## La gestion des fichiers avec PHP

Avec PHP, la création ou la lecture de fichiers est, une fois de plus, assez simple. Il existe une multitude de fonctions dédiées à l'utilisation des fichiers. La communication entre le script PHP et le fichier est repérée par une variable, indiquant l'état du fichier et que l'on peut passer en paramètre aux fonctions spécialisées pour le manipuler.

## La fonction *fopen()*

La fonction de base est la fonction fopen(). C'est elle qui permet d'ouvrir un fichier, que ce soit pour le lire, le créer, ou y écrire. Voilà sa syntaxe :

entier fopen(chaine nomdufichier, chaine mode);

Le mode indique le type d'opération qu'il sera possible d'effectuer sur le fichier après ouverture. Il s'agit d'une lettre (en réalité une chaîne de caractères) indiquant l'opération possible :

* **r** (comme *read*) indique une ouverture en lecture seulement
* **w** (comme *write*) indique une ouverture en écriture seulement (la fonction crée le fichier s'il n'existe pas)
* **a** (comme *append*) indique une ouverture en écriture seulement avec ajout du contenu à la fin du fichier (la fonction crée le fichier s'il n'existe pas)

Lorsque le mode est suivi du caractère **+** celui-ci peut être lu et écrit. Enfin, le fait de faire suivre le mode par la lettre **b** entre crochets indique que le fichier est traité de façon binaire.

Voici un tableau récapitulant l'ensemble des modes de fichiers possibles :

| Mode | Description |
| --- | --- |
| r | ouverture en lecture seulement |
| w | ouverture en écriture seulement (la fonction crée le fichier s'il n'existe pas) |
| a | ouverture en écriture seulement avec ajout du contenu à la fin du fichier (la fonction crée le fichier s'il n'existe pas) |
| r+ | ouverture en lecture et écriture |
| w+ | ouverture en lecture et écriture (la fonction crée le fichier s'il n'existe pas) |
| a+ | ouverture en lecture et écriture avec ajout du contenu à la fin du fichier (la fonction crée le fichier s'il n'existe pas) |

Voici des exemples d'utilisations possibles de cette fonction :

$fp = fopen("../fichier.txt","r"); //lecture  
//écriture depuis début du fichier  
$fp = fopen("ftp://phpfrance.com/pub/fichier.txt","w");  
  
//écriture depuis fin du fichier  
$fp = fopen("http://igalaxie.com/fichier.txt","a");

De plus, la fonction fopen permet d'ouvrir des fichiers présents sur le web grâce à leur URL. Voici un script permettant de récupérer le contenu d'une page d'un site web :

<?  
$fp = fopen("http://www.commentcamarche.net","r"); //lecture du fichier  
while (!feof($fp)) { //on parcourt toutes les lignes  
 $page .= fgets($fp, 4096); // lecture du contenu de la ligne  
}  
  
?>

Il est généralement utile de tester si l'ouverture de fichier s'est bien déroulée ainsi que d'éventuellement stopper le script PHP si cela n'est pas le cas :

<?  
if (!$fp = fopen("http://www.commentcamarche.net","r")) {  
echo "Echec de l'ouverture du fichier";  
  
exit;  
  
}  
  
else {  
// votre code;  
  
}  
  
?>

Un fichier ouvert avec la fonction *fopen()* doit être fermé, à la fin de son utilisation, par la fonction ***fclose()*** en lui passant en paramètre l'entier retourné par la fonction *fopen()*

## Lecture et écriture

Une fois que le fichier a été ouvert avec le mode désiré, il est possible de lire son contenu et d'y écrire des informations grâce aux fonctions :

* **fputs()** (aussi parfois appelée **fwrite()**, les deux noms sont équivalents, on parle d'alias) permettant d'écrire une chaîne de caractères dans le fichier
* entier fputs(entier Etat\_du\_fichier, chaine Sortie);
* La fonction *fputs()* renvoie le nombre de caractères effectivement écrits dans le fichier
* **fgets()** permettant de récupérer une ligne du fichier
* chaîne fgets(entier Etat\_du\_fichier, entier Longueur);
* Le paramètre *Longueur* désigne le nombre de caractères maximum que la fonction est sensée récupérer sur la ligne. La fonction *fgets()* renvoie 0 en cas d'échec, la chaîne dans le cas contraire

Etant donné que la fonction *fgets()* récupère à chaque appel une nouvelle ligne du fichier, il est essentiel, pour récupérer l'intégralité du contenu d'un fichier de l'insérer dans une boucle*while*.

Ainsi, on utilise la fonction ***feof()***, fonction testant si la fin du fichier n'a pas été atteinte, en tant que test de la boucle *while*. De cette façon, tant que la fin du fichier n'a pas été atteinte, on lit la ligne suivante du fichier...

<?  
if (!$fp = fopen("fichier.txt","r")) {  
echo "Echec de l'ouverture du fichier";  
  
exit;  
  
}  
  
else {  
 while(!feof($fp)) {  
 // On récupère une ligne  
 $Ligne = fgets($fp,255);  
  
 // On affiche la ligne  
 echo $Ligne;  
  
 // On stocke l'ensemble des lignes dans une variable  
 $Fichier .= $Ligne;  
  
 }  
 fclose($fp); // On ferme le fichier  
}  
  
?>

Pour stocker des infos dans le fichier, il faut dans un premier temps ouvrir le fichier en écriture en le créant si il n'existe pas. On a donc le choix entre le mode 'w' et le mode 'a'. On préférera le second puisque le pointeur se trouve en fin de fichier (autrement dit on écrit à la suite de ce qui se trouve dans le fichier au lieu d'écraser le contenu existant éventuellement déjà).

