de l'analyse du texte).
- ▷ `#INTRODUCTION` les 600 premiers caractères du texte, les enrichissements typographiques (gras, italique) sont supprimés.
- ▷ `#URL_RUBRIQUE` est l'URL de la page de la rubrique.
- ▷ **[SPIP 1.4]** `#DATE` affiche la date de la dernière publication effectuée dans la rubrique et/ou ses sous-rubriques (articles, brèves ...).
- ▷ `#FORMULAIRE_FORUM` fabrique l'interface permettant de poster un message répondant à cette rubrique.
- ▷ `#PARAMETRES_FORUM` fabrique la liste des variables exploitées par l'interface du formulaire permettant de répondre à cette rubrique. Par exemple :

```
[<A HREF="forum.php3?(#PARAMETRES_FORUM)">Répondre à cette rubrique</A>]
```

- ▷ #FORMULAIRE_SITE [SPIP 1.4] Le #FORMULAIRE_SITE affiche une interface permettant aux visiteurs du site de proposer des référencements de sites. Ces sites apparaîtront comme «proposés» dans l'espace privé, en attendant une validation par les administrateurs.
Ce formulaire ne s'affiche que si vous avez activé l'option «Gérer un annuaire de sites» dans la *Configuration sur site* dans l'espace privé, et si vous avez réglé «Qui peut proposer des sites référencés» sur «les visiteurs du site public».

Le logo

- ▷ #LOGO_RUBRIQUE le logo de la rubrique, éventuellement avec la gestion du survol. S'il n'y a pas de logo pour cette rubrique, SPIP va automatiquement chercher s'il existe un logo pour la rubrique dont elle dépend, et ainsi de suite de manière récursive.
Le logo s'installe de la manière suivante :

```
[ (#LOGO_RUBRIQUE|alignement|adresse) ]
```

- ▷ [SPIP 1.4] #LOGO_RUBRIQUE_NORMAL affiche le logo «sans survol»; #LOGO_RUBRIQUE_SURVOL affiche le logo de survol : ces deux balises permettent par exemple, quand on est dans une rubrique, de gérer un logo «avec survol» pour les liens vers les autres rubriques, et de laisser le logo de survol seul dans la rubrique active.

8.7 La boucle BREVES

20 mai 2001 par l'équipe de SPIP

La boucle BREVES, comme son nom l'indique, retourne une liste de brèves.

```
<BOUCLEn(BREVES) {critères...}>
```

8.7.1 Les critères de sélection

On utilisera l'un ou autre des critères suivants pour indiquer comment on sélectionne les éléments.

- ▷ {tout} les rubriques sont sélectionnés dans l'intégralité du site.
- ▷ {id_breve} retourne la brève dont l'identifiant est `id_breve`. Comme l'identifiant de chaque brève est unique, ce critère retourne une ou zéro réponse.
- ▷ {id_rubrique} retourne toutes les brèves contenues dans la rubrique en cours.
- ▷ [SPIP 1.2] {id_mot} retourne toutes les brèves liées au mot-clé en cours (à l'intérieur d'une boucle de type (MOTS).
- ▷ [SPIP 1.3] {titre_mot=xxxx}, ou {type_mot=yyyy} retourne les brèves liées au mot-clé dont le nom est «xxxx», ou liées à des mots-clés du groupe de mots-clés «yyyy». Attention, on ne peut pas utiliser plusieurs critères {titre_mot=xxxx} ou {type_mot=yyyy} dans une même boucle.
- ▷ [SPIP 1.4] {id_groupe=zzzz} permet de sélectionner les brèves liées à un groupe de mots-clés ; principe identique au {type_mot} précédent, mais puisque l'on travaille avec un identifiant (numéro du groupe), la syntaxe sera plus «propre».
- ▷ {recherche} retourne les brèves correspondant aux mots indiqués dans l'interface de recherche (moteur de recherche incorporé à SPIP). Voir la page consacrée au moteur de recherche (section 8.18 page 131).

8.7.2 Les critères d’affichage

Les critères communs à toutes les boucles (section 8.16 page 126) s’appliquent.

8.7.3 Les balises de cette boucle

Les balises tirées de la base de données

Les balises suivantes correspondent aux éléments directement tirés de la base de données. Vous pouvez les utiliser également en tant que critère de classement (généralement : {par titre}).

- ▷ #ID_BREVE affiche l’identifiant unique de la brève.
- ▷ #TITRE retourne le titre de la brève.
- ▷ #DATE retourne la date de publication de la brève.
- ▷ #TEXTE retourne le texte de la brève.
- ▷ #NOM_SITE le nom du site indiqué en références.
- ▷ #URL_SITE l’adresse (URL) du site indiqué en références.
- ▷ #ID_RUBRIQUE l’identifiant de la rubrique dont dépend cette brève.

Les balises calculées par SPIP

Les éléments suivants sont calculés par SPIP. (Ils ne peuvent pas être utilisés comme critère de classement.)

- ▷ #NOTES les notes de bas de page (calculées à partir de l’analyse du texte.
- ▷ #INTRODUCTION les 600 premiers caractères du texte, les enrichissements typographiques (gras, italique) sont supprimés.
- ▷ #URL_BREVE est l’URL de la page de la brève.
- ▷ #FORMULAIRE_FORUM fabrique l’interface permettant de poster un message répondant à cette brève.
- ▷ #PARAMETRES_FORUM fabrique la liste des variables exploitées par l’interface du formulaire permettant de répondre à cette brève. Par exemple :

```
<A HREF="forum.php3?(#PARAMETRES_FORUM)">Répondre à cette brève</A>
```

Le logo

- ▷ #LOGO_BREVE le logo de la brève, éventuellement avec la gestion du survol.
Le logo s’installe de la manière suivante :

```
LOGO_BREVE|alignement|adresse]
```

- ▷ #LOGO_BREVE_RUBRIQUE affiche, si il existe, le logo de la brève ; si ce logo n’a pas été attribué, SPIP affiche le logo de la rubrique [SPIP 1.4].

8.8 La boucle AUTEURS

18 mai 2001 par l’équipe de SPIP

La boucle AUTEURS, comme son nom l’indique, retourne une liste d’auteurs. Si l’on ne précise pas de critère de sélection, la boucle retournera tous les auteurs *ayant un article publié*.

```
<BOUCLEn(AUTEURS){critères...}>
```

8.8.1 Les critères de sélection

On utilisera l'un ou autre des critères suivants pour indiquer comment on sélectionne les éléments.

- ▷ {tout} les auteurs sont sélectionnés, qu'ils aient écrit un article ou non.
- ▷ {id_auteur} retourne l'auteur dont l'identifiant est id_auteur. Comme l'identifiant de chaque auteur est unique, ce critère retourne une ou zéro réponse.
- ▷ {id_article} retourne tous les auteurs de cet article.

8.8.2 Les critères d'affichage

Les critères communs à toutes les boucles s'appliquent (section 8.16 page 126).

8.8.3 Les balises de cette boucle

Les balises tirées de la base de données

Les balises suivantes correspondent aux éléments directement tirés de la base de données. Vous pouvez les utiliser également en tant que critère de classement (généralement : {par nom}).

- ▷ #ID_AUTEUR affiche l'identifiant unique de l'auteur.
- ▷ #NOM retourne le nom de l'auteur.
- ▷ #BIO retourne la biographie de l'auteur.
- ▷ #EMAIL retourne son adresse email.
- ▷ #NOM_SITE le nom de son site Web.
- ▷ #URL_SITE l'adresse (URL) de son site.
- ▷ #PGP sa clé publique pour PGP.
- ▷ #FORMULAIRE_ECRIRE_AUTEUR [SPIP 1.4] affiche un formulaire permettant d'écrire à l'auteur. Il faut que le serveur hébergeant le site accepte d'envoyer des mails. Ce système permet de ne pas divulguer l'adresse email de l'auteur.

Les balises calculées par SPIP

#NOTES les notes de bas de page (calculées à partir de l'analyse du texte).

Le logo

#LOGO_AUTEUR le logo de l'auteur, éventuellement avec la gestion du survol.

Le logo s'installe de la manière suivante :

```
[ (#LOGO_AUTEUR|alignement|adresse) ]
```

8.9 La boucle FORUMS

16 mai 2001 par l'équipe de SPIP

La boucle FORUMS retourne une liste de messages de forums.

```
<BOUCLEn(FORUMS) {critères...}>
```

8.9.1 Les critères de sélection

On utilisera l'un ou autre des critères suivants pour indiquer comment on sélectionne les éléments.

- ▷ {id_forum} retourne le message dont l'identifiant est id_forum. Comme l'identifiant de chaque message est unique, ce critère retourne une ou zéro réponse.
- ▷ {id_article} retourne les messages correspondant à cet article.
- ▷ {id_rubrique} retourne les messages correspondant à cette rubrique.
- ▷ {id_breve} retourne les messages correspondant à cette brève.
- ▷ {id_parent} retourne les messages dépendant d'un autre message. Indispensable pour gérer des *threads* dans les forums.
- ▷ {id_enfant} retourne le message dont dépend le message actuel (permet de «remonter» dans la hiérarchie des *threads*). (SPIP 1.3)
- ▷ {meme_parent} retourne les autres messages répondant à un même message. (SPIP 1.3)
- ▷ {plat} : par défaut, seuls les messages n'ayant pas de parent (i.e. à la racine d'un *thread*) sont affichés. En ajoutant ce critère, vous pouvez sélectionner tous les messages quelle que soit leur position dans un thread (dans la limite des autres critères, bien sûr). Cela permet d'afficher les messages par ordre strictement chronologique par exemple, ou de compter le nombre total de contributions dans un forum.
- ▷ {id_secteur} retourne les messages correspondant au secteur. A priori, peu utile ; mais cela permet par exemple de faire un grand forum thématique regroupant tous les messages d'un secteur, quel que soit l'endroit où l'on se trouve.
- ▷ À partir de la version SPIP 1.4, les messages des forums peuvent être liés à des mots-clés. Les critères de mots-clés peuvent donc être désormais utilisés dans les boucles (FORUMS) :
 - ★ {id_mot}, {titre_mot=xxx} récupèrent les messages liés au mot dont le numéro est id_mot ou dont le titre est titre_mot ;
 - ★ {id_groupe}, {type_mot=yyyy} récupèrent les messages liés à des mots du groupe id_groupe, ou du groupe dont le titre est type_mot.

8.9.2 Les critères d'affichage

Les critères communs à toutes les boucles (section 8.16 page 126) s'appliquent .

8.9.3 Les balises de cette boucle

Les balises tirées de la base de données

Les balises suivantes correspondent aux éléments directement tirés de la base de données. Vous pouvez les utiliser également en tant que critère de classement (généralement : {par titre}).

- ▷ #ID_FORUM affiche l'identifiant unique du message.
- ▷ #ID_BREVE affiche l'identifiant de la brève à laquelle ce message est attaché. Attention, cela n'est pas récursif : un message qui répond à un message attaché à une brève ne contient pas lui-même le numéro de la brève.
- ▷ #ID_ARTICLE est l'identifiant de l'article à laquelle le message répond.
- ▷ #ID_RUBRIQUE l'identifiant de la rubrique à laquelle le message répond.
- ▷ #DATE est la date de publication.
- ▷ #TITRE est le titre.
- ▷ #TEXTE est le texte du message.
- ▷ #NOM_SITE le nom du site Web indiqué par l'auteur.
- ▷ #URL_SITE l'adresse (URL) de ce site Web.
- ▷ #NOM est le nom de l'auteur du message.
- ▷ #EMAIL est l'adresse email de l'auteur.
- ▷ #IP est l'adresse IP de l'auteur du message au moment de l'envoi de sa contribution.

Les balises calculées par SPIP

- ▷ #FORMULAIRE_FORUM fabrique l'interface permettant de poster un message de réponse.
- ▷ #PARAMETRES_FORUM fabrique la liste des variables exploitées par l'interface du formulaire permettant de répondre à ce message. Par exemple :

```
[<a href="forum.php3?({PARAMETRES_FORUM})">Répondre à ce message</a>]
```

8.10 La boucle MOTS

14 mai 2001 par l'équipe de SPIP

La boucle MOTS retourne une liste de mots-clés.

```
<BOUCLen(MOTS){critères...}>
```

8.10.1 Les critères de sélection

On utilisera l'un ou autre des critères suivants pour indiquer comment on sélectionne les éléments.

- ▷ {tout} les mots sont sélectionnés dans l'intégralité du site.
- ▷ {id_mot} retourne le mot-clé dont l'identifiant est id_mot.
- ▷ {id_groupe} retourne les mots-clés associés au groupe de mots dont le numéro est id_groupe [SPIP 1.4].
- ▷ {id_article} retourne les mots-clés associés à cet article (c'est l'utilisation la plus courante de cette boucle).
- ▷ {id_rubrique} retourne les mots-clés associés à une rubrique [SPIP 1.4].
- ▷ {id_breve} retourne les mots associés à une brève [SPIP 1.4].
- ▷ {id_syndic} retourne les mots associés à un site référencé [SPIP 1.4].
- ▷ {id_forum} retourne les mots associés à un message de forum [SPIP 1.4] (attention, utilisation très spécifique).
- ▷ {titre=france} retourne le mot-clé intitulé france (par exemple).
- ▷ {type=pays} retourne les mots-clés du groupe de mots-clés intitulé pays (par exemple).

8.10.2 Les critères d'affichage

Les critères communs à toutes les boucles (section 8.16 page 126) s'appliquent.

8.10.3 Les balises de cette boucle

Les balises tirées de la base de données

Les balises suivantes correspondent aux éléments directement tirés de la base de données. Vous pouvez les utiliser également en tant que critère de classement (généralement : {par titre}).

- ▷ #ID_MOT affiche l'identifiant unique du mot.
- ▷ #TITRE est le titre (le mot-clé lui-même).
- ▷ #DESCRIPTIF est le descriptif du mot.
- ▷ #TEXTE est le texte associé au mot.
- ▷ #TYPE est la catégorie dans laquelle est installé ce mot clé (par exemple, le mot-clé «France» pourrait être associé à la catégorie «Pays»).

Le logo

#LOGO_MOT [SPIP 1.4] affiche le logo associé au mot-clé.

8.11 La boucle DOCUMENTS

[SPIP 1.4] La boucle DOCUMENTS retourne une liste de documents multimédia associés (à un article, à une rubrique, éventuellement les images incluses dans une brève). 15 juin 2002 par ARNO*

```
<BOUCLEn(DOCUMENTS) {critères...}>
```

Cette boucle gère non seulement les documents joints non installés dans le texte d'un article, mais peut aussi accéder aux *images* (depuis la version 1.4, les images sont gérées, au niveau du programme, comme un genre spécifique de documents), aux vignettes de prévisualisation et aux documents déjà insérés dans le corps de l'article.

Pour mémoire, on utilisera donc le plus fréquemment (utilisation courante) la boucle DOCUMENTS avec, au minimum, les critères suivants (explications ci-après) :

```
<BOUCLEn(DOCUMENTS) {mode=document} {doublons}>
```

8.11.1 Les critères de sélection

Une boucle DOCUMENTS s'utilise en général à l'intérieur d'un article ou d'une rubrique (éventuellement dans une brève, mais ici l'utilisation sera réservée à la récupération d'images, ce qui sera très spécifique).

- ▷ {id_article} retourne les documents de l'article dont l'identifiant est id_article.
- ▷ {id_rubrique} retourne les documents de la rubrique id_rubrique.
- ▷ {id_breve} retourne les documents de la brève id_breve (il n'est pas possible d'associer des documents multimédia à une brève, seulement des images ; l'utilisation d'une boucle DOCUMENTS dans ce cadre sera donc très spécifique).

Notez bien : il n'est pas possible d'utiliser ici le critère {id_secteur} ; les documents sont conçus pour être intimement liés aux articles et aux rubriques, et non à être appelés seuls sans ces éléments (on parle dans SPIP de «documents joints»).

8.11.2 Les critères d'affichage

- ▷ {mode=document} ou {mode=image} permet d'indiquer si l'on veut appeler les documents multimédia, ou les images (en effet, désormais les images associées à l'article et éventuellement insérées dans l'article sont traités comme des *documents* en *mode=image*).

N.B. Dans les sites SPIP existant avant la version 1.4, l'habitude a été prise de ne pas pouvoir afficher les images qui ne sont pas insérées à l'intérieur du texte de l'article. De fait, si vous ajoutez une boucle DOCUMENTS en *mode=image* sur un site déjà existant, vous risquez de voir réapparaître dans cette boucle des images qui n'étaient pas destinées à être publiées sur le site public. Donc, n'utilisez une telle boucle que sur un site créé avec la version 1.4, ou bien procédez avec beaucoup de précautions (vérifiez les anciens articles pour éviter la publication d'images parasites).

- ▷ {doublons} prend ici une importance particulière : elle permet non seulement de ne pas réafficher des documents déjà affichés par une autre boucle, mais également de ne pas réafficher les documents déjà intégrés à l'intérieur d'un article. Si l'on oublie ce critère, on affichera *tous* les documents associés à un article, y compris ceux qui auraient déjà été affichés à l'intérieur du texte.

- ▷ {extension=...} permet de sélectionner les documents selon leur terminaison (terminaison du fichier multimédia, par exemple «mov», «ra», «avi» ...). Cela peut être utilisé par exemple pour réaliser un *portfolio*, c'est-à-dire une boucle n'affichant que les documents de type image, une seconde boucle ensuite, avec une présentation graphique différente, les autres types de documents :

```
<BOUCLE_portfolio(DOCUMENTS) {id_article} {extension==jpg|png|gif}
                                     {mode=document} {doublons}>
```

Cette `BOUCLE_portfolio` récupère les documents joints à un article, non déjà affichés dans le texte de l'article, et donc les extensions des fichiers peuvent être «jpg», «png» ou «gif».

8.11.3 Les balises

- ▷ #LOGO_DOCUMENT affiche le logo (vignette de prévisualisation) associé à cet article ; si une vignette personnalisée n'a pas été installée manuellement par l'auteur de l'article, SPIP utilise une vignette standard selon le type du fichier.
- ▷ #URL_DOCUMENT est l'URL du fichier multimédia. Pour afficher une vignette cliquable pointant vers le document multimédia, on utilisera donc le code suivant :

```
[ (#LOGO_DOCUMENT|#URL_DOCUMENT) ]
```

- ▷ #TITRE affiche le titre du document.
- ▷ #DESCRIPTIF affiche le descriptif du document.
- ▷ #TYPE_DOCUMENT affiche le type (fichier Quicktime, fichier Real ...) du document multimédia.
- ▷ #TAILLE affiche la taille du fichier multimédia. Ce chiffre est fourni en octets. Pour de gros fichiers, cette valeur devient rapidement inutilisable ; on pourra donc lui appliquer le filtre `taille_en_octets`, qui affichera successivement en octets, en kilooctets, ou même en mégaoctets :

```
[ (#TAILLE|taille_en_octets) ]
```

- ▷ #LARGEUR et #HAUTEUR fournissent les dimensions en pixels.
- ▷ #EMBED_DOCUMENT est une balise à l'utilisation très spécifique : elle permet d'inclure directement les fichiers de formats autorisés (vidéo, sons) directement dans la page Web ; il faut éviter d'utiliser systématiquement cette balise, car il est déconseillé d'insérer systématiquement les documents dans les pages sans un contrôle strict (sauf à faire exploser la durée de chargement de vos pages Web ...). La balise peut être complétée de paramètres propres aux formats utilisés (encore une fois : utilisation très spécifique), par exemple :

```
[ (#EMBED_DOCUMENT|autostart=true) ]
```

8.12 La boucle SITES (ou SYNDICATION)

12 mai 2001 par l'équipe de SPIP

La boucle `SITES` ([SPIP 1.3](#)) retourne une liste de sites référencés. (Si l'on a syndiqué des sites référencés, cette boucle s'utilise, naturellement, associée à une boucle `SYNDIC_ARTICLES` qui permet de récupérer la liste des articles de ces sites.)

```
<BOUCLEn(SITES) {critères...}>
```

Avant la version 1.3 de SPIP, cette boucle était nommée SYNDICATION, car seuls des sites syndiqués pouvaient être référencés. Les deux dénominations sont rigoureusement équivalentes (mais «SITES» correspond mieux au fait que, depuis la version 1.3, il s’agit d’un système de *référencement* de sites, la syndication étant une option).

```
<BOUCLÉn(SYNDICATION) {critères...}>
```

8.12.1 Les critères de sélection

On utilisera l’un ou autre des critères suivants pour indiquer comment on sélectionne les éléments.

- ▷ {tout}, tous les sites référencés.
- ▷ {id_syndic} retourne le site référencés dont l’identifiant est id_syndic.
- ▷ {id_rubrique} retourne les sites référencés dans cette rubrique.
- ▷ {id_secteur} retourne les sites référencés dans ce secteur.
- ▷ [SPIP 1.3] {id_mot} retourne toutes les sites liés au mot-clé en cours (à l’intérieur d’une boucle de type (MOTS).
- ▷ [SPIP 1.3] {titre_mot=xxxx}, ou {type_mot=yyyy} retourne les sites liés au mot-clé dont le nom est «xxxx», ou liés à des mots-clés du groupe de mots-clés «yyyy». Attention, on ne peut pas utiliser plusieurs critères {titre_mot=xxxx} ou {type_mot=yyyy} dans une même boucle.
- ▷ [SPIP 1.4] {id_groupe=zzzz} permet de sélectionner les sites liés à un groupe de mots-clés ; principe identique au {type_mot} précédent, mais puisque l’on travaille avec un identifiant (numéro du groupe), la syntaxe sera plus «propre».

8.12.2 Les critères d’affichage

Les critères communs à toutes les boucles (section 8.16 page 126) s’appliquent.

- ▷ {moderation=oui} [SPIP 1.4] affiche les sites syndiqués dont les liens sont bloqués a priori («modérés»); l’inverse de ce critère est {moderation!=oui}.
- ▷ (SPIP 1.3) {syndication=oui}, {syndication=non} permet de n’afficher que les sites référencés faisant l’objet d’une syndication, ou les sites non syndiqués.
- ▷ (SPIP 1.2) {doublons} ou {unique} (ces deux critères sont rigoureusement identiques) permettent d’interdire l’affichage de sites référencés déjà affichés dans d’autres boucles.

8.12.3 Les balises de cette boucle

Les balises tirées de la base de données

Les balises suivantes correspondent aux éléments directement tirés de la base de données. Vous pouvez les utiliser également en tant que critère de classement (généralement : {par titre}).

- ▷ #ID_SYNDIC affiche l’identifiant unique du site syndiqué.
- ▷ #NOM_SITE est le nom du site syndiqué.
- ▷ #URL_SITE est l’adresse (URL) du site syndiqué.
- ▷ #DESCRIPTIF est le descriptif du site syndiqué.
- ▷ #ID_RUBRIQUE est le numéro de la rubrique contenant cette syndication.
- ▷ #ID_SECTEUR est le numéro de la rubrique-secteur (à la racine du site) contenant cette syndication.

Autre balise

- ▷ #LOGO_SITE affiche le logo attribué au site.

8.13 La boucle SYNDIC_ARTICLES

10 mai 2001 par l'équipe de SPIP

La boucle SYNDIC_ARTICLES retourne une liste des articles des sites syndiqués. On peut soit l'utiliser à l'intérieur d'une boucle SITES (cette dernière récupère une liste de sites référencés, ensuite on récupère chaque article de ces sites), soit directement à l'intérieur d'une rubrique (on récupère directement tous les articles syndiqués dans une rubrique, en court-circuitant le passage par la liste des sites).

```
<BOUCLEn(SYNDIC_ARTICLES) {critères...}>
```

(SPIP 1.3) À partir de la version 1.3 de SPIP, la boucle SITES (ou SYNDICATION) n'affiche plus uniquement des sites syndiqués, mais plus généralement des sites référencés (la syndication de certains sites référencés étant une option). On pourra donc, pour obtenir une présentation graphique plus précise, utiliser une boucle SYNDIC_ARTICLES uniquement à l'intérieur d'une boucle SITES utilisant le critère {syndication=oui}.

8.13.1 Les critères de sélection

On utilisera l'un ou autre des critères suivants pour indiquer comment on sélectionne les éléments.

- ▷ {tout}, tous les sites syndiqués.
- ▷ {id_syndic_article} retourne l'article syndiqué dont l'identifiant est id_syndic_article. (Dans la pratique, il y a très peu d'intérêt à fabriquer une page pour un article syndiqué, puisqu'on préférera renvoyer directement vers l'article en question.)
- ▷ {id_syndic} retourne la liste des articles du site syndiqué dont l'identifiant est id_syndic.
- ▷ {id_rubrique} retourne la liste des articles syndiqués dans cette rubrique.
- ▷ {id_secteur} retourne la liste des articles syndiqués dans ce secteur.

8.13.2 Les critères d'affichage

Les critères communs à toutes les boucles (section 8.16 page 126) s'appliquent.

8.13.3 Les balises de cette boucle

Les balises tirées de la base de données

Les balises suivantes correspondent aux éléments directement tirés de la base de données. Vous pouvez les utiliser également en tant que critère de classement (généralement : {par titre}).

- ▷ #ID_SYNDIC_ARTICLE affiche l'identifiant unique de l'article syndiqué.
- ▷ #ID_SYNDIC affiche l'identifiant unique du site syndiqué contenant cet article. syndiqué.
- ▷ #TITRE est le titre de l'article.
- ▷ #URL_ARTICLE est l'adresse (URL) du site syndiqué (sur son site original).
- ▷ #DATE est la date de publication de cet article.
- ▷ #LESAUTEURS, les auteurs de l'article syndiqué.
- ▷ #DESCRIPTIF le descriptif de l'article syndiqué.
- ▷ #NOM_SITE est le nom du site syndiqué contenant cet article.
- ▷ #URL_SITE est l'adresse (URL) du site

8.14 La boucle SIGNATURES

7 mai 2001 par l'équipe de SPIP

La boucle SIGNATURES retourne une liste de signataires d'une pétition associée à un article.

```
<BOUCLEn(SIGNATURES) {critères...}>
```

8.14.1 Les critères de sélection

On utilisera l'un ou autre des critères suivants pour indiquer comment on sélectionne les éléments.

- ▷ {tout} toutes les signatures sont sélectionnés dans l'intégralité du site.
- ▷ {id_signature}, la signature correspondant à l'identifiant courant.
- ▷ {id_article} retourne les signatures de la pétition de cet article.

8.14.2 Les critères d'affichage

Les critères communs à toutes les boucles (section 8.16 page 126) s'appliquent.

Attention. Dans ce type de boucles, certains critères de classement de la boucle ne sont pas identiques aux balises SPIP indiquées ci-dessous :

- ▷ {par nom_email} classe les résultats selon le #NOM du signataire ;
- ▷ {par ad_email} classe selon l'#EMAIL du signataire.

8.14.3 Les balises de cette boucle

Les balises tirées de la base de données

Les balises suivantes correspondent aux éléments directement tirés de la base de données. Vous pouvez les utiliser également en tant que critère de classement (généralement : {par titre}).

- ▷ #ID_SIGNATURE affiche l'identifiant unique du message.
- ▷ #ID_ARTICLE est l'identifiant de l'article pour cette pétition.
- ▷ #DATE est la date de publication.
- ▷ #MESSAGE est le texte du message.
- ▷ #NOM est le nom de l'auteur du message.
- ▷ #EMAIL est l'adresse email de l'auteur.
- ▷ #NOM_SITE le nom du site Web indiqué par l'auteur.
- ▷ #URL_SITE l'adresse (URL) de ce site Web.

8.15 La boucle HIERARCHIE

6 mai 2001 par l'équipe de SPIP

La boucle HIERARCHIE est basée sur un principe différent des autres boucles, car elle effectue une récursivité (au lieu de simplement tirer une liste de la base de données).

Cette boucle retourne la liste des RUBRIQUES qui mènent de la racine du site à la rubrique ou à l'article en cours.

```
<BOUCLEn(HIERARCHIE) {critères...}>
```

8.15.1 Les critères de sélection

On utilisera l'un ou autre des critères suivants pour indiquer comment on sélectionne les éléments.

- ▷ `{id_rubrique}` retourne la liste des rubriques depuis la racine jusqu'à la rubrique correspondant à cet identifiant.
- ▷ `{id_article}` retourne la liste des rubriques depuis la racine jusqu'à l'article correspondant à cet identifiant.

8.15.2 Les critères d'affichage

Attention : les critères communs à toutes les boucles (section 8.16 page 126) ne s'appliquent pas tous à ce type de boucle.

Les critères d'affichage utilisables sont : `{"inter"}` et `{a,b}`.

8.15.3 Les balises de cette boucle

Les éléments obtenus avec une boucle `HIERARCHIE` sont des rubriques. On peut donc utiliser toutes les balises proposées pour les boucles `RUBRIQUES` (section 8.6 page 114).

8.16 Les critères communs à toutes les boucles

5 mai 2001 par l'équipe de SPIP

Certains critères s'appliquent à (presque) tous les types de boucles. Ce sont des critères destinés à restreindre le nombre de résultats affichés ou à indiquer l'ordre d'affichage. On peut sans difficulté combiner plusieurs de ces critères de sélection.

8.16.1 Classer les résultats

`{par critère_de_classement}` indique l'ordre de présentation des résultats. Ce critère de classement correspond à l'une des balises tirées de la base de données pour chaque type de boucle. Par exemple, on pourra classer les articles `{par date}`, `{par date_redac}` ou `{par titre}`. (Notez que, si les balises sont en majuscules, les critères de classement sont en minuscules.)

Cas particulier : `{par hasard}` permet d'obtenir une liste présentée dans un ordre aléatoire.

Inverser le classement. De plus, `{inverse}` provoque l'affichage du classement inversé. Par exemple `{par date}` commence par les articles les plus anciens ; avec `{par date}{inverse}` on commence la liste avec les articles les plus récents.

Classer par numéro. (SPIP 1.3) Lorsqu'on réalise le classement selon un élément de texte (par exemple le *titre*), le classement est réalisé par ordre *alphabétique*. Cependant, pour forcer un ordre d'affichage, on peut indiquer un numéro devant le titre, par exemple : «1. Mon premier article», «2. Deuxième article», «3. Troisième . . .», etc ; avec un classement alphabétique, le classement de ces éléments donnerait la série «1, 10, 11, 2, 3 . . .». Pour rétablir le classement selon les numéros, on peut utiliser le critère :

`{par num critère}`

Par exemple :

```
<BOUCLE_articles(ARTICLES){id_rubrique}{par date}{inverse}>
```

affiche les articles d'une rubrique classés selon l'ordre chronologique inversé (les plus récents au début, les plus anciens à la fin), et :

```
<BOUCLE_articles(ARTICLES){id_rubrique}{par titre}>
```

les affiche selon l'ordre alphabétique de leur titre ; enfin :

```
<BOUCLE_articles(ARTICLES){id_rubrique}{par num titre}>
```

les affiche selon l'ordre du numéro de leur titre (remarque : l'option {par num titre} ne fonctionne pas pour les plus anciennes versions de MySQL, antérieures à la version 3.23).

8.16.2 Comparaisons, égalités

{critère < valeur} Comparaison avec une valeur fixée (on peut utiliser «>», «<», «=», «>=», «<=». Tous les *critères de classement* (tels que tirés de la base de données) peuvent également être utilisés pour limiter le nombre de résultats.

Par exemple :

```
<BOUCLE_art(ARTICLES){id_article=5}>
```

affiche l'article dont le numéro est 5 (utile pour mettre en vedette un article précis sur la page d'accueil).

```
<BOUCLE_art(ARTICLES){id_secteur=2}>
```

affiche les articles du secteur numéro 2.

Expressions régulières. Très puissant (mais nettement plus complexe à manipuler), le terme de comparaison «==» introduit une comparaison selon une expression régulière. Par exemple :

```
<BOUCLE_art(ARTICLES){titre==^[aA]}>
```

sélectionne les articles dont le titre commence par «a» ou «A».

Négation. (SPIP 1.2) On peut utiliser la notation {xxx != yyy}, le ! correspondant à la négation (opérateur logique NOT).

```
<BOUCLE_art(ARTICLES){titre!=^[aA]}>
```

sélectionne les articles dont le titre ne commence pas par «a» ou «A».

```
<BOUCLE_art(ARTICLES){id_secteur != 2}>
```

sélectionne les articles qui n'appartiennent pas au secteur numéro 2.

Pour faciliter l'utilisation des comparaisons sur les dates, on a ajouté des critères :

- ▷ `age` et `age_redac` correspondent respectivement à l'ancienneté de la publication et de la première publication d'un article, en jours : `{age<30}` sélectionne les éléments publiés depuis un mois ;
- ▷ les critères `mois`, `mois_redac`, `annee`, `annee_redac` permettent de comparer avec des valeurs fixes (`{annee<=2000}` pour les éléments publiés avant la fin de l'année 2000).

On peut combiner plusieurs de ces critères pour effectuer des sélections très précises. Par exemple :

```
<BOUCLE_art (ARTICLES) {id_secteur=2}{id_rubrique!=3}{age<30}>
```

affiche les articles du secteur 2, à l'exclusion de ceux de la rubrique 3, et publiés depuis moins de 30 jours.

Astuce. Le critère `age` est très pratique pour afficher les articles ou les brèves dont la date est située «dans le futur», avec des valeurs négatives (à condition d'avoir sélectionné, dans la Configuration précise du site, l'option «Publier les articles post-datés»). Par exemple, ce critère permet de mettre en valeur des événements futurs. `{age<0}` sélectionne les articles ou les brèves dont la date est située dans le futur («après» aujourd'hui) ...

(SPIP 1.3) Âge par rapport à une date fixée. Le critère `age` est calculé par rapport à la date d'aujourd'hui (ainsi `{age<30}` correspond aux articles publiés depuis moins d'un mois par rapport à aujourd'hui). Le critère `age_relatif` compare la date d'un article ou d'une brève à une date «courante»; par exemple, à l'intérieur d'une boucle `ARTICLES`, on connaît déjà une date pour chaque résultat de la boucle, on peut donc sélectionner par rapport à cette date (et non plus par rapport à aujourd'hui).

Par exemple :

```
<BOUCLE_article_principal (ARTICLES) {id_article}>

  <h1>#TITRE</h1>

  <BOUCLE_suivant (ARTICLES) {id_rubrique}{age_relatif<=0}{exclus}{par date}{0,1}>
  Article suivant: #TITRE
  </BOUCLE_suivant>

</BOUCLE_article_principal>
```

la `BOUCLE_suivant` affiche un seul article de la même rubrique, classé par date, dont la date de publication est inférieure ou égale à la date de l'«article_principal»; c'est-à-dire l'article de la même rubrique publié après l'article principal.

8.16.3 Affichage d'une partie des résultats

- ▷ `{a,b}` où `a` et `b` sont des chiffres. Ce critère permet de limiter le nombre de résultats. `a` indique le résultat à partir duquel on commence l'affichage (attention, le premier résultat est numéroté 0 - zéro); `b` indique le nombre de résultats affichés.

Par exemple `{0,10}` affiche les dix premiers résultats; `{4,2}` affiche les deux résultats à partir du cinquième (inclus).

- ▷ `{debut_xxx,b}` est une variante très élaborée de la précédente. Elle permet de faire commencer la limitation des résultats par une variable passée dans l'URL (celle variable remplace ainsi le `a` que l'on indiquait précédemment). C'est un fonctionnement un peu compliqué, que fort heureusement on n'a pas besoin d'utiliser trop souvent.

La variable passée dans l'URL commence forcément par `debut_xxx` (où `xxx` est un mot choisi par le webmestre). Ainsi, pour une page dont l'URL est :

```
petition.php?id_article=13&debut_signatures=200
```

avec un squelette (`petition.html`) contenant par exemple :

```
<BOUCLE_signatures (SIGNATURES) {id_article}{debut_signatures,100}>
```

on obtiendra la liste des 100 signatures à partir de la 200-ième. Avec l'URL :

```
petition.php3?id_article=13&debut_signatures=300
```

on obtient la liste des 100 signatures à partir de la 300-ième.

- ▷ `{a/b}` où `a` et `b` sont des chiffres. Ce critère permet d'afficher une partie `a` (proportionnellement) des résultats en fonction d'un nombre de «tranches» `b`.

Par exemple : `{1/3}` affiche le premier tiers des résultats. Ce critère est surtout utile pour présenter des listes sur plusieurs colonnes. Pour obtenir un affichage sur deux colonnes, il suffit de créer une première boucle, affichée dans une case de tableau, avec le critère `{1/2}` (la première moitié des résultats), puis une seconde boucle dans une seconde case, avec le critère `{2/2}` (la seconde moitié des résultats).

Attention. L'utilisation du critère `{doublons}` avec ce critère est périlleuse. Par exemple :

```
<BOUCLE_prem (ARTICLES) {id_rubrique}{1/2}{doublons}>
  <li> #TITRE
</BOUCLE_prem>
<BOUCLE_deux (ARTICLES) {id_rubrique}{2/2}{doublons}>
  <li> #TITRE
</BOUCLE_deux>
```

n'affichera pas tous les articles de la rubrique ! Imaginons par exemple qu'il y ait au total 20 articles dans notre rubrique. La `BOUCLE_prem` va afficher la première moitié des articles, c'est-à-dire les 10 premiers, et interdire (à cause de `{doublons}`) de les réutiliser. La `BOUCLE_deux`, elle, va récupérer la deuxième moitié des articles de cette rubrique *qui n'ont pas encore été affichés* par la `BOUCLE_prem` ; donc, la moitié des 10 articles suivants, c'est-à-dire les 5 derniers articles de la rubrique. Vous avez donc «perdu» 5 articles dans l'opération ...

8.16.4 Affichage *entre* les résultats

`{"inter"}` permet d'indiquer un code HTML (ici, `inter`) inséré entre les résultats de la boucle. Par exemple, pour séparer une liste d'auteurs par une virgule, on indiquera :

```
<BOUCLE_auteurs (AUTEURS) {id_article}{", "}>
```

8.17 Les formulaires

16 août 2002 par ARNO*

SPIP permet une grande interaction du site avec les visiteurs ; pour cela, il propose de nombreux formulaires sur le site public, permettant tantôt de gérer les accès à l'espace privé, tantôt d'autoriser l'ajout de messages et signatures.

Les formulaires s'insèrent dans les squelettes par une simple balise ; SPIP se charge ensuite de gérer le comportement (souvent complexe) de ces formulaires en fonction de l'environnement et des configurations effectuées dans l'espace privé.

#FORMULAIRE_INSCRIPTION

Sans doute le plus important, le #FORMULAIRE_INSCRIPTION gère l'inscription des nouveaux rédacteurs. Il n'affiche une interface d'inscription que si vous avez autorisé l'inscription automatique depuis le site public (sinon, cette balise n'affiche rigoureusement rien).

L'inscription nécessite l'envoi des informations de connexion (login et mot de passe) par email ; donc ce formulaire ne fonctionne que si votre hébergeur autorise l'envoi de mails par PHP.

#LOGIN_PRIVÉ

[SPIP 1.4] Tout aussi important (sinon plus), le #LOGIN_PRIVÉ affiche le formulaire d'accès à l'espace privé (la partie «/écrire» du site).

Important : cette balise doit impérativement être présente dans le squelette appelé par la page `spip_login.php3`, c'est-à-dire en standard par le squelette nommé `login-dist.html`. En effet, lors des accès directs à l'adresse «/écrire» de votre site, c'est vers `spip_login.php3` que SPIP va vous rediriger.

#LOGIN_PUBLIC

[SPIP 1.4] D'une utilisation beaucoup plus spécifique, #LOGIN_PUBLIC affiche un formulaire permettant à vos utilisateurs de s'identifier tout en restant sur le site public (sans entrer dans l'espace privé). Cette balise aura probablement, dans une future version de spip, un usage élargi, mais, pour l'instant, elle ne sert qu'à authentifier les visiteurs pour les sites proposant des forums *modérés sur abonnement*.

Le #LOGIN_PUBLIC, par défaut, «boucle sur lui-même», c'est-à-dire que le formulaire revient sur la page où il se trouve. On peut cependant indiquer une page vers laquelle le formulaire mènera, sous la forme :

```
[ (#LOGIN_PUBLIC|mapage.php3) ]
```

Si votre site offre une inscription automatique à l'espace privé, les données de connexion à l'espace public sont identiques à celles de l'espace privé ; c'est-à-dire que les données envoyées à l'utilisateur pour s'identifier à l'espace public lui permettent également d'accéder à l'espace privé. Si, au contraire, vous avez interdit l'inscription automatique à l'espace privé, *il faut impérativement avoir au moins un article dont les forums seront réglés en mode «sur abonnement»* pour activer cette balise ; dès lors, SPIP pourra fournir des informations de connexion pour le site public sans accès à l'espace privé.

(Note pour les *techos* qui veulent faire des essais avec ce système : il faudra s'appuyer sur du php dans les squelettes, qui exploitera le tableau associatif `$auteur_session`. Cette variable est positionnée si et seulement si le visiteur est authentifié. Elle contient alors divers champs, dont `id_auteur`, `statut`, `nom` et `login`. Mais, répétons-le : pour l'instant SPIP ne propose pas de système d'accès restreint à des documents, des articles ou des rubriques ...)

#FORMULAIRE_RECHERCHE

Il s'agit du formulaire du moteur de recherche intégré à SPIP. Il est présenté dans l'article sur les boucles de recherche (section 8.18 page 131).

#FORMULAIRE_FORUM

Le #FORMULAIRE_FORUM gère l'interface permettant de poster des messages dans les forums publics. Il concerne donc en premier chef la boucle FORUMS (section 8.9 page 118).

Le formulaire dépend évidemment du choix des forums modérés a posteriori, a priori ou sur abonnement.

Dans le cas (très spécifique) où l'on a autorisé la présence de mots-clés dans les forums publics, on peut affiner le comportement de ce formulaire avec des variables de personnalisation (section 8.22 page 136).

#FORMULAIRE_SIGNATURE

Le #FORMULAIRE_SIGNATURE autorise la signature des pétitions associées aux articles (ce formulaire se place donc dans une boucle ARTICLES).

N.B. La signature des pétitions réclame obligatoirement une validation des signataires par email. Ce formulaire n'a donc d'intérêt que si votre hébergeur autorise l'envoi de mails par PHP.

#FORMULAIRE_SITE

[SPIP 1.4] Le #FORMULAIRE_SITE affiche une interface permettant aux visiteurs du site de proposer des référencements de sites. Ces sites apparaîtront comme «proposés» dans l'espace privé, en attendant une validation par les administrateurs.

Ce formulaire ne s'affiche que si vous avez activé l'option «Gérer un annuaire de sites» dans la *Configuration sur site* dans l'espace privé, et si vous avez réglé «Qui peut proposer des sites référencés» sur «les visiteurs du site public».

Les sites référencés étant, dans SPIP, attachés aux rubriques, on ne peut placer ce #FORMULAIRE_SITE qu'à l'intérieur d'une boucle RUBRIQUES (section 8.6 page 114).

#FORMULAIRE_ECRIRE_AUTEUR

[SPIP 1.4] Placé à l'intérieur d'une boucle AUTEURS (section 8.8 page 117), ce formulaire permet d'envoyer un mail à l'auteur (d'un article). Cela permet, en modifiant les squelettes (qui, par défaut, affichent les liens contenant les adresses email des auteurs des articles), de pouvoir écrire aux auteurs sans afficher leur adresse email sur le site public.

8.17.1 Feuilles de style

On peut notablement modifier l'interface graphique des formulaires par l'intermédiaire des feuilles de style (section 7.6 page 92), notamment les classes `forml`, `spip_encadrer` et `spip_bouton`.

8.18 Les boucles de recherche

4 mai 2001 par l'équipe de SPIP

SPIP dispose d'un moteur de recherche intégré. Il faut donc prévoir une page permettant d'afficher les résultats des recherches.

8.18.1 L'interface de recherche

Pour afficher le formulaire de l'interface de recherche, il suffit d'insérer la balise :

```
#FORMULAIRE_RECHERCHE
```

Par défaut, le formulaire enverra les requêtes vers une page `recherche.php3` ; vous devez donc réaliser un squelette `recherche.html` permettant d'afficher les résultats.

Vous pouvez décider d'utiliser une autre page d'affichage des résultats. Pour cela, il faut utiliser la balise de la manière suivante :

```
[ (#FORMULAIRE_RECHERCHE|adresse.php3) ]
```

où `adresse.php3` est la page vers laquelle vous désirez envoyer l'utilisateur.

8.18.2 Le squelette des résultats

Les boucles permettant d'afficher les résultats de la recherche sont, en réalité, des boucles déjà abordées ici : `ARTICLES` (section 8.5 page 112), `RUBRIQUES` (section 8.6 page 114), `BREVES` (section 8.7 page 116). Vous pouvez en effet effectuer des recherches non seulement sur les articles, mais aussi sur les rubriques et les brèves.

La seule différence, par rapport à ce qui est documenté sur les pages de ces boucles, est le choix du critère de sélection, qui doit être `{recherche}`. Les critères d'affichage et les balises de ces boucles sont inchangées.

Cependant, afin de classer les résultats par pertinence, on utilisera de préférence ce nouveau critère d'affichage : `{par points}`.

Enfin, on pourra utiliser la balise `#POINTS`, qui indique la pertinence des résultats (attention, dans l'absolu cette valeur n'est pas très explicite, elle est surtout utile pour le classement des résultats).

8.19 Les filtres de SPIP

3 mai 2001 par l'équipe de SPIP

Nous avons vu dans la syntaxe des balises SPIP (section 8.4 page 109) qu'il était possible de modifier le comportement et l'affichage des balises en leur attribuant des filtres.

```
[ option avant (#BALISE|filtre1|filtre2|...|filtren) option après ]
```

Les filtres 1, 2, ..., n sont appliqués successivement à la `#BALISE`.

8.19.1 Les filtres de mise en page

- ▷ `majuscules` fait passer le texte en majuscules. Par rapport à la fonction de PHP, `majuscules` s'applique également aux lettres accentuées.
- ▷ `justifier` fait passer le texte en justification totale (`<P align=justify>`).
- ▷ `aligner_droite` fait passer le texte en justification à droite (`<P align=right>`).
- ▷ `aligner_gauche` fait passer le texte en justification à gauche (`<P align=left>`).
- ▷ `centrer` centre le texte (`<P align=center>`).

8.19.2 Les filtres des dates

Les filtres suivants s'appliquent aux dates (`[(#DATE|affdate)]` par exemple).

