

# PHP 3° PARTIE :

## GESTION DE FICHIERS ET DE REPERTOIRES

1. Les fonctions de gestion de fichiers
2. Exemple
3. L'appel `fopen`
4. Accès aux répertoires
5. Exemple
6. La classe `dir`
7. Exemple
8. Travaux pratiques

Source : <http://cyberzoide.developpez.com>

# Les fonctions de gestion de fichiers

---

La manipulation de fichier se fait grâce à un identifiant de fichier.

Quelques fonctions:

**fopen(\$file [, \$mode])** : ouverture du fichier identifié par son nom **\$file** et dans un mode **\$mode** particulier, retourne un identificateur **\$fp** de fichier ou **FALSE** si échec

**fclose(\$fp)** : ferme le fichier identifié par le **\$fp**

**fgets(\$fp, \$length)** : lit une ligne de **\$length** caractères au maximum

**fputs(\$fp, \$str)** : écrit la chaîne **\$str** dans le fichier identifié par **\$fp**

**fgetc(\$fp)** : lit un caractère

**feof(\$fp)** : teste la fin du fichier

**file\_exists(\$file)** : indique si le fichier **\$file** existe

**filesize(\$file)** : retourne la taille du fichier **\$file**

**filetype(\$file)** : retourne le type du fichier **\$file**

**unlink(\$file)** : détruit le fichier **\$file**

**copy(\$source, \$dest)** : copie le fichier **\$source** vers **\$dest**

**readfile(\$file)** : affiche le fichier **\$file**

**rename(\$old, \$new)** : renomme le fichier **\$old** en **\$new**

# Exemple

---

*Affichage du contenu d'un fichier :*

```
<?php
$file = 'fichier.txt' ;
// ouverture du fichier en lecture
if($fd = fopen($file, 'r'))
{
    // tant que la fin de fichier n'est pas atteinte
    while(!feof($fd))
    {
        $str .= fgets($fd, 1024);
        /* lecture jusqu'à fin de ligne où des 1024 premiers caractères */
    }
    fclose ($fd); // fermeture du fichier
    echo $str;    // affichage
}
else
    die('Ouverture du fichier <b>$file</b> impossible.');
```

?>

# L'appel fopen

---

La fonction **fopen** permet d'ouvrir des fichiers dont le chemin est relatif ou absolu. Elle permet aussi d'ouvrir des ressources avec les protocoles HTTP ou FTP. Elle renvoie FALSE si l'ouverture échoue.

*Exemples :*

```
$fp = fopen( '../docs/faq.txt' , 'r' );
```

```
$fp = fopen( 'http://www.php.net/' , 'r' );
```

```
$fp = fopen( 'ftp://user:password@cia.gov/' , 'w' );
```

Les modes d'ouverture :

'r' (lecture seule), 'r+' (lecture et écriture), 'w' (création et écriture seule),

'w+' (création et lecture/écriture),

'a' (création et écriture seule ; place le pointeur de fichier à la fin du fichier),

'a+' (création et lecture/écriture ; place le pointeur de fichier à la fin du fichier)

# Accès aux répertoires

---

Il est possible de parcourir les répertoires grâce à ces quelques fonctions :

**chdir(\$str)** : Change le dossier courant en **\$str**. Retourne TRUE si succès, sinon FALSE.

**getcwd()** : Retourne le nom du dossier courant (en format chaîne de caractères).

**opendir(\$str)** : Ouvre le dossier **\$str**, et récupère un pointeur **\$d** dessus si succès, FALSE sinon et génère alors une erreur.

**closedir(\$d)** : Ferme le pointeur de dossier **\$d**.

**readdir(\$d)** : Lit une entrée du dossier identifié par **\$d**. C'est-à-dire retourne un nom de fichier de la liste des fichiers du dossier pointé. Les fichiers ne sont pas triés. Ou bien retourne FALSE s'il n'y a plus de fichier.

**rewinddir(\$d)** : Retourne à la première entrée du dossier identifié par **\$d**.

# Exemple

---

*Affichage des noms de fichiers contenus dans le répertoire courant :*

```
<?php
// ouverture du dossier
if ($dir = opendir('.'))
{
    // lecture d'une entrée
    while($file = readdir($dir))
    {
        echo "$file<br />"; // affichage du nom de fichier
    }
    closedir($dir); // fermeture du dossier
}
?>
```

## Remarques :

**\$dir** est un pointeur vers la ressource dossier

**\$file** est une chaîne de caractères qui prend pour valeur chacun des noms de fichiers retournés par **readdir()**

# La classe `dir`

---

Il existe un autre moyen d'accéder aux dossiers : l'utilisation de la pseudo-classe **`dir`**.

En voici les attributs :

**`handle`** : valeur du pointeur

**`path`** : nom du dossier

En voici les méthodes :

**`read()`** : équivalent à **`readdir($d)`**

**`close()`** : équivalent à **`closedir($d)`**

Constructeur :

**`dir($str)`** : retourne un objet **`dir`** et ouvre le dossier **`$str`**

# Exemple

---

*Exemple équivalent au précédent :*

```
<?php
$d = dir('.');          // ouverture du dossier courant
echo ``Pointeur: ``.$d->handle.``<br />``;
echo ``Chemin: ``.$d->path.``<br />``;
while($entry = $d->read()) // lecture d'une entrée
{
    echo $entry.``<br />``;
}
$d->close();           // fermeture du dossier
?>
```



# Travaux pratiques

---

1. On vous demande de réaliser une page `visite1.php` qui affiche le nombre de fois que cette page a été visitée.

Le nombre de visite sera stocké dans un fichier qui sera mis à jour à chaque chargement de la page.

Contraintes : le nombre de visite sera limité à une taille de 8 octets. La fonction de comptage sera placée dans un script `compteur1.inc.php`.

Quelles sont les limites de ce script ?

2. Créer les pages `visite21.php` et `visite22.php` modifier la fonction de comptage afin de comptabiliser le nombre de visites pour chaque page vue. Chaque page affiche alors son propre compteur.

Contraintes : on peut penser qu'il y aura d'autres pages à l'avenir et que certaines pages soient renommées. La nouvelle fonction de comptage sera placée dans un script `compteur2.inc.php`.