<?  
$fp = fopen("php\_8\_fichier.txt","a"); // ouverture du fichier en écriture  
fputs($fp, "\n"); // on va a la ligne  
fputs($fp, "$nom|$email"); // on écrit le nom et email dans le fichier  
fclose($fp);  
  
?>

Voici un petit script permettant de récupérer le titre d'une page Web (le texte compris entre les balises <TITLE> et </TITLE>). Il utilise les [expressions régulières](http://www.commentcamarche.net/contents/php/phpreg.php3) pour localiser le texte.

<?  
$fp = fopen("http://www.commentcamarche.net","r"); //lecture du fichier  
while (!feof($fp)) { //on parcourt toutes les lignes  
 $page .= fgets($fp, 4096); // lecture du contenu de la ligne  
}  
  
$titre = eregi("<title>(.\*)</title>",$page,$regs); //on isole le titre  
echo $regs[1];  
  
fclose($fp);  
  
?>

## Les tests de fichiers

PHP fournit de nombreuses fonctions permettant de faire des tests sur les fichiers pour connaître leurs propriétés. Voici la liste des fonctions des tests :

* **is\_dir()** permet de savoir si le fichier dont le nom est passé en paramètre correspond à un répertoire
* booléen is\_dir(chaine Nom\_du\_fichier);
* La fonction *is\_dir()* renvoie 1 si il s'agit d'un répertoire, 0 dans le cas contraire
* <?  
  if (!is\_dir("install")) {  
  echo "Il ne s'agit pas d'un répertoire  
  ";  
    
  }  
    
  else {  
  echo "Il s'agit bien d'un répertoire  
  ";  
    
  }  
    
  ?>
* **is\_executable()** permet de savoir si le fichier dont le nom est passé en paramètre est exécutable
* booléen is\_executable(chaine Nom\_du\_fichier);
* La fonction *is\_executable()* renvoie 1 si le fichier est exécutable, 0 dans le cas contraire
* **is\_file()** permet de savoir si le fichier dont le nom est passé en paramètre ne correspond ni à un répertoire, ni à un lien symbolique
* booléen is\_file(chaine Nom\_du\_fichier);
* La fonction *is\_file()* renvoie 1 si il s'agit d'un fichier, 0 dans le cas contraire
* **is\_link()** permet de savoir si le fichier dont le nom est passé en paramètre correpond à un lien symbolique
* booléen is\_link(chaine Nom\_du\_fichier);
* La fonction *is\_link()* renvoie 1 si il s'agit d'un lien symbolique, 0 dans le cas contraire

## D'autres façons de lire et écrire

Dans certains cas, il peut être rébarbatif de devoir mettre en oeuvre les fonctions *fopen()* et *fgets* pour lire l'intégralité du contenu d'un fichier. Pour cette raison PHP fournit des fonctions supplémentaires permettant de faire directement certaines opérations.

La fonction **file()** permet de retourner dans un tableau l'intégralité d'un fichier en mettant chacune de ces lignes dans un élément du tableau (rappel : le premier élément d'un tableau est repéré par l'indice 0).

Voilà sa syntaxe :

Tableau file(chaine nomdufichier);

L'exemple suivant montre comment parcourir l'ensemble du tableau afin d'afficher le fichier.

<?  
$Fichier = "fichier.txt";  
  
if (is\_file($Fichier)) {  
 if ($TabFich = file($Fichier)) {  
 for($i = 0; $i < count($TabFich); $i++)  
 echo $TabFich[$i];  
  
 }  
 else {  
 echo "Le fichier ne peut être lu...<br>";  
  
 }  
}  
  
else {  
echo "Désolé le fichier n'est pas valide<br>";  
  
}  
  
?>

La fonction **fpassthru()** permet d'envoyer le contenu du fichier dans la fenêtre du navigateur. La syntaxe de cette fonction est la suivante :

booléen fpassthru(entier etat);

Cette fonction permet en réalité d'envoyer le contenu du fichier à partir de la position courante dans le fichier, c'est-à-dire qu'il est possible par exemple de lire quelques lignes avec *fgets()*, puis d'envoyer le reste au navigateur... Le script suivant permet de parcourir tous les fichiers [HTML](http://www.commentcamarche.net/contents/html/htmlintro.php3) contenus dans votre site et d'en afficher l'arborescence :

<HTML>  
<HEAD>  
<TITLE>Affichage de l'arborescence</TITLE>  
</HEAD>  
<BODY>  
  
<?  
function ScanDir($Directory){  
if (is\_dir($Directory) && is\_readable($Directory)) {  
 if($MyDirectory = opendir($Directory)) {  
 while($Entry = readdir($MyDirectory)) {  
 if (is\_dir($Directory."/".$Entry)) {  
 if (($Entry != ".") && ($Entry != "..")) {  
 echo "<li><b>Repertoire</b>: $Directory/$Entry</li>\n";  
 echo "<ul>";  
 ScanDir($Directory."/".$Entry);  
 echo "</ul>";  
  
 }  
 }  
 else {  
 echo "<li><b>Fichier</b>: $Directory/$Entry </li>\n";  
  
 if (eregi("(\.html)|(\.htm)",$Entry)){  
 $MetaTags = get\_meta\_tags($Directory."/".$Entry);  
  
 }  
 }  
 }  
 closedir($MyDirectory);  
  
}  
  
}  
  
}  
  
$open\_basedir=".";  
  
echo "<ul>";  
ScanDir(".");  
echo "</ul>";  
  
?>  
</BODY>  
</HTML>