- ▷ `affdate` affiche la date en français, par exemple «13 janvier 2001».
- ▷ `jour` affiche le jour (en nombre).
- ▷ `mois` affiche le mois (en nombre).
- ▷ `annee` affiche l'année.
- ▷ [SPIP 1.0.2] `heures` affiche les heures d'une date (les dates fournies par SPIP contiennent non seulement le jour, mais également les horaires).
- ▷ [SPIP 1.0.2] `minutes` affiche les minutes d'une date.
- ▷ [SPIP 1.0.2] `secondes` affiche les secondes.
- ▷ `nom_jour` affiche le nom du jour (lundi, mardi ...).
- ▷ `nom_mois` affiche le nom du mois (janvier, février ...).
- ▷ `saison` affiche la saison (hiver, été ...).

8.19.3 Filtres de logos

- ▷ `fichier` [SPIP 1.4]. Affecté à un logo, ce filtre permet de récupérer directement l'URL du fichier du logo.
- ▷ `||autres filtres` Contrairement aux versions précédentes, [SPIP 1.4] permet de passer des filtres «maison» sur les logos : la logique est un peu tordue, car il fallait respecter la compatibilité avec SPIP 1.3. L'analyse se déroule comme suit :
 - * si le premier «filtre» n'est pas un alignement, SPIP considère qu'il s'agit d'un URL et fait un lien du logo vers cette adresse ;
 - * si le premier filtre est un alignement, SPIP considère que le deuxième «filtre» est un URL ;
 - * les filtres suivants sont de vrais filtres au sens habituel (y compris des filtres «maison» déclarés dans `mes_fonctions.php3`) ;
 - * pour appliquer un filtre quelconque sans mettre d'URL, il faut mettre deux barres.
Par exemple : `<?php $logo = ' [(#LOGO_RUBRIQUE||texte_script)] ' ; ?>` permet de récupérer le logo dans la variable php `$logo`, pour traitement ultérieur (voir ci-dessous pour la signification de `|texte_script`).

8.19.4 Filtres de texte

La plupart de ces filtres ont été introduits dans la version [SPIP 1.4]

- ▷ `liens_ouvrants` transforme les liens SPIP qui donnent vers des sites extérieurs en liens de type «popup», qui ouvrent dans une nouvelle fenêtre ; c'est l'équivalent du `target=blank` du HTML. *N.B. : les développeurs de SPIP estiment qu'il s'agit en général d'une impolitesse, car les internautes savent très bien s'ils ont envie ou pas d'ouvrir une nouvelle fenêtre - or ce système le leur impose. Mais la demande était trop forte, et nous avons craqué ;-)*
- ▷ `supprimer_numero` sert à éliminer le numéro d'un titre, si par exemple on veut faire des tris d'article `{par num titre}` mais ne pas afficher les numéros (car ils ne servent qu'à ordonner les articles).
- ▷ `PtoBR` transforme les sauts de paragraphe en simples passages à la ligne, ce qui permet de «resserrer» une mise en page, par exemple à l'intérieur d'un sommaire
- ▷ `taille_en_octets` permet de transformer un nombre d'octets (25678906) en une chaîne de caractères plus explicite («24.4 Mo»).
- ▷ `supprimer_tags` est une suppression basique et brutale de tous les `<...>`
- ▷ `textebrut` s'apparente au filtre `supprimer_tags`, mais il agit de manière un peu plus subtile, transformant notamment les paragraphes et `
` en sauts de ligne, et les espaces insécables en espaces simples. Utilisation, par exemple, pour faire un descriptif META à partir du
`#DESCRIPTIF : [<meta name='description' content=' (#DESCRIPTIF|textebrut) ' >]`

8.19.5 Filtres techniques

Ces filtres ont été introduits par [SPIP 1.4].

- ▷ `entites_html` transforme un texte en entités HTML, que l'on peut donc implanter dans un formulaire, exemple :
`[<textarea>(#DESCRIPTIF|entites_html)</textarea>]`
- ▷ `texte_script` transforme n'importe quel champ en une chaîne utilisable en PHP ou Javascript en toute sécurité, exemple : `<?php $x = '[(#TEXTE|texte_script)]'; ?>`. Attention : utilisez bien le caractère ' et non " : en effet, dans le second cas, si votre texte contient le symbole \$, le résultat peut être catastrophique (affichage partiel, affichage d'autre chose, plantage php, etc.).
- ▷ `attribut_html` rend une chaîne utilisable sans dommage comme attribut HTML ; par exemple, si l'on veut ajouter un texte de survol au lien normal vers un article, on utilisera

```
<a href="#URL_ARTICLE" [ title = "(#DESCRIPTIF|supprimer_tags
|attribut_html)" ]>#TITRE</a>
```

8.19.6 Ajouter ses propres fonctions

Les filtres de SPIP sont des fonctions PHP à une seule variable. Vous pouvez utiliser directement les fonctions habituelles de PHP, mais également créer les vôtres (à la condition qu'elles n'aient qu'une seule variable), sur le modèle :

```
<?php
function mon_filtre($texte){
    $texte = (bidouillages en PHP) ...;
    return $texte;
}
?>
```

Afin de ne pas avoir à modifier des fichiers de SPIP (qui risqueraient d'être écrasés lors d'une prochaine mise à jour), vous pouvez installer vos fonctions personnelles dans un fichier `mes_fonctions.php3` : si SPIP repère un fichier ayant ce nom, il l'inclut automatiquement.

Par exemple, ARNO* a développé le filtre `enlettres`, qui n'est pas inclus dans la distribution standard de SPIP. Ce filtre écrit un nombre en toutes lettres (`[(#DATE|annee|enlettres)] = «deux mille deux»`); ce filtre peut être téléchargé sur <http://rezo.net/spip-dev/contrib/ARNO/> ; il suffit de l'ajouter dans votre fichier `mes_fonctions.php3` pour l'utiliser.

Autre exemple, pour faire un filtre qui coupe les textes à une longueur donnée (ex : 50 caractères), on peut créer le filtre personnalisé suivant dans `mes_fonctions.php3` :

```
<?php
function couper50($texte) {
    return couper($texte, 50);
}
?>
```

8.20 Les boucles récursives

Les boucles récursives sont une fonction très puissante pour gérer la mise en forme de l'interface. Leur programmation est particulièrement simple, mais leur utilisation demande une bonne maîtrise logique de l'enchaînement des boucles.

L'appel d'une boucle récursive est très simple : il suffit d'indiquer dans le `TYPE` de la boucle le nom d'une autre boucle :

```
<BOUCLEn(boucler) ></BOUCLEn>
```

Il n'y a ici aucun critère : en réalité, la boucle `n` correspond à une copie pure et simple de la boucle `x`. L'ensemble de la boucle fonctionne comme si l'on avait recopié l'intégralité de la boucle `x` (toutes les balises et le code HTML, ainsi que les textes conditionnels avant, après et alternatif) à l'endroit où l'on insère la boucle `n`. (Il faut bien entendu que la boucle `x` précède la boucle `n`.)

L'utilisation la plus simple consiste à «dupliquer» une boucle sans avoir à la recopier. Ainsi, toute modification de la boucle d'origine `x` sera automatiquement dupliquée dans la boucle `n`.

Tout l'intérêt, en réalité, consiste à placer la boucle `n` à l'intérieur de la boucle `x` : on obtient ainsi un comportement récursif : la boucle `x` contient une boucle `n`, qui elle-même reproduit la boucle `x` qui contient la boucle `n`, et ainsi de suite, jusqu'à ce que la boucle `x` ne donne plus aucun résultat.

Cette technique permet de créer notamment l'affichage des thread des forums. Cela devient très simple : une première boucle «fabrique» l'entrée des threads (les messages qui répondent directement à un article), une seconde boucle affiche les réponses à ces messages, et une boucle récursive provoque la récursivité sur cette seconde boucle :

```
<BOUCLE_forum(FORUMS){id_article}>
<P>#TITRE

    <B_reponses>
    <UL>
    <BOUCLE_reponses(FORUMS){id_parent}>
    <LI>#TITRE
        <BOUCLE_recursive(boucle_reponses)>
    </BOUCLE_recursive> </BOUCLE_reponses>
    </UL>
</B_reponses>
</BOUCLE_forum>
```

On peut ainsi, en très peu de lignes, provoquer l'affichage de l'intégralité de la structure (rubriques, sous-rubriques...) du site.

8.21 <INCLUDE> d'autres squelettes

16 août 2002 par ARNO*

[SPIP 1.4] Lorsque l'on a des éléments de texte et des boucles communs à plusieurs fichiers, on peut vouloir extraire ces éléments des pages où ils se trouvent, les installer dans un fichier séparé, et les appeler depuis les autres squelettes. De cette façon, le code commun est regroupé dans un unique fichier, ce qui facilite notamment les modifications qui concernent plusieurs squelettes d'un seul coup.

Les habitués de PHP connaissent la fonction `include`, dont le principe est similaire à ce qui est présenté ici.

Dans SPIP, on peut appeler un fichier depuis un autre squelette grâce à la balise `<INCLUDE>` (on peut aussi utiliser `<INCLUDE>`, qui est identique). Sa syntaxe complète est :

```
<INCLUDE (fichier.php3) {paramètre1} {paramètre2} ...>
```

Le «`fichier.php3`» est le nom du fichier que l'on veut intégrer dans sa page. Par exemple, imaginons que toutes les pages du site affichent les mêmes informations en bas de page. On regroupe alors le code HTML de ce « `pied de page`» dans un fichier : « `pied.html`», squelette lui-même appelé par la page « `pied.php3`» (toujours selon le principe des couples de fichiers destinés à appeler des squelettes). Il suffit d'intégrer dans chacun des squelettes voulant appeler ce fichier :

```
<INCLUDE (pied.php3)>
```

Certaines inclusions peuvent dépendre d'une variable.

Par exemple, imaginons un squelette «`hierarchie.html/php3`», qui affiche la hiérarchie menant à une rubrique ; on appellerait cette page par une URL de la forme :

```
hierarchie.php3?id_rubrique=xxx.
```

Dans les squelettes voulant afficher la hiérarchie à partir de la rubrique courante, il faut donc indiquer que le paramètre concerné est `{id_rubrique}` ; si nécessaire, on aura créé une boucle permettant de récupérer le numéro de la rubrique concernée, et on installera le code suivant à l'intérieur de cette boucle :

```
<INCLUDE (hierarchie.php3) {id_rubrique}>
```

On peut imaginer que, dans certains squelettes, on désire récupérer non pas la hiérarchie en fonction d'une rubrique «`variable`» (la rubrique dans laquelle se trouve l'appel d'inclusion), mais en fonction d'une rubrique dont on connaît à l'avance le numéro. Pour cela, on peut fixer la valeur du paramètre ainsi :

```
<INCLUDE (hierarchie.php3) {id_rubrique=5}>
```

N.B. Il est possible d'indiquer plusieurs paramètres dans la balise `<INCLUDE>` ; cependant ce cas est rare. En effet, il est déjà rare dans un site sous SPIP de créer volontairement des URL à plusieurs variables, il est donc tout aussi rare d'avoir besoin d'inclusions avec plusieurs paramètres. En tout état de cause ces variables se cumuleraient par un «`et logique`» comme lorsqu'on les juxtapose dans la définition des boucles.

N.B. Le fichier inclus est lui-même un couple de fichiers de squelette. Ce fichier disposera donc de sa propre valeur de `$delais`. Cela peut s'avérer pratique pour éviter des recalculs d'éléments lourds du site, qui ne changent pas, en y affichant une liste d'éléments qui, eux, sont très régulièrement remis à jour.

8.22 Les variables de personnalisation

16 août 2002 par ARNO*

Certains comportements des pages de votre site peuvent être modifiés au moyen de variables PHP. Ces variables sont normalement définies par SPIP, mais, pour obtenir une personnalisation plus fine du site, le webmestre peut les modifier.

8.22.1 Où indiquer ces variables ?

Inutile d'entrer dans le code source de SPIP lui-même pour fixer ces variables (ouf!).

Pour l'ensemble du site

Si vous voulez fixer ces variables pour l'intégralité du site, vous pouvez les indiquer, avec une syntaxe un peu différente, dans un fichier intitulé `mes_fonctions.php3`, placé à la racine du site. (Il faudra éventuellement créer ce fichier, et entourer les définitions de vos variables par les marqueurs `<?php` et `?>`, voir les exemples ci-dessous.)

Pour chaque type de squelette

[SPIP 1.4] Vous pouvez aussi définir ces variables squelette par squelette. Pour cela, il faut les installer *au début* du fichier PHP appelant le squelette (par exemple `article.php3`, `rubrique.php3` ...). Ils s'insèrent naturellement à côté des variables obligatoires `$fond` et `$delais`.

8.22.2 Les variables du texte

Ces variables sont utilisées lors du calcul de la mise en page (correction typographique) par SPIP.

- ▷ `$debut_intertitre` fixe le code HTML inséré en ouverture des intertitres (par le raccourci `{{{}}`). En standard, sa valeur est :

```
$debut_intertitre = "\n&nbsp;<h3 class=\"spip\">\n";
```

- ▷ `$fin_intertitre` est le code HTML inséré en fermeture des intertitres (raccourci `}}}`). Sa valeur normale est :

```
$fin_intertitre = "\n</h3><br>\n";
```

N.B. L'intertitre standard proposé par SPIP peut sembler laid ou inefficace ; il s'explique par l'histoire du développement de ce programme ; sur votre propre site, vous pouvez évidemment choisir de créer un code qui plus «élégant» (en faisant notamment appel à une simple feuille de style).

- ▷ `$ouvre_ref` est le code d'ouverture des appels des notes de bas de page ; par défaut, c'est une espace insécable et un crochet ouvrant ;
- ▷ `$ferme_ref` est le code de fermeture des appels des notes de bas de page ; par défaut, c'est un crochet fermant.
- ▷ `$ouvre_note` est le code d'ouverture de la note de bas de page (telle qu'elle apparaît dans `#NOTES`) ; par défaut, un crochet ouvrant ;
- ▷ `$ferme_note` est le code de fermeture des notes de bas de page (un crochet fermant).

Des choix alternatifs pourront être par exemple d'utiliser des parenthèses ; ou, plus joliment, d'ouvrir avec le tag HTML `^{`, et de fermer avec `}`.

- ▷ Le fichier `puce.gif` et la variable `$puce`. Lorsque vous commencez une nouvelle ligne par un tiret, SPIP le remplace par une petite «puce» graphique. Cette puce est constituée par le fichier `puce.gif` installé à la racine du site ; vous pouvez modifier ce fichier selon vos besoins. Mais vous pouvez aussi décider de fixer vous-même le choix de la puce, au travers de la variable `$puce`. Par exemple pour indiquer un autre fichier graphique :

```
$puce = "<img src='mapuce.gif' alt='-' align='top' border='0'>";
```

ou par un élément HTML non graphique :

```
$puce = "----";
```


8.22.3 Les variables pour les forums publics

Il existe des variables permettant de fixer le comportement des forums publics *avec des mots-clés*.

N.B. Ces variables ne sont utilisées que lorsque vous créez des forums publics dans lesquels les visiteurs peuvent sélectionner des mots-clés ; leur utilisation est donc extrêmement spécifique (et pas évidente ...).

- ▷ `$afficher_texte` («oui»/«non»). Par défaut, les forums publics sont conçus pour permettre aux visiteurs d'entrer le texte de leur message ; mais lorsque l'on propose le choix de mots-clés dans ces forums, on peut décider qu'aucun message n'est utile, seul la sélection des mots-clés importe. Dans ce cas, on pourra indiquer :

```
$afficher_texte = "non";
```

- ▷ `$afficher_groupe` permet d'indiquer les différents groupes de mots-clés que l'on souhaite proposer dans tel forum. En effet, tous les forums sur un site ne sont pas forcément identiques, et si, à certains endroits, on peut vouloir afficher une sélection de tous les groupes de mots-clés (ceux que l'ont a rendu accessibles aux visiteurs depuis l'espace privé), à d'autres endroits, on peut vouloir n'utiliser que certains groupes, voire aucune groupe (pas de sélection de mots-clés du tout).

La variable `$afficher_groupe` est un tableau (array), et se construit donc de la façon suivante :

```
$afficher_groupe[] = 3;
$afficher_groupe[] = 5;
```

impose l'affichage *uniquement* des groupes 3 et 5.

```
$afficher_groupe[] = 0;
```

interdit l'utiliser des mots-clés dans ces forums (puisqu'il n'existe pas de groupe de mots-clés numéroté 0.

Si l'on n'indique rien (on ne précise pas `$afficher_groupe`), tous les groupes de mots-clés indiqués, dans l'espace privés, comme «proposés aux visiteurs du site public» sont utilisés.

8.22.4 Exemples

- ▷ Codes modifiés globalement, fichier `mes_fonctions.php3` (attention, la syntaxe change un peu, il faut ajouter la mention `$GLOBALS['xxx']` pour chacune des variables à personnaliser.

```
<?php
  $GLOBALS['debut_intertitre'] = "<h3 class='mon_style_h3'>";
  $GLOBALS['fin_intertitre'] = "</h3>";

  $GLOBALS['ouvre_ref'] = ' &nbsp; (';
  $GLOBALS['ferme_ref'] = ')';
  $GLOBALS['ouvre_note'] = ' (';
  $GLOBALS['ferme_note'] = ') ';

  $GLOBALS['espace_logos'] = 0;
?>
```

- ▷ Codes modifiés pour les seules pages de rubrique, fichier `rubrique.php3` :

```
<?php
  $fond = "rubrique";
  $delais = 2 * 3600;
```

```
$espace_logos = 20;
include ("inc-public.php3");
?>
```

8.23 Utiliser des URLs personnalisées

1er mai 2001 par l'équipe de SPIP

Par défaut, les pages générées par SPIP utilisent des adresses relatives ressemblant à `article.php3?id_article=123`, donnant des URLs du type `http://www.minirezo.net/article.php3?id_article=123`. Ce type de syntaxe, courant chez les sites «dynamiques», n'est cependant pas très joli ni très évocateur. Il y a possibilité d'avoir des adresses plus à votre goût - par exemple `article123.html` - , et SPIP vous aide en partie dans cette tâche.

Cette fonctionnalité fait appel à la distinction entre deux types d'URLs :

- ▷ *l'URL apparente* d'une page, c'est-à-dire telle qu'elle est tapée et/ou affichée dans la barre d'adresse du navigateur. Par exemple `http://www.uzine.net/article765.html`. Ce sont ces URLs qu'on cherche à rendre plus «jolies» ou plus «signifiantes» ;
- ▷ *l'URL réelle* de la page, c'est-à-dire l'URL qui est «vue» par SPIP lorsque la page est calculée sur le serveur. Par exemple `http://www.uzine.net/article.php3?id_article=765` ; en général, cette URL peut aussi être tapée directement dans le navigateur (vous pouvez vérifier).

8.23.1 Choisir le type d'URLs apparentes

Le fichier `inc-urls.php3` à la racine de SPIP, contient la déclaration d'une variable PHP contenant le type d'URLs à utiliser. Par défaut :

```
$type_urls = "standard";
```

Cette variable détermine le nom du fichier PHP qui est appelé pour gérer les URLs. Avec la déclaration par défaut ci-dessus, c'est `inc-urls-standard.php3`. Vous remarquerez qu'il y a aussi un fichier `inc-urls-html.php3`. Il permet de traiter des adresses du type de celles que nous avons prises comme exemple («`article123.html`»). Vous pouvez donc décider d'utiliser plutôt ce fichier que le fichier «standard» en remplaçant la ligne sus-citée par la suivante :

```
$type_urls = "html";
```

Si vous voulez plutôt utiliser vos propres adresses (ce pour quoi vous devez savoir programmer en PHP), il est fortement conseillé de partir d'un des fichiers existants et de le recopier sous le nom que vous aurez choisi : `inc-urls-XXX.php3`.

Voyons maintenant les différents types de modifications à apporter.

8.23.2 Programmer la traduction des adresses apparentes en adresses réelles

Pour que l'adresse `article123.html` appelle bien en réalité le fichier PHP `article.php3` avec comme paramètre `id_article=123`, il va falloir configurer le serveur Web qui héberge votre site, soit dans un fichier `.htaccess` (ça ne marche pas toujours), soit dans le fichier de configuration centrale du serveur si vous y avez accès. Cela utilise,

sous le serveur Apache² (le plus utilisé), ce qu'on appelle des *Rewrite Rules* : des règles de réécriture d'adresses Web.

Savoir écrire ces règles n'est pas simple pour les non-programmeurs, et nous ne pouvons pas vous donner de solutions infaillibles car cela dépend de votre configuration : cette partie est entièrement entre vos mains (ou celles de votre hébergeur).

Néanmoins et à titre d'exemple, voici les règles utilisées sur le site d'uZine :

```
RewriteEngine on

# urls spip
RewriteRule ^/rubrique([0-9]+)\.html$ /rubrique.php3?id_rubrique=$1 [QSA,L]
RewriteRule ^/article([0-9]+)\.html$ /article.php3?id_article=$1 [QSA,L]
RewriteRule ^/breve([0-9]+)\.html$ /breve.php3?id_breve=$1 [QSA,L]
RewriteRule ^/secteur([0-9]+)\.html$ /secteur.php3?id_rubrique=$1 [QSA,L]
```

(note si vous écrivez vos propres règles : utilisez toujours l'option [QSA] car l'adresse peut présenter des paramètres supplémentaires - comme lorsque vous appuyez sur le bouton «recalcul», affiché si vous placez le cookie d'administration ; l'option [L] signifie, quant à elle, que la réécriture s'arrête là si la règle est appliquée : Last)

Il est conseillé de tester la validité de ces adresses (en les essayant depuis votre navigateur) dès que vous avez mis en place la configuration correspondante, car sinon la suite ne servirait à rien.

8.23.3 Générer les URLs apparentes dans les pages SPIP

SPIP incorpore une fonctionnalité permettant de générer automatiquement les URLs du type que vous aurez choisi, à l'intérieur même des pages calculées par SPIP. Ainsi, les liens vers des articles seront par exemple générés sous la forme `article123.html`.

Les fichiers `inc-urls-XXX.php3` sont là à cet effet. Si vous avez décidé de créer le vôtre propre, il va falloir programmer vous-mêmes la génération des URLs. Sinon, c'est déjà fait. Vous découvrirez qu'un certain nombre de fonctions PHP sont définies, dans le fichier, sous le nom `generer_url_type` d'objet (par exemple `generer_url_article`). Le rôle de ces fonctions devrait vous être évident pour peu que vous compreniez un minimum le PHP. A vous de les modifier à votre guise. A titre d'exemple, vous pouvez vous plonger dans le fichier `inc-urls du Monde diplo`³.

Dans tous les cas, n'oubliez pas d'aller modifier le type choisi à l'intérieur de `inc-urls.php3`.

Enfin, dans vos squelettes, vérifiez que pour calculer les liens à l'intérieur des boucles, vous utilisez toujours les raccourcis `#URL_ARTICLE`, `#URL_RUBRIQUE`, `#URL_BREVE`, etc. Ce sont en effet eux qui déclenchent l'utilisation des fonctions évoquées ci-dessus, permettant la génération des «bonnes» URLs.

8.24 La «popularité» des articles

31 juillet 2002 par l'équipe de SPIP

La notion de popularité, exposée ci-dessous, apparaît dans [SPIP 1.4](#).

²<http://httpd.apache.org/>

³<http://rezo.net/spip-dev/contrib/Fil/inc-urls-diplo.php3>

8.24.1 Comment décompter des visites

Des centaines de méthodes statistiques existent pour décompter des visites sur un site donné. La plupart donnent des courbes horaires, ou par jour, qui permettent de savoir si son site «monte» ou «descend», et de vérifier qu'il y a plus de gens sur le net en fin d'après-midi et dans la semaine, que le week-end ou la nuit . . .

Notre objectif est un peu différent : il s'agit d'attribuer à chaque article une valeur de «popularité» reflétant assez rapidement une tendance, et permettant de comparer l'activité de différents articles, soit de manière globale sur tout le site (hit-parade), soit à l'intérieur d'une rubrique, soit parmi les articles d'un même auteur, etc.

La méthode retenue est la suivante :

- ▷ chaque visite sur un article ajoute un certain nombre de points à cet article ; 1 point si c'est un article que l'on consulte depuis le site lui-même en suivant un lien, et 2 points si c'est une «entrée directe» depuis un site extérieur (moteur de recherche, lien hypertexte, syndication . . .)
- ▷ toutes les 10 minutes, le score obtenu est multiplié par un petit facteur d'escompte, qui fait qu'un point attribué par une visite à 10h12 le mercredi ne vaut plus, le lendemain à la même heure, qu'un demi-point, et, le vendredi à 10h12, un quart de point . . . ;
- ▷ le tout est calculé de manière à ce que, dans l'hypothèse où l'article reçoit toujours le même nombre x de visites par unité de temps, son score se stabilise sur cette valeur x . Autrement dit, si la fréquentation de l'article est stationnaire, sa popularité finira par refléter exactement son nombre de visites par jour (modulo le score 2 donné pour les entrées directes) ;
- ▷ cette popularité s'exprime de deux manières : l'une, la popularité_absolue, exprime le score en question (évaluation de la fréquentation quotidienne de l'article) ; l'autre, la popularité_relative, un pourcentage relatif à l'article du site ayant la plus forte popularité (popularité_max) ;
- ▷ enfin, la somme de toutes ces valeurs (absolues) sur le site donne la popularité_site, qui permet de comparer la fréquentation de deux sites sous spip . . .

8.24.2 Balises

Des balises permettent de récupérer et d'afficher ces valeurs. La boucle ci-dessous résume l'ensemble de ces balises :

```
<BOUCLE_pop(ARTICLES){id_article}{popularite>0}>
<h5>Popularité</h5>
Cet article a une popularité absolue égale à #POPULARITE_ABSOLUE, soit
#POPULARITE % de #POPULARITE_MAX. Au total, ce site fait environ
#POPULARITE_SITE visites par jour.
</BOUCLE_pop>
```

NB : bien que les données soient représentées, dans la base de spip, sous forme de nombres réels, le rendu de toutes ces balises est toujours donné sous la forme d'un nombre entier, ce qui donnera, sur des sites très peu fréquentés (sites de tests, notamment), des choses amusantes du genre :

«Cet article a une popularité absolue égale à 1, soit 17 % de 2.» Au total, ce site fait environ 5 visites par jour.»

8.24.3 Critères

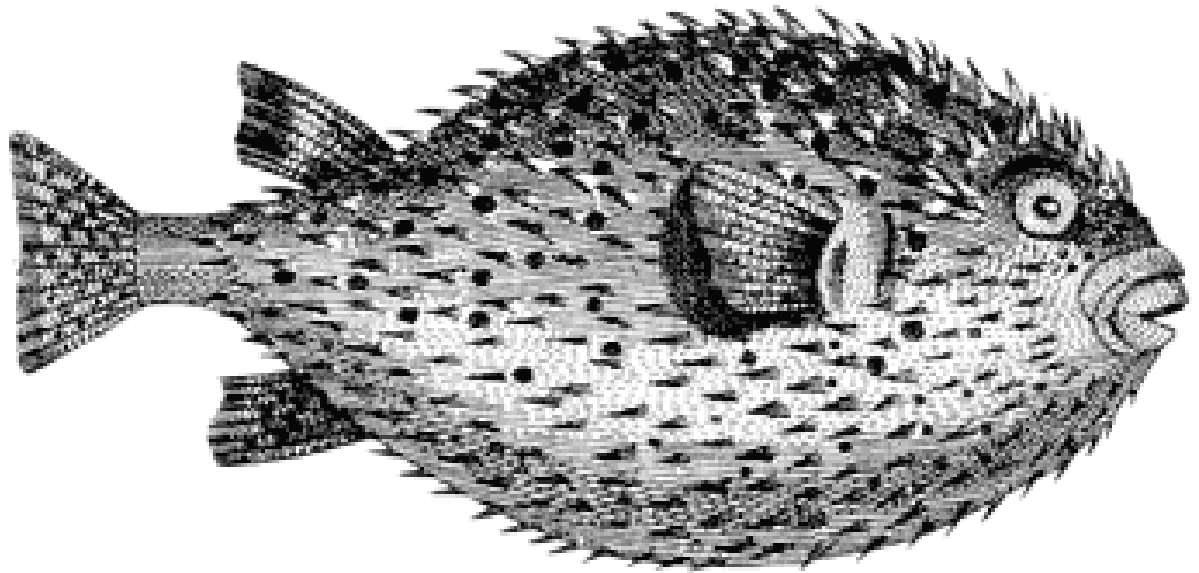
Enfin, un critère de tri peu se révéler utile : {par popularite}, que l'on utilisera par exemple de la manière suivante pour afficher la liste des 10 articles les plus populaires de la rubrique courante :

```
<BOUCLE_hitparade (ARTICLES) {id_rubrique}{par popularite}{inverse}{0,10}>  
...  
</BOUCLE_hitparade>
```

(On enlève {id_rubrique} pour afficher un hit-parade du site.)

Chapitre 9

SPIP pas à pas



SPIP pas à pas

9.1 Mon premier squelette

12 juin 2001 par l'équipe de SPIP

(je le sors du placard)

Comment écrire un premier squelette qui marchouille

Si le système de squelettes peut de prime abord paraître intimidant, c'est que ce qu'on lui demande est suffisamment riche pour l'obliger à être complexe. Mais ! Complexe ne veut pas dire compliqué. Voici un exemple minimal de squelette.

On supposera, pour commencer, que votre base SPIP contient au minimum une rubrique et deux articles publiés.

Avant d'utiliser un squelette, il faut pouvoir l'appeler : créez à la racine de votre site un fichier `tutoriel.php3` contenant les lignes suivantes

```
<?
$fond = "tutoriel";
$delais = 0;
include "inc-public.php3";
?>
```

Puis testez dans votre navigateur : `http://votresite.net/tutoriel.php3`. Pas très glorieux, n'est-ce pas ? Le message d'erreur vous informe qu'il manque un fichier. C'est la fameux squelette, que nous allons maintenant créer.

A la racine du site, déposez un fichier «`tutoriel.html`», qui contient ce qui suit :

```
<BOUCLE_article(ARTICLES){id_article=1}>
#TITRE
</BOUCLE_article>
```

Puis rechargez la page `http://votresite.net/tutoriel.php3`. C'est mieux, n'est-ce pas ? SPIP est allé chercher le titre de l'article n° 1 de votre base, et l'a inscrit à la place de `#TITRE`.

Si ça ne fonctionne pas, vérifiez :

- ▷ que votre article n°1 est bien «publié» (et pas «en attente» ou «en cours de rédaction»).
- ▷ que la `<BOUCLE...>` ne commence pas sur le premier caractère du squelette (bug de SPIP jusqu'à la version 1.0beta24.)

Puis ajoutez du HTML et d'autres appels de champs SPIP, et vous obtenez rapidement votre article n° 1 :

```
<BOUCLE_article(ARTICLES){id_article=1}>
<H1>#TITRE</H1>
<B>#CHAPO</B>
<P align=justify>#TEXTE
</BOUCLE_article>
```

Ajoutez ensuite les champs manquants : `#SURTTITRE`, `#LESAUTEURS`, `#SOUSTITRE`, `#NOTES`, etc.

Bien !

9.2 Un squelette, plusieurs articles

9 juin 2001 par l'équipe de SPIP

c'est à ça que ça sert ...

Et voici le premier contexte.

La leçon précédente nous a permis d'extraire des données de l'article n°1 de la base et d'en faire une page Web. Généralisons ...

Notre squelette est bien inutile s'il ne sert qu'à afficher l'article n°1. Apprenons-lui à afficher n'importe quel article :

Pour cela nous allons appeler notre page Web avec une variable `id_article=2` : pointez le navigateur sur l'URL `http://votresite.net/tutoriel.php3?id_article=2`.

S'affiche ... l'article 1. Modifions dans le squelette `tutoriel.html` la ligne qui définit la «boucle article» :

```
<BOUCLE_article(ARTICLES){id_article}>
```

(On remplace `{id_article=1}` par `{id_article}`.)

Voilà : `http://votresite.net/tutoriel.php3?id_article=2` vous donne l'article 2. Non ? Il devrait ...

La `BOUCLE_article` s'exécute dans un contexte où `id_article=2`. Si on lui précise `{id_article=1}` elle va chercher l'article n° 1, mais si on lui demande `{id_article}`, elle va chercher l'article dont le numéro est précisé dans le contexte.

Cliquez maintenant sur :

- ▷ `http://votresite.net/tutoriel.php3?id_article=1`,
- ▷ `http://votresite.net/tutoriel.php3?id_article=2` et
- ▷ `http://votresite.net/tutoriel.php3`.

Voyez-vous la différence ? Les deux premières pages vous donnent les articles n°1 et 2, la troisième n'a pas d'`id_article` dans son contexte, et génère une erreur.

Bravo ! Votre squelette est maintenant «contextuel».

9.3 Une rubrique

7 juin 2001 par l'équipe de SPIP

ou comment faire des listes du contenu de la base

Faire des listes avec une boucle SPIP

La leçon précédente nous a appris à afficher des éléments en fonction du contexte. Nous allons ici voir comment ce contexte varie au fur et à mesure des BOUCLES rencontrées.

Modifions notre squelette «`tutoriel.html`» de la manière suivante :

```
<BOUCLE_article(ARTICLES)>  
#TITRE<BR>  
</BOUCLE_article>
```

Là, on supprime carrément la condition `{id_article=1}`. Attention : cette BOUCLE va générer une page énorme si votre base contient déjà pas mal d'articles : mieux vaut prendre nos précautions et ajouter tout de suite `0,10` pour limiter aux 10 premiers articles ...

```
<BOUCLE_article(ARTICLES){0,10}>
```

Résultat : les titres des 10 premiers articles de votre base s'affichent, séparés par un saut de ligne. A partir de là, on voit comment on peut produire le sommaire d'une rubrique : affichons les 10 articles les plus récents appartenant à cette rubrique.

```
<BOUCLE_article(ARTICLES){id_rubrique}{par date}{inverse}{0,10}>
<a href=#URL_ARTICLE>#TITRE</A><BR>
</BOUCLE_article>
```

Prenons dans l'ordre :

- ▷ `{id_rubrique}` : ne prend que les articles appartenant à la rubrique `id_rubrique` (cf. ci-dessous pour que cette variable soit définie dans le contexte de notre `BOUCLE_article`).
- ▷ `{par date}{inverse}` : tri par date dans l'ordre décroissant ...
- ▷ `{0,10}` : ... et prend les 10 premiers résultats.
- ▷ Enfin, `#TITRE` va écrire non seulement le titre de l'article mais en plus créer un lien vers cet article.

Reste à invoquer le squelette, *en lui passant le contexte* `id_rubrique=1` :

`http://votresite.net/tutoriel.php3?id_rubrique=1.`

La magie de SPIP tient dans la combinaison de ce type de fonctionnalités. Si vous êtes arrivé jusqu'ici, c'est gagné !

9.4 Boucles en boucles

5 juin 2001 par l'équipe de SPIP

plusieurs niveaux de lecture

Affichons sur une même page des éléments en provenance de plusieurs endroits.

Nous savons générer une liste de titres dans une rubrique. Maintenant, nous allons afficher, sur la même page, les éléments de la rubrique elle-même : son titre, son texte de présentation, etc.

Essayez !

Et voici une solution :

```
<BOUCLE_rubrique(RUBRIQUES){id_rubrique}>
<H1>#TITRE</H1>

<BOUCLE_article(ARTICLES){id_rubrique}{par date}{inverse}{0,10}>
<a href=#URL_ARTICLE>#TITRE</A><BR>
</BOUCLE_article>

[ (#TEXTE|justifier) ]
</BOUCLE_rubrique>
```

On appelle la page avec `http://votresite.net/tutoriel.php3?id_rubrique=1`. Que s'est-il passé ici ?

Notre boucle `ARTICLES` est intégrée dans une boucle `RUBRIQUES`. Le contexte de la boucle `ARTICLES` est l'`id_rubrique` donné par la boucle `RUBRIQUES`, qui elle-même va chercher le contexte donné par l'URL (`?id_rubrique=1`). Donc nous sommes bien, au niveau des `ARTICLES`, avec l'`id_rubrique` demandé. De ce point de vue rien ne change.

En revanche, la boucle `RUBRIQUES` a permis à SPIP de sélectionner les valeurs des champs de la rubrique en question : on peut donc afficher le `#TITRE` et le `#TEXTE` de cette rubrique. Notez bien que le `#TEXTE` serait celui de la rubrique **même si** on appelait `#TEXTE` dans la boucle `ARTICLES`. Le fonctionnement arborescent de SPIP garantit que le `#TEXTE` d'un article ne déborde pas de la boucle `ARTICLES` ...

Dernière remarque : on a introduit un filtre `ljustifier` sur le champ `#TEXTE`. Ce filtre modifie le contenu du texte avant de l'installer dans la page finale. Ca vous fait saliver ?

9.5 Gérer le cache

3 juin 2001 par l'équipe de SPIP

et éviter de faire ramer le serveur qui n'a pas que ça à faire

Le cache, ou comment faire un site dynamique qui ne bouge pas trop.

Dans les leçons précédentes nous avons commencé à élaborer des squelettes. Le succès de notre site risque d'être fulgurant. Pensons tout de suite aux pauvres neurones de notre ordinateur. Dans cette leçon, rien d'amusant, rien d'essentiel non plus. Les flemmards en profiteront pour roupiller au fond près du radiateur ...

Résumé pour ceux-ci et pour les gens pressés :

dans les fichiers d'appel de type `tutoriel.php3`, réglez `$delais = 3600` ; au lieu de 0.

◇◇◇

Au moment où une page est demandée à SPIP, celui-ci regarde si, par hasard, il n'aurait pas déjà calculé cette page auparavant. Si l'URL demandée est `http://votresite.net/tutoriel.php3?id_article=12`, SPIP regarde dans son sous-répertoire `CACHE/` si ce fichier existe, et, le cas échéant, compare l'âge du fichier caché aux `$delais` fixés dans le fichier d'appel `tutoriel.php3`.

Dans notre exemple nous avons fixé des `$delais=0` ; - d'où un recalcul systématique des pages à chaque consultation du site. Passons à `$delais=3600` ; (c'est en secondes).

Notre page web n'est donc recalculée que si, lorsqu'un visiteur la demande, sa version cachée date de plus d'une heure (soit 3600 s.). Sinon, SPIP lit simplement le contenu du fichier caché¹, et renvoie le résultat sans se connecter à la base de données (sauf pour y insérer un «hit» dans les statistiques).

Comment fixer ces `$delais` de manière à optimiser le rapport réactivité/charge du serveur ? Pas de solution miracle, mais n'hésitez pas à fixer un délai d'une journée (i.e. `$delais=24*3600` ;) ou plus pour les articles et les rubriques. Les pages de navigation les plus importantes peuvent avoir des `$delais` plus courts (vingt minutes ou une heure par exemple) si votre site est censé réagir à la validation fréquente de nouvelles brèves et de sites syndiqués ... Si vous êtes sur un serveur partagé avec d'autres sites, soyez respectueux des autres et ne prenez pas tout le temps de calcul pour des pages qui changent rarement : ce serait d'autant plus idiot que, sur les gros articles ou sur les sommaires, le calcul des pages peut prendre quelques secondes, ce qui ralentit la consultation de vos pages ...

Comment provoquer une mise à jour hors délai ? Nous venons de décider de `$delais` extrêmement longs, et nous repérons une fôte d'ortographe dans une page. Correction dans le back-office ... Comment effacer tout de suite cette vilaine cicatrice du site ?

¹Pour les spécialistes, il s'agit en fait d'un `include PHP` du fichier correspondant, permettant d'exécuter du code depuis le cache ...

- ▷ Depuis le back-office, cliquer sur «Voir en ligne» déclenche le recalcul pour les pages correspondant à #URL_ARTICLE ou #URL_RUBRIQUE de l'article ou de la rubrique correspondante. C'est le cas le plus courant. Mais sinon ?
- ▷ Dans la partie «Sauvegarde/Restauration» du back-office, un bouton «vider le cache» efface tous les fichiers cachés (utile si vous faites plein de modifications et avez un site très complexe, à éviter sinon).
- ▷ Toutefois, la solution la plus simple est de demander à SPIP, dans la page d'accueil du back-office, de vous «poser un cookie». Ce cookie s'incrusterait sur votre navigateur, et SPIP vous reconnaîtra au moment de vous envoyer la page dans le site public : il vous proposera alors, en bas de page, un bouton «Recalculer cette page».

Retour au contexte : On revient ici à la notion de contexte. Si le squelette est appelé avec un contexte d'id_article, d'id_rubrique ou encore d'id_breve, un autre bouton vous est proposé quand SPIP détecte le cookie : «Modifier cet article (ou rubrique, ou brève)», qui vous mène directement sur la page correspondante dans le back-office. Merci qui ?

Derniers détails :

- ▷ pour des raisons évidentes, le moteur de recherche ne déclenche pas de cache, et les pages avec forum sont réactualisées dès qu'une nouvelle contribution est envoyée.
- ▷ le répertoire CACHE/ dans l'arborescence du site est découpé en 16 sous-répertoires numérotés 0, 1, 2 ... 9, A, B ... F, dans lesquels les fichiers cachés se distribuent quasi-aléatoirement ; cela s'appelle «*hasher le cache*» et rien que pour cela mérite bien qu'on le mentionne.
- ▷ les fichiers cachés sont exploités même si la base de données est «tombée», ce qui garantit le site contre des pannes transitoires du serveur MySQL.

9.6 Des filtres

1er juin 2001 par l'équipe de SPIP

Subtilités squelettiques

Les filtres transforment le contenu de la base de données en code HTML présentable.

Si les BOUCLES permettent de structurer la page de manière logique, reste à présenter les données de manière esthétique. Question dizahigne SPIP ne peut rien pour vous, mais sachez user des ces filtres ...

Une donnée stockée dans la base MySQL se présente comme un bloc, et on peut avoir envie de manipuler la variable correspondante avant de l'afficher à l'écran. Les filtres sont faits pour ça :

- ▷ les filtres les plus utilisés (ils sont appelés systématiquement) sont |typo et |propre ; le premier est un correcteur typographique, dont la mission principale est d'ajouter des espaces insécables où il en faut (*cf.* aide en ligne de SPIP) ; le second s'intéresse aux paragraphes, aux raccourcis SPIP (italiques, gras, intertitres, notes de bas de page, etc.) - il n'est appliqué par défaut qu'aux champs longs (#TEXTE, #CHAPO, etc.)
- ▷ d'autres filtres sont très utiles : citons |majuscules (à la fonctionnalité évidente), |justifier ou |aligner_droite (qui définissent l'alignement du texte), ou encore l'ésotérique |saison (qui affiche «été» si la variable est une date comprise entre le 21 juin et le 20 septembre) ...

Pour utiliser un filtre il faut entourer la variable de parenthèses et de crochets (on verra plus tard les implications) :
[blah blah (#VARIABLE|filtre) bloh bloh]

On peut enchaîner les filtres dans un pipeline : ainsi [(#DATE_REDAC|saison|majuscules)] affichera-t-il «HIVER».

Exercice portant sur l'ensemble des leçons précédentes : Afficher en majuscules les titres des 10 articles les plus récents de la rubrique passée en contexte, et mettre en tête de page la saison courante (c'est-à-dire la saison à laquelle a été publié l'article le plus récent de toute la base).

◇ ◇ ◇

Pourquoi ces crochets ? supposons que votre base contient des articles datés et d'autres non datés. La variable #DATE vaut «2001-07-01 10-53-01» (date au format MySQL) dans le premier cas, et «0000-00-00 00-00-00» dans le second. Pour afficher la date dans un joli (?) cadre, on va utiliser, dans le squelette, les lignes suivantes :

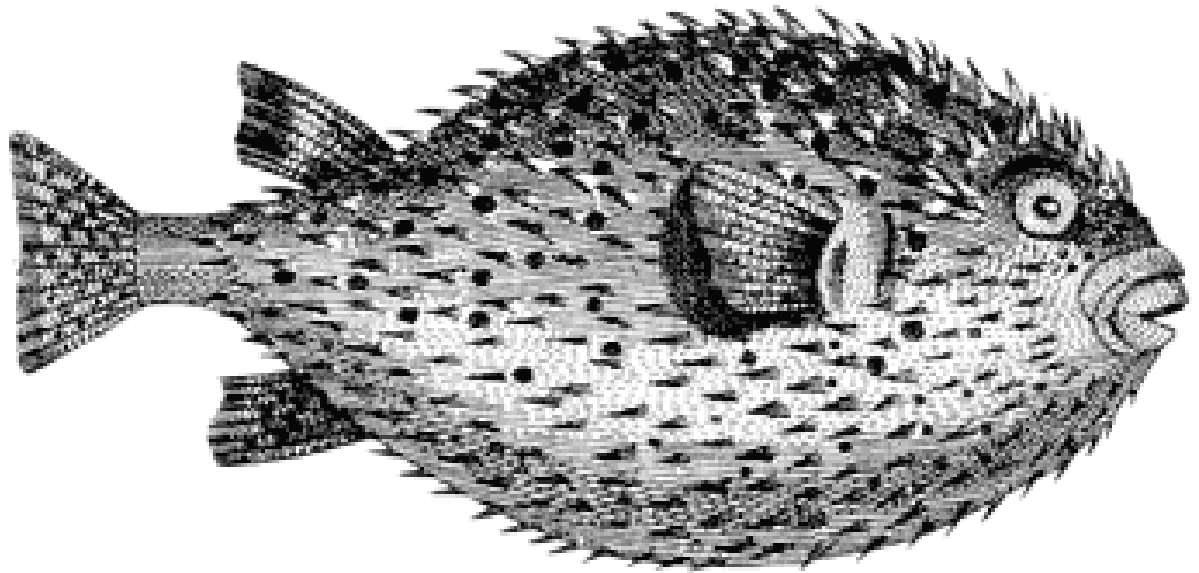
```
[<TABLE border=1><TR><TD>
(#DATE|affdate)
</TABLE>]
```

Ici le filtre |affdate affiche la date au format «1er juillet 2001», et renvoie une chaîne vide si la date est «0000...». Les crochets délimitent ce qu'il faut afficher *si le resultat entre parenthèses n'est pas une chaîne vide*.

Résultat : seuls les articles datés provoquent l'affichage d'un tableau contenant la date. Un squelette bien construit définira précisément ce qu'il faut afficher *ou pas* en fonction du contenu ... Les filtres servent aussi à ça.

