

Manipuler des répertoires en Php

Catégorie : Php

Publié par Jean-Marc [JM] le 8/5/2007

Dans une application internet, il est courant de devoir manipuler des données externes au programme. Ce tutoriel consiste essentiellement en une énumération de fonctions applicables aux répertoires. Mais rassurez-vous, nous mettrons nos nouvelles connaissances en application au travers d'un prochain cours !

Pour manipuler des répertoires, vous pouvez procéder de deux manières différentes... Celle que vous choisirez dépendra de vos habitudes de programmation. Présentement, la plupart d'entre vous choisiront certainement les fonctions listées si dessous; mais sachez qu'il existe également une classe *dir*, pour les habituées de la programmation orientée objet.

`chdir()` - Changer de répertoire
`int chdir (string directory)`

chdir() change le dossier courant de Php pour celui passé en paramètre, la fonction retourne *true* si l'opération se passe bien, et *false* pour le cas contraire.

`chroot()` - Changer la racine
`int chroot (string directory)`

chroot() permet de changer la racine du script en cours, et la remplacer par son paramètre. Si tout s'est bien passé, **chroot()** retourne *true*, ou *false* s'il n'a pas pu modifier la racine.

`opendir()` - Ouvrir un répertoire
`int opendir (string path)`

opendir() permet d'ouvrir un répertoire, dont vous avez passé le nom en paramètre de la fonction; il retourne un pointeur vers celui-ci. Si une erreur intervient (un répertoire non accessible, par exemple), une erreur Php est générée et la valeur *false* est retournée.

`closedir()` - Fermer un répertoire
`void closedir (ressource dir_handle)`

closedir() permet de fermer le pointeur de répertoire qu'il prend en paramètre. Celui-ci doit avoir été précédemment ouvert avec **opendir()**.

`getcwd()` - Connaître le répertoire de travail
`string getcwd (void)`

getcwd() permet de retourner le nom du répertoire courant.

`readdir()` - Lire une entrée de répertoire
`string readdir (ressource dir_handle)`

readdir () retourne le nom de fichier suivant dans le répertoire. Attention, les noms de fichier sont renvoyés dans un ordre imprévisible; si vous les souhaitez dans un ordre précis, vous devrez les stocker dans un tableau et seulement ensuite effectuer le tri désiré.

Voici un court programme permettant d'ouvrir le contenu d'un répertoire et d'afficher les noms de tous les fichiers qu'il contient :

```
<?php
    $r = opendir('.') ; // pointeur sur le répertoire courant

    echo ("Liste des fichiers dans ce répertoire :<br />");
    while($fichier = readdir($r))
        echo ("$fichier<br />");
    closedir($r);
?>
```

dir - Classe dossier
new dir (string directory)

Php accepte une classe *dir* qui représente un répertoire. Elle comporte trois méthodes :

- >**rewind** : renvoie le pointeur au début du répertoire.
- >**read** : retourne le prochain nom de fichier du répertoire.
- >**close** : ferme le répertoire

Notez qu'elle comporte, en outre, deux propriétés :

- >**path** : chemin du répertoire courant.
- >**handle** : permet de connaître le pointeur sur le répertoire courant.

Ci-dessous, vous pouvez retrouver un petit exemple permettant de lister les fichiers d'un répertoire avec la classe *dir*.

```
<?php
$d = dir("/etc");
echo "Pointeur: ".$d->handle."<br>\n";
echo "Chemin: ".$d->path."<br>\n";
while($entry=$d->read()) {
    echo $entry."<br>\n";
}
$d->close();
?>
```