Chapitre 10

Trucs et astuces



Trucs et astuces

10.1 Présenter les résultats d'une recherche par secteurs

11 avril 2002

Il suffit d'inclure la boucle de recherche dans une boucle de type rubriques sélectionnant les rubriques de premier niveau ; dans la boucle de recherche, on ajoute alors le critère `id_secteur` pour se limiter au secteur courant.

```
<BOUCLE_secteurs (RUBRIQUES) {racine}>

<B_recherche>
  <b>#TITRE</b>
  <ul>
<BOUCLE_recherche (ARTICLES) {recherche} {id_secteur} {par points} {inverse} {0,5}>
  <li><a href="#URL_ARTICLE">#TITRE </a>
</BOUCLE_recherche>
  </ul><hr>
</B_recherche>

</BOUCLE_secteurs>
```

On remarquera que le titre du secteur n'est affiché que si la recherche a donné des résultats pour ce secteur. D'autre part, pour chaque secteur on n'affiche que les cinq articles les mieux classés, par ordre décroissant de pertinence.

Attention cependant, comme la recherche est effectuée autant de fois qu'il y a de secteurs, le calcul risque d'être ralenti.

10.2 Afficher des éléments par lignes dans un tableau

11 avril 2002

Par exemple, on peut vouloir créer un tableau contenant les titres des articles d'une rubrique agencés sur trois colonnes, le nombre de lignes dépendant du nombre total d'articles ; sur le principe :

```
article 1  article 2  article 3
article 4  article 5  article 6
article 7  article 8  article 9
```

L'astuce consiste à jouer à la fois avec les doublons et avec les boucles récursives. On construit une première boucle qui affiche les trois premiers articles de la rubrique *une fois les doublons éliminés*. On voit qu'il suffit ensuite de réafficher cette boucle à chaque fois qu'il reste des articles pour afficher graduellement tous les articles, ceux déjà affichés venant à chaque fois grossir les rangs des doublons. Pour cela, dans le code conditionnel de cette boucle, on ajoute un appel récursif vers la boucle elle-même : elle sera affichée tant qu'elle produit des résultats.

```
<table>
<B_ligne>
  <tr>
    <BOUCLE_ligne (ARTICLES) {id_rubrique} {doublons} {par titre} {0,3}>
      <td width="33%">
        <a href="#URL_ARTICLE">#TITRE</a>
      </td>
    </BOUCLE_ligne>
  </tr>

  <BOUCLE_ligne_suite (BOUCLE_ligne)></BOUCLE_ligne_suite>
```



```
</B_ligne>
</table>
```

Le même type de boucle, en remplaçant l'appel du titre par le logo (avec la balise #LOGO_ARTICLE), permet d'afficher une galerie où chaque logo d'article donne un aperçu (dont la taille sera de préférence fixée afin d'avoir une belle mise en page), et le texte de l'article contient la ou les oeuvres exposées.

10.3 Afficher le nombre de messages du forum lié à un article

11 avril 2002

C'est un poil acrobatique.

À première vue, il est très simple de connaître le nombre d'éléments d'une boucle : il suffit d'utiliser le code SPIP : #TOTAL_BOUCLE. Ce code peut s'utiliser non seulement à l'intérieur de la boucle, mais aussi (c'est le seul dans ce cas) dans le *texte conditionnel après* (le texte qui s'affiche après la boucle si elle contient des éléments) et le *texte conditionnel alternatif* (le texte qui s'affiche si la boucle est vide).

Nous devons créer une boucle de type FORUMS, liée à un article, de façon à compter son nombre de résultats.

Première subtilité : nous voulons *tous* les messages des forums liés à l'article, en comptant les réponses aux messages, les réponses aux réponses...

Une simple boucle de type :

```
<BOUCLE_forum(FORUMS){id_article}>
```

contient uniquement les messages qui répondent à l'article. Habituellement, pour accéder aux réponses à ces messages, on inclut une seconde boucle à l'intérieur de celle-ci. Ici, nous voulons que la boucle sélectionne absolument tous les messages attachés à l'article, sans tenir compte de leur hiérarchie. Pour cela, il faut spécifier le critère «*plat*», qui comme son nom l'indique sert à afficher un forum à plat. Ce qui donne :

```
<BOUCLE_forum(FORUMS){id_article}{plat}>
```

Voyons maintenant comment compter les éléments qu'elle contient. La difficulté, ici, c'est que justement cette boucle ne doit rien afficher ! Elle n'affiche pas le titre des messages, on évitera même de lui faire afficher des espaces ou des retours à la ligne (sinon votre page HTML contiendra des dizaines de lignes vides, inélégantes) ; l'intérieur de la boucle n'affiche donc rigoureusement rien, mais on doit afficher, après la boucle, le nombre de résultats.

Une subtilité à bien comprendre : le texte conditionnel alternatif s'affiche *si la boucle n'affiche rien* ; il est donc affiché même si la boucle sélectionne des éléments (ici des messages de forum) mais qu'elle ne contient aucun affichage.

Nous devons donc placer #TOTAL_BOUCLE dans le texte conditionnel alternatif. S'il n'y a aucun message de forum attaché à l'article, #TOTAL_BOUCLE sera vide, il ne faut donc pas afficher le texte englobant («il y a N contributions au forum») dans ce cas.

```
<BOUCLE_nb_forums(FORUMS){id_article}{plat}></BOUCLE_nb_forums>
  [Il y a (#TOTAL_BOUCLE) contribution(s) au forum.]
</B_nb_forums>
```

10.4 Afficher les derniers articles de vos rédacteurs par rubrique

1er mai 2002 par Ecran de Bureau

Par défaut SPIP vous propose une page auteur qui vous permet de montrer la liste des auteurs/rédacteurs participant à votre site, ainsi que leurs dernières contributions.

Mais un problème vient à se poser quand vous avez plusieurs rédacteurs et que ceux-ci participent activement à votre site. Cela finit par être une page à rallonge.

Cependant il existe un moyen de montrer les dernières contributions de vos auteurs/redacteurs et ce pour chacun d'eux. Vous pouvez voir un exemple [ici](#)¹.

Comment procéder ?

Tout d'abord, on va créer deux fichiers : un fichier `myauteur.php3` et un fichier `myauteur.html`.

10.4.1 Création du fichier `myauteur.php3`

Dans le fichier `myauteur.php3` mettre le code suivant :

```
<?php
$fond = "myauteur";
$delais = 24*3600;

include ("inc-public.php3");

?>
```

10.4.2 Création du fichier `myauteur.html`

Dans le fichier `myauteur.php3` mettre les codes suivants :

▷ Juste après la balise `<body>`, mettre

```
<BOUCLE_principale(AUTEURS){id_auteur}{unique}>
```

▷ Juste avant la balise `</body>`, mettre

```
</BOUCLE_principale>
```

▷ Dans le corps de la page HTML, voici le code à installer (on ne peut déterminer une rubrique car par défaut l'auteur n'est pas associé à une rubrique mais à un article, le code peut paraître biscornu mais on va donc retrouver la rubrique par rapport à l'article) :

Code pour le dernier article

```
<B_appel_article>
Dernier article écrit par
<BOUCLE_nom_auteur(AUTEURS){id_auteur}>[ (#NOM) ]</BOUCLE_nom_auteur><br>
```

¹<http://zonewebmaster.membres.jexiste.org/>

```
<BOUCLE_appel_article(ARTICLES){id_auteur}>
  <BOUCLE_appel_rubrique_article(RUBRIQUES){id_rubrique}{par titre}{doublons}>
    [(#TITRE|majuscules)]
    <ul>
      <BOUCLE_rappel_article(ARTICLES){id_rubrique}{par date}{inverse}
                                                {doublons}{0,15}>
        <li><a href="#URL_ARTICLE">[(#TITRE)<br></a>]
      </BOUCLE_rappel_article>
    </ul>
  </BOUCLE_appel_rubrique_article>
</BOUCLE_appel_article>
</B_appel_article>

Cette auteur n'a pour l'instant écrit aucun article

</B_appel_article>
```

Code pour article choisi au hasard

```
<B_appel_article>

Dernier article écrit par
<BOUCLE_nom_auteur(AUTEURS){id_auteur}>[(#NOM)]</BOUCLE_nom_auteur><br>

<BOUCLE_appel_article(ARTICLES){id_auteur}>
  <BOUCLE_appel_rubrique_article(RUBRIQUES){id_rubrique}{par titre}{doublons}>
    [(#TITRE|majuscules)]
    <ul>
      <BOUCLE_rappel_article(ARTICLES){id_rubrique}{par hasard}{doublons}{0,15}>
        <li><a href="#URL_ARTICLE">[(#TITRE)<br></a>]
      </BOUCLE_rappel_article>
    </ul>
  </BOUCLE_appel_rubrique_article>
</BOUCLE_appel_article>
</B_appel_article>

Cette auteur n'a pour l'instant écrit aucun article

</B_appel_article>
```

10.4.3 Et enfin

Maintenant, il faut configurer votre page auteur (page où vous énumérez vos différents auteurs) pour que, en cliquant sur le lien auteur, celui-ci, dirigera vers la page *myauteur* ou sera inscrit les derniers articles écrits par l'auteur.

Le lien devra être écrit de la manière suivante :

```
<a href="myauteur.php3?id_auteur=#ID_AUTEUR">nom du lien</a>
```

10.5 Liens «Article précédent», «Article suivant»

1er mai 2002

Le but est d'afficher des liens vers l'article précédent, puis vers l'article suivant dans la page d'un article. Les squelettes par défaut présentent déjà la liste des articles présents dans la même rubrique, il s'agit donc ici d'obtenir une présentation plus précise.

Il n'est pas possible d'obtenir cet effet directement avec le code SPIP lui-même. Cependant, il est simple à réaliser en utilisant un peu de PHP.

Nous nous placerons à l'intérieur de la boucle principale de la page, celle qui affiche l'article «principal». Pour simplifier l'explication, nous résumerons le squelette de notre page d'article au code suivant :

```
<html>
<body>

<BOUCLE_principale(ARTICLES){id_article}>

  <h1>#TITRE</h1>
  #TEXTE

</BOUCLE_principale>

</body>
</html>
```

Nous allons avoir besoin, dans le code PHP qui affiche et sélectionne les articles précédents et suivants, de mémoriser le numéro de l'article principal. Pour cela, ajoutons simplement la définition de la variable PHP `$ze_article` à l'intérieur de la `BOUCLE_principale` :

```
<html>
<body>

<BOUCLE_principale(ARTICLES){id_article}>
  <?
    $ze_article = "#ID_ARTICLE";
  ?>

  <h1>#TITRE</h1>
  #TEXTE

</BOUCLE_principale>

</body>
</html>
```

Nous plaçons l'`#ID_ARTICLE` entre guillemets, de façon à éviter une erreur PHP lorsqu'on charge la page sans numéro de rubrique.

Il suffit maintenant d'insérer les boucles affichant l'article précédent et l'article suivant :

```
<html>
<body>
```

```

<BOUCLE_principale(ARTICLES){id_article}>
  <?
    $ze_article = "#ID_ARTICLE";
  ?>

  <BOUCLE_art_prec(ARTICLES){id_rubrique}{par titre}>
    <?
      if ($ze_article == #ID_ARTICLE) {
        echo stripslashes("$art_prec");
      }
      $art_prec = "<div>Article précédent : <a href=' #URL_ARTICLE'>
                  [ (#TITRE|addslashes) ]</a></div>";
    ?>
  </BOUCLE_art_prec>

  <h1>#TITRE</h1>
  #TEXTE

  <BOUCLE_art_suiv(ARTICLES){id_rubrique}{par titre}>
    <?
      $texte_suiv = "<div align='right'>Article suivant : <a href=' #URL_ARTICLE'>
                   [ (#TITRE|addslashes) ]</a></div>";
      if ($ze_article == $article_prec) {
        echo stripslashes("$texte_suiv");
      }
      $article_prec = #ID_ARTICLE;
    ?>
  </BOUCLE_art_suiv>

</BOUCLE_principale>

</body>
</html>

```

Les boucles «art_prec» et «art_suiv» s'effectuent autant de fois qu'il y a d'articles dans la même rubrique que notre article principal. À chaque occurrence, elles mémorisent le texte et le lien de l'article, sans l'afficher ; ce lien n'est affiché que lorsque l'identifiant de l'article de l'occurrence correspond à l'identifiant de notre article principal.

Pour l'article précédent, c'est très simple :

- ▷ on teste le numéro de l'article, et on affiche un texte éventuel ;
- ▷ puis on mémorise le texte du lien ; ainsi, dans le test, c'est bien une valeur récupérée dans l'occurrence précédente (donc l'article précédent) qui est affichée.

Pour l'article suivant, c'est à peine plus compliqué :

- ▷ on mémorise le lien vers l'article de l'occurrence ;
- ▷ on teste si le numéro de l'occurrence *précédente* correspond à l'article principal ;
- ▷ on mémorise le numéro de l'article en tant que \$article_prec, pour le test effectué lors de l'occurrence suivante.

Notez au passage qu'on mémorise une variable construite sur [(#TITRE|addslashes)]. En effet, sans cela, on obtiendrait un message d'erreur PHP pour les articles contenant des guillemets. Lorsqu'on affiche ce texte, on le passe donc par stripslashes pour supprimer les caractères indésirables.

Important : pour que votre navigation reste cohérente sur l'ensemble du site, il convient de bien choisir le critère de classement des boucles BOUCLE_art_prec et BOUCLE_art_suiv. Ici, nous avons indiqué {par titre}, car dans notre site nous affichons les listes d'articles classés selon leurs titres. Si vous préférez classer les articles selon leur

date, du plus récent au plus ancien, il faut modifier ces boucles avec les critères : `{par date}{inverse}`. (Pour être cohérent, il convient d'adopter les mêmes critères de classement que la boucle «Dans la même rubrique», ou que la boucle d'affichage des articles dans le squelette des rubriques.)

10.6 Ne pas afficher les articles publiés depuis plus d'un an

11 avril 2002

Cela s'effectue avec le critère «age», qui est l'âge de l'article (calculé depuis sa date de mise en ligne dans l'espace public) en nombre de jours.

Ainsi pour conserver tous les articles de moins d'un an dans la rubrique courante. Le critère de sélection qui nous intéresse ici est : «age».

```
<B_articles_recents>
  <ul>
<BOUCLE_articles_recents (ARTICLES) {id_rubrique}{age < 365}{par titre}>
  <li>#TITRE</li>
</BOUCLE_articles_recents>
  </ul>
</B_articles_recents>
```

Pour prendre en compte l'âge vis-à-vis de la date de première publication au lieu de la date de mise en ligne, il faut utiliser le critère «age_redac» au lieu de «age». L'âge est indiqué en nombre de jours.

Notons que cette manipulation est possible avec tous les types de données auxquels est associée une date (brèves, forums...).

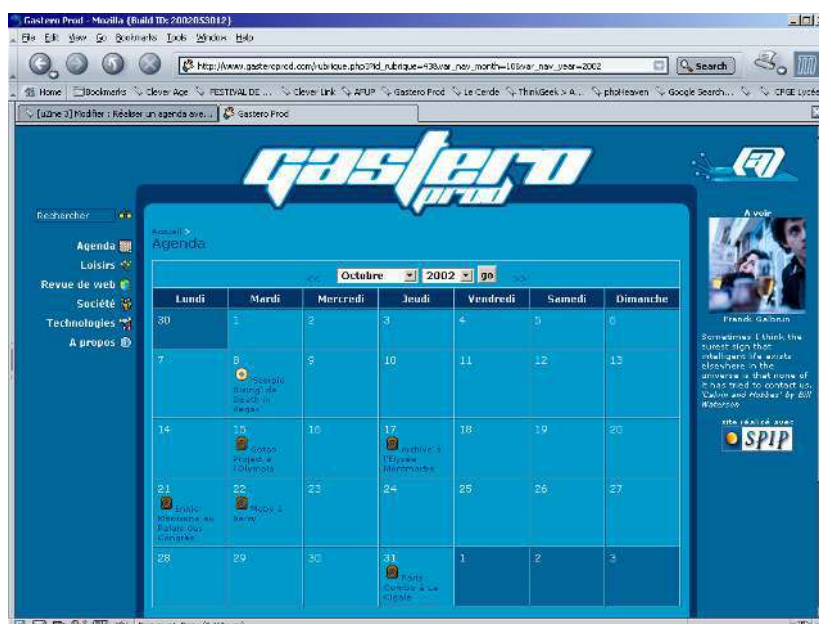
10.7 Réaliser un agenda avec SPIP

16 novembre 2002 par Nicolas "Brush" Hoizey

Un joli squelette avec du PHP dedans ...

Il peut être tentant d'utiliser SPIP pour présenter un agenda d'événements sous forme de calendrier. Plutôt que devoir intégrer un logiciel externe spécialisé, et en attendant que SPIP intègre un nouveau type de données, voyons comment réaliser cela avec un squelette mêlant boucles SPIP et code PHP.

[SPIP 1.4.2] Dans cet article nous allons utiliser SPIP pour présenter un agenda d'événements sous forme de calendrier. Plutôt que d'intégrer un logiciel externe spécialisé, voyons comment réaliser cela avec un squelette mêlant boucles SPIP classiques, et code PHP.



Exemple d'agenda
L'agenda de Gastero Prod²

10.7.1 De quelles informations avons-nous besoin ?

La première étape consiste à déterminer comment les informations seront stockées dans SPIP, et donc saisies par les rédacteurs.

L'objectif principal étant qu'un rédacteur puisse définir une date à laquelle un événement donné doit avoir lieu, nous avons besoin des éléments suivants :

- ▷ au moins un champ de type texte
- ▷ un champ de type date

Pour le champ texte, c'est simple, et nous allons même pouvoir nous régaler. SPIP nous en propose au moins 3 très utiles et systématiquement présents quel que soit le paramétrage du site : le titre, le descriptif et le texte.

10.7.2 La date de publication antérieure ou la vraie date de publication ?

La date de publication d'un article est déterminée automatiquement lors de sa validation, ce qui signifie que l'auteur ne peut pas la préciser à l'avance. Elle peut en revanche être modifiée par le valideur, mais elle n'est donc pas vraiment utile pour indiquer une date d'événement - surtout si on a configuré son site pour que SPIP ne publie pas les articles post-datés.

Heureusement, SPIP propose aussi une *date de publication antérieure*, normalement utilisée pour indiquer à quelle date a été publié un article repris ultérieurement dans SPIP, et qui peut être définie librement par le rédacteur³.

C'est cette date que nous allons utiliser pour définir les événements de l'agenda.

²http://www.gasteroprod.com/rubrique.php3?id_rubrique=43

³Vérifiez que vous avez configuré votre site pour gérer ce type de date supplémentaire.

Utilisation de la date de publication antérieure

10.7.3 Organiser les événements

Si l'agenda doit contenir beaucoup d'informations de natures diverses, il peut s'avérer utile de les qualifier (concert, film, réunion ?) pour les mettre en évidence et éventuellement ensuite les afficher de manière sélective.

Dans SPIP deux méthodes de classement viennent immédiatement à l'esprit :

- ▷ classement par rubrique (et éventuellement sous-rubriques) ;
- ▷ classement par mot-clé.

Dans cet exemple, nous allons utiliser les rubriques, ce qui nous permettra de constituer toute une arborescence pour stocker les articles événement ; cela va nous permettre de filtrer les types d'événements avec une notion de granularité progressive (c'est pas chic ça ?).

Dans l'agenda de Gastero Prod⁴, le rubricage n'est pas très développé, mais il est déjà possible de sélectionner uniquement les événements musicaux⁵, et éventuellement de filtrer encore plus en ne considérant que les concerts⁶.



Une hiérarchie de rubriques pour les thèmes

Voilà, les événements sont créés dans une arborescence de rubriques thématiques, sont positionnés à une date, passons aux choses sérieuses :

⁴http://www.gasteroprod.com/rubrique.php3?id_rubrique=43

⁵http://www.gasteroprod.com/rubrique.php3?id_rubrique=46

⁶http://www.gasteroprod.com/rubrique.php3?id_rubrique=44

10.7.4 L'affichage de l'agenda, PHP à la rescousse

Nous souhaitons afficher l'agenda sous la forme standard d'un tableau mensuel dont chaque colonne représente un jour de la semaine. Nous souhaitons aussi pouvoir naviguer d'un mois à l'autre, voire même choisir directement un mois arbitraire.

Tout cela n'est pas possible directement si l'on utilise exclusivement les boucles de SPIP, d'où l'usage de PHP.

Voici une explication, pas à pas, de tous les éléments du squelette (vous trouverez en fin d'article le squelette complet à télécharger) :

Préparation des informations à traiter

Définissons déjà deux tableaux contenant les noms des 12 mois de l'année et des 7 jours de la semaine.

```
$months = array('', 'Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin', 'Juillet', 'Août',  
                'Septembre', 'Octobre', 'Novembre', 'Décembre');  
$days = array('Dimanche', 'Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi', 'Samedi');
```

Voici maintenant une petite fonction toute bête permettant de simplifier la génération d'une date au format *timestamp* pour les plus fainéants d'entre nous qui ne veulent pas avoir besoin de préciser que l'heure, les minutes et les secondes sont nuls ... ;)

```
function mkdate($month, $day, $year)  
{  
    return mktime(0, 0, 0, $month, $day, $year);  
}
```

Gestion de la navigation

Les paramètres de navigation sont passés dans l'URL au travers des variables `var_nav_month` et `var_nav_year`⁷. La première chose à faire en début de script est donc de déterminer quel mois il faut afficher, le mois courant étant pris par défaut si aucun n'est précisé :

```
if (isset($GLOBALS['var_nav_month'])) {  
    $cal_day = mkdate($GLOBALS['var_nav_month'], 1, $GLOBALS['var_nav_year']);  
} else {  
    $cal_day = time();  
}  
  
$D = intval(date('d', $cal_day));  
$M = intval(date('m', $cal_day));  
$Y = intval(date('Y', $cal_day));
```

⁷On utilise la notation `var_xxx` pour indiquer à SPIP de ne pas prendre en compte ces variables pour le calcul du cache de la page, ce qui permet à la fois de limiter l'impact sur le cache (un seul fichier cache traitera toutes les pages de l'agenda) et d'accélérer l'affichage (puisque SPIP n'aura besoin d'aller chercher les données dans la base qu'une seule fois pour toutes ces pages).

Affichage du reste du calendrier

```

$TempD = 1;
while((date('m', mkdate($M, $TempD, $Y)) == $M) || (date('w', mkdate($M, $TempD, $Y)) != 1)) {
  if(date('w', mkdate($M, $TempD, $Y)) == 1) {
    echo '</tr><tr>';
  }
  echo '<td width="14%" height="50" valign="top" class="calendar_' .
    (date('m', mkdate($M, $TempD, $Y)) != $M ? 'not_' : '').'this_' .
    (date('Ymd', mkdate($M, $TempD, $Y)) == date('Ymd') ? 'day' : 'month').' ">';
  echo date('j', mkdate($M, $TempD, $Y));
  $date = date('Ymd', mkdate($M, $TempD, $Y));
  if (isset($events[$date])) {
    while (list(, $event) = each($events[$date])) {
      echo '<br />'. $event['logo'] . '<a href="' . $event['link'] . '" class="small">' .
        $event['title'] . '</a>';
    }
  }
  echo '</td>';
  $TempD++;
}
?>
</tr>
</table>

```

Affichage de l'arborescence thématique de l'agenda

```

<?php
function depth($depth, $type)
{
  for($i = 0; $i < $depth; $i++) {
    echo '';
  }
}

$depth = 1;
?>

<p class="big">Accès direct aux thèmes, et légende des icônes :</p>
<BOUCLE_sous_rubriques1(RUBRIQUES){id_parent=43}{par titre}>
  <p>
    #LOGO_RUBRIQUE
    <a href="#URL_RUBRIQUE">#TITRE</a>
    <B_sous_rubriques>
      <BOUCLE_sous_rubriques(RUBRIQUES){id_parent}{par titre}>
        <br />
        <?php depth($depth, 'rub'); ?>
        #LOGO_RUBRIQUE
        <a href="#URL_RUBRIQUE">#TITRE</a>
        <B_sous_rubriques_2>
          <?php $depth++; ?>
          <BOUCLE_sous_rubriques_2(BOUCLE_sous_rubriques)>
            </BOUCLE_sous_rubriques_2>
          <?php $depth--; ?>
        </B_sous_rubriques_2>

```

```
</BOUCLE_sous_rubriques>
</B_sous_rubriques>
</p>
</BOUCLE_sous_rubriques1>
```

Et bien sûr, fermeture finale de la boucle principale :

```
</BOUCLE_principale>
```

10.7.5 Un peu de personnalisation visuelle

Comme vous pouvez le voir dans le code proposé, la présentation est personnalisable via quelques styles.

Voici l'extrait de feuille de style de Gastero Prod correspondant à l'agenda :

```
.calendar_head
{
  background-color: #003366;
}

.calendar_this_day
{
  background-color: #00bbee;
}

.calendar_this_month
{
  background-color: #0099cc;
}

.calendar_not_this_month
{
  background: #006699;
}
```

10.7.6 Le code complet du squelette

Et voilà donc le code complet du squelette, à recopier et diffuser sans retenue !



agenda.html

Le fichier HTML du squelette⁸
(Zip, 1.6 ko)

⁸<http://www.uzine.net/IMG/zip/doc-710.zip>

10.8 Remplir les meta-tags HTML des pages d'article

11 avril 2002

Le but de cet exemple est d'installer dans les méta-tags de notre page, la liste des mots-clés associés à l'article ainsi que le nom des auteurs.

Si l'on veut optimiser le référencement du site par les moteurs de recherche, on peut par exemple mentionner le descriptif de l'article, les mots-clés associés, ainsi que le nom du ou des auteurs.

```
<head>
<BOUCLE_head(ARTICLES) {id_article}>
<title>#TITRE</title>
<meta name="Description" content="#DESCRIPTIF">
<meta name="Keywords" content="<BOUCLE_keywords(MOTS) {id_article}{", ">
                                     #TITRE </BOUCLE_keywords>">
<meta name="Author" content="<BOUCLE_author(AUTEURS) {id_article}{", ">
                                     #NOM </BOUCLE_author>">
</BOUCLE_head>
</head>
```

On remarquera que pour les mots-clés et l'auteur, on utilise une boucle imbriquée pour aller chercher ces informations à partir de l'`id_article` courant. De plus, on spécifie une virgule comme séparateur afin que le contenu du meta-tag soit compréhensible (y compris par un moteur de recherche).

Attention ! le code donné ci-dessus à titre d'exemple est un peu «naïf» : si le `#NOM` d'un auteur ou le `#DESCRIPTIF` d'un article peuvent contenir des tags html (mise en italiques, saut de paragraphe...) la page qui en résultera sera en effet pleine d'erreurs. Pour éviter cela, il faut penser à passer un filtre comme `|supprimer_tags` sur le champ en question :

Remplacer

```
#DESCRIPTIF
```

par

```
[ (#DESCRIPTIF|supprimer_tags) ]
```

...

10.9 Un menu déroulant pour présenter une liste d'articles

11 avril 2002

On souhaite réaliser un menu déroulant en utilisant les commandes HTML adaptées à la création de formulaire ; de plus on veut que ce menu serve à aller à l'URL de l'article sélectionné. Si l'URL des articles est du type `article.php3?id_article=123`, le bout de code suivant conviendra :

```
<FORM ACTION="article.php3" METHOD="get">
  <SELECT NAME="id_article">
    <BOUCLE_menu_articles(ARTICLES) {id_rubrique} {par titre}>
```

```
<OPTION VALUE="#ID_ARTICLE">#TITRE</OPTION>
</BOUCLE_menu_articles>
</SELECT>

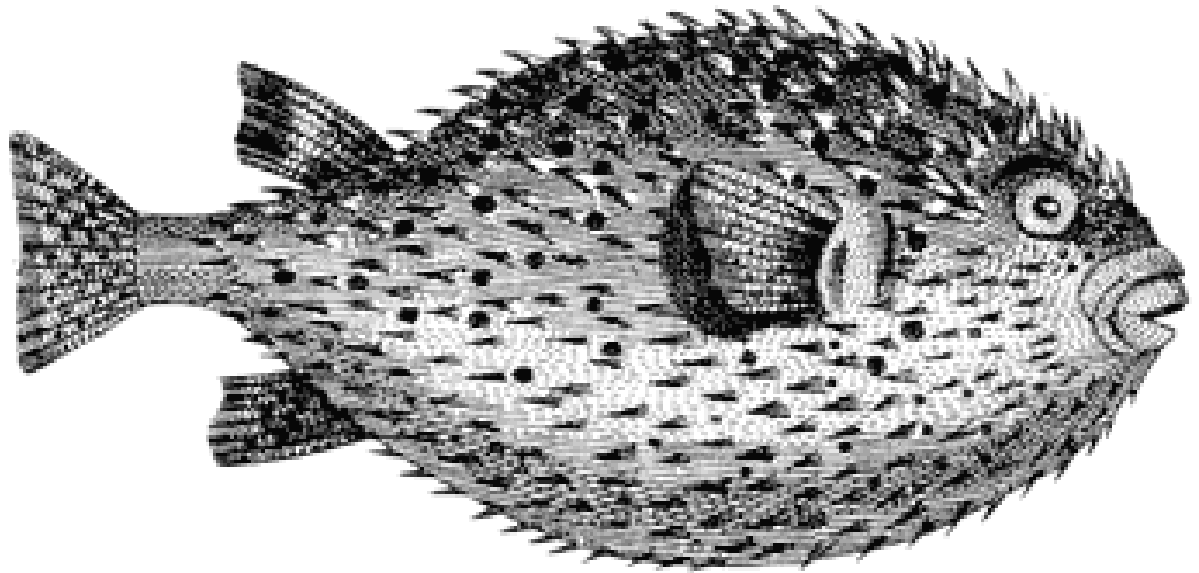
<INPUT TYPE="submit" NAME="Valider" VALUE="Afficher l'article">
</FORM>
```

Les critères de la boucle articles (ici : les articles de la rubrique courante, triés par titre) seront modifiés selon vos besoins. Ce type de construction marche bien sûr aussi pour les brèves, rubriques. . .

Selon le même principe, il est tout aussi facile de présenter une liste de rubriques, de brèves. . .ou même l'intégralité de la structure du site.

Chapitre 11

Tutorial : utilisation avancée des boucles et des mots-clés



Tutorial : utilisation avancée
des boucles et des mots-clés

Ce *tutorial* vous explique diverses méthodes destinées à dépasser les limites apparentes de SPIP. Attention : il s'adresse à des utilisateurs déjà expérimentés.

11.1 Introduction

23 septembre 2001 par l'équipe de SPIP

SPIP est un produit limité. Il y a des choses qu'il fait très bien, d'autres qu'il ne peut pas réaliser. Essayez par exemple de créer un site consacré au cinéma, avec des liens sur tous les noms (acteurs, réalisateur, équipe technique, à la façon des fiches de l'Internet Movie Database¹) vers d'autres films qu'ils ont réalisé, et vous réaliserez que SPIP n'a vraiment pas été conçu pour cela ! À l'inverse, créer et gérer un site de contenu éditorial à la structure simple, à la façon d'uZine, est très simple.

Cependant, entre ces deux extrêmes (des sites irréalisables avec SPIP aux sites pour lesquels SPIP est particulièrement adapté), il existe une multitude d'usages, de besoins, facilement réalisables, mais à priori inaccessibles avec la version standard des squelettes.

Certains webmestres (fort courageux), confrontés aux limites apparentes du produit, se lancent bille en tête dans le code source de SPIP dans le but de l'adapter à leurs besoins spécifiques. Si cette attitude très «*open source*» est louable, en revanche elle prive ces webmestres de la possibilité de suivre les évolutions du logiciel standard, et requiert des connaissances poussées en PHP.

Or, et c'est le but de ce tutorial, il existe de nombreuses possibilités pour dépasser les limites de SPIP, uniquement par une utilisation personnalisée des squelettes et de la structure du site.

11.1.1 À qui s'adresse ce document ?

Le présent tutorial est destiné aux webmestres qui désirent dépasser certaines limites apparentes de SPIP. Il est donc **impératif d'avoir déjà compris le fonctionnement des squelettes et des boucles** qui gèrent l'interface publique.

Si vous débutez, commencez avec le document *SPIP pas à pas* (chapitre 9 page 143), destiné aux webmestres qui s'initient au fonctionnement des squelettes.

Si vous savez déjà bien utiliser les squelettes, nous vous conseillons d'avoir à portée de la main une version imprimée du *Manuel de référence* (chapitre 8 page 97).

11.1.2 Quelle version de SPIP ?

Les exemples donnés ici utilisent des fonctionnalités présentes à partir de la version **SPIP 1.3**. Certaines peuvent être réalisées avec des versions précédentes, mais l'utilisation de la version 1.3 offre plus de souplesse.

11.1.3 Quelles autres connaissances techniques sont nécessaires ?

Ce tutorial est très progressif. Cependant, il présente la construction pas à pas de squelettes complets. Outre la compréhension du mécanisme général des squelettes (et du système de boucles), il convient de comprendre le code HTML. Le HTML utilisé ici sera volontairement rudimentaire mais, si vous n'avez jamais réalisé une page Web autrement qu'avec un logiciel Wysiwyg, vous risquez de souffrir.

Nous n'aborderons pas ici l'utilisation de PHP dans les squelettes ; la connaissance de ce langage est donc ici inutile.

¹<http://www.imdb.com>

11.1.4 Les impératifs

- ▷ Ne pas toucher à structure de SPIP lui-même. À aucun moment nous n'irons modifier le code source du produit, ni les tables de la base de données utilisées par SPIP. Cela garantira que les fonctionnalités apportées ici resteront compatibles avec les futures évolutions du logiciel, et que l'interface privée conservera sa cohérence.
- ▷ Réaliser un site dont l'interface de navigation reste cohérente. Un des buts ici étant de réaliser une navigation plus riche que celle proposée par les squelettes standards, il ne faut pas à l'inverse que l'interface devienne totalement incompréhensible pour le visiteur. (Vous verrez cependant que nous n'avons pas ici développé le graphisme de nos pages ; graphiquement, l'interface obtenue à la fin de ce tutorial sera hideuse ... mais en revanche, les liens entre les différents éléments structurels du site seront présents.)
- ▷ Réaliser un site dont les mises à jour restent simples. Ceux qui gèrent le site depuis l'espace privé ne doivent pas passer plusieurs heures pour ajouter un nouvel article ... Il s'agit bien de profiter des automatismes liés à un site dynamique.

11.1.5 Comment utiliser ce tutorial ?

Ce tutorial est très progressif, et les codes fournis sont complets : vous n'y trouverez pas des «bouts de code» sortis de nulle part à recopier, au contraire le code complet des squelettes se construira au fur et à mesure des articles. La méthode retenue ici est la création de pages de squelette de plus en plus complexes, chaque étape étant expliquée.

Nous vous conseillons donc de suivre ce tutorial dans l'ordre de ses articles, et de créer les fichiers en même temps que les explications, en intégrant à chaque fois les variations données en exemple.

Vous trouverez ces exemples beaucoup plus clairs si vous réalisez vous-même les pages présentées ici. Le code vous semblera plus simple, puisqu'à chaque nouvel élément intégré, vous verrez immédiatement l'impact des modifications.

Tous les squelettes réalisés dans ce tutorial sont regroupés dans nos archives². Vous pouvez les télécharger dès à présent, mais nous vous conseillons néanmoins de réaliser ces squelettes *ab initio* en suivant ce tutorial : si vous travaillez directement à partir des squelettes terminés, vous perdez la construction progressive et vous risquez de ne pas comprendre leur structure.

11.1.6 La progression de ce tutorial

- ▷ Les premiers articles (de «Le but du jeu» à «Écrire des articles») insistent sur l'importance bien penser la structure de son site et de définir la constitution des articles avant de démarrer un site complexe. Techniquement, cette partie est à la portée de tous les utilisateurs de SPIP (on peut donc la lire même si on n'a aucune connaissance du fonctionnement des squelettes).
Cette partie vous semblera peut-être un peu trop simple (il n'y a aucune astuce technique). Elle introduit cependant l'utilisation des mots-clés comme outil de structuration du site. Surtout, elle permet de comprendre que la structure du site est primordiale pendant le démarrage d'un projet de site ambitieux. Cette étape, lorsqu'elle est négligée (et elle l'est souvent), conduit à des sites qui deviennent ingérables et dont l'interface publique est incohérente (ces problèmes se posent rarement lorsque l'on démarre le site, avec une poignée d'articles). Enfin, nous y insistons (longuement ...) sur le fait que les choix techniques (programmation des squelettes, utilisation des mots-clés) dépendent directement des choix éditoriaux (cette évidence étant primordiale avec un système de publication).
- ▷ Les articles suivants (de «Première version du squelette des articles» à «Le site complet») proposeront la création pas à pas des squelettes. Il faut à partir de ce moment avoir une bonne connaissance du système de boucles de SPIP. Les difficultés à ce stade seront d'ordre logique, et non informatiques. Volontairement, nous fabriquerons les squelettes en plusieurs étapes successives : au départ des squelettes extrêmement simples, nous y reviendrons, en ajoutant plus de complexité dans la structure des boucles.

²http://rezo.net/spip-dev/SQUELETTES/tutorial_avance/

Vous trouverez dans cette partie quelques astuces logiques dans la gestion des boucles ; surtout, vous y verrez l'utilisation des mots-clés comme éléments de structuration de la navigation.

- ▷ Nous obtiendrons à la fin de ce tutorial un site complet et exploitable (moyennant la création d'une interface graphique un peu plus élaborée). Cependant, le dernier article («Le site complet»), vous fournira quelques idées de développements supplémentaires que vous pourrez ajouter à votre site, en exploitant les principes expliqués ici.

11.2 Le but du jeu : un site consacré aux jeux vidéo

23 septembre 2001 par l'équipe de SPIP

Les explications qui vont suivre sont basées sur la réalisation d'un site consacré aux jeux vidéo.

11.2.1 Pourquoi les jeux vidéo ?

Une notion à ne jamais perdre de vue lorsqu'on réalise des squelettes avec SPIP, c'est qu'il s'agit de présenter un contenu éditorial (penser en terme de possibilités techniques ou de base de données est le meilleur moyen de créer un site à l'interface incompréhensible). Il faut donc, en permanence, utiliser un vocabulaire et des images directement liées au contenu que l'on veut présenter.

Nous devons donc, pour que ce tutorial soit lisible, nous fixer un objectif *éditorial*, afin de pouvoir présenter chaque bidouille technique comme une réponse à un besoin éditorial.

Un site de jeux vidéo offre, pour notre démonstration, plusieurs avantages :

- ▷ le vocabulaire associé aux contenus de ce type de site est connu de tous,
- ▷ la navigation sur ces sites est très souple (il existe de nombreuses façons différentes de naviguer sur de tels sites) ;
- ▷ c'est l'exemple-type du site impossible à réaliser avec les squelettes standards de SPIP (sauf à se contenter d'un site très en deça de ce que l'on attend de ce genre de contenu) ;
- ▷ en revanche, c'est un très bon exemple de ce qu'il est possible de faire pour dépasser les limites de SPIP, tout en conservant la cohérence et la simplicité de son utilisation.

11.2.2 Quelles sont difficultés que présente un tel site ?

La principale difficulté concerne le rubriquage des articles :

- ▷ chaque jeu fait l'objet de plusieurs «articles» : un ou plusieurs tests (essayer le produit disponible dans le commerce), une *preview* (décrire un jeu qui n'est pas encore disponible dans le commerce, mais dont on a une version *beta*), des *news* (généralement, avant la sortie, il s'agit de répercuter quelques informations sur le futur jeu), des trucs et astuces (ou *tips*), une solution complète des énigmes du jeu ;
- ▷ certains jeux sont disponibles sur différentes machines (PC, Dreamcast, Playstation . . .) ; certains tests, previews, solutions concernent une seule version du jeu, parfois plusieurs (et cela n'est pas systématique même pour un même jeu : la solution d'un jeu est souvent commune à plusieurs versions, alors que les test doivent être faits pour chaque version) ;
- ▷ chaque jeu appartient à une catégorie (jeu d'action/aventure, jeu de réflexion, jeu de plateforme, jeu de simulation sportive, jeu de course, jeu de baston . . .). Avec SPIP, il n'est possible de créer qu'une seule structure hiérarchique des rubriques. De manière simple, il faudrait choisir une unique structure :
- ▷ une structure sur les types d'articles (une rubrique pour les tests, une rubrique pour les previews, une solution pour les solutions . . .) ;
- ▷ **ou** une structure selon les machines (une rubrique pour PC, une rubrique pour Playstation, une rubrique pour Dreamcast . . .) ;

▷ ou une structure selon la catégorie de jeu (une rubrique pour l'aventure, une pour la simulation sportive, une pour la baston ...).

Dans tous les cas, on voit bien qu'on perd la richesse de navigation que doit offrir un tel site, qui devrait permettre de passer à un autre article concernant un même jeu, mais aussi de naviguer selon une machine, ou de consulter d'autres jeux de la même catégorie ...

Une seconde difficulté tient dans les **informations systématiques** que l'on voudrait inclure pour chaque jeu, pour pouvoir les présenter comme des informations «à part» :

- ▷ la date de sortie du jeu ;
- ▷ une note d'appréciation.

On pourrait évidemment décider d'écrire «à la main» ces informations à l'intérieur des articles, mais alors on ne pourrait plus les présenter sur des pages spécifiques («les prochaines sorties», «les jeux sortis récemment», «les meilleurs jeux dans cette catégorie ...»).



Cette reproduction du sommaire que nous obtiendrons donne une idée de la diversité des informations présentées sur le site, et les possibilités de navigation :

- ▷ au centre, les tests ; pour chaque jeu testé, on indique la note attribuée, et la machine (Dreamcast, Playstation ...)
- ▷ en haut, on peut afficher des sommaires consacrés à chaque machine ;
- ▷ en haut à gauche, on pourra voir les prochaines sorties de jeux, ainsi que la liste des jeux ayant reçu les meilleures notes ;
- ▷ à gauche apparaît la navigation dans le site selon les catégories de jeux (survival-horror, simulation sportive ...)
- ▷ à droite, les previews, les soluces, les astuces, les news ...

11.3 La structure du site

23 septembre 2001 par l'équipe de SPIP

Si la souplesse de SPIP dans la gestion de la structures des rubriques autorise, généralement, à commencer à rédiger ses articles sans trop se soucier du rubriquage (il est toujours facile, ensuite, de déplacer des articles ou des rubriques), dans le cas qui nous intéresse, il faut au contraire commencer par définir la structure retenue. C'est, en effet, elle qui conditionne les possibilités d'un tel site.

Avec SPIP, il n'est possible de définir qu'une seule et unique structure hiérarchique des rubriques. Il va donc falloir décider comment nous allons créer les rubriques de notre site.

11.3.1 Les possibilités

Comme nous l'avons noté dans l'article précédent (section 11.2 page 173), chaque article dépend ici de plusieurs critères. On peut donc choisir la structure des rubriques parmi les structures suivantes :

- ▷ une structure par machines : une rubrique pour Dreamcast, une rubrique pour Playstation, une rubrique pour PC ...
- ▷ une structure par jeu : une rubrique pour *Resident Evil*, une rubrique pour *Gran Turismo*, une rubrique pour *Mortal Combat* ...
- ▷ une structure par catégorie d'articles : une rubrique pour les tests, une rubrique pour les news, une rubrique pour les solutions ...
- ▷ une structure par genre de jeu : une rubrique pour les jeux de sport, une pour les jeux de plateforme, une pour les jeux de baston ...

Si l'on était un peu tordu, on pourrait également adopter les structures suivantes :

- ▷ une structure par qualité des jeux (les meilleurs jeux dans une rubrique, les jeux moyens dans une autre ...);
- ▷ une structure par date de rédaction, adaptée notamment à la transposition d'un mensuel papier sur le Web (les articles du numéro de novembre 2001, une rubrique pour décembre 2001, une rubrique pour janvier 2002, etc.).

11.3.2 Les impératifs

Certaines caractéristiques de notre site permettent de choisir une structure plutôt qu'une autre.

- ▷ Il est naturel, lorsqu'on navigue sur un site, de pouvoir passer directement du test d'un jeu à sa solution, de la preview aux news ... Le visiteur, intéressé par la preview, va vouloir connaître les dernières informations sur le développement du produit (les news). Un joueur, après avoir lu l'opinion exprimée dans un test, voudra trouver rapidement des indications pour terminer son jeu (les solutions, les astuces).
Le plus simple, ici, consiste donc à réunir tous les articles d'un même jeu dans une même rubrique. La rubrique d'un jeu contiendra à la fois la preview, les tests, la solution ...
Ainsi, la plus petite rubrique du site sera la rubrique consacrée à tous les articles d'un seul jeu.
- ▷ On pourrait décider de créer des grandes rubriques par machine, et d'y placer, en tant que sous-rubriques, chacun des jeux avec leurs différents articles (un jeu existant sur plusieurs machines aurait une rubrique spécifique à chacun des machines). Cependant, il est fréquent que certains articles sur un même jeu soient communs à plusieurs machines. Si le test de la version Dreamcast de *Alone in the dark* est différent de celui de la version Playstation, en revanche la solution des énigmes est commune aux deux supports. Avec un rubriquage par machines, on aurait naturellement le test Dreamcast dans la rubrique Dreamcast, et le test Playstation dans la rubrique *ad hoc*; mais la solution, commune aux deux versions, devrait alors est publiée dans les deux rubriques; très inélégant et fastidieux.
- ▷ Les genres de jeux présentent une structure hiérarchique : une course de rallye est une course de voitures, qui est elle-même une simulation sportive; un «survival horror» est un jeu d'aventure/action; un jeu de baston en 2D est un jeu de baston ... Ce type de structure est donc particulièrement adapté à SPIP, car on peut imbriquer des sous-rubriques dans des rubriques autant que nécessaire. Une grande rubrique «Sport» contiendra plusieurs sous-rubriques (Course, Football, Glisse, Tennis ...), chacune de ces sous-rubriques pouvant à son tour contenir d'autres sous-rubriques.
- ▷ À l'inverse, les autres structures ne présentent pas réellement de structure hiérarchique. Les machines ne «dépendent» pas les unes des autres : on pourrait évidemment réunir les consoles de jeu à l'intérieur d'une grande rubrique, et créer une autre grande rubrique pour les PC, mais cela n'offre pas grand intérêt, et n'offrirait pas une grande profondeur hiérarchique. De la même façon, une catégorie d'article ne dépend pas d'une autre (les news ne dépendent pas des solutions qui ne dépendent pas des tests ...); un rubriquage par catégorie d'article ne permettrait donc pas une réelle structure de rubriques/sous-rubriques.

11.3.3 La structure retenue

En réalité, n'importe laquelle des possibilités évoquées précédemment est possible, car les techniques que nous aborderons plus loin permettraient de s'adapter à ces différentes situations. Cependant, il est plus agréable de choisir une structure plus directement adaptée à la logique de SPIP ; dit autrement : un site qui serait déjà très présentable avec une telle structure, même sans utiliser les astuces que nous exposerons plus loin (c'est-à-dire un site que l'on peut déjà visiter avec les squelettes standards). Ainsi, l'utilisation de l'espace privé et la gestion du site resteront cohérents et simples.

Nous allons donc privilégier la structure la plus hiérarchique : la structure par genres de jeux. Nous créerons de grandes rubriques thématiques (Action/aventure, Combats, Jeux pédagogiques, Plateforme, Sport, Stratégie ...), chacune de ces grandes rubriques ayant plusieurs sous-rubriques.

Dans chacune des sous-rubriques, nous créerons enfin une sous-rubrique pour chaque jeu, regroupant tous les types d'articles du même jeu sur toutes les plateformes de ce jeu (par exemple, tous les articles concernant *Alone in the Dark 4*, que ces articles soient des news, des previews, des tests, des solutions ... qu'ils concernent la version PC/Windows, Dreamcast ou Playstation ... seront regroupés dans l'unique rubrique consacrée à ce jeu).

Logiquement, les test de la version Dreamcast de *Alone in the Dark* se trouvera dans la rubrique *Alone in the Dark*, elle-même installée dans une rubrique «Survival horror», elle-même dépendant de la rubrique «Action/Aventure».

Nous verrons plus loin que, si cette structure s'utilise facilement avec SPIP (et qu'un tel site est déjà facile à visiter avec les squelettes standards), en revanche cela pose quelques difficultés de logique de présentation, puisque l'on mélange des rubriques dont le nom est un jeu avec des rubriques qui sont des grandes catégories de jeux.

11.3.4 Que deviennent les machines et les genres d'articles ?

Nous avons donc perdu, avec cette structure, la possibilité d'indiquer dans la navigation, pour chaque article, s'il agit d'un test, d'une solution, de news ... ou encore s'il concerne une version PC, Dreamcast, Playstation ...

Est-ce qu'il ne serait pas pratique, ici, d'avoir pour chaque article un menu déroulant proposant une liste des machines, et un menu déroulant avec la liste des types d'articles ? On sélectionnerait dans ces menus déroulants et, d'un clic, on pourrait indiquer qu'il s'agit d'un *test*, et que cela concerne la Dreamcast. On pourrait même indiquer que cela concerne la Dreamcast et la Playstation pour un même article.

(Vous n'imaginez pas le nombre de webmasters qui, à ce stade, se lancent à corps perdu dans le code source de SPIP pour ajouter les menus déroulants dont il a besoin !)

Ca tombe bien : ce menu déroulant existe déjà. C'est celui des mots-clés.

Les mots-clés, associés aux articles, permettent de créer des navigations *transversales* au sein de la structure hiérarchisée d'un site sous SPIP, c'est-à-dire de passer directement d'un article dans une rubrique à un article situé dans une autre rubrique (cela quelle que soit la position de l'autre rubrique dans la hiérarchie).

Certes, l'usage immédiat des mots-clés consiste à créer des mots-clés purement thématiques ; mais si l'on considère qu'il s'agit d'un moyen de navigation transversale dans le rubriquage, on peut tout autant les utiliser pour donner le nom de la machine utilisée ou le type d'article.

La situation s'y prête particulièrement bien ici : ce que nous devons indiquer ne présente aucune structure hiérarchique, et surtout le nombre d'éléments est relativement fixe. Une fois que l'on a déterminé la liste complète des machines, il n'est plus nécessaire d'en ajouter (l'apparition d'une nouvelle console n'est pas fréquente) ; et les types d'articles sur notre site seront déterminés à l'avance. On fixera donc la liste des mots-clés une bonne fois pour toute avant de commencer (les ajouts seront très rares), et nous aurons notre menu déroulant qui permettra, pour chaque article, d'indiquer la machine concernée et le type d'article.

N.B. Il existe d'autres cas où l'utilisation des mots-clés est un piège. Ce sont notamment les cas où la liste de mots n'est pas fixée à l'avance, et où il faudrait quasiment créer un nouveau mot-clé à chaque nouvel article. Certes, cela fonctionnerait avec SPIP, mais la mise à jour du site serait particulièrement pénible.

Par exemple, un site consacré au cinéma. Nous aurions une structure thématique, selon les grands genres du cinéma. Ensuite, nous voudrions pouvoir passer d'un film de tel réalisateur aux autres films du même réalisateur. Une solution (piège) consisterait donc à créer un mot-clé pour ce réalisateur : lorsqu'on ajoute la fiche d'un film, on lui associerait le mot-clé de son réalisateur. De même pour les acteurs principaux, l'auteur du scénario, etc. La navigation dans le site serait ainsi d'une très grande richesse. Mais la mise à jour avec SPIP serait infernale : une liste de mots-clés interminable, et surtout la nécessité de créer des mots-clés à chaque fois qu'on ajoute un film.

Imaginons encore que nous ayons retenu pour notre site une structure par grande catégorie d'articles : une rubrique pour les tests, une rubrique pour les previews, une rubrique pour les news ... Une fois dans le test de *Alone in the dark*, on voudrait afficher des liens vers sa solution, ses news ... Le seul moyen d'y arriver serait alors d'utiliser un mot-clé «Alone in the dark». On voit qu'alors, à chaque nouveau jeu abordé sur le site, on devrait créer un nouveau mot-clé : très fastidieux à gérer, difficile à utiliser (rapidement le menu déroulant deviendrait interminable), interface inutilement surchargée (certains jeux n'étant plus du tout traités quelques mois après leur sortie, leur présence dans la liste des mots-clés affichée à chaque fois devient inutile).

11.4 Mise en place de la structure

23 septembre 2001 par l'équipe de SPIP

Puisque nous avons déterminé la structure du site dans l'article précédent, rendons-nous dans l'espace privé de SPIP, et entrons les éléments nécessaires.

Tâche toujours ingrate, mais indispensable avant de rédiger le moindre article et même de créer les squelettes. En effet, les éléments que nous allons créer dès maintenant serviront de base à la suite.

11.4.1 La structure des rubriques

Commençons par créer le rubriquage thématique (selon les grandes catégories de jeux).

C'est une question de choix personnels, mais voici une idée du résultat (non exhaustif) :

- ▷ Action/aventure
 - * Exploration et énigmes
 - * FPS (Quake-like)
 - * RPG (Jeux de rôle)
 - * Survival horror
- ▷ Combats
 - * 2D
 - * 3D
- ▷ Jeux pédagogiques
 - * Lecture
 - * Mathématiques
 - * Physique
 - * Culture
- ▷ Plateforme
 - * 2D
 - * 3D
- ▷ Sports
 - * Course
 - ◇ Formule 1
 - ◇ Rallye
 - ◇ Grand Tourisme

- ★ Football
- ★ Glisse
- ★ Tennis
- ▷ Stratégie
 - ★ Casse-têtes
 - ★ Jeux de cartes
 - ★ Jeux de plateau
 - ★ Wargames
 - ★ Stratégie en temps réel

Cette structure est incomplète, mais c'est là son intérêt : on peut facilement la modifier en fonction des besoins, sans pour autant devoir repenser le principe général ni les squelettes. Quand une rubrique contient trop de jeux, on peut créer des sous-rubriques pour éviter d'obtenir des listes trop longues (par exemple, si l'on a trop de «jeux de cartes», on peut créer des sous-rubriques «Poker», «Bridge», «Belotte» dans cette rubrique et y transférer les jeux).

Nous avons prévu de créer ensuite des rubriques pour chaque jeu (une rubrique «Resident Evil», une rubrique «Alone in the dark» ...). Mais comme ces rubriques ne concernent pas la structure «fixe» du site, nous verrons cela dans l'article suivant.

11.4.2 Les machines

Comme nous l'avons expliqué précédemment, les machines seront indiquées par des mots-clés.

Il faut d'abord créer un «groupe de mots», que nous nommerons «Machines».

Ensuite, nous allons créer un mot-clé pour chaque machine, chacun de ces mots-clés étant installé dans le groupe «Machines». Ce qui donne :

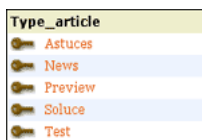


Il est fréquent, sur ce genre de site, de signaler la machine concernée par un petit logo. Nous allons donc attribuer à chaque mot-clé un logo lui correspondant (considérons ce logo comme obligatoire ; en effet, dans les squelettes, nous afficherons fréquemment le logo sans indiquer le nom de la machine en clair). Pour assurer la cohérence graphique, ces logos doivent être approximativement de même taille (par exemple environ 25 pixels x 25 pixels) :



11.4.3 Les types d'articles

De la même façon, nous allons maintenant définir une liste des types d'articles. Cela donne :



Nous n'attribuons pas de logo à ces mots-clés (vous pouvez décider de la faire, vous pourrez ainsi créer des éléments de navigation graphiques).

Une nuance (que nous verrons de manière plus évidente par la suite) par rapport aux mots-clés des machines : ces mots seront utilisés nominativement dans les squelettes ; il faut donc choisir des noms et s'y tenir (si, dans le cas précédent, «Playstation» pourra être ultérieurement changé en «PS One» sans difficulté, ici on ne pourra plus changer «Soluce» en «Solution» sans devoir modifier les squelettes).

Notez que le groupe de mots se nomme «Type_article», et non «Type d'article» ; en effet, nous appellerons spécifiquement ce titre dans les squelettes, et les noms avec des espaces et des apostrophes sont moins faciles à manipuler.

11.5 Écrire des articles

23 septembre 2001 par l'équipe de SPIP

Il est temps désormais de créer les premiers articles. Non seulement cela nous permettra dans ce tutorial d'expliquer la démarche, mais de toute façon cette étape est indispensable avant de pouvoir travailler sur les squelettes.

N.B. Si vous avez déjà manipulé les squelettes de SPIP, vous savez qu'on ne peut réellement travailler que lorsqu'au moins un article est publié. Nous vous conseillons même, dans le cadre de la création de l'interface graphique, de créer le maximum d'articles possibles, avec plusieurs articles par rubriques, plusieurs rubriques ... et même des articles de longueurs différentes, avec ou sans surtitre et sous-titre, avec ou sans chapeau, descriptif ... Plus ces articles seront nombreux et de formats divers, plus vous pourrez créer une interface «souple» (c'est-à-dire adaptée à tous les cas de figure possible). Evidemment, pour les tests d'interface, ces articles peuvent très bien être constitué de «faux texte» (des copier-coller d'articles pris sur le Web, du charabia, du latin de cuisine ...).

11.5.1 Une rubrique par jeu

Nous avons décidé que tous les articles concernant un jeu seraient regroupés dans une même rubrique. Avant même d'écrire le premier article au sujet d'un jeu, il faut donc créer une rubrique portant le nom de ce jeu (pour les articles suivants, évidemment on se placera dans la même rubrique).

La rubrique va donc prendre un rôle très important : c'est elle qui donne le nom du jeu, et c'est elle qui fournit le logo, pour tous les articles qu'elle contient. Par exemple, nous allons créer une rubrique «Resident Evil : Code Veronica» et nous lui attribuerons un logo. Ensuite, lorsque nous insérerons un article contenant la solution du jeu dans cette rubrique, cet article n'aura pas besoin de rappeler le nom du jeu (l'article se nommera simplement «Solution complète»), et il ne sera pas obligatoire de lui attribuer un logo.

Plaçons-nous dans la rubrique :

▷ Action/aventure | Survival horror

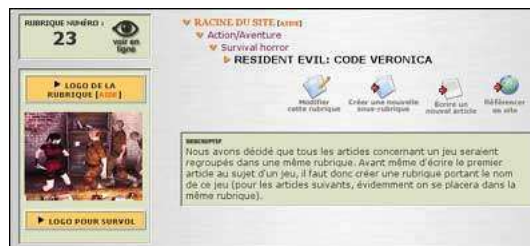
et créons une nouvelle sous-rubrique intitulé «Resident Evil : Code Veronica».

Nous plaçons un «descriptif» pour cette rubrique, présentant la trame général du jeu.

Enfin nous utilisons une copie d'écran pour créer le logo de la rubrique.

N.B. Ces logos de rubrique seront par la suite très utilisés pour obtenir l'interface graphique du site. Nous vous conseillons donc de choisir arbitrairement (en fonction de vos exigences graphiques) un format unique pour tous les logos. Cela facilitera la création d'une interface cohérente. Nous choisissons par exemple que tous les logos de rubriques font 120 x 160 pixels.

Cela donne :



11.5.2 Le test complet du jeu !

Voilà, la rubrique créée, nous pouvons écrire notre premier article concernant ce jeu.

N.B. Soulignons ici un premier inconvénient sur ce site : il faut s'attreindre (et l'expliquer aux autres participants du site) à créer une rubrique avant de pouvoir écrire le premier article concernant un jeu. Or le comportement naturel serait d'écrire l'article directement dans la rubrique «Survival horror», et de lui donner pour titre (ou pour surtitre) le nom du jeu. Cela conviendrait pour un site plus simple (traitant d'une seule machine, ne proposant que des tests ou des soluces ...). Mais puisque nous sommes partis sur un site plus ambitieux, cela induit plus de formalisme dans la structuration.

Dans un premier temps, il suffit de créer du texte, d'intégrer des images ... rien d'inhabituel.

Attention cependant, il y a deux points à déterminer immédiatement, et s'y tenir systématiquement par la suite :

- ▷ le nom du jeu est-il indiqué dans les «cases» de l'article lui-même ? (en titre, surtitre, sous-titre ...)
- ▷ le «descriptif» est-il obligatoire ?

Cela détermine la façon dont nous traiterons les squelettes par la suite. Vous pouvez choisir de façon arbitraire, il suffira de réaliser vos squelettes en conséquence.

Pour les exemples qui seront fournis dans ce tutorial, nous avons décidé que l'article lui-même n'indique pas le nom du jeu (ni dans le titre, ni dans les sur/sous-titres) ; ce nom sera récupéré systématiquement depuis la rubrique qui contient l'article. Et dans nos squelettes, nous utiliserons le #DESCRIPTIF ; donc il est conseillé d'en mettre un, sinon l'espace attribué au descriptif sera vide.

Enfin, il n'est pas obligatoire d'attribuer un logo à l'article lui-même, puisque nous récupérerons celui de la rubrique. Mais si on décide de placer un logo à l'article, nous conseillons de respecter le format adopté pour les logos des rubriques (dans notre cas 120 x 160 pixels).

En gros, cela donne :

- ▷ L'indication de la machine n'est pas réellement obligatoire (l'article apparaîtra tout de même au sommaire) ; mais son absence rendra l'interface incohérente (certains articles indiquant des machines, d'autres non) et l'article ne sera plus référencé dans la navigation «par machine». Il est possible d'indiquer plusieurs machines pour un même article.
- ▷ Le type d'article est indispensable. Nous verrons par la suite que nous faisons un usage intensif de ces types d'articles. En cas d'oubli, l'article risque purement et simplement de ne pas apparaître sur le site public. Il ne faut indiquer qu'un seul type d'article pour un article.

Cela fait, créez d'autres articles, à des formats différents, certains avec plusieurs machines, selon différents types d'articles ... En particulier, essayez de créer une rubrique complète (avec tous les types d'articles), et enrichissez les rubriques proches de «Survival horror», de façon à pouvoir travailler sur des squelettes avec des rubriques représentatives de l'activité du site. Pensez de plus à modifier les dates de publication, afin de pouvoir tester les classements des articles et des rubriques.

L'idéal étant, dans cette situation, de créer un maximum d'articles dans tout le site, avec des dates de publication couvrant une période assez large.

11.6 Première version du squelette des articles

23 septembre 2001 par l'équipe de SPIP

Nous avons désormais un site très structuré depuis l'espace privé, et de nombreux articles publiés. Il est temps de nous attaquer à la création du premier squelette.

Beaucoup de webmestres commencent cette étape en travaillant à partir d'un squelette standard, qu'ils modifient par petites touches.

De notre côté, non seulement pour les besoins de ce tutorial, mais aussi par habitude, nous construisons les squelettes à partir d'un document totalement vide. Étape par étape, les différentes boucles sont ajoutées, puis finalement l'interface graphique (au départ du code HTML extrêmement simple, la construction graphique avec des tableaux étant ajoutée en dernière étape). Cette méthode présente plusieurs avantages :

- ▷ cela permet de bien comprendre sa propre construction intellectuelle lors du développement d'une page : quelle boucle dépend de quelle autre, etc. ;
- ▷ initialement débarrassé des éléments d'interface graphique, il est plus simple de bien percevoir la hiérarchie des informations sur la page ; ainsi l'interface graphique découle-t-elle de cette hiérarchie d'une manière plus cohérente ;
- ▷ le code HTML généré est souvent plus clair : les redondances sont moins nombreuses (défaut fréquent du copier-coller de code HTML).

Dans ce qui suit, nous utiliserons donc du code HTML très simple (notamment : pas de tableaux). La création d'une interface graphique plus élaborée, en fin de processus, est laissée au soin du lecteur.

Volontairement, nous construirons le code étape par étape. Pour éviter des pages interminables, lorsque nous enrichissons une boucle existante, nous ne reproduisons pas l'intégralité de la page, mais uniquement la partie qui nous intéresse. Pour s'y retrouver, on se référera souvent au nom des boucles déjà existantes.

11.6.1 La page de base

Commençons donc avec un fichier «article.html» vierge.

Extrêmement simple, voici la structure minimale d'un article, sans aucun élément de navigation :

```
<html>
<title>[#NOM_SITE_SPIP]
<BOUCLE_titre(ARTICLES){id_article}>#TITRE</BOUCLE_titre></title>
</head>

<body>
<blockquote>
  <BOUCLE_principale(ARTICLES){id_article}>

  [<h3>(#SURTITRE)</h3>]
  <h2>#TITRE</h2>
  [<h3>(#SOUSTITRE)</h3>]

  [(#DATE|affdate)]

  [(#LOGO_ARTICLE_RUBRIQUE|right)]
  [<b>(#CHAPO|justifier)</b>]

  [(#TEXTE|justifier)]

  [<p align="right" align='justify'>(#LESAUTEURS)]

  [<hr>(#PS)]
  [<hr>(#NOTES)]

  </BOUCLE_principale>
</blockquote>
</body>
</html>
```

Facile :

- ▷ la `BOUCLE_titre` permet d'afficher le titre de l'article dans l'entête de la page ;
- ▷ la `BOUCLE_principale` contient d'intégralité de la page ; elle permet de positionner toutes les autres boucles à l'intérieur de l'article sélectionné.

11.6.2 Récupérer les infos sur le jeu

Le squelette précédent serait une bonne base pour un webzine standard (tout l'information nécessaire est dans l'article), mais dans notre cas, une information vitale ne se trouve pas dans l'article : de quel jeu s'agit-il ?

En effet, nous avons décidé que le titre du jeu n'était pas dans l'article, mais dans la rubrique contenant cet article. On pourrait se contenter de la boucle `HIERARCHIE`, mais nous préférons ici prendre une habitude qui reviendra systématiquement dans nos exemples : **il faut passer de l'article à sa rubrique pour récupérer le nom du jeu.** (C'est le même principe pour récupérer le logo ; à la différence près que la balise `#LOGO_RUBRIQUE_ARTICLE` effectue automatiquement cette opération.)

Ce sera donc une constante sur ce site : les boucles `ARTICLES` contiennent souvent une boucle `RUBRIQUES` permettant de « remonter d'un cran », c'est-à-dire de récupérer le titre du jeu. Cela sera le cas dans les pages des rubriques, ainsi que sur la page de sommaire.

Reprenons le début du code, tout au début de la `BOUCLE_principale` :

```
<BOUCLE_principale(ARTICLES){id_article}>
```

```
<BOUCLE_rubrique (RUBRIQUES) {id_rubrique}>
<h1>#TITRE</h1>
</BOUCLE_rubrique>
...
```

Notez que le #TITRE de la rubrique est affiché avec la taille maximale (h1); logique, puisque c'est le titre du jeu. Le titre de la rubrique, qui est d'ordinaire seulement un élément de navigation dans le site, est ici l'information principale de l'article.

Afin d'obtenir une esquisse de navigation, affichons la hiérarchie. Nous allons démarrer cette hiérarchie depuis la rubrique qui contient la sous-rubrique actuelle, puisque ce qui nous intéresse est la liste des catégories de jeux (et nous connaissons déjà le nom de la dernière sous-rubrique - le titre du jeu). Nous nous plaçons donc à l'intérieur de la BOUCLE_rubrique, ajoutons une BOUCLE_rub_parent (qui permet encore de «remonter d'un cran» dans la hiérarchie), et à partir de là nous demandons la hiérarchie :

```
<BOUCLE_rubrique (RUBRIQUES) {id_rubrique}>
  <BOUCLE_rub_parent (RUBRIQUES) {id_enfant}>
    <BOUCLE_hierarchie (HIERARCHIE) {id_rubrique}{" : ">
      <a href="#URL_RUBRIQUE">#TITRE</a>
    </BOUCLE_hierarchie>
  </BOUCLE_rub_parent>
<h1>#TITRE</h1>
</BOUCLE_rubrique>
```

Laissons cette page «article.html» en l'état. Rien de compliqué pour l'instant. Il manque la gestion des mots-clés, et les liens entre les articles d'un même rubrique (c'est-à-dire traitant du même jeu).

Seule subtilité : il faut remonter à la rubrique qui contient l'article pour obtenir le titre du jeu, et il faut encore «remonter d'un cran» (la rubrique qui contient la rubrique qui contient le jeu) avant de démarrer la hiérarchie.

11.7 La page des rubriques

23 septembre 2001 par l'équipe de SPIP

Commençons avec un fichier «rubrique.html» vierge.

La première version de notre fichier sera très simple. Nous intégrons tout de suite la BOUCLE_hierarchie, qui ne présente ici aucune particularité.

```
<html>
<title>[#NOM_SITE_SPIP]
<BOUCLE_titre (RUBRIQUES) {id_rubrique}>#TITRE</BOUCLE_titre></title>
</head>

<body>
<blockquote>
  <BOUCLE_principale (RUBRIQUES) {id_rubrique}>

    <BOUCLE_hierarchie (HIERARCHIE) {" : ">
      <a href="#URL_RUBRIQUE">#TITRE</a>
    </BOUCLE_hierarchie>

  </BOUCLE_principale>
</blockquote>
<h1>#TITRE</h1>
```

```
[ (#LOGO_RUBRIQUE|right) ]

[ (#TEXTE|justifier) ]

[<hr> (#NOTES) ]

</BOUCLE_principale>
</blockquote>
</body>
</html>
```

Si vous installez cette page sur votre site, ça ne va pas être spectaculaire ...

11.7.1 Les articles contenus dans cette rubrique

Ajoutons la liste des articles contenus dans cette rubrique.

Plaçons cette BOUCLE_articles à la fin de la BOUCLE_principale (après le texte et ses notes).

```
...
<B_articles>Les articles :
<ul>
<BOUCLE_articles(ARTICLES){id_rubrique}>
  <li> <a href="#URL_ARTICLE">#TITRE</a>
</BOUCLE_articles>
</ul>
</B_articles>

</BOUCLE_principale>
...
```

En réalité, cette boucle ne présente pas grand intérêt : elle affiche les articles sans classement particulier (elle ne tient pas compte des mots-clés de type d'articles). Nous l'installons pour l'instant telle quelle, afin d'avoir une première navigation possible sur le site. Nous reprendrons cette partie un peu plus loin dans ce tutorial.

11.7.2 Les sous-rubriques

Installons maintenant la boucle des sous-rubriques (après la BOUCLE_articles) :

```
...
<B_sous_rubriques>Les sous-rubriques :
<ul>
<BOUCLE_sous_rubriques(RUBRIQUES){id_parent}{par titre}>
  <li> <a href="#">#TITRE</a>
</BOUCLE_sous_rubriques>
</ul>
<B_sous_rubriques>

</BOUCLE_principale>
...
```


Et là, on rencontre le vrai premier problème de notre tutorial : les rubriques définissant des genres de jeux sont mélangées avec des rubriques consacrées directement à des jeux.

Par exemple, dans la structure de démonstration, la rubrique «Action/aventure» contient à la fois, présentés exactement de la même manière :

- ▷ deux sous-rubriques thématiques («Exploration et énigmes» et «Survival horror»),
- ▷ et deux rubriques de jeux («Devil May Cry» et «Soul Reaver 2»).

L’affichage de la boucle ci-dessus fournit donc la liste :

- ▷ Devil May Cry,
- ▷ Exploration et énigmes,
- ▷ Soul Reaver 2,
- ▷ Survival horror.

Or il faut impérativement différencier très nettement les rubriques «catégories de jeux» (utiles essentiellement à la navigation) et les rubriques «jeux» (qui donnent le titre des jeux).

Pour résoudre cette difficulté, nous allons utiliser le système des boucles. Aucune bidouille informatique donc, mais il va faut bien comprendre la logique du site et la logique des boucles.

La logique du site : une rubrique «Jeu» est une rubrique qui contient des articles ; une rubrique de «catégorie de jeux» ne contient pas directement d’articles. C’est la définition de notre site : tout article concernant un jeu est placé dans une rubrique portant le nom du jeu ; ainsi une rubrique de catégorie de jeu ne contient que des sous-rubriques de catégories et des rubriques portant le nom des jeux, mais jamais directement des articles (est-ce bien clair ?). Par exemple, dans notre cas :

- ▷ «Soul Reaver 2» et «Devil May Cry» contiennent les articles concernant ces jeux (tests, previews, etc.) ;
- ▷ «Survival horror» et «Exploration et énigmes» ne contiennent par définition pas d’articles. S’il y a des jeux dans «Survival horror», les articles les concernant sont dans des sous-rubriques portant le nom de ces jeux, mais pas directement dans la rubrique «Survival ...».

Le but de la manoeuvre va donc consister à séparer les rubriques *sans* articles des rubriques *avec* des articles. Et le plus simple est de faire deux boucles différentes.

11.7.3 Les rubriques sans articles

Commençons par les sous-rubriques qui ne contiennent pas d’articles (c’est-à-dire les grandes catégories de type «Survival horror», «Exploration» ...). Modifions la `BOUCLE_sous_rubriques` ainsi :

```
<BOUCLE_sous_rubriques(RUBRIQUES){id_parent}{par titre}>
  <BOUCLE_sans_articles(ARTICLES){id_rubrique}>
</BOUCLE_sans_articles>
<h3><a href="#URL_RUBRIQUE">#TITRE</a></h3>
</B_sans_articles>
</BOUCLE_sous_rubriques>
```

Le principe de cette astuce est simple :

- ▷ à l’intérieur de la `BOUCLE_sous_rubriques`, on insère une nouvelle `BOUCLE_sans_articles`, qui va récupérer les articles qu’elle contient. Notez que l’intérieur de la `BOUCLE_sans_articles` est vide : si la sous-rubriques contient des articles, ces derniers ne sont pas affichés (la boucle ne sert qu’à contrôler leur présence) ;

▷ le #TITRE de la BOUCLE_sous_rubriques est inséré dans le texte optionnel «sans» de la BOUCLE_sans_articles. Juste avant le </B_sans_articles>.

Ainsi, le titre de la rubrique ne s'affiche que si BOUCLE_sans_articles ne contient aucun article (c'est-à-dire si la BOUCLE_sous_rubriques elle-même ne contient aucun article).

La BOUCLE_sous_rubriques, dans l'absolu, récupère bien toutes les rubriques (y compris celle des jeux). Mais le #TITRE (avec le lien hypertexte) n'est affiché que si la BOUCLE_sans_articles est vide.

11.7.4 Les rubriques de jeu (avec des articles)

Nous appliquons le principe exactement inverse à une BOUCLE_jeux (que nous pouvons placer juste avant BOUCLE_sous_rubriques).

```
<BOUCLE_jeux(RUBRIQUES){id_parent}{par titre}>

  <B_avec_articles>
  <h4><a href="#URL_RUBRIQUE">#TITRE</a></h4>
  [(#LOGO_RUBRIQUE|right)]
  <ul>
  <BOUCLE_avec_articles(ARTICLES){id_rubrique}>
  <li><a href="#URL_ARTICLE">#TITRE</a>
  </BOUCLE_avec_articles>
  </ul>
  </B_avec_articles>

</BOUCLE_jeux>
```

Cette fois, le #TITRE de la rubrique est placé dans la *texte conditionnel* «avant» ; on affiche le #LOGO_RUBRIQUE (évidemment, à terme, on fera de beaux tableaux pour éviter les chevauchements disgracieux). Et au passage, on en profite pour afficher la liste des articles de la BOUCLE_avec_articles.

La BOUCLE_jeux et la BOUCLE_sous_rubriques sont au départ totalement identiques : leurs critères sont les mêmes. À l'intérieur de ces boucles, ce sont donc les mêmes rubriques qui sont traitées.

En revanche, les boucles BOUCLE_avec_articles et BOUCLE_sans_articles servent de «filtres» : ils prennent chacune des rubriques de BOUCLE_jeux et BOUCLE_sous_rubriques, et vérifient la présence ou non d'articles.

Est-ce que l'on aurait pu ici utiliser le critère doublons ?

Non, car la première boucle affichée (BOUCLE_sous_rubriques) traite bien *toutes* les sous-rubriques (seulement, le titre de ces sous-rubriques est affiché ou non selon les cas). Lui attribuer un critère doublons interdirait par la suite de traiter aucune de ces sous-rubriques. Ensuite, la BOUCLE_jeux associée à ce critère de doublons tenterait de récupérer toutes les sous-rubriques qui n'ont pas été traitées par la première boucle ; or toutes les sous-rubriques ont bien été traitées. La BOUCLE_jeux ne pourrait donc rigoureusement rien afficher.

Laissons maintenant cette page «rubrique.html», nous y reviendrons plus tard.

11.8 Les mots-clés dans les articles

23 septembre 2001 par l'équipe de SPIP

Revenons à nos articles ... Nous allons maintenant exploiter les mots-clés pour déterminer la ou les machines concernés, puis gérer «intelligemment» les liens vers les autres articles concernant le même jeu.

11.8.1 Les machines

Indiquer quelles machines sont concernées est très simple : il s'agit d'une simple application de la boucle (MOTS). Nous insérons la boucle suivante dans l'article (par exemple entre le #SOUSTITRE et la #DATE) :

```
<BOUCLE_machines (MOTS) {id_article}{type=Machines}>
  [ (#LOGO_MOT|left) ]
</BOUCLE_machines>
```

Nous n'affichons que le logo de la machine. (Évidemment, une fois tout le processus de création des boucles terminé, il conviendra de figoler l'interface graphique ...)

11.8.2 Dans la même rubrique, au sujet du même jeu

Afficher les autres articles contenus dans la même rubrique n'est guère plus compliqué.

Commençons par modifier l'appel de la BOUCLE_principale avec un {doublons}, pour éviter ensuite de créer un lien vers l'article déjà affiché (ça n'est pas bien grave, mais ça ne fait pas très propre) :

```
<BOUCLE_principale (ARTICLES) {id_article}{doublons}>
```

Nous pourrions désormais nous contenter de récupérer tous les articles de la même rubrique (très simplement, une boucle (ARTICLES){id_rubrique}{doublons}). Cependant, nous voulons ici différencier clairement les types d'articles, non seulement pour rendre l'interface plus claire, mais aussi pour pouvoir réaliser des liens et des présentations différents selon les cas.

Commençons par une boucle vers les autres tests du même jeu. Nous insérons cette boucle toujours à l'intérieur de la BOUCLE_principale ; par exemple juste après #NOTES.

```
<B_tests><p>Autres tests de ce jeu :
  <ul>
  <BOUCLE_tests (ARTICLES) {id_rubrique}{titre_mot=Test}{doublons}>
    <li><a href="#URL_ARTICLE">#TITRE</a>
  </BOUCLE_tests>
  </ul>
</B_tests>
```

(Pour vérifier le fonctionnement de cette boucle, placez plusieurs articles de type «Test» dans la même rubrique. Vous pouvez même leur attribuer des machines différentes - un test de la version PC, un test de la version Dreamcast, un test de la version Playstation ...)

Comme vous le constatez, c'est très simple :

- ▷ on appelle les (ARTICLES) selon le critère {id_rubrique} (c'est-à-dire les articles appartenant à la même rubrique) ;
- ▷ on ajoute la restriction suivante : le titre d'un des mots-clés liés à ces articles doit être «Test» (notez ici l'importance de respecter le choix initial du nom du mot-clé, et de ne plus le modifier par la suite) ;
- ▷ le critère {doublons} interdit d'afficher à nouveau l'article principal.

Les textes optionnels avant et après (les ...) ne servent que pour la présentation graphique.

N.B. Cette boucle utilise une fonctionnalité introduite dans le version **SPIP 1.3** : dans une boucle (ARTICLES), vous pouvez sélectionner les articles selon le titre d'un mot-clé (avec le critère titre_mot), ou selon un groupe de mot-clé (critère type_mot).

Cependant cette boucle n'est pas suffisante : puisqu'il s'agit de tests du même jeu, mais sur d'autres machines, il faut indiquer de quelle machine il s'agit. Il suffit d'ajouter une boucle de mots (comme ci-dessus) à l'intérieur de cette `BOUCLE_tests` et d'afficher les logos correspondants. Ajoutons au passage la date de publication de l'article.

Le code devient :

```
<B_tests><p>Les tests de ce jeu :
<ul>
<BOUCLE_tests(ARTICLES){id_rubrique}{titre_mot=Test}>
  <li>
    <BOUCLE_tests_machines(MOTS){id_article}{type=Machines}>
      [ (#LOGO_MOT) ]
    </BOUCLE_tests_machines>
    <a href="#URL_ARTICLE">#TITRE</a> - [ (#DATE|affdate) ]
  </BOUCLE_tests>
</ul>
</B_tests>
```

Même chose pour les previews, les soluces et les astuces :

```
<B_previews><p>Les previews de ce jeu :
<ul>
<BOUCLE_previews(ARTICLES){id_rubrique}{titre_mot=Preview}>
  <li>
<BOUCLE_previews_machines(MOTS){id_article}{type=Machines}>
    [ (#LOGO_MOT) ]
  </BOUCLE_previews_machines>
  <a href="#URL_ARTICLE">#TITRE</a> - [ (#DATE|affdate) ]
</BOUCLE_previews>
</ul>
</B_previews>

<B_soluces><p>Les soluces de ce jeu :
<ul>
<BOUCLE_soluces(ARTICLES){id_rubrique}{titre_mot=Soluce}>
  <li>
<BOUCLE_soluces_machines(MOTS){id_article}{type=Machines}>
    [ (#LOGO_MOT) ]
  </BOUCLE_soluces_machines>
  <a href="#URL_ARTICLE">#TITRE</a> - [ (#DATE|affdate) ]
</BOUCLE_soluces>
</ul>
</B_soluces>

<B_astuces><p>Les trucs et astuces de ce jeu :
<ul>
<BOUCLE_astuces(ARTICLES){id_rubrique}{titre_mot=Astuces}>
  <li>
<BOUCLE_astuces_machines(MOTS){id_article}{type=Machines}>
    [ (#LOGO_MOT) ]
  </BOUCLE_astuces_machines>
  <a href="#URL_ARTICLE">#TITRE</a> - [ (#DATE|affdate) ]
</BOUCLE_astuces>
</ul>
</B_astuces>
```

(Ces trois boucles fonctionnent exactement sur le même principe que la `BOUCLE_tests`.)

Restent les news ... Nous allons ici adopter un comportement très différent. Les news étant généralement des informations «courtes» (annonce d'un délais, rumeurs ...), on aura avantage à les présenter toutes réunies sur une même page (peut-être pas, mais pour les besoins de ce tutorial, on dira que c'est mieux !).

Le petit morceau de code fera l'affaire :

```
<BOUCLE_news (ARTICLES) {id_rubrique}{titre_mot=News}{doublons}{par date}
                                     {inverse}{0,1}>
  <p><a href="news_jeu.php3?id_rubrique=#ID_RUBRIQUE">Les news de ce jeu</a> -
                                     dernière mise-à-jour : [(#DATE|affdate)]
</BOUCLE_news>
```

L'astuce qui permet d'obtenir un code aussi court :

- ▷ tout d'abord, on ne s'intéresse plus à la machine concernée ; cela supprime une boucle ;
- ▷ la `BOUCLE_news` récupère le dernier article (`{par date}{inverse}{0,1}`) associé au mot-clé «News». Un seul article, cela suffit largement pour savoir s'il y a une page consacrée aux news ; et en récupérant la plus récente news, afficher sa date revient à annoncer la «dernière mise-à-jour» de la page des news ...

De manière plus élégante, on aurait pu réaliser une `BOUCLE_news` vide et insérer le lien hypertexte dans le *texte optionnel* de la boucle.

Important : le lien hypertexte pointe vers une page «`news_jeu.php3`», avec l'`id_rubrique` de la rubrique actuelle. Nous allons devoir, par la suite, créer le squelette correspondant.

11.8.3 Les jeux dans la même catégorie

Nous voulons, pour finir, afficher les jeux de la même catégorie (par exemple, les autres jeux du genre «Survival horror»).

Nous allons d'abord utiliser la méthode qui consiste à «remonter d'un cran» (pour passer de la rubrique de l'article à la rubrique de la catégorie de ce jeu). Puis nous allons récupérer la méthode utilisée dans les rubriques pour afficher séparément les rubriques qui contiennent des articles (c'est-à-dire des rubriques directement consacrées à un jeu).

Le code est :

```
<BOUCLE_rub_act (RUBRIQUES) {id_rubrique}{doublons}>
<BOUCLE_jeux (RUBRIQUES) {meme_parent}{doublons}{par date}{inverse}>
  <B_avec_articles>
  <h4><a href="#URL_RUBRIQUE">#TITRE</a></h4>
<BOUCLE_avec_articles (ARTICLES) {id_rubrique}{age < 90}>
  </BOUCLE_avec_articles>
</BOUCLE_jeux>
</BOUCLE_rub_act>
```

Les subtilités de ce code :

- ▷ la `BOUCLE_rub_act` (qui renvoie la rubrique actuelle) contient un `{doublons}` pour éviter ensuite d'afficher le jeu principal ;
- ▷ logiquement, la `BOUCLE_jeux` contient elle aussi un `{doublons}` ; attention : multiplier les critères `doublons` peut conduire à des difficultés. Si vous dupliquez ces boucles dans la même page pour un autre usage, prenez soin à éviter ce critère ;
- ▷ la `BOUCLE_avec_articles` ne récupère que les articles agés de moins de 3 mois (90 jours). En effet, nous ne voulons pas ici afficher l'intégralité des jeux de la même catégorie, mais uniquement ceux qui ont été modifiés récemment.

11.9 Les mots-clés dans les rubriques

23 septembre 2001 par l'équipe de SPIP

Revenons au fichier «rubrique.html». Nous l'avons laissé avec une méchante `BOUCLE_articles`, sans intérêt pour notre site, puisqu'elle ne classait pas les articles par catégories.

Le monde est bien fait : nous venons de programmer une telle fonctionnalité pour les articles !

Il suffit donc de copier-coller le bloc qui nous intéresse, de l'insérer à la place de la `BOUCLE_articles` (qui disparaît donc), et de voir s'il y a des modifications nécessaires. Ce qui nous donne :

```
<BOUCLE_les_articles (RUBRIQUES) {id_rubrique}>
  <B_tests><p>Les tests de ce jeu :
  <ul>
  <BOUCLE_tests (ARTICLES) {id_rubrique}{titre_mot=Test}>
    <li>
    <BOUCLE_tests_machines (MOTS) {id_article}{type=Machines}>
      [ (#LOGO_MOT) ]
    </BOUCLE_tests_machines>
    <a href="#URL_ARTICLE">#TITRE</a> - [ (#DATE|affdate) ]
  </BOUCLE_tests>
  </ul>
</B_tests>

  <B_previews><p>Les previews de ce jeu :
  <ul>
  <BOUCLE_previews (ARTICLES) {id_rubrique}{titre_mot=Preview}>
    <li>
    <BOUCLE_previews_machines (MOTS) {id_article}{type=Machines}>
      [ (#LOGO_MOT) ]
    </BOUCLE_previews_machines>
    <a href="#URL_ARTICLE">#TITRE</a> - [ (#DATE|affdate) ]
  </BOUCLE_previews>
  </ul>
</B_previews>

  <B_soluces><p>Les soluces de ce jeu :
  <ul>
  <BOUCLE_soluces (ARTICLES) {id_rubrique}{titre_mot=Soluce}>
    <li>
    <BOUCLE_soluces_machines (MOTS) {id_article}{type=Machines}>
      [ (#LOGO_MOT) ]
    </BOUCLE_soluces_machines>
    <a href="#URL_ARTICLE">#TITRE</a> - [ (#DATE|affdate) ]
  </BOUCLE_soluces>
  </ul>
</B_soluces>

  <B_astuces><p>Les trucs et astuces de ce jeu :
  <ul>
  <BOUCLE_astuces (ARTICLES) {id_rubrique}{titre_mot=Astuces}>
    <li>
    <BOUCLE_astuces_machines (MOTS) {id_article}{type=Machines}>
      [ (#LOGO_MOT) ]
    </BOUCLE_astuces_machines>
    <a href="#URL_ARTICLE">#TITRE</a> - [ (#DATE|affdate) ]
```

```
</BOUCLE_astuces>
</ul>
</B_astuces>

<BOUCLE_news (ARTICLES) {id_rubrique}{titre_mot=News}{par date}{inverse}{0,1}>
<p><a href="news_jeu.php3?id_rubrique=#ID_RUBRIQUE">Les news de ce jeu</a> -
                                dernière mise-à-jour : [(#DATE|affdate)]
</BOUCLE_news>
</BOUCLE_les_articles>
```

Les modifications apportées :

- ▷ inutile de changer la requête {id_rubrique}, puisque c'était déjà sur cette rubrique que l'on se basait dans les articles ;
- ▷ supprimons tous les {doublons}, ils ne sont d'aucune utilité ici (il n'y a pas d'article déjà affiché) ; autant ne pas prendre le risque de conflits avec de futurs développements de notre page s'ils ne servent à rien ;
- ▷ l'ensemble est installé dans une grande BOUCLE_les_articles. Que fait cette boucle ? Strictement rien : elle renvoie la rubrique dans laquelle nous nous trouvons déjà. Mais elle nous sera utile ci-après ...

L'autre point faible de notre page était la BOUCLE_jeux, dans laquelle la BOUCLE_avec_articles profitait de son passage pour afficher la liste des articles de la sous-rubrique sans effectuer de tri. Or, trier les articles d'une rubrique, c'est ce que nous venons de faire ci-dessus.

Modifions donc la BOUCLE_jeux ainsi :

```
<BOUCLE_jeux (RUBRIQUES) {id_parent}{par titre}>

  <B_avec_articles>
  <h4><a href="#URL_RUBRIQUE">#TITRE</a></h4>
  [(#LOGO_RUBRIQUE|right)]
  <ul>
  <BOUCLE_avec_articles (ARTICLES) {id_rubrique}{0,1}>
    <BOUCLE_rec (boucle_les_articles)></BOUCLE_rec>
  </BOUCLE_avec_articles>
  </ul>
  </B_avec_articles>

</BOUCLE_jeux>
```

La BOUCLE_avec_articles est désormais effectuée une seule fois (sur un seul article), ce qui est suffisant pour savoir si elle contient un article. On voit apparaître une BOUCLE_repetier_articles : il s'agit d'un boucle qui reproduit exactement le comportement de la BOUCLE_les_articles, à partir de l'endroit où elle se trouve (c'est-à-dire qu'elle démarre non plus depuis la rubrique principale, mais de la sous-rubrique). L'intérêt désormais de limiter la BOUCLE_avec_articles à un unique élément est alors clair : sans cette limitation, la BOUCLE_rec serait exécutée autant de fois qu'il y a d'articles dans la sous-rubrique.

Évidemment, le résultat graphique de cet exemple est hideux : trop chargé, peu clair. Disons qu'il était intéressant d'utiliser une «boucle récursive» dans ce tutorial ...

11.10 L'interface des news

11.10.1 Toutes les news d'un jeu

Nous avons inséré, dans les squelettes des articles et des rubriques, un lien vers une page «news_jeu.php3», qui doit présenter d'un coup toutes les news concernant un jeu.

Créons donc ce squelette ...

Tout d'abord (principe du couple de fichiers pour gérer les squelettes de SPIP), nous allons créer le fichier d'appel «news_jeu.php3». Il suffit par exemple de copier le fichier «article.php3» (fourni avec SPIP) et d'en modifier quelques éléments. Voici le contenu du fichier «news_jeu.php3» :

```
<?
$fond = "news_jeu";
$delais = 24 * 3600;

include ("inc-public.php3");
?>
```

Le fichier de squelette utilisé sera donc «news_jeu» («.html», ou une des variantes localisées à une rubrique si nécessaire, telle «news_jeu-52.html» ...).

Le squelette sera donc un fichier «news_jeu.html». Ce fichier est utilisé avec une variable de *rubrique*, puisque c'est bien la rubrique qui contient le nom du jeu et l'intégralité des articles concernant le jeu. Vite fait, bien fait, on va recopier le contenu du fichier «rubrique.html» et supprimer de dont nous n'avons plus besoin :

```
<html>
<title>[#NOM_SITE_SPIP]
<BOUCLE_titre(RUBRIQUES){id_rubrique}>#TITRE</BOUCLE_titre></title>
</head>

<body>
<blockquote>
<BOUCLE_principale(RUBRIQUES){id_rubrique}>

  <BOUCLE_hierarchie(HIERARCHIE){" : ">
  <a href="#URL_RUBRIQUE">#TITRE</a>
  </BOUCLE_hierarchie>

  <h1><a href="#URL_RUBRIQUE">#TITRE</a></h1>

  [ (#LOGO_RUBRIQUE|right) ]

</BOUCLE_principale>
</blockquote>
</body>
</html>
```

Cela affiche le titre de la rubrique (c'est-à-dire le nom du jeu), le logo de la rubrique, et la hiérarchie. Seule modification : le titre du jeu est un lien hypertexte pour revenir à la page générale (lors de la finition graphique du site, on pourra évidemment préférer un lien plus discret).

Tout ce qu'il nous reste à faire : insérer une unique boucle qui affichera tous les articles de cette rubrique liés au mot-clé «News». À l'intérieur de la BOUCLE_principale, sous le #LOGO_RUBRIQUE, il suffit d'indiquer :


```
<BOUCLE_news (ARTICLES) {id_rubrique}{titre_mot=News}{par date}{inverse}>
  <hr><h3>#TITRE</h3>
  [ (#DATE|affdate) ]

  <BOUCLE_news_machines (MOTS) {id_article}{type=Machines}>
    [ (#LOGO_MOT|right) ]
  </BOUCLE_news_machines>

  [ (#TEXTE|justifier) ]
  [<p><font size=2>(#PS)</font>]
  [<p>(#NOTES) ]

</BOUCLE_news>
```

La `BOUCLE_news` fait tout le travail : elle affiche le titre de chaque news, son texte, et si nécessaire le post-scriptum et des notes de bas de page.

Comme à l'habitude, la `BOUCLE_news_machines` affiche le logo des machines concernées par la news.

Considérons notre page des news terminée. On pourra évidemment y ajouter les liens vers les articles de la même rubrique (les tests, les previews, les solutions ...).

11.11 Et encore d'autres moyens de se compliquer la vie !

23 septembre 2001 par l'équipe de SPIP

Noter les jeux, annoncer les dates de sortie

Notre site utilise désormais ce qui était prévu à l'origine :

- ▷ le rubriquage thématique (par genre de jeux) ;
- ▷ l'indication des machines concernées par chaque article ;
- ▷ la gestion des types d'articles (tests, news, previews ...).

L'étape suivante consistera à créer des navigations parallèles (naviguer selon une unique machine, ou selon un type d'articles).

Mais avant de nous lancer dans le développement de ces navigations transversales, compliquons-nous un peu l'affaire, en ajoutant deux autres éléments indispensables aux sites de jeux vidéo :

- ▷ la note attribuée à un jeu ;
- ▷ la date de sortie officielle du jeu.

Ces deux éléments devraient être, désormais, simples à installer avec le système des mots-clés. Ils permettront eux-même de proposer des navigations transversales (notamment une page des jeux ayant les meilleures notes, et une page d'annonce des prochaines sorties).

11.11.1 L'attribution d'une note à un jeu

Il est logique d'attribuer une note à la fin d'un test :

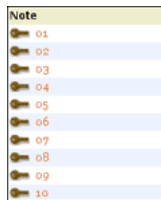
- ▷ pour les previews, c'est trop tôt, puisqu'il ne s'agit pas d'une version définitive ; pour les autres types d'articles (news, soluces ...), ça n'a tout simplement rien à voir ;
- ▷ il est impossible d'attribuer une note à la rubrique-jeu, puisque cette rubrique peut présenter différentes versions d'un jeu.

Donc, nous décidons que, dans le fonctionnement de notre site, la note est liée à un article de test. Dans une même rubrique d'un jeu, puisqu'il peut y avoir plusieurs tests différents (selon les critiques, selon les machines ...), il y aura donc plusieurs notes, associées à ces tests.

Arbitrairement, nous choisissons une notation sur 10 (de 1 pour nul, à 10 pour génial).

Retour à la page de gestion des mots-clés :

- ▷ nous créons un groupe de mots intitulé «Note» ;
- ▷ dans ce groupe de mots, nous créons 10 mots-clés, dont les noms sont successivement «01», «02», ..., «09», «10». Ce qui donne :



On peut choisir, comme pour les machines, d'attribuer un logo différent à chaque note (par exemple des étoiles), et ensuite dans les squelettes utiliser les logos plutôt que le nombre indiqué par le mot-clé en toutes lettres. Par pure fainéantise, nous nous contenterons ici d'utiliser le texte des mots-clés.

Désormais, avant la publication d'un test, en plus de sélectionner les mots-clés correspondant au type de l'article («Test», donc) et ceux des machines concernées, il faut penser à choisir une note. Si l'on veut un fonctionnement cohérent, il est évident qu'on ne doit sélectionner qu'une seule note par article.

11.11.2 Intégrer la note dans les squelettes

La boucle permettant d'afficher la note d'un article est très simple.

- ▷ Dans «article.html», plaçons la note de l'article sous la signature de l'auteur (avant le #PS et les #NOTES) :

```
<BOUCLE_note (MOTS) {id_article}{inverse}{type=Note}>
  [<p align="right"><b>NOTE : (#TITRE)/10</b>]
</BOUCLE_note>
```

Ajoutons également la mention de la note dans les «autres tests de ce jeu». La `BOUCLE_tests` est ainsi modifiée :

```
<B_tests><p>Les tests de ce jeu :
<ul>
  <ul>
    <BOUCLE_tests (ARTICLES) {id_rubrique}{titre_mot=Test}>
      <li>
        <BOUCLE_tests_machines (MOTS) {id_article}{type=Machines}>
          [ (#LOGO_MOT) ]
        </BOUCLE_tests_machines>
        <a href="#URL_ARTICLE">#TITRE</a>
        <BOUCLE_autre_note (MOTS) {id_article}{inverse}{type=Note}>
          [ - NOTE : (#TITRE)/10 ]
        </BOUCLE_autre_note>
        - [ (#DATE|affdate) ]
      </BOUCLE_tests>
    </ul>
  </ul>
</B_tests>
```

- ▷ Dans «rubrique.html», cette même `BOUCLE_tests` est à remplacer directement.
Voilà, l'intégration d'une note de test n'a pas pris cinq minutes ! Tout cela étant trop facile, nous verrons par la suite comment exploiter ailleurs cette note attribuée aux jeux.

11.11.3 La date de sortie officielle des jeux

L'affichage de la date de sortie officielle des jeux est plus problématique. En effet, une date ne peut pas être elle-même un mot-clé (nous avons déjà souligné que les mots-clés devaient être des éléments stables de la structure : on ne va donc pas créer des mots-clés de dates !).

Il existe une possibilité : utiliser la date de première publication de certains articles. Problème : quels articles ?

- ▷ Les dates de sortie sont connues (de manière plus ou moins approximative) longtemps à l'avance, donc on ne peut pas attendre qu'il y ait des articles consacrés au jeu pour pouvoir l'annoncer.
- ▷ Les articles d'un jeu peuvent traiter d'une plateforme ou de plusieurs. Or les dates de sorties selon les plateformes est très variables. On peut donc difficilement utiliser des articles communs à plusieurs plateformes pour annoncer les dates. Par exemple, *Alone in the Dark : the new nightmare*, est très similaire entre ses versions Playstation 2 et Dreamcast, il est donc logique de consacrer un test unique aux deux versions ; en revanche, les dates de sorties sont différentes sur ces plateformes.
- ▷ Tant que le jeu n'est pas commercialisé, les dates annoncées doivent être corrigées régulièrement (retards, plus grande précision ...). Le webmestre doit donc facilement trouver où il doit modifier ces informations.

Dans la structure de notre site, on peut donc considérer que la date de sortie d'un jeu est une information *indépendante* des autres informations (tests, previews, etc.) ; nous ne pouvons donc pas utiliser la «date de première publication» d'un autre article pour indiquer cette date de sortie. Nous allons procéder de la façon suivante (il en existe certainement d'autres) :

- ▷ nous créons un nouveau type d'article (en plus des tests, previews, etc.) ; installé dans le groupe «Type_article», nous le nommerons «Date_sortie» ;
- ▷ dans la rubrique du jeu, nous créons autant d'articles nécessaires qu'il y a de versions du jeu avec des dates de sortie différentes. Ces articles sont *vides* : ils n'ont qu'un titre et une date de «publication» (fixée manuellement à la date de sortie du jeu). Son titre lui-même n'a aucune importance, nous ne l'afficherons pas sur le site public ; en revanche, ce titre facilitera la navigation dans le site privé ;
- ▷ à chaque article associé à «Date_sortie», on attribue également le mot-clé de la machine concernée par cette date.

Voici par exemple comment procéder avec *Resident Evil : Code Veronica* :

- ▷ créer un nouvel article dans la rubrique «Code Veronica» ;
- ▷ titrer cet article «Sortie Veronica Dreamcast» (titre de convenance, sans autre intérêt que d'identifier ces articles dans l'espace privé) ;
- ▷ attribuer les mots-clés «Dreamcast» et «Date_sortie» à cet article ;
- ▷ «publier» l'article, et modifier sa date en «mai 2000» (si l'on ne connaît pas le jour exact, on peut sélectionner «n.c.» dans le menu déroulant du jour) ;
- ▷ recommencer l'opération avec un nouveau article «Sortie Veronica PS2» ;
- ▷ lui attribuer les mots-clés «Playstation 2» et «Date_sortie» ;
- ▷ publier et modifier la date (13 septembre 2001).
Pour le premier article, cela donne :

L'auteur de ces articles est totalement indifférent (inutile de perdre du temps à le supprimer, de toute façon il ne sera pas utilisé).

Notez l'intérêt de notre façon d'afficher les articles d'une rubrique *en fonction de certains types d'articles* (tests, previews ...): les articles ayant pour mot-clé «Date_sortie» ne sont pas affichés (avec les squelettes standards de SPIP, ces «articles» pourtant vides de tout contenu seraient affichés comme les autres).

11.11.4 Afficher la date dans les articles

Lorsque l'on se trouve dans un article (d'un jeu), la (les) date(s) de sortie sont contenus dans des articles associés au mot «Date_sortie», situés dans la même rubrique. Pour connaître la date de sortie d'un jeu depuis un article de test (par exemple), il faut donc récupérer le (ou les) article(s) installés dans la même rubrique et ayant pour mot-clé «Date_sortie».

Dans le squelette «article.html», insérons (par exemple juste après le titre) :

```
<BOUCLE_sortie(ARTICLES){id_rubrique}{titre_mot=Date_sortie}>
  <li> #TITRE :
  [<b>(#DATE|affdate)</b>]
</BOUCLE_sortie>
```

Pendant, nous avons décidé de ne pas utiliser le titre de l'article contenant la date. Nous souhaitons en effet unifier la présentation de cette information, sans pour autant nous obliger à saisir le titre toujours de la même façon. Nous n'allons donc pas afficher le #TITRE de cet article-bidon (ni même le titre de la rubrique, nous connaissons déjà le titre du jeu, inutile de le rappeler ici). En revanche, pour différencier (éventuellement) les différentes dates de sorties en fonction des machines, nous allons insérer :

```
<BOUCLE_sortie(ARTICLES){id_rubrique}{titre_mot=Date_sortie}>
  <li> Date de sortie
    <BOUCLE_machine_sortie(MOTS){id_article}{type=Machines}{", "}>#TITRE
    </BOUCLE_machine_sortie> :
    [<b>(#DATE|affdate)</b>]
  </BOUCLE_sortie>
```

Ce qui lors de la visite donnera :

- ▷ Date de sortie Playstation : 22 août 2001
- ▷ Date de sortie Dreamcast : mai 2001

Cela ne nous satisfait pas encore (c'est un tutorial, nous avons le droit d'être exigeants!) : la BOUCLE_sortie affiche les dates de sortie sur toutes les machines ; or, notre article peut ne pas traiter de toutes ces machines. Nous voulons donc n'afficher que les dates de sortie correspondant aux machines traitées par l'article.

Attention, ça devient coton, et cela utilise une subtilité des boucles :

```
<BOUCLE_mac2 (MOTS) {id_article}{type=Machines}>
  <BOUCLE_sortie (ARTICLES) {id_rubrique}{titre_mot=Date_sortie}>
  <BOUCLE_verifier_mot (ARTICLES) {id_article}{id_mot}>
  <li> Date de sortie
    <BOUCLE_machine_sortie (MOTS) {id_article}{type=Machines}{", "}>#TITRE
  </BOUCLE_machine_sortie> :
  [<b> (#DATE|affdate)</b>]
</BOUCLE_verifier_mot>
</BOUCLE_sortie>
</BOUCLE_mac2>
```

Vous reconnaissez les `BOUCLE_sortie` et `BOUCLE_machine_sortie`, que nous n'avons pas modifiées. En revanche, deux boucles apparaissent.

1. L'ensemble est placé dans une `BOUCLE_mac2`, qui va récupérer les mots-clés des machines associées à cet article. À l'intérieur de cette boucle, nous avons donc une certaine valeur pour «`id_mot`» (l'identifiant de chaque machine concernée par l'article).
2. La `BOUCLE_sortie`, inchangée, récupère tous les articles de la rubrique associés au mot-clé «`Date_sortie`». Notez bien : cette boucle n'utilise pas «`id_mot`», donc les articles de cette boucle sont à la fois des boucles associés au mot-clé dont «`id_mot`» et qui n'y sont pas associés. Second point important, c'est une boucle (`ARTICLES`), cette boucle ne renvoie aucune information sur les mots-clés ; donc la valeur «`id_mot`» fournie par `BOUCLE_mac2` reste inchangée à l'intérieur de cette boucle.
3. La `BOUCLE_verif_mot` est une subtilité intéressante : c'est une boucle (`ARTICLES`) sélectionnant l'article dont l'identifiant est «`id_article`» ; or nous sommes déjà dans la `BOUCLE_sortie`, qui a déjà retourné un article. Donc la `BOUCLE_verif_mot` retourne exactement le même résultat que la boucle précédente !
À cette différence que la `BOUCLE_verif_mot` instaure un critère supplémentaire, {`id_mot`}, l'article doit donc être associé au mot-clé «`id_mot`».
Résultat : si l'article retourné par `BOUCLE_sortie` est associé au mot-clé «`id_mot`», on continue ; si l'article de `BOUCLE_sortie` n'est pas associé, on s'arrête et on passe au suivant.

- ▷ On peut ainsi dire que la `BOUCLE_verif_mot` est un filtre : elle prend l'article et vérifie s'il correspond au critère du mot-clé.
- ▷ On peut le présenter autrement : `BOUCLE_verif_mot` est seulement un critère de sélection supplémentaire pour la `BOUCLE_sortie`. Le plus simple en effet aurait été de ne pas utiliser de `BOUCLE_verif_mot`, mais d'ajouter un critère à `BOUCLE_sortie`, ainsi :

```
<BOUCLE_sortie (ARTICLES) {id_rubrique}{id_mot}{titre_mot=Date_sortie}>
```

C'est-à-dire :

récupérer les articles de la même rubrique associés à la machine «`id_mot`» et au mot-clé «`Date_sortie`». Cela aurait été plus direct, malheureusement SPIP n'accepte pas ce genre de constructions (il faut un seul critère portant sur les mots-clés dans une boucle). Une telle boucle est donc refusée par SPIP.

Vous remarquerez que nous avons empilé quatre boucles successives pour afficher une information aussi peu importante que la date de sortie d'un jeu ! On peut le déplorer, mais on peut préférer penser que, si l'on comprend bien la logique des boucles, les possibilités de SPIP sont beaucoup plus étendues que ce que les squelettes standards laissent supposer.

N.B. En utilisant un peu de code PHP très simple, on aurait obtenu le même résultat avec une seule boucle. Dans le développement d'un site, on aura certainement avantage à gagner du temps en préférant la version raccourcie utilisant un peu de PHP. Mais nous tenions à montrer que des certains résultats peuvent très bien se passer de PHP.

11.12 Quelques sommaires alternatifs

23 septembre 2001 par l'équipe de SPIP

Déjà une tripotée d'articles dans ce tutorial, et nous n'avons pas encore réalisé le sommaire !

Tenez-vous bien : nous n'allons pas encore nous y attaquer. Le sommaire général du site, c'est en effet la partie la plus difficile à réaliser graphiquement, et l'interface doit présenter les informations les plus importantes de manière cohérente (ni trop, ni pas assez ...). Avant de créer le sommaire, il faut donc déjà connaître tous les types d'information présents sur le site, et les différentes structures de navigation. Le sommaire général, c'est donc pour la fin ...

Nous allons donc commencer à réellement exploiter les informations de notre site pour créer des «sommaires» alternatifs. Les sites de jeux vidéo regorgent en effet de ces pages récapitulatives présentant les jeux selon différents critères : le calendrier des sorties, les meilleures notes ...

11.12.1 Le tableau des meilleures notes

Créons un couple de fichiers pour afficher la liste des meilleurs jeux.

La page d'appel sera «notes.php3», dont le contenu est :

```
<?
$fond = "notes";
$delais = 24 * 3600;

include ("inc-public.php3");
?>
```

Nous pouvons maintenant commencer le squelette, «notes.html» :

```
<html>
<title>[#NOM_SITE_SPIP] Les meilleurs jeux</title>
</head>

<body>
<blockquote>

<h2>Les meilleurs jeux</h2>

<BOUCLE_notes (MOTS) {type=Note}{par titre}{inverse}>
<h1>#TITRE/10</h1>

</BOUCLE_notes>

</blockquote>
</body>
</html>
```

Nous avons installé une première `BOUCLE_notes`, qui affiche tous les mots-clés de type «Note» (c'est-à-dire «01», «02», ..., «10») par ordre inversé («10» en haut de liste).

Affichons le titre des articles correspondant à chaque note (c'est-à-dire les articles associés au mot-clé renvoyé par la `BOUCLE_notes`) :

```
<BOUCLE_notes (MOTS) {type=Note}{par titre}{inverse}>

  <B_articles>
  <h1>#TITRE/10</h1>

  <BOUCLE_articles (ARTICLES) {id_mot}{par date}{inverse}>
    <li>#TITRE
  </BOUCLE_articles>

</BOUCLE_notes>
```

La seule subtilité : le mot (la note) est désormais affiché dans le *texte conditionnel avant* de la BOUCLE_articles. De ce façon, seules les notes qui ont été réellement attribuées à des articles sont affichées.

Nous affichons ici le titre de l'article. Ce n'est pas ce qui nous intéresse. Nous voulons afficher :

- ▷ le titre du jeu, c'est-à-dire le titre de la rubrique qui contient l'article ;
- ▷ la (les) machine(s) sur lesquelles tournent ces jeux ;
- ▷ la (les) date(s) de sortie du jeu.

Rien de bien compliqué, nous avons déjà réalisé ces opérations sur d'autres pages (pour la date de sortie, nous recopions directement ce que nous avons développé pour «`article.html`») :

```
<BOUCLE_notes (MOTS) {type=Note}{par titre}{inverse}>

  <B_articles>
  <h1>#TITRE/10</h1>
  <ul>

  <BOUCLE_articles (ARTICLES) {id_mot}{par date}{inverse}>
    <BOUCLE_rubrique_jeu (RUBRIQUES) {id_rubrique}>
    <li><b><a href="#URL_ARTICLE">#TITRE</a></b>
    </BOUCLE_rubrique_jeu>
    <BOUCLE_machines_jeu (MOTS) {id_article}>
    #LOGO_MOT
    </BOUCLE_machines_jeu>
    <BOUCLE_mac2 (MOTS) {id_article}{type=Machines}>
    <BOUCLE_sortie (ARTICLES) {id_rubrique}{titre_mot=Date_sortie}>
    <BOUCLE_verifier_mot (ARTICLES) {id_article}{id_mot}>
    <br> Date de sortie
      <BOUCLE_machine_sortie (MOTS) {id_article}{type=Machines}{", "}>#TITRE
    </BOUCLE_machine_sortie> :
    [<b>(#DATE|affdate)</b>]
    </BOUCLE_verifier_mot>
    </BOUCLE_sortie>
  </BOUCLE_mac2>

  </BOUCLE_articles>
  </ul>
  </B_articles>

</BOUCLE_notes>
```

Seule petite subtilité : la balise #URL_ARTICLE est utilisée dans une boucle de type RUBRIQUES, de façon en réaliser le lien vers la page du test plutôt que sur la page générale de la rubrique du jeu. Cette utilisation d'un élément propre

aux ARTICLES est possible ici, car l' #URL_ARTICLE est réalisé à partir de la variable «id_article». Cet id_article est fourni par la BOUCLE_articles, puis la BOUCLE_rubrique_jeu ne la modifie pas (puisqu'il s'agit d'une boucle de RUBRIQUES). C'est le même principe qui permet d'utiliser «id_mot» dans la BOUCLE_verifier_mot.

Dernier problème à régler : cette page fonctionne bien au lancement du site et pendant les essais, avec une dizaine d'articles, mais quand votre site aura un an et plusieurs milliers de jeux tests (au minimum !), la liste sera trop longue. De toute façon, afficher indifféremment des jeux sortis il y a une semaine et des jeux sortis depuis plusieurs mois n'a pas grand intérêt : un jeu noté «9/10» il y a un an serait certainement moins bien noté aujourd'hui. Introduisons donc une limitation de temps en modifiant la BOUCLE_articles :

```
<BOUCLE_articles (ARTICLES) {id_mot}{age<90}{par date}{inverse}>
```

{age<90}, c'est-à-dire les articles publiés depuis moins de 3 mois.

11.12.2 Le calendrier des sorties

Incontournable sur les pages de jeux vidéo, la page indiquant les sorties récentes et les prochaines sorties.

La page d'appel sera «sorties.php3» :

```
<?
$fond = "sorties";
$delais = 24 * 3600;

include ("inc-public.php3");
?>
```

Commençons le fichier «sorties.html» ainsi :

```
<html>
<title>[#NOM_SITE_SPIP] Les sorties</title>
</head>

<body>
<blockquote>

<h2>Les prochaines sorties</h2>

<BOUCLE_sorties (ARTICLES) {titre_mot=Date_sortie}{par date}{age < 0}>
<p> #TITRE - [(#DATE|affdate)]
</BOUCLE_sorties>

</blockquote>
</body>
</html>
```

Le #TITRE est celui de l'article contenant la date de sortie ; c'est un titre bidon utilisé par commodité dans l'espace privé. Il faut donc le remplacer par le titre du jeu, c'est-à-dire le titre de la rubrique qui contient cet article. Toujours le même processus. Nous voulons de plus afficher le logo des machines concernées par cette sortie. Là non plus rien de compliqué.


```
<BOUCLE_sorties(ARTICLES){titre_mot=Date_sortie}{par date}{age < 0}>
<p>
  <BOUCLE_rubrique_jeu(RUBRIQUES){id_rubrique}>
    <b><a href="#URL_RUBRIQUE">#TITRE</a></b>
  </BOUCLE_rubrique_jeu>

  <BOUCLE_machines_jeu(MOTS){id_article}>
    #LOGO_MOT
  </BOUCLE_machines_jeu>

  - [ (#DATE|affdate) ]
</BOUCLE_sorties>
```

Il y a ici un défaut de navigation important : il y a systématiquement un lien vers la rubrique du jeu. Or, et surtout pour les jeux qui ne sont pas encore sortis, il est très possible qu'il n'y ait aucune information publiée sur ce jeu, en dehors de sa date de sortie. On risque alors de pointer vers une rubrique qui ne présente aucune information.

Modifions donc encore ce code, pour afficher non plus un lien vers la rubrique elle-même, mais vers chaque type d'articles.

Afin d'obtenir une présentation un peu plus condensée, commençons par créer une interface sous forme de lignes et de colonnes (et au passage, nous supprimons le lien hypertexte fautif) :

```
<B_sorties>
<table cellpadding="4">
<BOUCLE_sorties(ARTICLES){titre_mot=Date_sortie}{par date}{age < 0}>
<tr>

  <BOUCLE_rubrique_jeu(RUBRIQUES){id_rubrique}>
    <td><b>#TITRE</b></td>
  </BOUCLE_rubrique_jeu>

  <td align="center">
    <BOUCLE_machines_jeu(MOTS){id_article}>
    #LOGO_MOT
  </BOUCLE_machines_jeu>
  &nbsp;
  </B_machines_jeu>
  </td>

  <td>[ (#DATE|affdate) ]</td>
</tr>
</BOUCLE_sorties>
</table>
</B_sorties>
```

Ce code ne présente pas de grosse difficulté. Expliquons rapidement le positionnement des balises du tableau.

- ▷ `<table>` et `</table>` sont placés en *texte optionnel* de la `BOUCLE_sorties` ; de cette façon, si la liste des jeux à venir est vide, on évite d'afficher un `<table></table>` peu élégant.
- ▷ Les `<tr>` et `</tr>` sont à l'intérieur de la `BOUCLE_sorties`, et englobent l'intégralité de son contenu ; il s'agit ici de créer les lignes du tableau.
- ▷ Dans la `BOUCLE_rubrique_jeu`, nous plaçons les `<td>` et `</td>` directement dans la boucle, autour du `#TITRE`. Nous savons en effet que, l'article étant forcément dans une rubrique, il y aura forcément une (et une seule)

réponse dans `BOUCLE_rubrique_jeu` (inutile de placer ces codes dans du *texte optionnel*; et comme il n'y aura qu'une seule réponse, on peut se placer à l'intérieur de la boucle).

▷ C'est différent pour la `BOUCLE_machines_jeu` : les `<td align="center">` et `</td>` sont à l'extérieur de la boucle, et ne sont même pas placés en *code optionnel*. En effet :

1. ils sont à l'extérieur de la boucle, car il peut y avoir plusieurs machines concernées par cette date de sortie (c'est-à-dire plusieurs mots-clés de machines associés à l'article); si on avait placé ces balises à l'intérieur de la boucle, comme précédemment, le tableau perdrait son alignement, puisqu'on créerait des «cases» supplémentaires à chaque nouveau logo de mot-clé;
2. ils ne sont pas placés dans du texte optionnel, car ce ne sont pas des balises optionnelles; si on a oublié d'indiquer une machine pour la date de sortie, il faut tout de même créer cette «case» du tableau, certes vide, pour conserver l'alignement. Enfin, nous avons placé un ` ` en *texte optionnel alternatif* : s'il n'y a pas de mot-clé de machine associé à la date de sortie, on affiche cet espace insécable pour que la «case» ait un contenu (dans de nombreux butineurs, une case `<td></td>` n'est pas affichée de la même façon qu'une case `<td> </td>`).

Nous allons maintenant afficher pour chaque jeu la liste des Tests qui concernent ce jeu (insérons cela juste avant le `</tr>` final de la ligne) :

```
<td>
  <B_tests_jeu>
  <b>Test :</b>
  <BOUCLE_tests_jeu(ARTICLES){id_rubrique}{titre_mot=test}>
    <br><a href="#URL_ARTICLE">#TITRE</a>
  </BOUCLE_tests_jeu>
  &nbsp;
</B_tests_jeu>
</td>
```

Cette `BOUCLE_tests_jeu` utilise des méthodes déjà expliquées.

Cependant, cela ne nous convient pas encore : nous affichons ici tous les tests, pour toutes les machines, alors que notre date de sortie ne concerne pas forcément toutes les versions du jeu (on peut ici afficher la date de sortie de la versions Dreamcast, alors qu'on a déjà publié des tests pour les versions Playstation, Dreamcast, Window ...).

On ne veut donc afficher que les tests qui concernent les versions concernées par cette date de sortie. Nous avons déjà réalisé un processus similaire dans la page «`article.html`» pour n'afficher que les dates de sortie sur les mêmes machines que l'article affiché (`BOUCLE_mac2`). Cela donne :

```
<td>
  <BOUCLE_mac2(MOTS){id_article}{type=Machines}>

  <BOUCLE_tests_jeu(ARTICLES){id_rubrique}{titre_mot=test}>
  <BOUCLE_verifier_mot(ARTICLES){id_article}{id_mot}>
    <li><a href="#URL_ARTICLE">#TITRE</a>
  </BOUCLE_verifier_mot>
  </BOUCLE_tests_jeu>

  </BOUCLE_mac2>
  &nbsp;
</td>
```

La `BOUCLE_mac2` récupère la liste des machines de la date de sortie. La `BOUCLE_tests_jeu` récupère l'ensemble des tests de jeu de la rubrique (toutes machines confondues). La `BOUCLE_verifier_mot` vérifie pour chaque test récupéré dans `BOUCLE_tests_jeu` qu'il est associé à la machine récupérée par `BOUCLE_mac2`.

Nous avons pour l'instant supprimé l'affichage de «Tests :» en texte conditionnel pour des raisons de lisibilité. Voici maintenant le code, exactement sur le même principe, qui permet d'afficher cette indication (ainsi que le):

```
<td>
  <B_mac2>
  <b>Tests :</b>
    <BOUCLE_mac2 (MOTS) {id_article}{type=Machines}><BOUCLE_tests_jeu (ARTICLES)
      {id_rubrique}{titre_mot=test}>
    <BOUCLE_verifier_mot (ARTICLES) {id_article}{id_mot}>
      <li><a href="#URL_ARTICLE">#TITRE</a>
    </BOUCLE_verifier_mot></BOUCLE_tests_jeu></BOUCLE_mac2>
  &nbsp;
</B_mac2>
</td>
```

Notez bien : les BOUCLE_mac2, BOUCLE_tests_jeu et BOUCLE_verifier_mot sont «collées», c'est-à-dire qu'elles ne sont plus séparées par des retours à la ligne ni par des espaces. En effet, nous testons le «contenu» de la BOUCLE_mac2 pour savoir si nous affichons l'indication «Tests :» ou le (les textes conditionnels). S'il nous avions conservé les retours à la ligne ou des espaces blancs entre les boucles, quel que soit le résultat de BOUCLE_tests_jeu et de BOUCLE_verifier_mot, la BOUCLE_mac2 aurait été considérée comme ayant un contenu à cause de l'affichage de ces retours-chariot.

Pour afficher une liste des previews, dupliquons simplement ce code, en modifiant le nom des boucles et en remplaçant titre_mot=test par titre_mot=preview :

```
<td>
  <B_mac3>
  <b>Previews :</b>
  <BOUCLE_mac3 (MOTS) {id_article}{type=Machines}><BOUCLE_preview_jeu (ARTICLES)
    {id_rubrique}{titre_mot=preview}>
  <BOUCLE_verifier_mot3 (ARTICLES) {id_article}{id_mot}>
    <li><a href="#URL_ARTICLE">#TITRE</a>
  </BOUCLE_verifier_mot3></BOUCLE_preview_jeu></BOUCLE_mac3>
  &nbsp;
</B_mac3>
</td>
```

Ajoutons enfin une boucle pour les «news» concernant ce jeu. Ici nous ne nous préoccupons pas des machines, puisque toutes les news sont affichées sur une unique page. Nous n'afficherons donc pas de #TITRE, mais directement la mention «Les news» ; le lien hypertexte pointe vers la page spécifique que nous avons créée pour ce jeu. Notez enfin la restriction {0,1} qui permet de ne récupérer qu'une seule «news» pour créer un unique lien.

```
<td>
  <BOUCLE_news_jeu (ARTICLES) {id_rubrique}{titre_mot=news}{0,1}>
  <br><a href="news_jeu.php3?id_rubrique=#ID_RUBRIQUE">Les news</a>
</BOUCLE_news_jeu>
  &nbsp;
</B_news_jeu>
</td>
```

Pour être exhaustif, il resterait à recommencer l'opération avec les Soluces et les Astuces. Mais puisqu'il s'agit ici d'afficher une liste de sorties, considérons que ces informations ne sont pas pertinentes ici. Même si, techniquement, on peut les afficher facilement, faisons un choix éditorial : l'information n'est pas intéressante ici et surchargerait l'interface, donc nous préférons ne pas l'indiquer. (Mais si cela vous amuse, vous pouvez vous entraîner en les intégrant vous-même.)

11.13 Une première version du sommaire

23 septembre 2001 par l'équipe de SPIP

Nous commençons à avoir une idée plus précise de la navigation de notre site. Réalisons une première version du sommaire.

```
<html>
<head>
<title>[#NOM_SITE_SPIP] Sommaire</title>
</head>
<body>

<center>
<a href="sorties.php3">Les prochaines sorties</a>
| <a href="notes.php3">Les meilleurs jeux du moment</a>
</center>

<BOUCLE_secteurs(RUBRIQUES){id_parent=0}{par titre}>
  <p><b><a href="#URL_RUBRIQUE">#TITRE</a></b>

  <B_sous_rubriques>
  <ul>
  <BOUCLE_sous_rubriques(RUBRIQUES){id_parent}>
    <li><a href="#URL_RUBRIQUE">#TITRE</a>
  </BOUCLE_sous_rubriques>
  </ul>
  </B_sous_rubriques>

</BOUCLE_secteurs>

</body>
</html>
```

Avant de placer les boucles qui permettront d'afficher les nouveautés du site, nous nous contentons de créer (à la main) les liens vers les deux sommaires alternatifs, et d'afficher la structure du site (uniquement les grandes rubriques et leurs sous-rubriques).

Installons la liste des cinq plus récents tests publiés sur le site. Pour chacun, nous afficherons (en plus du logo, du titre du test, et son descriptif) :

- ▷ une boucle (RUBRIQUES) pour récupérer le titre du jeu ;
- ▷ une boucle (MOTS) avec pour type de mot «note» pour afficher la note ;
- ▷ une seconde boucle (MOTS) avec pour type «machines» pour afficher le logo des machines concernées.

Rien que nous ne sachions déjà faire.

```
<BOUCLE_tests(ARTICLES){titre_mot=test}{par date}{inverse}{0,5}>
  <p><div style="border:1px solid black">
  [ (#LOGO_ARTICLE_RUBRIQUE|left|#URL_ARTICLE) ]

  <BOUCLE_rub_tests(RUBRIQUES){id_rubrique}>
  <h3>#TITRE</h3>
  </BOUCLE_rub_tests>

  <h4>#TITRE</h4>
```

```
<BOUCLE_note_tests (MOTS) {id_article} {type=Note}>
<b>NOTE : #TITRE/10</b><p>
</BOUCLE_note_tests>

<BOUCLE_mac_tests (MOTS) {id_article} {type=Machines}>
[ (#LOGO_MOT|left) ]
</BOUCLE_mac_tests>

[ (#DESCRIPTIF) ]
<p align="right"><a href="#URL_ARTICLE">Lire ce test...</a>
</div>
</BOUCLE_tests>
```

Ajoutons maintenant les cinq dernières previews, cinq soluces et cinq astuces. Les présentations sont différentes à chaque fois, mais le principe est rigoureusement similaire à la boucle précédente (dans des versions simplifiées, puisque nous voulons afficher moins d'informations) :

```
<B_previews>
<p><b>Previews :</b>
<ul>
<BOUCLE_previews (ARTICLES) {titre_mot=preview} {par date} {inverse} {0,5}>

  <BOUCLE_rub_previews (RUBRIQUES) {id_rubrique}>
  <li><b>#TITRE</b> /
  </BOUCLE_rub_previews>

  <a href="#URL_ARTICLE">#TITRE</a>

  <BOUCLE_mac_previews (MOTS) {id_article} {type=Machines}>
  [ (#LOGO_MOT) ]
  </BOUCLE_mac_previews>

</BOUCLE_previews>
</ul>
</B_previews>

<B_soluces>
<p><b>Soluces :</b>
<ul>
<BOUCLE_soluces (ARTICLES) {titre_mot=soluce} {par date} {inverse} {0,5}>

  <BOUCLE_rub_soluces (RUBRIQUES) {id_rubrique}>
  <li><b><a href="#URL_ARTICLE">#TITRE</a></b>
  </BOUCLE_rub_soluces>

  <BOUCLE_mac_soluces (MOTS) {id_article} {type=Machines}>
  (#TITRE)
  </BOUCLE_mac_soluces>

</BOUCLE_soluces>
</ul>
</B_soluces>

<B_astuces>
```

```
<p><b>Astuces :</b>
<ul>
<BOUCLE_astuces (ARTICLES) {titre_mot=astuces}{par date}{inverse}{0,5}>

  <BOUCLE_rub_astuces (RUBRIQUES) {id_rubrique}>
  <li><b><a href="#URL_ARTICLE">#TITRE</a></b>
  </BOUCLE_rub_astuces>

  <BOUCLE_mac_astuces (MOTS) {id_article}{type=Machines}>
  (#TITRE)
  </BOUCLE_mac_astuces>

</BOUCLE_astuces>
</ul>
</B_astuces>
```

Enfin, la liste des news. Toujours le même principe ; cette fois, le lien hypertexte pointe vers la page commune aux news d'un jeu. Le lien se fait sur le titre du jeu (c'est-à-dire le titre de la rubrique), et nous affichons le titre de la news.

```
<B_news>
<p><b>News :</b>
<ul>
<BOUCLE_news (ARTICLES) {titre_mot=news}{par date}{inverse}{0,5}>

  <BOUCLE_rub_news (RUBRIQUES) {id_rubrique}>
  <li><b><a href="news_jeu.php?id_rubrique=#ID_RUBRIQUE">#TITRE</a>:</b>
  </BOUCLE_rub_news>
  #TITRE

</BOUCLE_news>
</ul>
</B_news>
```

Il nous reste désormais à indiquer la structure thématique du site (par grand genre de jeux : Action/aventure, Plateforme, Sport ...). Là encore rien de bien compliqué :

```
<BOUCLE_secteurs (RUBRIQUES) {id_parent=0}{par titre}>
  <p><b><a href="#URL_RUBRIQUE">#TITRE</a></b>

  <B_sous_rubriques>
  <ul>
  <BOUCLE_sous_rubriques (RUBRIQUES) {id_parent}>
    <BOUCLE_art_sous (ARTICLES) {id_rubrique}>
    </BOUCLE_art_sous>
    <li><a href="#URL_RUBRIQUE">#TITRE</a>
    </B_art_sous>
  </BOUCLE_sous_rubriques>
  </ul>
  </B_sous_rubriques>

</BOUCLE_secteurs>
```

Notons tout de même cette subtilité : dans les sous-rubriques, la `BOUCLE_art_sous` vérifie la présence d'articles (cette boucle n'affiche rien). L'affichage du titre et du lien et du titre de la sous-rubrique n'est effectif que s'il n'y

pas de rubrique dans la sous-rubrique, puisque cet affichage est placé dans le *texte optionnel alternatif*. En effet, il ne faudrait pas afficher les rubriques de jeu placés directement dans une rubrique.

Notre sommaire est complet. Pas terminé pour autant (dans les articles suivants, nous étendrons encore ses possibilités), mais complet :

- ▷ la structure thématique du site par grande rubriques de type de jeux est présentée ;
- ▷ les tests récents (dont nous considérons - et c'est un choix éditorial - qu'ils constituent l'élément le plus important du site) sont affichés de manière très visible ;
- ▷ nous exploitons l'indication des machines ;
- ▷ nous exploitons les «types d'articles» (les news, previews, tests, soluces, astuces sont présentés séparément) ;
- ▷ la note des jeux est exploitée, ainsi que les dates des sorties ...

Tout cela, notez bien, sans utiliser la moindre ligne de code PHP, et sans modifier la structure de la base de données. Nous n'avons même pas détourné certaines indications des articles pour y arriver (on aurait pu par exemple décider que le surtitre des articles serait utilisé pour indiquer la machine, et le soustitre pour indiquer la note ; nous aurions ainsi pu afficher ces informations).

11.14 Un sommaire pour chaque machine

23 septembre 2001 par l'équipe de SPIP

Les amateurs de jeux vidéo possèdent généralement une seule machine (deux pour les accros, trois pour les collectionneurs ...). Une page de sommaire général, si elle se justifie à l'arrivée sur le site (le visiteur indique l'URL du site, on ne sait donc pas a priori ce qu'il utilise comme machine), contient trop d'informations superflues pour le propriétaire d'une seule machine : le possesseur d'une Playstation 2 n'est pas concerné par le dernier jeu sorti sur Dreamcast.

Donc, il faut pouvoir présenter un sommaire différent pour chaque machine. Un sommaire qui n'affiche que les jeux de cette machine.

Nous allons donc créer un squelette dont le contenu change en fonction de la machine. Nous nommerons ces fichiers : «`sommaire_machine.php3`» et «`sommaire_machine.html`». La machine étant indiquée, dans notre structure, par un mot-clé, ces pages dépendront donc d'un «`id_mot`». Chaque machine différente sera donc appelée par une URL du type :

- ▷ `sommaire_machine.php3?id_mot=23` (23 étant le numéro du mot-clé de la machine).

Revenons à notre sommaire général : nous allons y inclure les liens vers ces sommaires spécialisés ; la création automatique de ces liens (avec les «`id_mot`» correspondants) nous évitera de devoir effectuer les tests en cherchant les numéros (`id_mot`) de chaque machine. Dans «`sommaire.html`», insérons la boucle suivante :

```
<table cellpadding=5><tr>
<BOUCLE_somm_mac (MOTS) {type=Machines}{par titre}>
<td><a href="sommaire_machine.php3?id_mot=#ID_MOT">#LOGO_MOT</a></td>
</BOUCLE_somm_mac>
</tr></table>
```

Attention : si l'on a réalisé des logos avec survol, le lien ne fonctionnera pas correctement, car SPIP ajoutera une commande javascript. Le code ci-dessus ne fonctionne qu'à la condition que le logo de chaque mot ne soit constitué que d'un unique fichier (pas d'image de survol).

Le moyen le plus simple et le plus efficace d'éviter un tel désagrément consiste à ne pas utiliser directement `#LOGO_MOT`, qui construit l'intégralité de la balise HTML de l'image, éventuellement avec un petit code javascript

pour gérer le survol, mais à la place l'appel d'une image dont le fichier est : [(#LOGO_MOT|fichier)] (ce qui retourne uniquement le nom du fichier de l'image).

La boucle devient :

```
<table cellpadding=5><tr>
<BOUCLE_somm_mac(MOTS){type=Machines}{par titre}>
<td><a href="sommaire_machine.php3?id_mot=#ID_MOT">
    <img src='IMG/[(#LOGO_MOT|fichier)]' border=0></a>
</td>
</BOUCLE_somm_mac>
</tr></table>
```

11.14.1 Le fichier «`sommaire_machine.php3`»

Comme toujours :

```
<?
$fond = "sommaire_machine";
$delais = 24 * 3600;

include ("inc-public.php3");
?>
```

11.14.2 Le squelette du sommaire par machine

Inutile de nous compliquer la vie :

dupliquons le fichier «`sommaire.html`», et renommons-le «`sommaire_machine.html`».

Puisque cette page dépend d'un «`id_mot`», nous commençons par ajouter une `BOUCLE_principale` qui va contenir l'intégralité des autres boucles. Il suffit donc d'ajouter l'appel de cette boucle juste après `<body>` et de la refermer juste avant `</body>` (tout en fin de fichier) :

```
<body>
<BOUCLE_principale(MOTS){id_mot}>

    ...

</BOUCLE_principale>
</body>
```

Le titre de la page, qui jusqu'à présent ne contenait que : `<h1>#NOM_SITE_SPIP</h1>`, est modifié pour afficher le nom et le logo de la machine (c'est-à-dire du mot-clé correspondant à «`id_mot`») :

```
<h1>#NOM_SITE_SPIP / #LOGO_MOT #TITRE</h1>
```

La `BOUCLE_somm_mac`, qui affiche tous les logos des machines pour créer les liens vers les sommaires spécifiques, ne doit plus contenir le logo de la machine principale (nous sommes déjà sur cette page, inutile donc de créer un lien vers elle-même). Il suffit d'insérer le critère `{exclus}` dans la définition de cette boucle :


```
<BOUCLE_somm_mac (MOTS) {type=Machines}{exclus}{par titre}>
...
</BOUCLE_somm_mac>
```

La `BOUCLE_tests`, qui affiche les derniers tests publiés, ne doit désormais afficher que les tests qui concernent des jeux sur la machine sélectionnée. Pour cela, nous allons ajouter à l'intérieur de cette boucle une `BOUCLE_verif_test`, qui va vérifier pour chaque article qu'il concerne bien la machine choisie.

Inconvénient désormais : la `BOUCLE_tests` affichait les cinq derniers tests. Désormais, sur ces cinq articles, nous ne sélectionnons que ceux qui concernent notre machine. Il est fort probable que cela donne moins de cinq articles, et dans certains cas extrêmes, cela n'affichera plus aucun article. Modifions le critère, et décidons d'afficher non pas les cinq derniers tests (`{0,5}`), mais tous les tests publiés depuis moins de trois mois (`{age<90}`) ; on pourra modifier cette valeur en fonction de l'activité réelle du site (si le site est très actif, on réduira le délai de cette boucle, si le site publie peu d'articles, on pourra le rallonger).

Au passage, supprimons la petite boucle qui affichait le logo des machines concernées : cela devient inutile, puisque cette page n'affiche que les informations d'une unique machine.

La boucle devient :

```
<BOUCLE_tests (ARTICLES) {titre_mot=test}{par date}{inverse}{age < 90}>
<BOUCLE_verif_test (ARTICLES) {id_article}{id_mot}>
  <p><div style="border:1px solid black">

  [ {#LOGO_ARTICLE_RUBRIQUE|left|#URL_ARTICLE} ]

  <BOUCLE_rub_tests (RUBRIQUES) {id_rubrique}>
  <h3>#TITRE</h3>
  </BOUCLE_rub_tests>

  <h4>#TITRE</h4>

  <BOUCLE_note_tests (MOTS) {id_article}{type=Note}>
  <b>NOTE : #TITRE/10</b><p>
  </BOUCLE_note_tests>

  [ {#DESCRIPTIF} ]
  <p align="right"><a href="#URL_ARTICLE">Lire ce test...</a>
  </div>
</BOUCLE_verif_test>
</BOUCLE_tests>
```

Le principe est identique pour les previews, soluces, astuces, news : on ajoute une boucle de vérification, et on supprime la boucle qui affiche la liste des machines concernées :

```
<B_previews>
<p><b>Previews :</b>
<ul>
<BOUCLE_previews (ARTICLES) {titre_mot=preview}{par date}{inverse}{age < 90}>
<BOUCLE_verif_previews (ARTICLES) {id_article}{id_mot}>
  <BOUCLE_rub_previews (RUBRIQUES) {id_rubrique}>
  <li><b>#TITRE</b> /
  </BOUCLE_rub_previews>
  <a href="#URL_ARTICLE">#TITRE</a>
```

```

</BOUCLE_verif_previews></BOUCLE_previews>
</ul>
</B_previews>

<B_soluces>
<p><b>Soluces :</b>
<ul>
<BOUCLE_soluces(ARTICLES){titre_mot=soluce}{par date}{inverse}{age < 90}>
<BOUCLE_verif_soluces(ARTICLES){id_article}{id_mot}>
  <BOUCLE_rub_soluces(RUBRIQUES){id_rubrique}>
  <li><b><a href="#URL_ARTICLE">#TITRE</a></b>
  </BOUCLE_rub_soluces>
</BOUCLE_verif_soluces></BOUCLE_soluces>
</ul>
</B_soluces>

<B_astuces>
<p><b>Astuces :</b>
<ul>
<BOUCLE_astuces(ARTICLES){titre_mot=astuces}{par date}{inverse}{0,5}>
<BOUCLE_verif_astuces(ARTICLES){id_article}{id_mot}>
  <BOUCLE_rub_astuces(RUBRIQUES){id_rubrique}>
  <li><b><a href="#URL_ARTICLE">#TITRE</a></b>
  </BOUCLE_rub_astuces>
</BOUCLE_verif_astuces></BOUCLE_astuces>
</ul>
</B_astuces>

<B_news>
<p><b>News :</b>
<ul>
<BOUCLE_news(ARTICLES){titre_mot=news}{par date}{inverse}{0,5}>
<BOUCLE_verif_news(ARTICLES){id_article}{id_mot}>
  <BOUCLE_rub_news(RUBRIQUES){id_rubrique}>
  <li><b><a href="news_jeu.php?id_rubrique=#ID_RUBRIQUE">#TITRE</a>:</b>
  </BOUCLE_rub_news>
  #TITRE
</BOUCLE_verif_news></BOUCLE_news>
</ul>
</B_news>

```

Important. Vous noterez ci-dessus que la boucle de vérification est «collée» à la première boucle. Par exemple, entre `<BOUCLE_previews(ARTICLE) . . . >` et `<BOUCLE_verif_previews(ARTICLE) . . . >`, il n'y a ni espace ni retour à la ligne). En effet, le *texte conditionnel* avant de la première boucle (ici, la mention «Previews») est activé si la boucle contient au moins un caractère. S'il y a un espace ou un retour chariot avant l'appel de la boucle de vérification, alors même si la boucle n'affiche aucun article (car, depuis trois mois, on n'a publié aucun article de ce type pour cette machine), cet espace parasite activerait l'affichage du code conditionnel.

Voyons maintenant l'affichage des prochaines sorties. On peut décider de créer une nouvelle page des sorties, concernant uniquement la machine sélectionnée, mais nous préférons ici afficher directement ces sorties sur notre sommaire spécifique (les sorties n'étant pas si nombreuses par machines). En appliquant les mêmes méthodes que précédemment, on remplace le lien «Les prochaines sorties» par :

```

<B_sorties>
<b>Les sorties #TITRE du prochain mois:</b>
<ul>

```

```
<BOUCLE_sorties(ARTICLES){titre_mot=Date_sortie}{par date}{age < 0}>
<BOUCLE_sorties_verif(ARTICLES){id_article}{id_mot}>
  <li>[ (#DATE|affdate)] :

  <BOUCLE_rub_sorties(RUBRIQUES){id_rubrique}>
  #TITRE
  </BOUCLE_rub_sorties>

</BOUCLE_sorties_verif></BOUCLE_sorties>
</ul>
</B_sorties>
```

Sur le même principe, affichons la liste des jeux préférés pour cette machine depuis trois mois :

```
<B_notes>
<b>Nos jeux préférés depuis 6 mois:</b>
<ul>
<BOUCLE_notes(ARTICLES){type_mot=note}{par titre_mot}{inverse}{age<120}>
<BOUCLE_notes_verif(ARTICLES){id_article}{id_mot}>
  <BOUCLE_rub_notes(RUBRIQUES){id_rubrique}>
  <li> #TITRE
  </BOUCLE_rub_notes>
  <BOUCLE_lanote(MOTS){id_article}{type=note}>
  (#TITRE/10)
  </BOUCLE_lanote>

</BOUCLE_notes_verif></BOUCLE_notes>
</ul>
</B_notes>
```

La seule subtilité pour l'instant est le classement des articles associés aux mots-clés de type «note» selon le critère {par titre_mot}{inverse} (fonction apparue dans la version 1.3 de SPIP). En effet, les mots de type «note» sont de la forme «01», «02» ..., «10» ; on peut donc demander à les classer de 10 à 1.

Cependant, cette présentation ne nous convient pas encore : nous prétendons afficher nos «jeux préférés», alors qu'il s'agit seulement d'un classement par note. Si un jeu est sorti depuis moins de six mois et s'est ramassé une note catastrophique, il sera dans les «jeux préférés» (en bas de liste, mais présent tout de même) !

Nous devons donc trouver un moyen d'afficher uniquement les jeux ayant eu une note supérieure à «07» (choix arbitraire). Si vous pensez utiliser pour cela la `BOUCLE_notes`, en y ajoutant simplement la mention {titre_mot>07}, cela ne fonctionnera pas : SPIP n'accepte qu'un seul critère basé sur les mots-clés dans les boucles autres que les boucles de type (MOTS).

Nous allons utiliser la `BOUCLE_lanote` (qui affiche la note), et lui demander de ne sélectionner que les notes supérieures à 7 (puisque'il s'agit ici d'une boucle de type (MOTS), on peut lui fournir plusieurs contraintes sur les mots-clés). La `BOUCLE_rub_notes`, qui affiche le titre du jeu, est alors placé en *texte optionnel après* de la `BOUCLE_lanote` :

- ▷ si la note est supérieure à 7, la `BOUCLE_rub_notes` est effectuée ;
- ▷ sinon rien n'est affiché.

Attention : il est toujours délicat de placer une boucle dans le *texte optionnel* d'une autre boucle (SPIP peut se mélanger facilement les pinceaux). Pour une raison étrange, cela est possible dans le *texte optionnel après*, mais pas dans le *texte optionnel avant*.

Sachant que nous ne récupérerons que les articles notés «08», «09», «10», classés du meilleur au moins bon, nous considérons désormais inutile de rappeler la note elle-même. La `BOUCLE_lanote` est donc vide, elle ne sert plus qu'à activer ou désactiver la `BOUCLE_rub_notes`.

Enfin, puisque la `BOUCLE_lanote` devient un nouveau «filtre» appliqué à la sélection d'articles, il faut supprimer les blancs et les retours chariot autour de cette boucle, de façon à ne réellement activer l'expression «Nos jeux préférés» que si au moins un article est vraiment affiché.

Cela nous donne :

```
<B_notes>
<b>Nos jeux préférés depuis 6 mois:</b>
<ul>
<BOUCLE_notes (ARTICLES) {type_mot=note}{par titre_mot}{inverse}{age<120}>
<BOUCLE_notes_verif (ARTICLES) {id_article}{id_mot}><BOUCLE_lanote (MOTS)
      {id_article}{titre>07}{type=note}>
  </BOUCLE_lanote>
  <BOUCLE_rub_notes (RUBRIQUES) {id_rubrique}>
  <li> <a href="#URL_RUBRIQUE">#TITRE</a>
  </BOUCLE_rub_notes>
</B_lanote></BOUCLE_notes_verif></BOUCLE_notes>
</ul>
</B_notes>
```

Tout cela peut sembler un peu indigeste, mais le tout est que cela fonctionne, et toujours sans la moindre fonction PHP !

Maintenant que nos pages de sommaires spécifiques sont presque terminées, nous constatons, en passant de l'une à l'autre, que certaines pages sont vides. En effet, nous n'affichons ici que des informations de moins de trois mois (pour les tests, les news ...). Si une machine n'a été concernée par aucun article depuis trois mois, ou même si une machine n'a encore aucun article du tout (même si cela est prévu par un mot-clé, il est très possible que personne sur votre site n'ait encore écrit au sujet des jeux Linux).

Il faut donc modifier la `BOUCLE_somm_mac` (celle qui affiche les logos des différentes machines) pour que seules les machines ayant réellement des articles publiés depuis trois mois soient affichés. Il suffit pour cela d'ajouter une boucle vérifiant la présence d'article datés de moins de trois mois associés à ce mot-clé. Ce qui nous donne :

```
<BOUCLE_somm_mac (MOTS) {type=Machines}{exclus}{par titre}>
<BOUCLE_verif_somm (ARTICLES) {id_mot}{0,1}{age < 90}>
  <td>
  <a href="sommaire_machine.php3?id_mot=#ID_MOT">
    <img src='IMG/[ (#LOGO_MOT|fichier)]' border=0></a>
  </td>
</BOUCLE_verif_somm>
</BOUCLE_somm_mac>
```

On pensera à effectuer le même remplacement dans la page «`sommaire.html`» du sommaire général (penser à supprimer le critère `{exclus}` dans ce cas).

11.15 Référencer des articles sur le Web

23 septembre 2001 par l'équipe de SPIP

Monter son site consacré aux jeux vidéo, c'est bien ; mais d'autres y ont pensé avant vous, et une tripotée de sites existe déjà. Pour enrichir encore l'information présentée sur notre site, nous allons donc indiquer des articles consacrés aux jeux que nous présentons.

Cela se fait, dans l'espace privé, par le lien «Référencer un site Web» accessible depuis chaque rubrique. Notre structure, qui regroupe tous les articles consacrés à un jeu dans une rubrique dédiée, facilite l'opération ; il suffit de «référencer un site Web» dans la rubrique qui traite du jeu concerné par le lien.

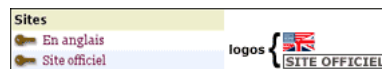
Tirons à nouveau profit de nos mots-clés. Pour chaque page référencée sur le Web, nous indiquerons :

- ▷ la machine concernée par cet article (la page référencée concerne une version Playstation, Xbox ...);
- ▷ le type d'article (test, preview ...); si la page n'entre dans aucune catégorie, nous n'indiquerons pas de type d'article ;
- ▷ éventuellement, la note attribuée au jeu sur ce site extérieur.

Afin d'enrichir encore l'information, nous souhaitons indiquer, le cas échéant :

- ▷ qu'il s'agit du site officiel du jeu ;
- ▷ que l'article est en anglais (vous pouvez prévoir d'autres langues si vous le souhaitez). En effet, certaines solutions de jeu circulent sur le Net en anglais bien avant d'apparaître en français.

Pour cela, créons un nouveau groupe de mots-clés, intitulé «Sites», et deux mots : «En anglais» et «Site officiel» (on pourra aussi créer «En japonais», «En italien» ...). Nous attribuons un petit logo pour chacun de ces mots-clés (ce qui permettra de créer une interface plus compacte). Ce qui nous donne :



Reprenons notre fichier «rubrique.html». Nous ajoutons la boucle suivante à la fin de la BOUCLE_les_articles :

```
<B_sites><p>SUR LE WEB :
<ul>
  <BOUCLE_sites (SYNDICATION) {id_rubrique}{doublons}>
    <li>[ (#LOGO_SITE|#URL_SITE) ] <a href="#URL_SITE">#NOM_SITE</a>

    <BOUCLE_note_sites (MOTS) {type=Note} {id_syndic}>
      #TITRE/10
    </BOUCLE_note_sites>
    <BOUCLE_type_sites (MOTS) {type=Sites} {id_syndic}>
      #LOGO_MOT
    </BOUCLE_type_sites>
    <BOUCLE_mac_sites (MOTS) {type=Machines} {id_syndic}>
      #LOGO_MOT
    </BOUCLE_mac_sites>
  </BOUCLE_sites>
</ul>
</B_sites>
```

La BOUCLE_sites affiche le logo et le nom des sites référencés. Pour chaque site, les trois boucles incluses affichent successivement :

- ▷ BOUCLE_note_sites : la note attribuée au jeu par cet article ;
- ▷ BOUCLE_type_sites : s'il s'agit du site officiel, ou si l'article est en anglais ;
- ▷ BOUCLE_mac_sites : la machine concernée par le site.

Vous noterez le critère {doublons} dans la BOUCLE_sites. Il interdit d'afficher des sites référencés qui auraient déjà été affichés. Pour l'instant, c'est inutile, puisque c'est le seul endroit de ce squelette où nous récupérons des sites référencés.

Complétons cependant notre processus. Puisque nous nous autorisons, si cela est pertinent, à indiquer pour chaque site référencé s'il s'agit d'un test, d'une preview, d'une soluce ..., exploitons cette information. Nous allons intégrer, à la suite de nos propres tests, soluces ... les liens vers les pages référencées de ce type.

Juste après la `BOUCLE_tests`, nous allons recopier la boucle précédente (en renommant les boucles), et légèrement la modifier, pour n'afficher que les sites référencés auxquels nous avons attribué le type «Test» :

```
<B_sites_tests><p>Des tests sur le Web :
  <ul>
<BOUCLE_sites_tests(SYNDICATION) {id_rubrique}{titre_mot=Test}{doublons}>
  <li>[ (#LOGO_SITE|#URL_SITE) ] <a href="#URL_SITE">#NOM_SITE</a>

  <BOUCLE_note_sites_tests (MOTS) {type=Note}{id_syndic}>
  #TITRE/10
  </BOUCLE_note_sites_tests>
  <BOUCLE_type_sites_tests (MOTS) {type=Sites}{id_syndic}>
  #LOGO_MOT
  </BOUCLE_type_sites_tests>
  <BOUCLE_mac_sites_tests (MOTS) {type=Machines}{id_syndic}>
  #LOGO_MOT
  </BOUCLE_mac_sites_tests>
</BOUCLE_sites_tests>
</ul>
</B_sites_tests>
```

Nous reproduisons cela pour les previews, les soluces et les astuces.

Le critère `{doublons}` devient utile : chaque type de sites étant affiché successivement (d'abord les tests, puis les previews ...), notre toute dernière boucle (`BOUCLE_sites`) n'affiche plus que les sites référencés qui ne sont d'aucun type. Par exemple, le site officiel d'un jeu, ou le site d'un fan du jeu, ne peut pas être décrit comme uniquement un test ou une solution ; ils seront donc affichés par la `BOUCLE_sites`.

Pour terminer l'exploitation des sites référencés, recopions ces nouvelles boucles dans `<article.html>` (où nous affichons déjà les autres articles concernant le même jeu).

11.16 Un forum pour chaque jeu

23 septembre 2001 par l'équipe de SPIP

Notre site n'autorise pas, pour l'instant, les visiteurs du site à participer. Or ce genre de site se prête particulièrement bien aux forums ouverts aux visiteurs :

- ▷ d'abord parce que sur ce genre de sujet, les utilisateurs aiment à donner leur avis ; les forums de jeux vidéo sont très fournis ...
- ▷ parce que c'est un élément indispensable au contenu éditorial du site : il est en effet impossible de prévoir à l'avance toutes les questions concernant un jeu (trucs, astuces, bloqué à tel niveau, comment reconfigurer la carte graphique RadeTI 2525 de son PC ...); les forums sont le lieu indispensable pour ce genre de questions/réponses.

Avant d'intégrer des forums sur notre site, il faut d'abord définir où et comment les intégrer.

Première question : où ?

Avec les squelettes fournis en standard avec SPIP, on a pris l'habitude d'installer les forums directement sous chaque *article*. Or, ici, notre structure éditoriale est totalement différente : tous les articles concernant un jeu sont

regroupés dans une *rubrique*. Nous pourrions certes continuer à installer les forums sous chaque article, mais cela serait sans doute peu efficace : plus qu'une réaction à un article précis, le visiteur voudra certainement discuter du jeu «en général».

Nous décidons (encore un choix éditorial arbitraire !) que les forums ne concerneront pas chaque article (chaque test, chaque preview, etc.), mais le jeu dans son ensemble. C'est-à-dire la rubrique du jeu. Dans chaque article, nous ferons un lien vers une page commune affichant le forum de la rubrique.

Deuxième question : comment ?

La présentation du forum est importante : faut-il tout afficher d'un seul coup (c'est toujours très pratique), ou bien afficher d'abord une liste des titres des messages, et une page spécifique pour chaque message.

Sur un site de jeux vidéo, nous pouvons prévoir des forums très actifs, avec de nombreux messages, certains très longs (certains visiteurs vont certainement poster la critique complète d'un jeu, une soluce interminable ...). Impossible dès lors d'afficher tout le contenu de tous les messages sur une seule page.

Nous choisissons donc de présenter d'abord la liste des messages, avec uniquement leur titre ; puis nous réalisons une seconde page affichant chaque message spécifique.

11.16.1 La page forum_jeu.php3

La page qui affichera la liste complète des messages d'un forum sera donc la page «forum_jeu.php3». Cette page est appelée avec un numéro de rubrique (puisque c'est la rubrique qui contient tous les articles du jeu).

Commençons par intégrer le lien vers cette page dans la navigation de notre site.

Sur la page «article.html», ajoutons le lien vers le forum du jeu. Insérons (par exemple après l'affichage des #NOTES) :

```
<p><b><a href="forum_jeu.php3?id_rubrique=#ID_RUBRIQUE">
    Le forum de ce jeu
</a></b>
```

Contentons-nous pour l'instant de ce seul lien de navigation. En effet, pour réaliser un affichage plus complet (nombre de messages notamment), nous avons besoin de créer des messages de forum afin de tester notre interface, et pour l'instant ces forums sont totalement vides.

Créons donc notre squelette des forums ...

Commençons, comme d'habitude, par le fichier d'appel : «forum_jeu.php3» :

```
<?
$fond = "forum_jeu";
$delais = 24 * 3600;

include ("inc-public.php3");
?>
```

11.16.2 Le squelette forum_jeu.html

Créons maintenant le fichier «forum_jeu.html». Pour l'instant très simple :

```
<html>
<title>[#NOM_SITE_SPIP]
<BOUCLE_titre(RUBRIQUES){id_rubrique}>#TITRE</BOUCLE_titre></title>
</head>

<body>

<a href="index.php3">#NOM_SITE_SPIP</a>

<blockquote>
<BOUCLE_principale(RUBRIQUES){id_rubrique}>

  <BOUCLE_hierarchie(HIERARCHIE){" : ">
  <a href="#URL_RUBRIQUE">#TITRE</a>
  </BOUCLE_hierarchie>

  <h1>FORUM: <a href="#URL_RUBRIQUE">#TITRE</a></h1>

</BOUCLE_principale>
</blockquote>
</body>
</html>
```

Ce premier squelette très simple affiche :

- ▷ le retour à la page d'accueil du site ;
- ▷ la hiérarchie menant à ce jeu ;
- ▷ le titre du jeu.

Grâce à la BOUCLE_principale, nous sommes bien dans la *rubrique* qui concerne ce jeu.

Pour l'instant, notre forum est totalement vide. Avant de créer l'affichage prévu (afficher uniquement la liste des messages), créons une interface permettant de créer un forum complet. Grâce à lui, nous pourrions «alimenter» notre forum, et ainsi créer l'interface ensuite.

Insérons le lien qui nous permettra de poster notre premier message :

```
<h3>[<a href="forum.php3?(#PARAMETRES_FORUM)">Poster un message</a>]</h3>
```

Automatiquement,

#PARAMETRES_FORUM indiquera qu'il s'agit de messages liés à une rubrique, puisque notre BOUCLE_principale est bien de type (RUBRIQUES).

Affichons le titre de ces premiers messages :

```
<B_thread>
  <ul>
  <BOUCLE_thread(FORUMS){id_rubrique}{par date}{inverse}>
<li>#TITRE
  </BOUCLE_thread>
</ul>
</B_thread>
```


Ajoutons la possibilité de répondre à ces messages :

```
<B_thread>
  <ul>
  <BOUCLE_thread(FORUMS) {id_rubrique}{par date}{inverse}>
  <p><li>#TITRE
  <p align="right"><a href="forum.php3?#PARAMETRES_FORUM">
    répondre à ce message
    </a>
  </BOUCLE_thread>
  </ul>
</B_thread>
```

De cette façon, nous pouvons répondre aux messages de «premier» niveau (ce qu'on nomme les «*threads*», c'est-à-dire les messages qui lancent une nouvelle discussion), mais nous ne les affichons pas.

Affichons les réponses :

```
<B_thread>
  <ul>
  <BOUCLE_thread(FORUMS) {id_rubrique}{par date}{inverse}>
  <p><li>#TITRE

  <p align="right"><a href="forum.php3?#PARAMETRES_FORUM">
    répondre à ce message</a>

  <B_reponses>
  <ul>
  <BOUCLE_reponses(FORUMS) {id_parent}{par date}>
  <p><li>#TITRE
  <a href="forum.php3?#PARAMETRES_FORUM">répondre à ce message</a>
  </BOUCLE_reponses>
  </ul>
  </B_reponses>

  </BOUCLE_thread>
  </ul>
</B_thread>
```

Nous affichons désormais les réponses aux threads, il est possible de répondre à ces réponses, mais sans affichage. Pour afficher ces réponses, nous allons désormais ajouter une boucle *récursive*, c'est-à-dire que la `BOUCLE_reponses` va s'appeler elle-même, ce qui nous permettra d'afficher d'un seul coup l'intégralité des messages.

Modifions la `BOUCLE_reponses` ainsi :

```
<B_reponses>
  <ul>
  <BOUCLE_reponses(FORUMS) {id_parent}{par date}>
  <p><li>#TITRE
  <a href="forum.php3?#PARAMETRES_FORUM">répondre à ce message</a>

  <BOUCLE_rep_messages(boucle_reponses)></BOUCLE_rep_messages>

  </BOUCLE_reponses>
```

```
</ul>
</B_reponses>
```

Nous avons ajouté à l'intérieur de la `BOUCLE_reponses` une `BOUCLE_rep_messages`, qui appelle la `BOUCLE_reponses`. Ce qui permet d'afficher les réponses aux messages de la `BOUCLE_reponses`, de manière récursive (on exécute cette boucle jusqu'à ce qu'il n'y ait plus aucune réponse).

Arrivé à ce stade, pour les besoins de notre développement, il faut faire une pause, et consacrer un peu de temps à créer de nombreux messages dans un forum, avec plusieurs fils de discussion (*threads*), des réponses à des réponses ... de façon à simuler un forum complet. Cela nous permettra de créer plus logiquement notre interface graphique.

11.16.3 On recommence ...

Après cette étape, pendant laquelle nous avons alimenté notre forum avec une tripotée de messages, nous allons recommencer notre mise en page, cette fois dans l'optique de ce que nous avons prévu au départ : la page «forum_jeu» n'affichera que les titres des messages, et une autre page affichera chaque message.

Comme nos boucles précédentes n'affichaient pas d'informations, les modifications sont peu importantes :

- ▷ il faut afficher le `#TITRE` en lien hypertexte vers la page de ce message (nous décidons que nous utiliserons une page intitulée «message.php3» pour cela) ;
- ▷ il faut afficher le `#NOM` de l'auteur du message, et la `#DATE` d'envoi du message ;
- ▷ il faut supprimer la mention «répondre à ce message».

Nous en profitons pour modifier la présentation graphique. Les `` et `` suffisent pour présenter la structure logique, mais graphiquement ça n'est pas très joli. Nous les remplaçons donc par quelques feuilles de style très simples.

Finalement, notre code devient :

```
<BOUCLE_thread(FORUMS){id_rubrique}{par date}{inverse}>
  <p>
  <div style="border: solid black 1px; background-color: #CCCCCC;
                padding: 3px; width:100%">
  <a href="message.php3?id_forum=#ID_FORUM">#TITRE</a>
  [| (#NOM) ] | [(#DATE|affdate)]
  </div>

  <B_reponses>
  <div style="margin-left: 15px">
  <BOUCLE_reponses(FORUMS){id_parent}{par date}>
    <div style="border-left: solid black 1px; border-bottom: solid black 1px;
                border-right: solid black 1px; padding: 3px; width:100%">
    <a href="message.php3?id_forum=#ID_FORUM">#TITRE</a>
    [| (#NOM) ] | [(#DATE|affdate)]
    </div>

    <BOUCLE_rep_messages(boucle_reponses)></BOUCLE_rep_messages>

  </BOUCLE_reponses>
  </div>
  </B_reponses>

</BOUCLE_thread>
```

Remarque sur les styles. Nous avons inséré ici les styles directement dans la déclaration des `<div>`. On pourra préférer les regrouper dans des feuilles de style en début de fichier. Cependant, pendant le développement de la page, il est toujours pratique de pouvoir les modifier directement à l'endroit où se trouve l'information. Par ailleurs, l'usage des styles pose des difficultés de compatibilité avec Netscape 4.7 : non seulement celui-ci n'utilise pas l'intégralité des styles, mais en plus certains styles provoquent des bugs d'affichage. Dans l'exemple ci-dessus, nous avons choisi des styles qui ne provoquent pas ces bugs.

Notre page d'affichage de tous les messages d'un forum est terminée. Dans le cadre d'un véritable site, on gagnerait à afficher de plus les liens vers les articles de cette rubrique (les tests, les previews, un rappel des dates de sortie) ; comme cela a déjà été réalisé par ailleurs, nous laissons ce soin au lecteur.

11.16.4 Afficher chaque message : `message.php/message.html`

Il nous faut désormais afficher le texte de chaque message, avec la possibilité d'y répondre. Comme nous avons créé les liens sur la page précédente, nous savons que cela se fera avec un couple de fichiers : `«message.php3»` et `«message.html»`, appelés par un `«id_forum»` (`id_forum` étant le numéro de chaque message du forum).

Le fichier `«message.php3»` :

```
<?
$fond = "message";
$delais = 24 * 3600;

include ("inc-public.php3");
?>
```

Le fichier `«message.html»` contient le squelette :

```
<html>
<title>[#NOM_SITE_SPIP]
<BOUCLE_titre (FORUMS) {id_forum}>#TITRE</BOUCLE_titre></title>
</head>

<body>

<a href="index.php3">#NOM_SITE_SPIP</a>

<blockquote>
<BOUCLE_principale (FORUMS) {id_forum}>

  <h3>#TITRE</h3>
  [(#DATE|affdate)][, par <A HREF="mailto:#EMAIL">(#NOM)</A>]
  [<BR><A HREF="#URL_SITE">(#NOM_SITE)</A>]

  <p>#TEXTE

</BOUCLE_principale>
</blockquote>
</body>
</html>
```

Cette première version affiche :

▷ le titre de la page (dans `<title>...</title>`);

- ▷ le lien vers la page d'accueil du site ;
- ▷ le titre, le texte, la date, l'auteur et un site Web de ce message.

Du côté de la navigation dans le site, tout reste à faire ... Nous allons afficher la mention du jeu concerné par ce message. Dans la `BOUCLE_principale`, juste avant le `#TITRE` du message, ajoutons l'indication du titre du jeu (avec un lien vers la rubrique du jeu) et un lien vers la page affichant tous les messages :

```
<BOUCLE_larubrique (RUBRIQUES) {id_rubrique}>
  <h3>forum du jeu: <a href="#URL_RUBRIQUE">#TITRE</a></h3>
  afficher <a href="forum_jeu.php3?id_rubrique=#ID_RUBRIQUE">
    tous les messages</a>
</BOUCLE_larubrique>
```

Nous recopions notre traditionnelle `BOUCLE_hierarchie` permettant d'indiquer la hiérarchie des styles de jeux :

```
<BOUCLE_larubrique (RUBRIQUES) {id_rubrique}>
  <BOUCLE_hierarchie (HIERARCHIE) {" : ">
  <a href="#URL_RUBRIQUE">#TITRE</a>
</BOUCLE_hierarchie>
  <h3>forum du jeu: <a href="#URL_RUBRIQUE">#TITRE</a></h3>
  afficher <a href="forum_jeu.php3?id_rubrique=#ID_RUBRIQUE">
    tous les messages</a>
</BOUCLE_larubrique>
```

Nous désirons maintenant afficher la hiérarchie des messages qui mènent à notre message (s'il s'agit d'une réponse à un autre message, nous voulons afficher le lien vers ce message «parent»). Nous insérons (après le lien «afficher tous les messages», et avant le `#TITRE` du message actuel) :

```
<BOUCLE_parents (FORUMS) {id_enfant}>
  <br><a href="message.php3?id_forum=#ID_FORUM">#TITRE</a>
  [| (#NOM)] | [(#DATE|affdate)]
</BOUCLE_parents>
```

Note : le critère `{id_enfant}` appliqué aux boucles `(FORUMS)` a été introduit dans la version 1.3 de SPIP. Ce critère permet d'appeler le message auquel le message actuel répond. (Un critère similaire existe pour les boucles `(RUBRIQUES)`, nous l'utilisons d'ailleurs dans «`article.html`» pour «remonter d'un cran» dans la hiérarchie des rubriques. Curieusement, nous avons oublié ce critère dans les boucles des forums ...)

Nous affichons ici le message auquel le message de la `BOUCLE_principale` répond. Cela ne nous suffit pas, nous voulons afficher toute la succession des messages jusqu'à notre message principal (si le message affiché grâce à la `BOUCLE_parents` est lui-même une réponse, nous voulons encore remonter d'un cran, et ainsi de suite jusqu'au premier message du *thread*).

Pour cela nous utilisons tout simplement une boucle récursive, comme nous l'avons fait avec la `BOUCLE_reponses` de «`forum_jeu.html`». Nous inversons cependant la logique, puisqu'au lieu de «descendre» la hiérarchie des messages, nous la «remontons». Pour que la présentation soit cohérente, nous plaçons l'appel de la boucle récursive *avant* l'affichage du `#TITRE` du message (les message «parents» doivent s'afficher avant le titre de leur réponse) :

```
<BOUCLE_parents (FORUMS) {id_enfant}>
  <BOUCLE_par_parents (boucle_parents)></BOUCLE_par_parents>
  <br><a href="message.php3?id_forum=#ID_FORUM">#TITRE</a>
  [| (#NOM)] | [(#DATE|affdate)]
</BOUCLE_parents>
```

Cette boucle suffit à afficher la navigation et la structure logique dans les messages. Cependant, graphiquement cela ne nous convient pas. En effet, pour bien marquer le fait qu'il s'agit d'un enchaînement de réponses successives, nous voulons rétablir le petit décalage vers la droite à chaque réponse (comme sur la page «forum_jeu.html»).

Nous allons donc utiliser un style provoquant, à chaque appel de la boucle récursive, un décalage vers la gauche (de 15 points, puisque c'est le décalage utilisé dans la page «forum_jeu.html»). Nous en profitons pour afficher la #TITRE du message dans la même style que sur cette page (liseret à gauche et en dessous) :

```
<B_parents>
  <div style="margin-left:-15px">
    <BOUCLE_parents (FORUMS) {id_enfant}>
      <BOUCLE_par_parents (boucle_parents)></BOUCLE_par_parents>

      <div style="border-left: solid black 1px; border-bottom: solid black 1px;
                border-right: solid black 1px; padding: 3px; width:100%">
        <a href="message.php?id_forum=#ID_FORUM">#TITRE</a>
        [| (#NOM)] | [(#DATE|affdate)]
      </div>

    </BOUCLE_parents>
  </div>
</B_parents>
```

L'affichage devient plus cohérent, avec le décalage vers la gauche et les liserets, mais cela n'est pas encore parfait : ce décalage rentre dans la marge gauche de la page, puisque l'on décale vers la gauche, sans avoir auparavant décalé l'ensemble de la page vers la droite (si l'on considère une valeur de marge, on peut ainsi considérer que la marge gauche devient «négative» : au départ une marge égale à 0, puis -15, puis -30 ...).

Nous allons donc devoir décaler l'ensemble vers la droite.

Nous ne pouvons pas le faire à l'intérieur de la BOUCLE_parents, puisque nous décalerions, à chaque récursivité, une fois vers la gauche, et immédiatement une fois vers la droite ; les décalages d'annuleraient à chaque fois.

Nous allons donc créer une nouvelle boucle, avant la boucle d'affichage des parents, dont le seul but sera d'afficher des <div> destinés à provoquer l'affichage vers la droite.

L'ensemble de la page sera décalé vers la droite (s'il y a trois messages avant d'arriver au message principal, tout sera décalé de 45 points vers la droite). Les messages parents partiront donc de cette valeur pour se décaler vers la gauche : le premier message parent sera décalé vers la gauche, avec donc une marge global de 30 points vers la droite (45 - 15), le second sera à 15 points, puis 0 points.

Notre code devient :

```
<BOUCLE_decaler_droite (FORUMS) {id_enfant}>
  <div style="margin-left: 15px">
    <BOUCLE_rec_decaler_droite (boucle_decaler_droite)>
      </BOUCLE_rec_decaler_droite>
    </BOUCLE_decaler_droite>

  <B_parents>
    <div style="margin-left:-15px">
      <BOUCLE_parents (FORUMS) {id_enfant}>
        <BOUCLE_par_parents (boucle_parents)></BOUCLE_par_parents>

        <div style="border-left: solid black 1px; border-bottom: solid black 1px;
                  border-right: solid black 1px; padding: 3px;
                  background-color: #CCCCCC; width:100%">
          <a href="message.php?id_forum=#ID_FORUM">#TITRE</a>
```

```
[| (#NOM)] | [(#DATE|affdate)]
</div>

</BOUCLE_parents>
</div>
</B_parents>
```

La `BOUCLE_decaler_droite` est encore une boucle récursive, similaire à celle destinée à afficher les messages parents. La différence est ici dans le contenu : à chaque message parent, au lieu d'afficher le titre du message, on affiche uniquement un `<div>` provoquant le décalage vers la droite.

Puisque nous avons provoqué un décalage vers la droite, il faut ensuite réaliser l'opération inverse en fermant tous ces `<div...>`. Cela se fait par la même boucle, contenant cette fois les balises `</div>`. Mais où placer cette boucle ? Il nous semble logique d'afficher le texte du message principal avec un décalage vers la droite : il apparaîtra ainsi très clairement comme une réponse au dernier message affiché par la `BOUCLE_parents`.

C'est donc juste après la `#TEXTE` du message principal que nous plaçons le code suivant :

```
<BOUCLE_decaler_gauche (FORUMS) {id_enfant}>
</div>
  <BOUCLE_rec_decaler_gauche (boucle_decaler_gauche)>
  </BOUCLE_rec_decaler_gauche>
</BOUCLE_decaler_gauche>
```

Nous voulons maintenant afficher les réponses au message principal. De cette façon, nous pourrions nous seulement «remonter» la hiérarchie (avec la `BOUCLE_parents`), mais aussi la «descendre».

Pour cela, il nous suffit de recopier la `BOUCLE_reponses` récursive de la page «forum_jeu.html». Nous insérons ce code juste après le `#TEXTE` du message principal, et avant le décalage vers la gauche (il est logique que ces réponses soient décalées à droite du message principal) :

```
<B_reponses>
<div style="margin-left: 15px">
<BOUCLE_reponses (FORUMS) {id_parent}{par date}>
  <div style="border-left: solid black 1px; border-bottom: solid black 1px;
            border-right: solid black 1px; padding: 3px; width:100%">
  <a href="message.php3?id_forum=#ID_FORUM">#TITRE</a>
  [| (#NOM)] | [(#DATE|affdate)]
  </div>

  <BOUCLE_rep_messages (boucle_reponses)></BOUCLE_rep_messages>

</BOUCLE_reponses>
</div>
</B_reponses>
```

Enfin il nous reste à afficher les réponses situées au «même niveau» hiérarchique que le message principal (c'est-à-dire : si notre message principal est une réponse à un message, nous voulons afficher les liens vers les autres réponses à ce message).

Affichons toutes les autres réponses, en insérant ce code juste avant le `#TITRE` de notre message principal :

```
<BOUCLE_niveau (FORUMS) {meme_parent}{par date}{exclus}>
  <div style="border-left: solid black 1px; border-bottom: solid black 1px;
```

```
border-right: solid black 1px; padding: 3px; width:100%">
<a href="message.php3?id_forum=#ID_FORUM">#TITRE</a>
[| (#NOM)] | [(#DATE|affdate)]
</div>
</BOUCLE_niveau>
```

Avec le décalage que nous avons créé précédemment, tout cela est du plus bel effet ...

Améliorons encore notre présentation : nous allons afficher au dessus de notre message principal uniquement les messages postés *avant* ce message. Pour cela, nous ajoutons un critère à notre BOUCLE_niveau :

```
<BOUCLE_niveau(FORUMS){meme_parent}{par date}{age_relatif>0}{exclus}>
```

Le critère `{age_relatif}` (apparu dans SPIP 1.3) est similaire au critère habituel `{age}`. Mais, là où `{age}` compare la date de l'élément (article, message de forum, etc.) à la date actuelle (aujourd'hui), `{age_relatif}` compare cette date à la date de l'élément courant : dans notre cas, puisque nous sommes dans la BOUCLE_principale qui affiche le message principal, nous sélectionnons les autres messages en comparant leur date à la date du message principal. Avec `{age_relatif>0}`, nous récupérons donc uniquement les messages plus anciens que le message principal.

Fort logiquement,

nous créons une boucle avec le critère d'âge relatif opposé, après les messages de BOUCLE_reponses.

```
<BOUCLE_niveau_apres(FORUMS){meme_parent}{par date}{age_relatif<=0}{exclus}>
  <div style="border-left: solid black 1px; border-bottom: solid black 1px;
border-right: solid black 1px; padding: 3px; width:100%">
  <a href="message.php3?id_forum=#ID_FORUM">#TITRE</a>
  [| (#NOM)] | [(#DATE|affdate)]
  </div>
</BOUCLE_niveau_apres>
```

Il ne nous reste plus qu'à retravailler un peu la présentation du message principal (nous le plaçons lui aussi à l'intérieur d'un liseret noir), et à ajouter le lien «Répondre à ce message» :

```
<div style="border-left: solid black 1px; border-bottom: solid black 1px;
border-right: solid black 1px; padding: 3px; width:100%">
  <h2>#TITRE</h2>
  [(#DATE|affdate)]|, par <A HREF="mailto:#EMAIL">(#NOM)</A>]
  [<BR><A HREF="#URL_SITE">(#NOM_SITE)</A>]

  <p>#TEXTE
  <p align="right"><b>[<a href="forum.php3?(#PARAMETRES_FORUM)">
Répondre à ce message</a>]</b>
</div>
```



Nos deux squelettes permettant de gérer les forums d'un jeu sont terminés. Tout cela peut paraître un peu compliqué, mais il faut bien considérer que nous avons décidé de figoler la présentation de la structure. Nous aurions pu nous contenter d'une interface de navigation beaucoup plus spartiate.

11.16.5 Les liens depuis les autres pages

Reprenons le référencement de ce forum dans les autres pages du site.

Dans la page «`article.html`», modifions notre lien et ajoutons une boucle :

```
<p><b><a href="forum_jeu.php3?id_rubrique=#ID_RUBRIQUE">
    Le forum de ce jeu
  </a></b>
  <BOUCLE_forum(FORUMS) {id_rubrique} {plat}>
  </BOUCLE_forum>
  (#TOTAL_BOUCLE messages)</B_forum>
```

La `BOUCLE_forum` n'affiche rien, nous n'utiliserons que le *texte conditionnel après*. Le critère `{plat}` de cette boucle de forum indique que nous voulons récupérer tous les messages du forum (sans ce critère, nous ne récupérerions que les *threads* ; ici nous récupérerons même les messages qui sont des réponses à d'autres messages). Cette boucle (qui n'affiche rien) effectuée, nous pouvons afficher en texte optionnel le nombre total de résultats, grâce à `#TOTAL_BOUCLE`, c'est-à-dire le nombre de messages dans le forum.

Dans «`rubrique.html`», cela devient un peu plus compliqué. En effet, de la même manière que nous avons établi une différence de présentation entre les rubriques de navigation (par grandes catégories de jeux) et les rubriques de jeux (contenant des articles), nous ne voulons afficher un lien vers un forum de discussion que s'il s'agit bien de la rubrique d'un jeu. Pour cela, nous créons une boucle qui va tester la présence d'au moins un article dans la rubrique, et notre lien vers le forum ne sera affiché que si cette boucle (`BOUCLE_test_jeu`) contient au moins un élément. Ce qui nous donne, inséré après la `BOUCLE_sites` et juste avant la fin de la `BOUCLE_les_articles` :

```
<BOUCLE_test_jeu(ARTICLES) {id_rubrique} {0,1}>
  </BOUCLE_test_jeu>
  <p><b><a href="forum_jeu.php3?id_rubrique=#ID_RUBRIQUE">
    Le forum de ce jeu
```



```
</a></b>  
</B_test_jeu>
```

Nous supprimons ici l’affichage du nombre de messages : nous sommes ici à l’intérieur du texte optionnel de la BOUCLE_test_jeu, et l’insertion d’une nouvelle boucle à cet endroit est toujours déconseillée (notamment, la valeur #TOTAL_BOUCLE serait celle de la boucle BOUCLE_test_jeu, et non plus le nombre de messages dans le forum).

Evidemment, on pourra compléter ce processus en ajoutant ces liens vers les forums sur d’autres pages. Par exemple le tableau des annonces des sorties de jeux.

11.17 Le site complet

23 septembre 2001 par l’équipe de SPIP

Il y aurait encore beaucoup à faire pour enrichir la navigation sur notre site, uniquement en utilisant les boucles de SPIP. Mais ce tutorial est déjà bien assez long ... nous en resterons donc là.

L’intégralité des fichiers réalisés dans ce tutorial sont disponibles dans nos archives³. Ces fichiers contiennent quelques petits éléments de mise en page supplémentaires (des tableaux pour les sommaires), mais rien de fondamentalement complexe.

Si vous voulez bidouiller à partir des exemples fournis ici, n’hésitez pas, il reste de nombreuses possibilités de navigation inexploitées dans notre site :

- ▷ vous pouvez créer des pages de sommaire par type d’article : toutes les dernières previews, tous les derniers tests, toutes les soluces ...
- ▷ vous pouvez vouloir créer des «événements», c’est-à-dire des articles que vous voulez mettre particulièrement en valeur sur la page d’accueil (ces articles n’étant finalement remplacés que par un «événement» suivant). Pour cela, vous pouvez créer un nouveau mot-clé, et effectuer des sélections supplémentaires (notamment sur les pages de sommaire) en fonction de ce mot-clé ;
- ▷ vous pouvez créer de nouveaux types d’articles (par exemple les «produits dérivés» tels que les figurines, les films adaptés des films ...)
- ▷ vous pouvez créer de nouvelles grandes rubriques, destinées à recevoir un contenu éditorial différent, tel que des analyses générales sur les évolutions des machines («faut-il encore acheter une Dreamcast?», «la Xbox est-elle une bonne machine?» ...), des comparatifs ... là encore l’utilisation des mots-clés permettra d’enrichir la structure éditoriale de votre site ; ils suffira de créer des squelettes localisés à ces nouvelles rubriques pour obtenir des interfaces adaptées à ces nouveaux besoins ... il sera de plus intéressant d’intégrer l’affichage de ces nouveaux types d’articles dans votre sommaire, en exploitant ou non les mots-clés des machines (par exemple, un article «faut-il acheter une Dreamcast?» serait affiché dans le sommaire spécifique de cette machine) ;
- ▷ sur la page d’un article, vous pouvez améliorer l’affichage des jeux de même genre (les autres jeux de survival horror, les autres jeux de plateforme) : vous pouvez limiter cet affichage qu’aux jeux ayant reçu une bonne note, ou sortis depuis moins de six mois ;
- ▷ vous pouvez réaliser une navigation de rubrique en rubrique en n’affichant que ce qui concerne une unique machine (attention, cela devient particulièrement complexe à réaliser).

Comme vous pouvez le constater, il y a beaucoup à faire avec les squelettes de SPIP. C’est même le point important à retenir de ce tutorial : *pour exploiter pleinement les possibilités de SPIP, il faut réaliser ses propres squelettes*. On peut dès lors considérer les squelettes fournis en standard avec SPIP comme un pis-aller : afin que chacun puisse commencer à exploiter immédiatement un site avec SPIP, nous fournissons des squelettes ; mais l’intérêt de SPIP, justement, réside dans la réalisation de ses propres squelettes, adaptés à la structure éditoriale dont on a besoin.

³http://rezo.net/spip-dev/SQUELETTES/tutorial_avance/

Cinquième partie

Aide en ligne



Chapitre 12

Aide en ligne



12.1 Installation de SPIP

12.1.1 Régler les droits d'accès

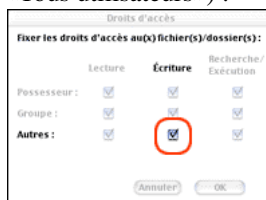
Lorsque vous installez, par FTP, les fichiers de SPIP sur votre propre serveur, certains **répertoires** ne sont pas configurés correctement : vous devez donc intervenir sur ces répertoires, avec votre logiciel de FTP habituel, pour modifier leur configuration.

Il s'agit pour vous de «fixer les droits d'accès» des répertoires suivants :

- ▷ /CACHE
- ▷ /IMG
- ▷ /ecrire
- ▷ /ecrire/data

Les logiciels («clients») FTP ont des fonctionnements différents, mais la procédure est généralement la suivante :

- ▷ sélectionnez le dossier dont vous voulez modifier les «droits d'accès» ;
- ▷ trouvez dans votre logiciel de FTP une fonction intitulée «changer (ou *modifier*, ou *fixer*) les droits d'accès» ;
- ▷ si cette fonction se présente sous la forme d'une interface graphique, vous devez cocher la case correspondant à «Ecriture», pour l'utilisateur «Autres» (ou «Tous utilisateurs») :



- ▷ si cette modification se fait en mode «texte», la configuration numérique est «777».

Lorsque vous avez effectué cette opération pour chaque répertoire signalé par le système d'installation, rechargez la page, la procédure se poursuit automatiquement.

12.1.2 Votre connexion MySQL

Cette étape consiste à indiquer les informations nécessaires à la connexion de SPIP au serveur MySQL.

- ▷ *Adresse de la base de données* : en fonction des choix de votre hébergeur, cette information sera simplement «localhost», ou bien l'adresse de votre site («www.monsite.org»).
- ▷ *Le login de connexion* : souvent, il s'agit du même login que celui utilisé pour le téléchargement de vos fichiers par FTP.
- ▷ *Le mot de passe de connexion* : souvent, il s'agit du même mot de passe que celui de votre accès FTP au site.
- ▷ Ces informations ne s'inventent pas : si vous ne les connaissez pas, c'est que votre hébergeur ne vous les a pas fournies. Donc : si n'avez pas sous la main ces trois informations, vous ne pouvez pas continuer.
- ▷ Ces informations vous sont fournies **par l'hébergeur de votre site** : pour les obtenir, vous devez le contacter directement (ne demandez pas à l'équipe de développement de SPIP de vous les fournir, nous ne les connaissons pas !). Chez certains hébergeurs, ces informations vous sont fournies dans un mode d'emploi en ligne.
- ▷ Notez encore que, très souvent, il faut **demander** à votre hébergeur d'activer votre accès MySQL, ou effectuer vous-même une opération spécifique dans ce but. Beaucoup d'hébergeurs annonçant un hébergement doté d'une base de données MySQL n'activent pas automatiquement cette connexion lors de l'ouverture du site ; une démarche supplémentaire de votre part (après l'ouverture de votre compte chez cet hébergeur) est souvent nécessaire. Dans ce cas, voyez la documentation de votre hébergeur pour savoir comment *activer* votre connexion MySQL. (Là encore : l'équipe de SPIP ne peut pas vous aider.)

12.1.3 Choix de votre base

- ▷ Vous devez indiquer le nom de la base de données que vous a attribué votre hébergeur. Cette information vous est fournie par votre hébergeur : si vous ne l'avez pas, demandez-la lui, ou bien consultez la documentation en ligne de cet hébergeur (inutile de la demander à l'équipe de développement de SPIP, nous ne connaissons pas cette information).
- ▷ Très souvent, cette information correspond au *login* de votre compte d'hébergement (celui que vous utilisez pour vous connecter à votre site par FTP).
- ▷ Première possibilité (le plus souvent) : une liste de comptes est affichée (parfois un peu longue, selon la configuration fixée par votre hébergeur). Parmi les comptes indiqués, un seul correspond au vôtre, il vous suffit donc de le sélectionner et de valider pour passer à l'étape suivante.
- ▷ Deuxième possibilité : un seul nom, correspondant à votre compte, s'affiche (automatiquement, votre hébergeur a configuré son système pour que s'affiche uniquement votre propre compte). Facile : sélectionnez ce compte et validez.
- ▷ Troisième possibilité (généralement sur un hébergement pour professionnels, ou s'il s'agit d'une machine sur laquelle vous avez des droits élevés) : vous devez *créer* votre propre base. Dans ce cas (après avoir bien vérifié qu'un tel compte n'existait pas dans la liste indiquée ci-dessus), vous pouvez indiquer le nom de votre choix à la suite de l'indication «Créez une nouvelle base».
- ▷ Dernière possibilité (échec) : votre compte n'apparaît pas dans la liste proposée, et vous ne pouvez pas en créer vous-même. C'est assez rare : cela signifie que votre hébergeur a créé pour vous un accès au serveur de la base, mais a oublié de vous créer un compte. Dans ce cas, vous devez directement consulter votre hébergeur.

12.1.4 Informations personnelles

Cette étape est très simple, mais doit être effectuée attentivement. En effet, c'est elle qui détermine vos identifiants de connexion à votre propre site ! Si vous procédez trop vite, vous risquez de ne pas pouvoir vous connecter . . .

Notez qu'une fois votre site installé, vous aurez la possibilité de modifier ces informations.

- ▷ *Votre identité publique* : c'est ce qui apparaîtra en tant que signature de vos articles.
- ▷ *Vos identifiants de connexion* : il s'agit ici des identifiants que **vous choisissez vous-même** pour vous connecter à votre propre système SPIP. Ces informations n'ont aucunement besoin d'être identiques à celles que votre hébergeur vous a indiquées pour vous connecter à votre compte par FTP. Au contraire, *nous vous recommandons très fortement de ne pas utiliser les mêmes identifiants* pour votre accès SPIP (que vous choisissez librement) et pour votre connexion FTP (imposées par votre hébergeur).

12.1.5 Vérification par FTP

Certaines fonctionnalités de SPIP influent directement sur la structure et le contenu mêmes de la base de données (notamment : mise-à-jour, sauvegarde et restauration de la base . . .). Pour ces fonctionnalités, particulièrement sensibles, il a été ajouté une procédure d'identification par FTP, de façon à les réserver aux quelques personnes ayant accès au serveur du site par FTP (généralement, si un site peut comporter plusieurs administrateurs pour SPIP, l'accès FTP est réservé à un webmestre principal).

Pour utiliser ces fonctionnalités, vous devez simultanément :

- ▷ vous connecter par le Web avec votre butineur habituel ; lorsque vous lancez l'action, SPIP vous indique un nom du type "admin_XXXXX", qu'il vous faut recopier ;
- ▷ vous connecter au serveur de votre site par FTP ; là, dans le dossier /ecrire/data, créez un fichier (ou un répertoire vide) portant le nom "admin_XXXXX" ;
- ▷ une fois ce fichier ou ce dossier créé, retournez à votre butineur Web, et rechargez la page. Dès lors, l'authentification par FTP est réalisée, et l'action désirée est déclenchée.

12.1.6 Un problème de squelette ?

Lorsque SPIP se trouve confronté à une erreur dans sa communication avec la base de données MySQL, il affiche à l'écran la requête qui a échoué, ainsi que le message d'erreur renvoyé par la base de données (en rouge).

Le problème peut provenir :

- ▷ soit d'une erreur dans la définition de votre squelette, si vous êtes en train de modifier votre site ;
- ▷ soit d'une panne de la base de données.

Par exemple, un message du type

```
\textcolor{red}{> Unknown column 'articles.chapi' in 'where clause'}
```

signale que la boucle fait appel à un critère de sélection (`chapi`) qui n'est pas prévu.

En revanche, un message du type

```
> Can't open file: 'spip_articles.MYD'
```

signale un grave problème de la base MySQL elle-même : voyez alors avec votre hébergeur pour qu'il révise son installation et/ou répare votre base de données. Si vous disposez d'une version récente de MySQL (à partir de 3.23.14), vous pouvez aussi tenter une auto-réparation de la base (voir référence INCONNUE).

12.2 Les articles

12.2.1 Les raccourcis typographiques de SPIP

Pour faciliter la mise en page des documents publiés avec SPIP, le système propose un certain nombre de «raccourcis SPIP» destinés :

- ▷ à simplifier l'utilisation par des utilisateurs en connaissant pas le HTML ;
- ▷ à faciliter le traitement automatique de la mise en page.

Ainsi, vous pouvez naturellement utiliser du code HTML dans vos documents SPIP, mais nous vous conseillons d'utiliser de préférence ces quelques raccourcis SPIP, plus simples à mémoriser, et permettant surtout quelques manipulations automatiques par le système.

Typographie française automatique

SPIP respecte automatiquement les principales règles d'espacement de la typographie française - ainsi des espaces insécables sont ajoutées devant les caractères « :», « ;», « !», « ?» -, et place des espaces insécables avant et après les guillemets « à la française ».

Par exemple : «L'occasion a tous ses cheveux au front : quand elle est outre passée, vous ne la pouvez plus révoquer ; elle est chauve par le derrière de la tête, et jamais plus ne retourne.» (François Rabelais) s'affiche ainsi : « L'occasion a tous ses cheveux au front : quand elle est outre passée, vous ne la pouvez plus révoquer ; elle est chauve par le derrière de la tête, et jamais plus ne retourne. » (ici ce sont les deux points et le point-virgule qui sont corrigés).

Créer des paragraphes

Pour créer des paragraphes, il suffit de laisser une ligne vide, un peu comment on sépare les paragraphes dans un email (on «saute» une ligne).

Le fait de simplement «revenir à la ligne» (retour-chariot) sans séparer les deux paragraphes par une ligne vide ne suffit pas pour provoquer un changement de paragraphe (cela ne provoque même pas un retour à la ligne).

Vous pouvez laisser plusieurs lignes vides à la suite sans que cela modifie la présentation. Pour SPIP, laisser une ligne vide ou plusieurs lignes vides revient au même : un changement de paragraphe, sans influence sur l'espacement entre les paragraphes.

Fabriquer des listes ou des énumérations

On peut fabriquer des listes dans SPIP de la même manière que dans un email : il suffit de revenir à la ligne et de commencer la nouvelle ligne avec un tiret («-»).

Notez : ici un simple retour à la ligne suffit (on peut faire des énumérations dans le même paragraphe); mais si l'on «saute» une ligne avant la ligne commençant par un tiret, une ligne vide est affichée avant l'énumération.

Par exemple,

```
- Qu'est-ce que cela peut faire que je lutte pour la mauvaise cause
  puisque je suis de bonne foi?
- Et qu'est-ce que ça peut faire que je sois de mauvaise foi puisque
  c'est pour la bonne cause.
```

(Jacques Prévert)

sera affiché ainsi :

- ▷ Qu'est-ce que cela peut faire que je lutte pour la mauvaise cause puisque je suis de bonne foi ?
- ▷ Et qu'est-ce que ça peut faire que je sois de mauvaise foi puisque c'est pour la bonne cause.

Variantes :

- ▷ Un simple saut de ligne sans puce s'obtient en tapant _ (le trait de soulignement ou *underscore*) au début de la ligne, suivi d'une espace.
- ▷ On peut faire des énumérations imbriquées en ajoutant des étoiles après le tiret d'énumération.

Essayez par exemple :

```
-* Ton cheval est
  -** alezan;
  -** bai;
  -** noir;
-* mais mon lapin est
  -** blanc:
  -*** angora ou
  -*** à poil ras.
```

- ▷ Enfin, on peut faire des listes numérotées en utilisant le # à la place de l'étoile :

```
-# premier
-# deuxi&egrave;me
-# troisi&egrave;me
```

donnera :

1. premier
2. deuxième
3. troisième

Gras et italique

On indique simplement du texte en *italique* en le plaçant entre des accolades simples :

```
... du texte {en italique} en ...
```

On indique du texte **en gras** en le plaçant entre des accolades doubles :

```
... du texte {{en gras}} en ...
```

Intertitres

Les intertitres sont des titres à l'intérieur d'un texte permettant d'en indiquer la structure. Dans SPIP, on les indique très simplement en les plaçant entre des accolades triples :

```
{{{Un titre de partie}}}
```

donnera :

Un titre de partie

Trait de séparation horizontal

Il est très simple d'insérer un trait de séparation horizontal sur toute la largeur du texte : il suffit de placer une ligne ne contenant qu'une succession d'au moins quatre tirets, ainsi :

◇◇◇

Les liens hypertextes

On fabriquera facilement un lien hypertexte avec le code suivant :

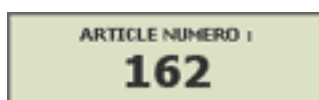
```
SPIP est une initiative du [minirézo->http://www.minirezo.net/].
```

devient «SPIP est une initiative du [minirézo](http://www.minirezo.net/).».

L'adresse du lien peut être une adresse absolue (commençant, comme ici, par `http://`), une adresse relative (vers une autre page du même site), un lien vers un document utilisant un protocole de l'internet (`ftp://...`), une adresse email («`mailto:minirezo@rezo.net`»)...

Liens hypertextes à l'intérieur du site

Ce même système de liens hypertextes facilite, de plus, la création de liens à l'intérieur de votre site sous SPIP. La seule subtilité consiste à repérer le *numéro* de l'article, de la rubrique, ou de la brève vers laquelle vous voulez mener votre lien hypertexte : NUMERO lorsque vous «visitez», dans l'espace privé, un article, une brève ou une rubrique, la colonne de gauche contient un pavé indiquant, en gros caractères, ce numéro.



C'est ce numéro que vous allez indiquer dans le lien hypertexte :

Lien vers l'article 342 (quatre possibilités) :

lien vers [l'article->342]

lien vers [l'article->art342]

lien vers [l'article->article 342]

pour sa part, [->art342] affichera le titre de l'article 342 avec un lien vers cet article.

Lien vers la rubrique 12 :

lien vers [la rubrique->rub12]

lien vers [la rubrique->rubrique 12]

Lien vers la brève 65 :

lien vers [la brève->br65]

lien vers [la brève->breve 65]

lien vers [la brève->brève 65]

Auteurs, mots-clés, sites :

[->aut13] ou [->auteur13]

[->mot32]

[->site1]

A noter : pour un site le raccourci pointe vers l'adresse URL du site référencé.

Notes de bas de page

Une note de bas de page est, habituellement, signalée par un numéro placé à l'intérieur du texte, numéro repris en bas de page et proposant un complément d'information.

Dans SPIP, cette fonctionnalité (assez lourde à gérer manuellement en HTML) est automatisée : les notes sont numérotées par SPIP, qui gère également des liens hypertextes à l'intérieur du document pour passer directement de l'appel de note au texte de la note correspondante, et vice-versa.

Une note de bas de page est indiquée, dans SPIP, entre doubles crochets :

Une note[[Voici un complément d'information.]] de bas de page.

sera affiché sous la forme : «Une note [2] de bas de page.»

Des notes non automatiques

Dans la plupart des cas, le système de notes automatiques indiqué ci-dessus suffit amplement. Cependant, vous pouvez gérer les notes d'une manière non automatique.

Par exemple :

```
"Vous pouvez utiliser les notes numérotées automatiques [[En plaçant
    le texte de la note entre crochets.]],
- mais aussi forcer la numérotation de la note [[<23> En indiquant
    le numéro de la note entre les symboles "<" et ">".]],
- utiliser des notes sous forme d'astérisques [[<*> En plaçant
    simplement une astérisque entre les symboles "<" et ">".]],
- fabriquer des notes sans références (non numérotées) [[<> En
    n'indiquant rien entre les symboles "<" et ">".]],
- donner un nom (en toutes lettres) à une note [[<Rab> François
    Rabelais.]];
- rappeler une note déjà existante [[<23>]] en indiquant le numéro
    de cette note entre les symboles <<<>> et <<<>{>>>. et en laissant
    vide le reste de la note."
```

Ce qui donne :

- «Vous pouvez utiliser les notes numérotées automatiques [3],
- ▷ mais aussi forcer la numérotation de la note [23],
 - ▷ utiliser des notes sous forme d'astérisques [*],
 - ▷ fabriquer des notes sans références (non numérotées) ,
 - ▷ donner un nom (en toutes lettres) à une note [Rab] ;
 - ▷ rappeler une note déjà existante [23] en indiquant le numéro de cette note entre les symboles «<>» et «>>». et en laissant vide le reste de la note.»

Tableaux

Pour réaliser des tableaux très simples dans SPIP, il suffit de faire des lignes dont les «cases» sont séparées par le symbole «|» (pipe, un trait vertical), lignes commençant et se terminant par des traits verticaux. Il est impératif de laisser des lignes vides avant et après ce tableau.

Par exemple, le tableau :

Nom	Prénom	Age
Marso	Ben	23 ans
Capitaine		non connu
Philant	Philippe	46 ans
Cadoc	Bébé	4 mois

se code ainsi :

```
| {{Nom}} | {{Prénom}} | {{Age}} |
| Marso | Ben | 23 ans |
| Capitaine | | non connu |
| Philant | Philippe | 46 ans |
| Cadoc | Bébé | 4 mois |
```

Court-circuiter les raccourcis SPIP

Dans certains cas, il peut être utile d'indiquer à SPIP que certaines parties d'un document ne doivent pas être «traitées» par le filtre des raccourcis typographiques : vous ne voulez pas corriger la typographie, afficher du code source (par exemple en PHP, JavaScript ...) ...

Le code de ce raccourci est :

```
<HTML>texte à ne pas transformer; attention!</HTML>
```

ce qui donne : «texte à ne pas transformer; attention !» (ici, notez l'absence d'espaces avant le point-virgule et le point d'exclamation).

Par exemple, le présent article est écrit sous SPIP, et il affiche le code source des raccourcis SPIP ; si l'on n'avait pas utilisé ce code, ces raccourcis auraient été interprétés, et vous ne pourriez pas lire ici leur code source !

Afficher du code informatique

Certains utilisateurs de SPIP veulent parfois afficher du code informatique dans leurs pages.

Le raccourci `<code>...</code>` est là pour ça.

Exemple :

```
<code><?php // ceci est du langage php
echo "bonjour";
?></code>
```

donne

```
<?php // ceci est du langage php echo "bonjour" ; ?>
```

◇ ◇ ◇

[2] Voici un complément d'information.

[3] En plaçant le texte de la note entre crochets.

[23] En indiquant le numéro de la note entre les symboles «<» et «>».

[*] En plaçant simplement une astérisque entre les symboles «<» et «>».

En n'indiquant rien entre les symboles «<» et «>».

[Rab] François Rabelais.

12.2.2 Titre, surtitre, soustitre

▷ Le *titre* est obligatoire.

▷ Le *surtitre* et le *soustitre* sont totalement optionnels. Si vous n'en avez pas besoin, laissez-les vides, la présentation du site s'adaptera automatiquement à leur présence ou absence.

Les administrateurs du site peuvent, s'ils le désirent, totalement supprimer l'utilisation du surtitre et/ou du soustitre, via la page «Configuration précise» (section 12.10.2 page 259).

12.2.3 Choisir la rubrique

Le menu indique toute la hiérarchie des rubriques (section 12.3.1 page 244) (telles qu'elles ont été créées par les administrateurs du site) : sélectionnez celle dans laquelle vous voulez placer l'article.

Beaucoup d'utilisateurs débutants semblent ignorer complètement cette fonction pourtant simple, et laissent leur article «n'importe où» dans la structure du site. Il est donc conseillé aux administrateurs, avant validation d'un article, de toujours vérifier que celui-ci se trouve dans la bonne rubrique.

12.2.4 Descriptif rapide

Le *descriptif rapide* est utilisé pour la navigation à l'intérieur du site : il permet d'indiquer brièvement, dans les sommaires par exemple, le thème de l'article.

Ce descriptif est optionnel ; de même, on peut le rédiger de la longueur que l'on veut. Cependant, il a été prévu à l'origine pour des textes très courts (une ou deux phrases), qui figureront dans les listes d'articles (sommaires, listes des articles de tel auteur, tri d'articles par mot-clé, réponses du moteur de recherche, etc.).

Les administrateurs du site peuvent, via l'interface de «Configuration précise» (section 12.10 page 259), désactiver totalement l'affichage du descriptif.

12.2.5 Chapeau

Le *chapeau*, dans le jargon de la presse, désigne le court texte qui «chapeaute» l'article. Son utilisation est optionnelle.

Les administrateurs peuvent, via l'interface de «Configuration précise», désactiver totalement l'utilisation du chapeau (section 12.10 page 259).

12.2.6 Redirection d'article

Cette option (accessible uniquement aux administrateurs) permet de créer un «article virtuel» : il s'agit d'un article dont le titre, la date et les auteurs sont enregistrés dans votre site sous SPIP, mais qui pointe vers une autre adresse.

Cette fonctionnalité vous permet de référencer dans votre système SPIP des articles qui ne sont pas réalisés avec SPIP (par exemple pour intégrer à la navigation de votre site sous SPIP des pages réalisées antérieurement à l'installation de SPIP).

Pour indiquer que votre article est un «article virtuel», il suffit d'indiquer l'URL de l'article-cible dans la case correspondante.

Pour supprimer la redirection, il suffit de «vider» la case de redirection (effacer l'adresse de l'article-cible).

12.2.7 Texte

Rien de bien compliqué : il s'agit du texte de votre article, comme son nom l'indique.

- ▷ Il y a cependant un point qui peut parfois poser problème : la longueur des textes. Pour certains textes très longs (dans notre expérience, à partir de 32 ko), il arrive que ce texte soit tronqué ou carrément refusé lors du transfert dans SPIP. Ce problème n'est pas dû à SPIP, mais aux navigateur Web que vous utilisez. Lorsque vous êtes confronté à un texte «trop long» pour un navigateur, essayez avec un autre logiciel.
- ▷ Le texte de l'article se prête particulièrement bien à l'utilisation des raccourcis typographiques (section 12.2.1 page 233) de SPIP.

12.2.8 Date

La *date* de l'article correspond, en général, à la date de sa *mise en ligne* (ou publication sur le site Web).

- ▷ Cette date est fixée automatiquement au moment où l'article est *validé* par un administrateur (donc au moment où il apparaît sur le site public).
- ▷ Après validation, toutefois, l'admin peut modifier cette date.

12.2.9 Date de publication antérieure

Cette fonction a été ajoutée pour des cas très spécifiques, lorsque SPIP est utilisé pour installer des archives et que ces archives doivent avoir une date de publication différente de la date de mise en ligne.

Il s'agit de pouvoir indiquer qu'un document a déjà fait l'objet d'une publication antérieure : article de journal, livre ...

- ▷ Par défaut, cette date ne s'affiche pas : le bouton «Ne pas afficher de date de publication antérieure» est sélectionné lors de la création de l'article.
- ▷ Si l'on veut indiquer une telle date, il faut à la fois sélectionner l'option «Afficher la date» et utiliser les menus déroulants pour la fixer.

Cette date, contrairement à la «date de publication» de l'article, n'est pas calculée automatiquement lors de la validation de l'article. C'est pourquoi elle peut être modifiée à tout moment.

Les administrateurs peuvent, via l'interface de «Configuration précise», désactiver l'utilisation de la date de publication antérieure (section 12.10 page 259).

12.2.10 Les auteurs

Lorsqu'un administrateur ou un rédacteur crée un article, il est considéré automatiquement comme l'auteur de cet article. Dans bien des cas il faudra changer les auteurs (quand on intègre au site le texte d'une autre personne, quand un texte ne doit pas être signé, ou encore quand on l'édite à plusieurs ...)

▷ Ajouter un auteur

Un menu déroulant proposant la liste de tous les rédacteurs du site permet de sélectionner et d'ajouter un nouvel auteur.

Si plus de 50 rédacteurs participent au site, il est difficile de les présenter sous forme de menu déroulant (temps de chargement interminable) ; dans ce cas, le menu déroulant est remplacé par une case de recherche : indiquez le nom de l'auteur que vous voulez ajouter, et cliquez sur «Chercher». Si plusieurs rédacteurs correspondent à cette recherche, le système vous proposera de sélectionner celui qui vous convient.

▷ Retirer un auteur

A la suite de chaque auteur de l'article, un lien «Retirer l'auteur» permet de simplement effacer cet auteur.

C'est uniquement en tant qu'auteur de cet article précis que le rédacteur est supprimé ; il reste bien entendu présent dans la liste des rédacteurs du site.

▷ Remplacer un auteur par un autre

Cela se fait en deux étapes : il faut simplement ajouter le nouvel auteur, et retirer le précédent (voir ci-dessus).

- ▷ Notez finalement que les administrateurs ont beaucoup plus de possibilités sur cette liste d'auteurs que les rédacteurs. Les rédacteurs ne peuvent pas, en particulier, se retirer eux-mêmes d'un article. Si un rédacteur veut publier un article anonyme (sans auteur), il doit demander à un administrateur d'effectuer cette opération.

12.2.11 Logo de l'article

SPIP vous permet d'installer un logo correspondant à l'article. De cette façon, dans l'interface de navigation du site public, il sera possible d'afficher un bouton graphique menant à l'article.

Pour un article, il est possible :

- ▷ de ne pas utiliser de logo ;
- ▷ d'installer un logo graphique simple ;
- ▷ d'utiliser un logo animé gérant le «survol»

(logo «2 positions» : le logo change lorsque que la souris le survole).

Formats d'images

Lorsque vous créez vos images (avec votre logiciel habituel), vous devez les créer à l'un des formats suivants :

- ▷ GIF (le fichier GIF peut être un «GIF animé»);
- ▷ JPEG;
- ▷ PNG (déconseillé, de nombreux butineurs ne permettant pas de les afficher correctement).

Afin d'éviter les erreurs graves de manipulation, SPIP rejette les fichiers-image d'une taille supérieure à 256 ko. Conseil : puisque ces «boutons» sont des éléments de l'interface graphique, veillez à ce que leur poids (nombre de kilo-octets) ne soit pas trop élevé (en général, moins de 10 ko) si vous voulez que la navigation sur votre site reste fluide.

Veillez particulièrement à ce que le nom de vos fichiers ait une *terminaison* indiquant leur format : **.gif**, **.jpg** ou **.png**. Le nom du fichier n'a aucune importance, à condition de *ne pas oublier cette terminaison*.

Si vous créez un bouton gérant le «survol», créez deux fichiers graphiques différents (un pour le bouton «normal», et un autre fichier s'affichant lorsque le bouton est survolé par la souris) : il convient alors que ces deux fichiers aient exactement la même taille (en pixels).

Logo simple (pas de survol)

Pour ajouter un bouton, une interface est disponible dans la colonne de gauche de l'article, sous la mention «LOGO DE L'ARTICLE».



Selon la version de votre navigateur, cliquez sur le bouton «Browse», «Sélectionner», «File», «Fichier» ... ce qui déclenche l'ouverture d'une interface permettant de sélectionner, sur votre disque dur, le fichier graphique correspondant à votre bouton.

Une fois ce fichier sélectionné, cliquez sur le bouton «Télécharger». Votre logo s'affiche alors. Il est suivi d'un bouton «Supprimer le logo», qui vous permet, simplement, de supprimer ce logo.

Si vous ne comptez pas obtenir un logo gérant le survol, aucune autre opération n'est nécessaire.

Remplacer le logo

Il peut arriver que vous vouliez remplacer le logo par un autre fichier.



Cela se fait en deux étapes :

- ▷ commencez par «Supprimer le logo» : l'interface précédente, dotée du bouton «Télécharger», apparaît à nouveau ;
- ▷ téléchargez le nouveau fichier, selon la procédure déjà décrite.

Du fait du fonctionnement des butineurs, l'image qui s'affiche alors est erronée, puisqu'il s'agit encore de la version précédente (l'image est «en cache» de votre navigateur). Cliquez sur cette image (avec le bouton droit de votre souris, ou en maintenant la touche «ctrl» sur Macintosh) afin de faire apparaître un menu déroulant local : sélectionnez l'option «Recharger cette image» (ou, en anglais, «Reload image»). La nouvelle version de votre logo devrait alors s'afficher.

Logo pour le survol

Après l'installation du premier fichier (logo simple), l'interface affiche non seulement le logo que vous avez installé sur le serveur, mais ajoute une seconde interface, sous le titre «LOGO POUR LE SURVOL».



C'est par cette interface que vous pouvez indiquer le second fichier nécessaire à la gestion du survol.

Si, lorsque vous avez installé les deux fichiers, vous supprimez le premier (le bouton «simple»), l'interface du second logo ne s'affiche plus. En effet, en l'absence du premier logo, il n'y a plus de raison de gérer un quelconque «survol» !

Aucune intervention dans le «texte» de votre article n'est nécessaire. Lors de l'affichage sur le site public, la gestion des logos des rubriques est entièrement automatique. Le code HTML sera généré en fonction de la taille du logo, et la fonction de survol en JavaScript sera également créée automatiquement.

12.2.12 Le statut de l'article

Le *statut* de l'article correspond à sa situation éditoriale sur le site. L'article peut être :

- ▷ en cours de rédaction ;
- ▷ proposé à l'évaluation ;
- ▷ publié en ligne ;
- ▷ à la poubelle ;
- ▷ refusé.

Ces étapes, que seuls les administrateurs peuvent modifier, permettent la gestion du site.

N.B. Le statut des articles est symbolisé par des puces colorées.

▷  **Article en cours de rédaction**

Lorsqu'il est créé, un article est naturellement considéré comme étant «en cours de rédaction» : ses auteurs sont en train de le rédiger, de le modifier . . .


L'article «en cours de rédaction» n'est visible que par les auteurs de l'article et par les administrateurs. Les autres rédacteurs du site n'y ont pas accès.

▷  **Article proposé à l'évaluation**

Lorsque l'auteur juge que son article est prêt, il le «propose» aux autres participants, afin qu'il soit éventuellement discuté collectivement, avant d'être validé (publié en ligne) ou refusé.


Lorsque l'article est «proposé à l'évaluation», il est indiqué sur la page «à suivre» de tous les utilisateurs de l'espace privé, qui sont ainsi invités à venir le discuter par l'intermédiaire du forum de discussion interne placé à la suite de l'article.

Un tel article est alors visible par tous les rédacteurs. En revanche, il ne peut être modifié que par son auteur ou un administrateur.

▷  **Article publié en ligne**

Après avoir été éventuellement discuté par les rédacteurs (pendant la phase de «proposition»), un article peut être «validé», c'est-à-dire publié en ligne par un administrateur. Dès lors, tous les visiteurs du site public y ont accès.


Lorsqu'un article est publié en ligne, seuls les administrateurs peuvent le modifier. Son auteur, s'il n'est pas rédacteur, ne peut plus le modifier, et doit donc demander à un administrateur s'il veut y porter des corrections.

▷  **Article refusé**

Un article «proposé» qui ne conviendrait pas à la ligne éditoriale du site peut être «refusé» si les administrateurs refusent de le publier en ligne.

Un article «refusé» n'est plus visible que par son auteur et par les administrateurs.

Un article «refusé» ne peut cependant plus être modifié par son auteur, qui ne pourra donc pas plus le repropose à la publication. Dans le cas d'un article qui demanderait des retouches, on préférera donc repasser l'article «en cours de rédaction» au lieu de le «refuser» purement et simplement, afin que son auteur puisse le modifier et le représenter ultérieurement.

▷  **Article à la poubelle**

Un article peut être mis à la poubelle, uniquement par un administrateur.

Un article «à la poubelle» n'est plus visible dans l'espace privé, même par les administrateurs. Attention donc, cette option est «violente» : l'article disparaît complètement.

En réalité, l'article est toujours stocké dans la base de données, mais devient très difficilement accessible avec les outils de SPIP.

Cette option est donc réservée aux articles créés par erreur, que l'on veut totalement détruire. On préférera, le plus souvent, l'option «Article refusé», qui est moins définitive.

- ▷ Notez enfin que les administrateurs peuvent à tout moment modifier le *statut* d'un article. Un article publié peut ainsi être repassé en «rédaction». Cependant, une fois un article publié en ligne, n'abusez pas de ces changements de statut : vous obtiendriez en effet un site «à trou», avec des pages qui apparaissent et disparaissent, ce qui est très pénalisant pour le visiteur.

12.2.13 Proposer son article

Lorsque l'article est «en cours de rédaction» (voir la rubrique «Le statut de l'article» section 12.2.12 page 242), il est suivi d'un bouton «Demander la publication de cet article».

Seul l'auteur de l'article peut effectuer cette opération.

Cela signifie que l'article est alors «Proposé à l'évaluation», c'est-à-dire présenté à tous les autres rédacteurs, qui seront invités à le commenter, dans l'attente d'une validation (publication) ou d'un refus de la part des administrateurs.

Attention : une fois l'article «proposé à l'évaluation», il n'est plus possible à l'auteur de revenir sur sa décision et de repasser le texte «en cours de rédaction». Donc l'opération «Demander la publication de cet article» ne doit être effectuée par l'auteur qu'une fois qu'il considère son texte comme complet et définitif. Seul un administrateur peut alors remettre le texte «en cours de rédaction».

12.2.14 Articles en cours de modification

Lorsqu'un rédacteur intervient sur un article pour le modifier, les autres participants au site qui se rendent sur la page de cet article sont prévenus et il leur est déconseillé d'intervenir à leur tour sur l'article.

En effet, si deux rédacteurs interviennent en même temps sur le même article, les modifications de l'un risquent d'«écraser» les modifications de l'autre.

Si vous voyez la mention «Attention, un rédacteur est intervenu sur cet article», il est donc fortement déconseillé de le modifier à ce moment. Revenez plus tard sur cet article, pour intervenir lorsque l'autre rédacteur aura effectué et enregistré ses modifications.

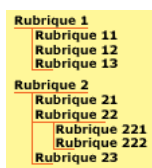
À l'inverse, lorsque vous-mêmes intervenez pour modifier un article, les autres participants au site qui visiteraient cette page seraient prévenus de votre intervention. Tant que vous modifiez l'article, et pendant une durée d'**une heure**, les autres rédacteurs sont invités à ne pas intervenir sur ce texte. Lorsque vous considérez que vous avez terminé de travailler sur cet article, et que d'autres peuvent à leur tour intervenir dessus, vous pouvez «débloquent» cet article. L'avertissement aux autres participants disparaîtra, et ils pourront si nécessaire effectuer leurs propres modifications.

12.3 Les rubriques

12.3.1 Une structure hiérarchisée

La structure des rubriques constitue l'ossature de votre site ; c'est elle qui va déterminer son interface, le mode de navigation, les relations entre articles et entre brèves ...

Dans SPIP, cette structure est de type *hiérarchique* : une rubrique peut contenir des sous-rubriques qui, elles-mêmes, contiennent des sous-rubriques, etc.



Dans l'exemple ci-dessus, on comprend bien que la rubrique 222 dépend de la rubrique 22, qui elle-même dépend de la rubrique 2, laquelle ne dépend d'aucune autre rubrique (dans ce cas, on considérera que la rubrique 2 se trouve à la *racine du site*).

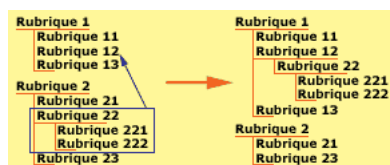
Par structure hiérarchisée, on entend le fait qu'une rubrique ne dépend que d'une seule autre rubrique (et non de plusieurs), et qu'une rubrique ne peut dépendre d'une de ses propres sous-rubriques (c'est-à-dire : SPIP n'autorise pas les structures circulaires). Cette structure, très classique, a été retenue en raison de sa simplicité d'utilisation.

Seuls les administrateurs peuvent créer, modifier ou supprimer des rubriques.

12.3.2 Choisir la rubrique

Le fonctionnement de ce menu déroulant est très simple : le menu indique toute la hiérarchie des rubriques (telles qu'elles sont créées par les administrateurs du site), il suffit de sélectionner celle dans laquelle on veut placer la sous-rubrique.

Déplacer une rubrique



Vous pouvez, par l'intermédiaire de ce menu déroulant, faire dépendre la présente rubrique d'une autre rubrique. Dans ce cas, il faut comprendre que l'ensemble des sous-rubriques contenues dans la présente rubrique se «déplacent» avec elle dans la hiérarchie du site. De la même façon, les articles contenus dans cette rubrique et ses sous-rubriques se déplacent avec elle.

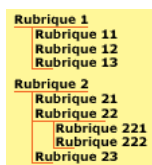
12.3.3 Logo de la rubrique

Vous pouvez installer sur votre site un logo pour chaque rubrique. Ce logo peut être soit un logo unique (image fixe), soit un logo à deux positions gérant le survol de la souris.



L'installation d'images pour ce logo de rubrique se déroule exactement de la même façon que pour le logo d'un article (section 12.2.11 page 240).

N.B. Les logos des rubriques ont un fonctionnement *récuratif* : en l'absence d'un logo pour une rubrique donnée, SPIP va tenter de lui substituer le logo d'une rubrique dont elle dépend :



Dans la hiérarchie ci-dessus, en l'absence d'un logo pour la rubrique 221, SPIP lui substituera (uniquement pour la visite du site public) le logo de la rubrique 22 ou encore, s'il n'y a pas de logo pour cette rubrique, celui de la rubrique 2. Sinon, SPIP affichera le logo installé à la racine du site.

Notez encore que, si le webmestre l'a programmé ainsi, le logo d'une rubrique pourra être utilisé comme logo de substitution pour les articles qu'elle contient.

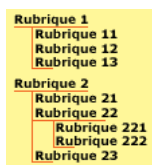
12.4 Les brèves

12.4.1 Les brèves

Les brèves constituent une méthode simple et rapide de publication dans SPIP. Contrairement aux articles, les brèves sont constituées d'un nombre très réduit d'informations : un titre, un texte et un lien hypertexte. Ainsi, le système de brève est-il idéal pour un suivi de l'actualité, une revue de presse en ligne, etc.

12.4.2 Choisir la rubrique

Afin de simplifier leur utilisation (et pour éviter que les brèves ne fassent double-emploi avec les articles), l'intégration des brèves dans la hiérarchie des rubriques est réduite au strict minimum : les brèves dépendent uniquement des rubriques situées à la racine du site.



Dans notre exemple, on pourra placer des brèves dans les rubriques 1 et 2, mais pas dans leurs sous-rubriques (contrairement aux articles, que l'on peut placer n'importe où). De ce fait, la présentation de la page des brèves se fait directement en fonction de ces grandes rubriques, et le menu déroulant permettant d'indiquer la position des brèves est très court.

12.4.3 Le lien hypertexte

Afin de faciliter l'utilisation des brèves dans le cadre d'une revue de presse en ligne, chaque brève peut recevoir l'indication d'un lien hypertexte. Il suffit d'indiquer le titre du site ou de l'article référencé, et son adresse (URL).

Ces informations sont bien entendu optionnelles.

N.B. Ce système de lien n'empêche pas d'intégrer des liens hypertextes dans le corps du texte de la brève, mais le lien hypertexte *séparé* permet au webmestre du site de lui appliquer un traitement graphique spécifique.

12.4.4 Le statut de la brève

La gestion d'une brève est plus simple que celle d'un article. Une brève n'a pas d'auteur. Son statut est soit «Proposée», «Validée» ou «Refusée». Seuls les administrateurs peuvent en modifier le statut.

▷ **Brève proposée**

Les brèves «proposées» sont signalées sur la page «A suivre» : tous les rédacteurs peuvent les consulter et les modifier. Les administrateurs se voient proposer deux boutons - permettant de les valider ou de les refuser.

▷ **Brève validée**

Les brèves «validées» sont celles qui apparaissent sur le site public. Seuls les administrateurs peuvent dès lors les modifier.

▷ **Brève refusée**

Une brève «refusée» n'est pas publiée sur le site public, et seuls les administrateurs peuvent y accéder dans le site privé.

12.4.5 Le logo de la brève

Vous pouvez installer sur votre site un logo pour chaque brève.



Ce logo peut être soit un logo unique (image fixe), soit un logo à deux positions gérant le survol de la souris.

L'installation d'images pour ce logo de brève se déroule exactement de la même façon que pour le logo d'un article (section 12.2.11 page 240).

12.5 Images et documents

12.5.1 Insérer des images

SPIP vous offre la possibilité d'illustrer vos articles et vos brèves avec des images. Cela s'effectue en plusieurs étapes : vous devez envoyer le fichier de votre image vers le site, puis insérer l'image à l'intérieur du texte.

◇ ◇ ◇

Préparation : Formats d'images

Lorsque vous créez vos images (avec votre logiciel habituel), vous devez les créer à l'un des formats suivants :

- ▷ GIF (extension **.gif**),
- ▷ JPEG (extension **.jpg**),
- ▷ PNG (extension **.png**).

Veillez particulièrement à ce que le nom de vos fichiers ait une terminaison indiquant leur format : **.gif**, **.jpg** ou **.png**. Si vous installez un fichier dont le nom ne contient cette extension, le système ne saura pas utiliser l'image.

◇ ◇ ◇

Étape 1 : Installation des images sur le serveur

Avant de pouvoir insérer vos images à l'intérieur du texte, il faut bien entendu installer ces images sur le serveur. Cela se fait, dans SPIP, par l'interface graphique.



Lorsque vous «modifiez» un article ou une brève, la colonne de gauche vous propose une interface intitulée : «Ajouter une image». Cela se présente sous la forme d'un *champ* suivi d'un bouton nommé, suivant les navigateurs, «Browse», «Sélectionner», «File», «Fichier» ...

Lorsque vous cliquez sur ce bouton, une interface s'ouvre, vous permettant de visiter votre disque dur et d'indiquer quel fichier graphique vous voulez sélectionner.

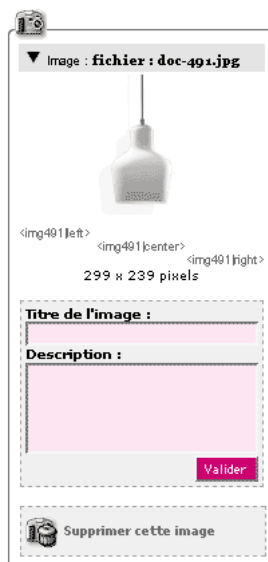
Cela fait, cliquez sur le bouton intitulé «Télécharger».

Si l'opération a réussi, votre image apparaît dans la colonne de gauche, complétée de plusieurs indications ...

◇ ◇ ◇

Étape 2 : Les informations liées à votre image

Une fois votre image envoyée au serveur, une case apparaît sur la gauche de l'écran. Il y a là toutes les informations nécessaires qui la concernent. (Une partie de ces informations apparaît masquée, cliquer sur le triangle pour «déplier» la boîte d'information.)



▷ *Affichage sous forme de vignette.*

Une prévisualisation de votre image apparaît. Si l'image est de grande taille (plus de 200 pixels de large), c'est une version de taille réduite qui est affichée.

▷ *Raccourcis SPIP.*

Voir ci-après : SPIP vous rappelle les 3 «raccourcis» qui vous permettent d'insérer cette image à l'intérieur de votre texte.

Notez que chaque image est «numérotée» ainsi : «IMG1», «IMG2» ... Ces «raccourcis» sont utilisés dans la troisième étape.

▷ *Taille de l'image.*

Juste au-dessus de l'image, la largeur et la hauteur de votre image (en pixels - ou «points») sont rappelées.

▷ *Titre et description de l'image.*

Vous pouvez, si vous le désirez, indiquer un nom et une description pour chaque image. Par exemple une explication, ou une mention du copyright du photographe ...

▷ *Supprimer cette image.*

Comme son nom l'indique, le bouton «Supprimer cette image» permet d'effacer ce fichier, si vous avez fait une erreur de manipulation, ou si vous décidez finalement de ne pas utiliser l'image dans ce texte. Il est conseillé d'effacer les images inutilisées, afin d'éviter d'encombrer votre serveur avec des fichiers inutiles.

◇ ◇ ◇

Étape 3 : Insérer une image à l'intérieur du texte

A ce stade, les fichiers graphiques sont bien présents sur le serveur, mais il reste à indiquer à quel endroit de votre texte vous voulez les insérer. Pour cela, inutile de faire du HTML, SPIP vous propose un «raccourci» permettant d'insérer votre image simplement.

▷ *Images sans commentaire*

Pour chaque image, voyez la mention des 3 raccourcis :

★ `<img1|left>`

★ `<img1|center>`

★ `<img1|right>`

Recopiez l'un de ces raccourcis (le nombre correspond au numéro de l'image, il change donc pour chaque fichier), et recopiez-le à l'intérieur de la case «Texte», là vous désirez le situer dans votre article. «left» aligne l'image à gauche, «right» à droite, et «center» place votre image au centre du texte.

Lors de l'affichage à l'écran, SPIP remplacera ces raccourcis par le code HTML correspondant, en calculant automatiquement la taille des images.

▷ *Images avec titre et description*

Si vous avez indiqué un titre et/ou une description, les mentions `<img...>` sont remplacées par :

★ `<doc1|left>`

★ `<doc1|center>`

★ `<doc1|right>`

Elles s'utilisent de la même façon que ci-dessus ; cependant, lorsque vous insérez un «raccourci» de ce type, SPIP insère dans votre texte non seulement l'image, mais le titre et la description que vous lui avez donnée. Votre image apparaît ainsi avec, éventuellement, une explication et des mentions de copyright, le nom de l'artiste, etc.

12.5.2 Joindre des documents

L'interface de SPIP vous permet de proposer sur votre site des fichiers multimédia (son, vidéo, textes ...).

Les rédacteurs peuvent joindre des documents aux articles. Ces documents peuvent être soit présentés à la suite du texte (à la façon de «pièces jointes»), soit présentés à l'intérieur du texte sous la forme d'une vignette de prévisualisation.

Les administrateurs du site peuvent, de plus, installer des documents directement dans les rubriques.

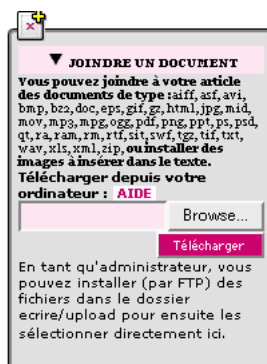
Notez bien la différence importante entre ces deux utilisations : joints aux articles, les documents sont des «pièces jointes», qui n'ont pas d'intérêt sans l'article auquel ils sont associés (dans la navigation dans le site, on peut

consulter de tels fichiers à partir des articles) ; lorsqu'ils sont installés directement dans les rubriques, les documents deviennent des éléments du site comparables aux articles et aux brèves, et non plus des compléments d'information.

◇ ◇ ◇

Étape 1 : Installation des documents sur le serveur

L'installation des fichiers sur le serveur se fait au travers de l'interface «Joindre un document» pour les articles, et «Publier un document dans cette rubrique» pour les rubriques.



Vous remarquerez que, pour les articles, cette interface apparaît à deux endroits différents : en bas de la page de chaque article, et dans la colonne de gauche (sous les images) lorsque vous modifiez un article. Ces interfaces ont exactement la même fonction, mais vous utiliserez l'une ou l'autre en fonction de vos besoins. Pour les rubriques, l'installation des documents se fait sur la page de la rubrique concernée.

Avant d'installer vos fichiers, vous devez les créer sur votre ordinateur. L'interface d'envoi des documents vous rappelle la liste des formats autorisés sur ce système. *Veillez absolument à nommer vos fichiers avec la terminaison correcte (par exemple, «xxxxx.mp3» pour un fichier au format MP3.*

L'interface est la même que pour les images : le bouton «Fichier», ou «File», ou «Parcourir», «Browse» (selon les navigateurs) ouvre une fenêtre qui vous permet de sélectionner le fichier sur votre disque dur. Une fois ce fichier sélectionné, cliquez sur «Télécharger» pour envoyer le fichier.

Attention : selon la taille de votre fichier, cette opération peut prendre du temps. Notez aussi que, selon les réglages de l'hébergeur de votre site, les fichiers trop gros pourront être refusés ; dans ce cas, vous pourrez contourner cette limitation en installant vos fichiers par FTP (section 12.5.3 page 252).

◇ ◇ ◇

Étape 2 : Informations et vignette

Une fois le fichier transféré sur le serveur, une boîte d'information apparaît.



Plusieurs opérations peuvent y être réalisées.

▷ *Vignette de prévisualisation*

Cette notion est très importante : contrairement aux images, que l'on insère dans le corps du texte, les documents n'apparaissent pas directement. Une vignette de prévisualisation est présentée au visiteur, sur laquelle il pourra cliquer pour obtenir le document correspondant.

La partie supérieure de la boîte d'information vous permet de choisir la vignette de prévisualisation. Vous pouvez opter pour une *vignette par défaut*, ou installer un *logo personnalisé*.

La vignette par défaut est installée automatiquement par le système, en fonction du format du document. L'avantage de laisser une telle vignette est que la présentation des documents d'un même type sur l'ensemble de votre site sera uniforme.

Si vous préférez, vous pouvez installer un logo (de taille réduite de préférence, et au format GIF, JPG ou PNG), qui apparaîtra à la place de la vignette par défaut. Une fois un tel logo installé, un lien «Supprimer la vignette personnalisée» vous permettra de revenir à la vignette par défaut si nécessaire.

- ▷ Dans la page de modification des articles, les «raccourcis» vous permettant d'insérer le document dans le corps du texte sont présentés, identiques à ceux des images.
- ▷ La partie inférieure vous permet de donner un titre et de fournir un descriptif pour votre document. Inutile d'indiquer ici le format et le poids du fichier multimédia, cette information sera fournie automatiquement par le système de publication.
- ▷ Enfin, le bouton «Supprimer ce document» permet d'effacer les documents inutiles.
Notez bien : il est impératif de supprimer les documents non désirés, faute de quoi ils apparaîtront sur le site public.
- ▷ **Dans le cas des documents installés dans les rubriques**, vous pouvez de plus modifier la date de mise en ligne du document (sur le même principe que vous modifiez la date de publication d'un article ou d'une brève). Une fois ces réglages effectués, les documents des rubriques sont immédiatement disponibles sur le site public (il n'est pas nécessaire de les «valider» comme les brèves ou les articles).

◇ ◇ ◇

Étape 3 : Insérer les documents dans le texte des articles

Pour les documents associés aux articles, vous pouvez vous contenter de les installer et de préciser les informations (étapes 1 et 2 ci-dessus). Lorsque vous publierez l'article, ces documents apparaîtront à la suite du texte sous la forme d'une liste de *documents joints*.

Cependant, vous pouvez également décider d'insérer les vignettes de prévisualisation à l'intérieur du texte. Vous obtiendrez ainsi des images cliquables à l'intérieur de l'article.

Ici, la procédure est exactement la même que pour les images, à la différence près que les vignettes seront des éléments cliquables. Insérez un raccourci de la forme `<imgxx|yy>` ou `<docxx|yy>` selon que vous voulez afficher uniquement la vignette, ou bien le titre et le descriptif.

Notez bien : les documents que vous aurez installé à l'intérieur du texte n'apparaîtront plus *sous l'article*. Pour les articles, il y a donc deux emplacements où apparaissent les documents : à l'intérieur du texte (vignette cliquable), ou à la suite de l'article sous la mention «Document joint».

◇ ◇ ◇

Cas particulier : Les documents vidéo et sonores

Certains formats de fichiers multimédia sont conçus pour être affichés directement dans une page Web (par exemple une vidéo insérée directement dans l'article).

Pour pouvoir insérer de tels documents dans le corps de l'article, non plus sous forme de vignette cliquable, mais en tant qu'animation multimédia, vous devez indiquer ses dimensions : largeur et hauteur absolument supérieures à zéro (pour les fichiers audio, on choisira pour largeur la dimension que l'on souhaite attribuer au curseur de défilement, et une hauteur réduite, par exemple 25 pixels).

Notez bien : les cases vous permettant d'indiquer les dimensions n'apparaissent que pour les documents dont le fichier correspond à certains formats acceptés par SPIP pour l'intégration à l'intérieur des articles (notamment : avi, quicktime, real, flash).

Une fois ces dimensions fixées, un raccourci SPIP supplémentaire vous sera proposé, de la forme `<embxx|yy>` (pour mémoire : «**embed**»).

Si vous connaissez bien le fonctionnement de ces types d'inclusion, sachez que vous pouvez ajouter des paramètres supplémentaires, par exemple :

```
<emb54|center|autostart=true|quality=hight>
```

12.5.3 Installer des fichiers par FTP

Certains serveurs interdisent l'envoi de fichiers au travers d'un interface Web. Surtout, il peut être pénible d'envoyer des fichiers lourds de cette façon. SPIP permet de contourner ces limitations en installant directement les fichiers que vous souhaitez utiliser comme images ou comme documents par FTP.

Cette opération est donc évidemment réservée aux personnes qui possèdent les codes de connexion au serveur par FTP.

▷ Le dossier /ecriture/upload

Il suffit, avec votre logiciel de FTP habituel, d'installer vos fichiers (images, documents multimédia) dans le dossier /ecriture/upload de votre site SPIP.

▷ Le menu déroulant des fichiers

Cela fait, un menu déroulant apparaîtra automatiquement après l'interface de téléchargement des fichiers par le Web, vous proposant la liste des fichiers contenus dans ce dossier. Il vous suffit de sélectionner le fichier qui vous intéresse, et de valider votre choix.

Si l'opération est réussie, pensez à supprimer ce fichier du dossier /ecrire/upload (le système en a créé une copie à un autre endroit du serveur, votre fichier d'origine n'est donc plus utile), afin de ne pas allonger indéfiniment la liste du menu déroulant.

▷ **Tous les fichiers à la fois**

Si vous installez plusieurs fichiers à la fois dans le dossier /ecrire/upload, une fonctionnalité supplémentaire vous est proposée dans l'interface du site : vous pourrez installer tous ces fichiers en une seule opération. Cela peut se révéler pratique pour créer rapidement des portfolios.

12.6 Les mots-clés

12.6.1 Principe des mots-clés

L'une des limitations importantes de SPIP est sa structure hiérarchique : chaque article ne peut se trouver que dans une seule rubrique, ce qui pose parfois des difficultés de classement.

Les mots-clés offrent un moyen de navigation transversal à l'intérieur du site. En associant un ou plusieurs mots-clés à un article, on dispose en effet d'un moyen de créer des liens avec d'autres articles aux thèmes similaires, mais situés dans d'autres rubriques.

Les mots-clés n'ont réellement d'intérêt que si chaque mot est associé à plusieurs articles, afin de pouvoir relier ces différents articles entre eux.

Seuls les administrateurs peuvent créer et modifier les mots-clés.

Fréquemment, la structure des rubriques, si elle est bien conçue, permet de se passer des mots-clés : les articles de thèmes similaires se trouvent simplement dans les mêmes rubriques, il est alors inutile d'ajouter des mots-clés pour indiquer le thème de chacun. Les administrateurs peuvent donc, dans la page de Configuration précise, désactiver totalement l'utilisation des mots-clés (section 12.10 page 259).

12.6.2 Les mots-clés

Afin de diversifier la navigation dans le site, il est possible d'associer des mots-clés (section 12.6.1 page 253) aux articles, aux brèves et aux sites référencés. Ainsi le visiteur du site pourra non seulement naviguer de rubrique en rubrique, mais également d'un article traitant d'un thème (indiqué par un mot clé) à un autre article associé au même mot-clé.

Il est possible d'indiquer, pour chaque article, brève ou site, autant de mots-clés que nécessaire.

Un menu déroulant indique la totalité des mots-clés du site. Son utilisation est très simple. *Attention : lorsqu'il existe plus de 50 mots-clés, ce menu déroulant est remplacé par un moteur de recherche : indiquez le mot désiré, et cliquez sur «Recherche».*

N.B. Seuls les administrateurs peuvent créer des mots-clés, à partir de la page qui est consacrée à leur gestion (bouton «Les mots-clés» dans l'interface de navigation supérieure).

Les administrateurs peuvent désactiver l'utilisation des mots-clés pour l'ensemble du site, via l'interface «Configuration précise» (section 12.10 page 259).

12.6.3 Les groupes de mots

Lorsque l'on utilise beaucoup de mots-clés, il peut devenir difficile de les manipuler rapidement. C'est pourquoi on peut créer des groupes réunissant ces mots. L'interface devient alors plus claire (par exemple, un groupe «Pays» regroupera «Namibie», «Allemagne», «Pérou», tandis qu'un groupe «Thèmes» regroupera «Chômage», «Poésie», «Animaux» ...).

12.7 Les sites référencés

12.7.1 Référencer un site

SPIP propose un système complet permettant de gérer des listes de liens vers d'autres sites. Ce système est très complet et permet notamment :

- ▷ de regrouper ces listes dans des rubriques (dans les mêmes rubriques que des articles, ou des rubriques spécifiques dédiées à cet usage, à la façon d'un annuaire de liens) ;
- ▷ d'attribuer un logo à chaque site ;
- ▷ d'attribuer des mots-clés pour chaque site référencé ;
- ▷ d'ajouter un descriptif personnalisé pour chaque site.

On peut de plus, pour les sites qui l'autorisent, récupérer automatiquement les derniers articles publiés (voir «Les sites syndiqués» section [12.7.2](#) page [254](#)).

Référencer un nouveau site

Un bouton «référencer un nouveau site» dans la page de chaque rubrique de votre site vous permet d'indiquer un nouveau site.

La méthode «traditionnelle» consiste à indiquer le nom du site, l'URL de cette page, puis d'insérer une description. Il est également possible de choisir la rubrique dans laquelle ce référencement sera inséré dans votre propre site.

Un cadre en bas de page vous permet de gérer, éventuellement, la syndication de contenu. Pour plus de précisions sur ce sujet, voyez l'explication sur les sites syndiqués. Pour un référencement simple, il suffit de laisser l'option «Pas de syndication».

Référencement rapide

Lors de la création d'un nouveau référencement de site, un cadre apparaît en haut de page vous permettant de référencer rapidement un site, sans indiquer vous-même son titre et son descriptif. Pour cela, il vous suffit d'indiquer l'URL de la page à référencer et de valider. Dans la mesure du possible, SPIP va récupérer automatiquement à cette adresse le titre de la page et un descriptif. Vous pourrez ultérieurement modifier ces informations.

Qui peut proposer des sites référencés ?

Dans la page de «Configuration précise du site», les administrateurs peuvent indiquer que seuls les administrateurs peuvent proposer des sites, ou les rédacteurs, ou même les visiteurs du site (dans ce cas, un formulaire sur le site public permettra aux visiteurs de proposer des sites).

Dans tous les cas, seuls les administrateurs pourront valider ces propositions de référencement. Lorsqu'un référencement de site est proposé, tous les participants à l'espace privé peuvent discuter dans un forum lié à chaque site la pertinence du référencement.

12.7.2 Sites syndiqués

Les sites fabriqués à l'aide d'un système de publication automatique (tel SPIP ou phpNuke) peuvent facilement créer un fichier indiquant en permanence la liste de leurs dernières publications. Il existe en particulier un format standardisé pour un tel fichier, intitulé «fichier backend».

Ce fichier peut être facilement analysé de manière automatique, afin de récupérer en permanence la liste des nouveautés de tels sites. De cette manière, SPIP vous permet d'afficher sur votre propre site la liste des derniers articles publiés sur d'autres sites.

Pour chaque site référencé (section 12.7.1 page 254) dans vos propres rubriques, vous avez la possibilité d'indiquer qu'il faut récupérer la liste des derniers articles publiés sur ce site. Cela, évidemment, si le site référencé propose un fichier *backend*.

Comment trouver ces fichiers «backend.php3» ?

Pour les sites gérés sous SPIP ou phpNuke, ces fichiers *backend* sont faciles à localiser : il s'agit tout simplement du fichier situé à la racine du site, et portant le nom «backend.php3» (éventuellement, «backend.php»). Par exemple, pour uZine (<http://www.minirezo.net/>), l'adresse du fichier backend est :

<http://www.minirezo.net/backend.php3>

Autres exemples de fichiers de backend :

<http://www.davduf.net/backend.php>

<http://www.vacarme.eu.org/backend.php3>

<http://www.vakooler.com/backend.php3>

Notez enfin que *L'autre portail*¹ fournit de tels fichiers pour les sites qu'il référence, même si ces sites ne comportent pas leur propre système de backend. Vous trouverez sur cette page une trentaine de fichiers backend pour les sites référencés par *L'autre portail*, ainsi qu'une poignée de fichiers thématiques.

Ajouter un site syndiqué à votre propre système

Lorsque vous référencez un site dans une de vos rubriques, en plus d'indiquer le nom du site, l'URL de sa page d'accueil et une description, vous pouvez choisir de le syndiquer (un site syndiqué est donc avant tout un site référencé, pour lequel on demande à SPIP de récupérer la liste des dernières publications).

Pour cela, sélectionnez l'option «Syndication» et indiquez l'adresse du fichier *backend* du site désiré. Après validation, un message vous indiquera immédiatement si la syndication a fonctionné correctement.

Si la syndication a échoué :

- ▷ vérifiez l'URL que vous avez indiquée pour ce site ;
- ▷ vérifiez que le site que vous souhaitez syndiquer est actuellement accessible en ligne.

Référencement rapide

La fonction de référencement rapide d'un site (indiquer directement l'URL d'un site, SPIP se chargeant de récupérer automatiquement les informations nécessaires) est particulièrement adaptée aux sites syndiqués. En effet, au lieu d'indiquer lors du référencement rapide l'URL de la page d'accueil, préférez alors indiquer l'URL du fichier *backend* de ce site : SPIP récupérera automatiquement un grand nombre d'informations et procédera directement à la syndication.

Les articles syndiqués

Lorsque la syndication fonctionne, SPIP affiche la liste des derniers articles publiés par ce site. Voyez la page de l'aide consacrée à la gestion de ces liens (section 12.7.3 page 256).

¹<http://rezo.net/backend>

Le fichier backend de votre propre site

SPIP fabrique automatiquement le fichier backend de votre propre site. Cependant, n'oubliez pas de paramétrer le nom et l'adresse de votre site sur la page de configuration précise (section 12.10 page 259).

12.7.3 Articles syndiqués

Lorsque vous réclamez la syndication d'un site (section 12.7.2 page 254), SPIP affiche la liste des derniers articles publiés sur ce site, sous la mention «Articles syndiqués tirés de ce site».

Pour chaque article, SPIP indique :

- ▷ le titre de l'article (il suffit de cliquer sur ce titre pour accéder à l'article sur son site d'origine) ;
- ▷ éventuellement les auteurs des articles ;
- ▷ éventuellement un descriptif de l'article.

Ces informations, tirées automatiquement du site référencé, ne peuvent pas être modifiées.

De plus, pour chaque article, un bouton «bloquer ce lien» vous permet d'en bloquer l'affichage sur votre propre site (parce qu'un article ne vous convient pas, parce qu'une erreur rend ce lien inopérant ...). Vous pourrez à tout moment rétablir l'affichage de cet article sur votre site.

Il est possible de demander que chacun des futurs liens en provenance du site soit a priori bloqué. Les articles ainsi récupérés ne s'afficheront qu'une fois que vous les aurez, un par un, validés «à la main».

12.7.4 Utiliser un proxy

Si votre site se situe derrière un pare-feu (*firewall*), il peut être nécessaire de configurer un proxy HTTP pour aller chercher sur Internet les nouveautés de sites syndiqués.

Ce proxy doit autoriser les requêtes vers l'extérieur sans authentification aucune.

Indiquez, dans la configuration de votre site (section «fonctionnalités de SPIP»), le proxy au format suivant :

```
http://nomproxy:port/
```

où *nomproxy* est le nom du serveur faisant office de proxy, et *port* le numéro de port TCP (le plus souvent 3128, 8080, voire 80) sur lequel envoyer les requêtes.

Attention : ce réglage est global : SPIP ira chercher tous les sites syndiqués à travers ce proxy. Si vous avez besoin de réglages plus fins, il faudra impérativement vous tourner vers l'administrateur de votre réseau.

12.8 La messagerie interne


12.8.1 Les messages entre utilisateurs

SPIP facilite l'envoi de messages entre utilisateurs du site, sans passer par l'échange d'emails.

Lorsqu'un message est «envoyé» par un utilisateur à un ou plusieurs autres utilisateurs, il devient un forum de discussion privé. Ainsi, une fois un message envoyé, une discussion peut avoir lieu, sous la forme d'un forum placé sous le texte de ce message. Dans SPIP, on peut donc considérer qu'un message est également un forum privé (c'est-à-dire qu'il est inutile de s'envoyer une multitude de messages pour discuter, il suffit de «rester» dans un même message avec son correspondant pour «discuter» grâce au forum privé qui lui est associé).

Note : Les messages entre utilisateurs et les forums qui y sont associés sont privés, c'est-à-dire que SPIP n'offre pas d'interface permettant aux administrateurs du site de voir ces messages. Cependant, nous attirons votre attention sur le caractère très relatif de cette confidentialité : un administrateur du site, doté d'un outil d'accès direct à la base de données, pourra toujours consulter ces messages.

Créer un message/discussion

La méthode la plus simple pour envoyer un message consiste à cliquer sur le logo vert  (un petit «M» complété d'un triangle) qui est affiché à côté du nom de votre destinataire. Cela ouvre immédiatement un nouveau message.

La seconde méthode consiste à utiliser le bouton «Nouveau message» présent sur chaque page de SPIP. Cela crée un nouveau message sans destinataire. Avant de pouvoir envoyer ce message, vous devrez évidemment indiquer à qui il est destiné.

L'interface de rédaction de ces messages est très simple.

La seule erreur courante consiste à oublier d'«envoyer» ce message : tant que le message est indiqué «en cours de rédaction», seul son auteur peut y accéder. Il faut donc l'envoyer pour qu'il soit présenté à ses destinataires (attention : une fois un message envoyé, il ne peut plus être modifié).

Ajouter/supprimer un destinataire

À tout moment il est possible d'ajouter un destinataire : soit pendant la rédaction du message, soit lorsqu'il est déjà envoyé (par exemple pour inscrire un nouveau participant à une discussion dans un forum qui l'intéresse).

De la même façon, on peut retirer un participant à tout moment. Un bouton «Ne plus participer à cette discussion» permet d'ailleurs à l'un des participants de s'exclure de lui-même d'une discussion.

Fixer un rendez-vous

N'importe quel message peut être transformé en rendez-vous : c'est-à-dire qu'il est lié à une date et affiché dans le calendrier de SPIP.

Peut-on écrire à n'importe quel utilisateur du site ?

Certains rédacteurs ne peuvent pas être joints (ils n'apparaissent pas dans la liste «Ajouter un participant», et leur nom n'est pas accompagné d'un logo de messagerie) :

- ▷ les rédacteurs décident individuellement de ne pas utiliser la messagerie interne ;
- ▷ les rédacteurs qui ne se sont pas connectés à l'espace privé du site depuis plus de 15 jours ne sont pas joignables (pour ces utilisateurs qui se connectent peu souvent, il est préférable d'utiliser un email classique).

12.8.2 Les pense-bête

Un pense-bête est, dans sa forme, similaire à un message : mais il ne comporte aucun destinataire. Il est uniquement accessible à son auteur.

Comme son nom l'indique, le pense-bête est destiné à noter des éléments que l'on souhaite conserver.

Placer un pense-bête dans le calendrier


L'utilisation la plus pratique du pense-bête consiste à lui fixer une date. De cette façon, le pense-bête est rappelé à son auteur jusqu'à cette date (et pendant les 24 heures qui suivent), et il apparaît dans le calendrier de SPIP.

Note : Comme pour les messages entre utilisateurs, nous attirons votre attention sur la confidentialité toute relative de ces pense-bête. SPIP n'offre aucune possibilité aux administrateurs du site de voir vos messages ; cependant, d'autres outils d'accès direct à la base de données le permettent.

12.8.3 Le calendrier

Le calendrier de SPIP présente deux sortes d'informations :

- ▷ **des informations communes à l'ensemble du site** ;
il s'agit des articles et des brèves publiés - de cette façon le calendrier permet de retrouver des articles en fonction de leur date de mise en ligne ;
- ▷ **des informations privées** ;
il s'agit des messages entre utilisateurs et des pense-bête dotés d'une date de «rendez-vous». Ce calendrier permet ainsi de rappeler et de noter des rendez-vous.

Notez que chaque jour du calendrier est accompagné d'un petit logo bleu  : ce logo permet de créer un pense-bête directement associé à cette date (on pourra régler plus précisément l'heure d'un rendez-vous grâce à l'interface d'édition de ce pense-bête).

12.8.4 Configuration personnelle de la messagerie

Chaque utilisateur du site peut configurer individuellement l'utilisation qu'il souhaite faire de la messagerie interne.

Notez bien : Les administrateurs du site peuvent décider de ne pas utiliser la messagerie sur leur site, ou simplement la liste des utilisateurs concernés. Dans tous les cas, si une option est désactivée de manière générale pour l'ensemble du site (par un administrateur), ces options ne sont pas accessibles aux rédacteurs.

Ne pas utiliser la messagerie interne

Si la messagerie interne est disponible pour l'ensemble du site (option réservée aux administrateurs), chaque utilisateur peut décider individuellement de ne pas l'utiliser (c'est-à-dire qu'il ne souhaite tout simplement pas échanger de messages avec d'autres utilisateurs par ce biais).

Ne pas utiliser la liste des rédacteurs connectés

Lorsque cette fonction est disponible (choix des administrateurs), et lorsqu'il utilise la messagerie interne, un rédacteur peut décider de ne pas participer à la «liste des rédacteurs connectés».

Cette fonction affiche en permanence la liste des rédacteurs connectés en direct, facilitant ainsi les discussions rapides entre utilisateurs. Certains utilisateurs trouvent que cette fonction est intrusive et/ou ne veulent pas être «dérangés» lorsqu'ils se connectent. Il leur suffit alors de désactiver cette option : ils n'apparaîtront plus dans la liste des rédacteurs connectés, et cette liste ne s'affichera plus sur leurs pages.

Note : Lorsqu'un administrateur indique qu'il ne souhaite pas apparaître dans la liste des rédacteurs connectés, la liste est cependant toujours affichée : il «voit» les autres, mais les autres ne le «voient» pas.

12.9 Suivi des forums

12.9.1 Suivi des forums

La page de suivi des forums est un élément important de votre site, si vous autorisez l'utilisation des forums publics (à ce sujet, voir la documentation sur la configuration des forums publics section 12.10.4 page 260). C'est là, en effet, que se déroule la modération de ces forums.

Présentation des messages

Les messages ne sont pas ici présentés selon leur structure hiérarchique (par *thread*), mais les uns à la suite des autres, dans l'ordre chronologique inversé (les plus récents en haut). En revanche, chaque message est accompagné du nom de l'article auquel il se réfère.

Supprimer un message

La principale possibilité consiste à *supprimer* les contributions. **Attention** : cette opération est irréversible. Un message supprimé ne peut plus être remis en ligne. Un message supprimé n'est cependant pas effacé de la base de données : il apparaît sur cette page encadré de rouge, avec l'indication de la date d'envoi du message et l'adresse IP de l'expéditeur.

Si vous avez configuré les forums publics pour fonctionner avec une modération *a priori*, les articles en attente sont encadrés de jaune, et vous proposent deux boutons : *Supprimer ce message* et *Valider ce message*.

12.10 Configuration précise

12.10.1 Nom et adresse de votre site

Le nom et l'adresse (URL) de votre site sont, en particulier, utilisés pour la génération du fichier «backend.php3» qui permet la syndication de votre site (c'est-à-dire l'affichage sur un site extérieur des 10 derniers articles publiés sur votre site).

L'adresse de votre site doit être celle du *dossier* de la page d'accueil, et non celle du fichier HTML correspondant ; elle doit donc se terminer par le caractère «/». Si l'adresse de votre page d'accueil est :

`http://www.monsite.net/index.html,`

l'URL de votre site doit être indiquée, ici, de la manière suivante :

`http://www.monsite.net/`

12.10.2 Contenu des articles

Les articles sont constitués d'un certain nombre d'éléments : le titre, le surtitre, le soustitre, le descriptif, le chapeau, un post-scriptum . . . Certains sites, cependant, n'ont pas besoin de tous ces éléments : les rédacteurs n'en ont aucun usage, ou l'interface graphique du site public ne les intègre pas.

Afin d'alléger l'interface de gestion du site et/ou d'empêcher purement et simplement les rédacteurs d'utiliser certains éléments que le webmestre ne souhaite pas intégrer, la page de «Configuration précise» permet de désactiver totalement l'utilisation de ces éléments.

N.B. Il est important de comprendre que, par rapport au choix interface simplifiée / interface complète (section 12.11.1 page 264), qui n'influe que sur l'interface de chaque utilisateur, le choix des options de la «configuration précise» influe sur l'ensemble des utilisateurs. Ainsi, si vous décidez de désactiver l'utilisation du surtitre, plus aucun rédacteur ni aucun administrateur ne pourra utiliser de surtitre dans ses articles.

L'interface s'adapte à la présence ou l'absence de ces éléments. Si vous désactivez l'utilisation des mots-clés, le bouton correspondant dans la barre de navigation supérieure disparaît carrément.

12.10.3 Articles post-datés

Les administrateurs ont la possibilité de modifier la date de publication en ligne d'un article (section 12.2.8 page 239) (lorsque celui-ci est déclaré «publié en ligne»).

Comment SPIP doit-il se comporter lorsque l'on fixe cette date de publication en ligne à une date future ? SPIP doit-il publier tous les articles, quelque soit la date de publication fixée (au risque d'un affichage farfelu, un article indiquant «31 mai 2001» alors qu'on n'est encore que le 21 mai), ou doit-il attendre l'échéance fixée (ici le 31 mai) avant d'afficher cet article sur le site public ?

- ▷ L'intérêt principal de cette man¹/₂uvre est de pouvoir échelonner à l'avance la publication d'une série d'articles. Cas pratique : le webmestre part en vacances pendant un mois ; s'il a déjà rédigé quelques articles, il peut les passer, dans l'espace privé, dans l'état «publié en ligne», mais leur fixer des dates de publication réparties pendant le mois de son absence. De cette façon, plutôt que de mettre en ligne un *paquet* d'articles d'un seul coup, puis plus aucun pendant un mois, le site publie régulièrement de «nouveaux» articles, malgré l'absence de leur auteur.
- ▷ Un site de science-fiction qui publierait des *chroniques martiennes* aurait, pour sa part, intérêt à désactiver cette fonction, sauf à vouloir attendre l'année 2030. Même remarque pour un mensuel dont le numéro d'avril est publié le 20 mars.

12.10.4 Fonctionnement des forums

La façon de gérer les forums publics d'un site varie énormément d'un webmestre à l'autre, en fonction notamment des besoins réels du site. Certains webmestres ne veulent pas de forums, d'autres veulent des forums d'accès libre, d'autres encore préfèrent modérer les forums *a priori*, ne publiant les messages qu'une fois qu'ils ont été validés par un administrateur.

SPIP vous permet de déterminer le fonctionnement de vos forums publics (les forums internes à la gestion du site sont toujours gérés comme des forums ouverts à tous les rédacteurs, et modérés *a posteriori*).

Désactiver les forums publics

Lorsque les forums sont désactivés, l'interface d'envoi de contributions disparaît, et les anciennes contributions ne sont plus affichées (elles ne sont pas effacées de la base, mais leur affichage est interrompu). Cette option suspend le fonctionnement des forums, même si l'affichage des forums est prévu dans la mise en page (squelettes) du site.

Vous pouvez l'utiliser en permanence (le site n'offre jamais de forums de discussion), ou temporairement (suspendre l'activité des forums, le temps de calmer un spammeur fou ou de partir en vacances à la chasse au troll ... ou encore lors d'un éventuel transfert de votre site sur un nouveau serveur).

Forums modérés *a posteriori*

Lorsque les forums sont modérés *a posteriori*, les contributions s'affichent dès qu'elles sont postées par les utilisateurs. Libre à vous, ensuite, d'utiliser la page de suivi des forums de SPIP pour modérer plus ou moins sévèrement ces messages. *La modération a posteriori est le mode par défaut de SPIP.*

Forums modérés *a priori*

Dans les forums modérés *a priori*, les contributions des utilisateurs sont stockées, mais pas affichées. Les administrateurs doivent utiliser la page de suivi des forums de SPIP pour valider (ou refuser) chaque message.

Forums accessibles sur abonnement

Si les forums sont *accessibles sur abonnement*, les utilisateurs désirant participer doivent s'inscrire en fournissant leur adresse email. Ils reçoivent alors leur identifiant par email. Pour les rédacteurs qui ont déjà accès au site privé, cet identifiant correspond à leur login habituel.

Ce mode est un compromis entre le besoin de responsabilisation (les participants doivent fournir une adresse email valide), et l'absence de modération a priori (une fois inscrits, ces utilisateurs peuvent poster leurs contributions directement).

Ce mode permet de plus d'exclure les utilisateurs qui abuseraient des forums (*black-list*). En effet, lorsque vous supprimez (via la page «Suivre les forums») une contribution postée dans le mode «sur abonnement», vous avez accès à la «fiche» (extrêmement réduite) de l'auteur de cette contribution. Vous pouvez alors simplement placer cet auteur «à la poubelle» : son identifiant ne «fonctionnera» plus, et il ne pourra en obtenir d'autre avec cette adresse email.

Attention : le mode «forum sur abonnement» suppose que votre hébergeur supporte la fonction d'envoi de mail automatique. Si ce n'est pas le cas, changez d'hébergeur ;)

12.10.5 Système de brèves

Certains sites n'utilisent pas les *brèves*, ces courts articles sans auteur. Il se peut notamment que le webmestre du site ne les ait pas incluses dans l'interface de navigation du site public.

Dans ce cas, vous pouvez purement et simplement décider de les désactiver. Les rédacteurs ne pourront pas plus en créer. L'interface s'en trouvera d'autant allégée.

12.10.6 Messagerie interne

SPIP propose un système de messagerie interne (une rubrique de la présente documentation est consacrée aux messages entre utilisateurs (section 12.8.1 page 256), aux pense-bête (section 12.8.2 page 257) et au calendrier (section 12.8.3 page 258)).

Vous pouvez décider d'utiliser tout ou partie de ce système.

Activer/désactiver la messagerie interne

Une raison pour décider de ne pas utiliser la messagerie interne est la place que les messages occupent dans la base de données : ces messages (comme par exemple les messages des forums liés à vos articles) sont conservés dans la base de données, et ainsi occupent de l'espace chez votre hébergeur. De plus, les fonctions de messagerie interne provoquent un travail supplémentaire sur la machine qui vous héberge (interrogations de la base de données) : sur une machine peu puissante (et/ou très lente), vous pouvez préférer alléger le travail de la machine en désactivant la messagerie.

Activer/désactiver la liste des rédacteurs connectés

Si vous activez cette fonctionnalité, la liste des utilisateurs connectés à l'espace privé de votre site apparaît en permanence. Cela facilite notamment l'échange de messages instantanés entre utilisateurs.

Cette fonction provoque des appels supplémentaires à la base de données ; sur une machine peu puissante, vous pouvez préférer la désactiver. Notez également que certains utilisateurs trouvent cette fonctionnalité intrusive.

Notez bien : Lorsque vous activez pour l'ensemble du site les fonctionnalités ci-dessus, il reste possible, pour chaque utilisateur, de désactiver ces fonctions pour lui-même (section 12.8.4 page 258). De cette façon, si un utilisateur trouve inutiles ou intrusives les fonctionnalités de la messagerie interne, il peut simplement désactiver cette fonction pour son usage propre.

12.10.7 Statistiques des visites

SPIP intègre un système simple vous permettant de compter et suivre le nombre de visites pour le site et pour chaque article. Il permet aussi de connaître quels sont les autres sites qui ont mené des visiteurs vers votre site et vers chaque article.

Le nombre de visites

SPIP identifie chaque jour les «visiteurs uniques» de votre site en fonction de leur adresse IP. Le système est rapide et *relativement* fiable (il s'agit d'une estimation *relativement* correcte du nombre de visiteurs du site, et non des simples «hits» ou des «pages vues» ; un visiteurs qui visite plusieurs fois la même page est bien compté pour un unique «visiteur unique»).

Les entrées directes, ou referers

On nomme «entrée directe» une arrivée sur le site ou sur la page d'un article depuis un autre site Web qui affiche un lien hypertexte vers votre propre site (ce site étant lui-même considéré comme un «referer»).

Pour l'intégralité du site et pour chaque article, SPIP affiche la liste des principaux «referers» (les pages qui affichent un lien hypertexte vers votre site), accompagnés du nombre d'«entrées directes» (le nombre de visiteurs qui ont suivi ce lien).

◇ ◇ ◇

Un système «relativement» fiable

Un système complet d'analyse du trafic d'un site est un logiciel très gourmand en puissance et en mémoire ; le système de SPIP est donc très simplifié, afin d'être le plus rapide possible, et d'occuper peu d'espace disque sur le serveur. De plus, le comptage des «visiteurs uniques» se base sur l'adresse IP des visiteurs chaque jour, ce qui ne constitue pas la méthode la plus précise dans l'absolue ; on considère cependant que cela fournit une information *relativement* fiable.

Pour une information absolument complète sur le trafic du site, on pourra donc préférer se tourner vers un système d'analyse des statistiques plus spécialisé.

Un calcul quotidien

Le système de suivi du trafic intégré à SPIP effectue un calcul du nombre de visiteurs et des referers chaque jour (et non en temps réel). Certaines informations pourront ainsi vous sembler parfois incohérentes, car elles ne prennent pas en compte les visites de la journée en cours ; en cas de doute, la page spécifique affichant les statistiques est la plus fiable et la plus détaillée. Dans cette logique, la page des statistiques d'un article n'est disponible qu'après le premier jour de parution d'un article (les chiffres ne sont pas connus auparavant, car SPIP ne les a pas encore analysés).

Activer/désactiver les statistiques et les referers

Le comptage du nombre de visiteurs uniques ne devrait pas occuper beaucoup de place, ni utiliser beaucoup de puissance machine. Il n'y a donc d'intérêt à le désactiver que pour des serveurs très lents.

Le système de comptage des referers et des entrées directes est, lui, nettement plus gourmand. Il est donc désactivé par défaut. Il est conseillé de ne l'activer que sur les serveurs ne posant aucun problème de puissance de calcul (les serveurs qui ont déjà du mal à calculer les articles très longs ne pourront certainement pas, en plus, calculer les referers).

N.B. L'espace disque occupé et le temps de calcul utilisé pour le suivi des visites et des referers augmentent avec le trafic de votre site. Plus un site est visité, plus les besoins techniques pour effectuer ces tâches augmentent.

12.10.8 Envoi automatique de mails

Les rédacteurs et les administrateurs ne passent pas forcément leur vie dans l'espace de gestion de leur site. Afin de faciliter le travail coopératif et le suivi de la vie du site, le système peut vous prévenir par email de certains événements du site ...

Attention : certains hébergeurs désactivent la fonction qui permet l'envoi automatique de mails. Si vous êtes dans cette situation, les options suivantes ne pourront pas être activées.

Suivi des messages des forums publics

Afin de permettre aux auteurs de suivre les discussions provoquées par leurs articles, cette option permet de faire suivre automatiquement, à l'auteur d'un article, chaque message posté à la suite de son article.

Si cette option est activée, lorsqu'un message est posté dans le site public à la suite d'un article, le (ou les) auteur(s) de l'article reçoivent par mail le texte de la contribution, et le rappel de l'adresse (URL) de cet article ; d'un clic il(s) peut (peuvent) donc se rendre sur la page de l'article et éventuellement répondre aux commentaires.

Suivi de l'activité éditoriale du site

Lorsqu'un article est proposé pour la validation ou publié, vous pouvez demander à SPIP de le signaler par mail. De cette façon, les participants à la vie du site sont informés en temps réel des évolutions importantes du site.

Pour un site coopératif (plusieurs rédacteurs), nous vous conseillons de créer une liste de diffusion des rédacteurs (la fonction de liste de diffusion n'est pas fournie par SPIP), vers laquelle envoyer de tels messages.

Annnonce des nouveautés

Cette fonctionnalité de SPIP permet de créer des emails du type «Quoi de neuf?» : si vous l'activez, et après avoir fixé l'intervalle de temps entre les différentes annonces, un mail est envoyé régulièrement à l'adresse indiquée, récapitulant les derniers articles et brèves publiés.

Le fonctionnement est très simple : si vous activez cette option en indiquant un intervalle de 7 jours, SPIP enverra à l'adresse désirée, tous les 7 jours, la liste des articles et brèves publiés depuis 7 jours.

Un bouton «Envoyer maintenant» provoque l'envoi immédiat de ce mail récapitulatif (et relance un nouveau délai avant l'envoi du prochain mail).

Vous pouvez envoyer ce mail d'annonce des nouveautés vers l'adresse du webmestre principal (qui fera suivre), ou, si vous aimez les sites qui se gèrent entièrement tous seuls, directement à la liste de vos abonnés (la fonction de liste de diffusion n'est pas fournie par SPIP).

12.10.9 Moteur de recherche intégré

SPIP intègre un moteur de recherche. Lorsque celui-ci est activé, un système d'indexation des articles analyse le contenu de tous les articles. Cette opération, si elle permet ensuite des recherches extrêmement rapides, nécessite beaucoup de travail de la part du serveur qui héberge le site. Dans le cas d'un hébergeur lent, cela peut poser quelques difficultés.

Pour cette raison, vous pouvez activer ou désactiver le système d'indexation.

Les données générées par le moteur de recherche intégré à SPIP **triplent** environ l'espace disque occupé par la base de données. D'autre part, sur des systèmes lents ou chargés, l'indexation risque de donner lieu à une légère dégradation des performances, voire des erreurs d'exécution (cas extrême).

D'une manière générale, si votre site est très gros, nous vous conseillons de ne pas utiliser le moteur de recherche intégré à SPIP, et de vous orienter vers des produits spécialisés, tels que ht://Dig².

Notez aussi que le moteur de recherche n'indexe pas toutes les pages d'un coup. Si vous l'activez alors que votre site contient déjà un grand nombre d'articles, il faudra attendre que votre site enregistre un nombre de connexions *grosso modo* égal au nombre de textes à indexer pour que le moteur soit à jour.

12.11 Configuration de l'interface personnelle

12.11.1 Interface simplifiée / complète

Chaque utilisateur de SPIP peut modifier son propre affichage (sans influencer, à l'inverse de la page de «Configuration précise» (section 12.10 page 259), sur l'affichage pour les autres utilisateurs.)

De nombreuses fonctions de SPIP sont rarement (ou pas du tout) utilisées par certains rédacteurs. Passer en mode «interface simplifiée» permet donc d'alléger l'interface et de faciliter sa compréhension. Dans ce mode, seuls les éléments réellement indispensables à la gestion du site sont affichés. Par exemple, peu d'utilisateurs faisant appel à la «date de rédaction antérieure» (section 12.2.9 page 240), celle-ci n'apparaît donc pas en «affichage simplifié».

N.B. Les différences d'affichage sont beaucoup plus évidentes pour les administrateurs, ceux-ci ayant en effet à leur disposition beaucoup plus de fonctions que les rédacteurs du site.

Pour que ces modifications soient appliquées, vous devez accepter l'utilisation des cookies.

12.11.2 Le cookie de correspondance

Les administrateurs peuvent activer un cookie qui provoquera l'affichage d'informations supplémentaires lors de la visite du site public. Ces informations facilitent la gestion du site.

Recalculer cette page

Sur toutes les pages du site public apparaît un bouton «Recalculer cette page». SPIP intégrant un système de *cache*, certaines modifications peuvent ne pas apparaître immédiatement en ligne. (Les pages affichées sur le site public ne sont pas directement tirées de la base de données : elles sont calculées à intervalle régulier et stockées dans le cache.)

En *recalculant* une page, l'administrateur provoque l'affichage de la page en fonction des éléments contenus dans la base de données, sans attendre la prochaine mise à jour du cache.

²<http://www.htdig.org/>

Modifier ...

Les pages des articles, des rubriques et des brèves contiennent un bouton «Modifier cet article» (ou «rubrique» ...). Ce bouton mène directement du site public à la page de l'espace privé correspondant à cet article (ou rubrique ...). Ce bouton facilite donc la correction des erreurs repérées en ligne, ou la mise à jour de tout élément du site.

Nombre de visites

Si le système de statistiques intégré à SPIP est activé, les pages des articles sont complétées de l'information suivante : nombre de visites (estimation) et nombre de *referers* différents.

Les referers sont les liens vers cet article précis depuis l'extérieur du site (c'est-à-dire : lorsqu'un autre site a fait un lien directement vers cet article, ou lorsque l'adresse de cet article a été communiqué par email).

Identifiant de connexion

Le cookie d'admin permet aussi à SPIP de reconnaître votre navigateur lorsque vous vous reconnectez : vous n'avez donc qu'à entrer votre mot de passe pour entrer dans l'espace privé.

(NB : Si la connexion se fait par cookie - c'est le cas le plus courant -, ce cookie est posé dès votre arrivée dans l'espace privé.)

12.11.3 Se déconnecter

Le bouton «Se déconnecter» sert à annuler votre identification à l'espace privé. Lorsque vous effectuez cette opération, les informations d'identification que vous aviez indiquées pour accéder à l'espace privé du site sont «oubliées» par le système ; SPIP vous propose d'entrer à nouveau ces informations, ou de revenir à l'espace public du site.

L'intérêt principal de cette fonction est d'interdire qu'une autre personne, utilisant la même machine que vous, puisse accéder à l'espace privé du site en exploitant vos propres informations d'identification.

▷ **Vous vous connectez depuis chez vous et vous êtes la seule personne à accéder à votre ordinateur**

Dans ce cas, le fait de vous déconnecter avec cette fonction lorsque vous avez fini de travailler sur l'espace privé du site peut sembler une opération superflue. L'opération de déconnexion est évidemment conseillée, mais si vous oubliez de l'effectuer, ça n'est pas très grave.

▷ **Vous accédez au site privé depuis une machine commune à plusieurs utilisateurs** (machine au travail, dans un cyber-café ...)

Dans ce cas, il est vivement conseillé de vous déconnecter systématiquement en utilisant cette fonction lorsque vous avez terminé d'utiliser l'ordinateur. Cela interdit totalement à toute personne qui utiliserait cette machine après vous de pouvoir accéder à l'espace privé en se faisant passer pour vous.

▷ **Utiliser plusieurs identifications sur un même site**

Certains utilisateurs peuvent avoir besoin de se connecter à l'espace privé sous plusieurs identités. Dans ce cas, en utilisant cette fonction de déconnexion, ils peuvent se reconnecter immédiatement au site avec de nouvelles informations d'identification.

Historique de la documentation papier

Ce document est une tentative de création d'un manuel imprimable pour SPIP. Il s'agit de la compilation des articles publiés en ligne sur le site de SPIP³. Je ne revendique évidemment aucune propriété intellectuelle sur ce document (il ne manquerait plus que ça).

Une grande partie de ce document à été créée comme support de documentation pour un projet de site de communication en entreprise. Site tournant sur un intranet, sans accès vers l'extérieur, donc inaccessible par Internet. Comme le support existait déjà, je me suis dit qu'il pourrait intéresser d'autres utilisateurs de SPIP. Ceci explique pourquoi j'ai souhaité le proposer comme contribution au projet SPIP.

Evolution du document :

Je vais essayer de le maintenir à jour. La mise en page du document à partir de zéro m'a pris pas mal de temps. Si j'arrive à trouver un moyen d'identifier facilement les modifications d'articles ainsi que la publication de nouveaux articles sur le site de SPIP, le maintien à jour de cette version «papier» devrait être assez facile à réaliser.

Pour la petite histoire, ce document à été écrit avec L^AT_EX (compilé avec PdfT_EX plus précisément).

Philippe Charlier



³<http://www.spip.org/>

versions

20021122

Première version. Correspond aux articles et aide en ligne publiés sur le site avant le 22 novembre 2002.

A faire :

- ▷ Améliorer la qualité des images.
- ▷ Trouver la possibilité de créer des publications séparées (Trucs, FAQ technique, Manuel de référence, ...). Plutôt qu'une seule «brique monolithique» : adapter les références internes à des publications séparées.
- ▷ augmenter la précision des renvois de l'index.

Index

Symboles

#BIO	117
#CHAPO	112
#DATE	112, 114, 116, 118, 124, 125
#DATE_REDAC	112
#DESCRIPTIF	112, 114, 120–122, 124
#EMAIL	117, 118, 125
#EMBED_DOCUMENT	121
#FORMULAIRE_ECRIRE_AUTEUR ...	117, 129
#FORMULAIRE_FORUM	112, 114, 116, 118, 129
#FORMULAIRE_INSCRIPTION	129
#FORMULAIRE_RECHERCHE	129, 131
#FORMULAIRE_SIGNATURE	112, 129
#FORMULAIRE_SITE	129
#HAUTEUR	121
#ID_ARTICLE	112, 118, 125
#ID_AUTEUR	117
#ID_BREVE	116, 118
#ID_FORUM	118
#ID_MOT	120
#ID_RUBRIQUE	112, 114, 116, 118, 122
#ID_SECTEUR	112, 114, 122
#ID_SIGNATURE	125
#ID_SYNDIC	122, 124
#ID_SYNDIC_ARTICLE	124
#INTRODUCTION	112, 114, 116
#IP	118
#LARGEUR	121
#LESAUTEURS	112, 124
#LOGIN_PRIVÉ	129
#LOGIN_PUBLIC	129
#LOGO_ARTICLE	112
#LOGO_ARTICLE_NORMAL	112
#LOGO_ARTICLE_RUBRIQUE	112
#LOGO_ARTICLE_SURVOL	112
#LOGO_AUTEUR	117
#LOGO_BREVE	116
#LOGO_BREVE_RUBRIQUE	116
#LOGO_DOCUMENT	121
#LOGO_MOT	120
#LOGO_RUBRIQUE	112, 114
#LOGO_RUBRIQUE_NORMAL	114
#LOGO_RUBRIQUE_SURVOL	114
#LOGO_SITE	122
#MESSAGE	125
#NOM	117, 118, 125
#NOM_SITE	116–118, 122, 124, 125
#NOTES	112, 114, 116, 117
#PARAMETRES_FORUM	112, 114, 116, 118
#PGP	117
#POINTS	131
#POPULARITE	112, 140
#POPULARITE_ABSOLUE	140
#POPULARITE_MAX	140
#POPULARITE_SITE	140
#PS	112
#SOUS_TITRE	112
#SURTITRE	112
#TAILLE	121
#TEXTE	112, 114, 116, 118, 120
#TITRE	112, 114, 116, 118, 120, 121, 124
#TYPE	120
#TYPE_DOCUMENT	121
#URL_ARTICLE	112, 124
#URL_BREVE	116
#URL_DOCUMENT	121
#URL_RUBRIQUE	114
#URL_SITE	116–118, 122, 124, 125
#VISITES	112
{“inter”}	126
{a,b}	126
{a/b}	126
A	
ad_email	125
affdate	132
age	126, 159
age_relatif	126, 159
aligner_droite	132
aligner_gauche	132
annee	132
attribut_html	132
B	
Boucle récursive	134
BOUCLE(ARTICLES)	112

BOUCLE(AUTEURS)	117
BOUCLE(BREVES)	116
BOUCLE(DOCUMENTS)	121
BOUCLE(FORUMS)	118
BOUCLE(HIERARCHIE)	125
BOUCLE(MOTS)	120
BOUCLE(RUBRIQUES)	114
BOUCLE(SIGNATURES)	125
BOUCLE(SITES)	122
BOUCLE(SYNDIC_ARTICLES)	124
C	
centrer	132
D	
debut_	126
doublons	112, 114, 121, 122
E	
entites_html	132
exclus	112, 114
extension	121
F	
fichier	132
H	
heures	132
I	
id_article	112, 117, 118, 120, 121, 125
id_auteur	112, 117
id_breve	116, 118, 120, 121
id_document	121
id_enfant	114, 118
id_forum	118, 120
id_groupe	112, 114, 116, 118, 120, 122
id_mot	112, 114, 116, 118, 120, 122
id_parent	114, 118
id_rubrique	112, 114, 116, 118, 122, 124, 125
id_secteur	112, 114, 118, 122, 124
id_signature	125
id_syndic	120, 122, 124
id_syndic_article	124
INCLURE	135
inverse	126
J	
jour	132
justifier	132
L	
liens_ouvrants	132
M	
majuscules	132
meme_parent	114, 118
minutes	132
mode	121
moderation	122
mois	132
N	
nom_email	125
nom_jour	132
nom_mois	132
P	
par hasard	126
par num	126, 132
par points	131
par	126
plat	118
PtoBR	132
R	
recherche	112, 114, 116, 131
S	
saison	132
secondes	132
supprimer_numero	132
supprimer_tags	132
syndication	122
T	
taille_en_octets	121, 132
texte_script	132
texteb brut	132
titre	120
titre_mot	112, 114, 116, 118, 122
type	120
type_mot	114, 116, 118, 122