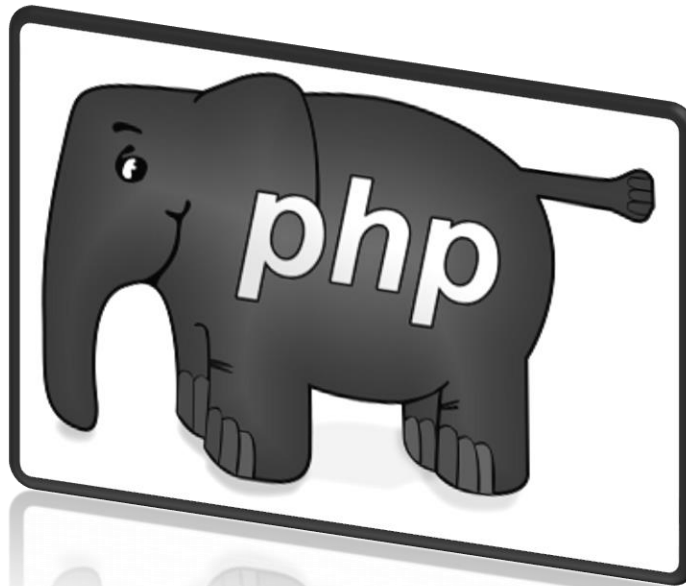


# Langage PHP

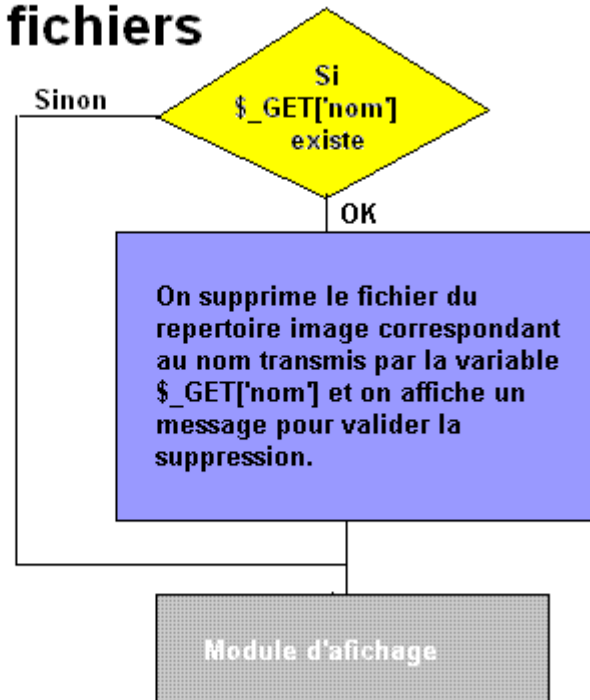


**FAQIR NADA**

([nadafaqir@gmail.com](mailto:nadafaqir@gmail.com))

# Manipuler des fichiers

## Gestion de fichiers



# Ouvrir un fichier fopen()

- **fopen(string filename, string mode)**

```
<?php
$source="fichier.txt";
if($fichier=@fopen($source,"r")) // Le @ permet de ne pas
afficher l'erreur et renvoie le statut que nous définissons en
dessous
{
echo "L'ouverture du fichier est possible car la fonction fopen
retourne TRUE<br />";
}
else
{
echo "Ouverture du fichier impossible car fopen retourne
FALSE";
}
?>
```

# Valeur Opérations permises :

mode	Description
'r'	Ouvre en lecture seule, et place le pointeur de fichier au début du fichier.
'r+'	Ouvre en lecture et écriture, et place le pointeur de fichier au début du fichier.
'w'	Ouvre en écriture seule ; place le pointeur de fichier au début du fichier et réduit la taille du fichier à 0. Si le fichier n'existe pas, on tente de le créer.
'w+'	Ouvre en lecture et écriture ; place le pointeur de fichier au début du fichier et réduit la taille du fichier à 0. Si le fichier n'existe pas, on tente de le créer.
'a'	Ouvre en écriture seule ; place le pointeur de fichier à la fin du fichier. Si le fichier n'existe pas, on tente de le créer. Dans ce mode, la fonction <a href="#">fseek()</a> n'a aucun effet, les écritures surviennent toujours.
'a+'	Ouvre en lecture et écriture ; place le pointeur de fichier à la fin du fichier. Si le fichier n'existe pas, on tente de le créer. Dans ce mode, la fonction <a href="#">fseek()</a> n'affecte que la position de lecture, les écritures surviennent toujours.
'x'	Crée et ouvre le fichier en écriture seulement ; place le pointeur de fichier au début du fichier. Si le fichier existe déjà, <b>fopen()</b> va échouer, en retournant <b>FALSE</b> et en générant une erreur de niveau <b>E_WARNING</b> . Si le fichier n'existe pas, <b>fopen()</b> tente de le créer. Ce mode est l'équivalent des options <b>O_EXCL/O_CREAT</b> pour l'appel système <b>open(2)</b> sous-jacent.
'x+'	Crée et ouvre le fichier pour lecture et écriture; le comportement est le même que pour 'x'.
'c'	Ouvre le fichier pour écriture seulement. Si le fichier n'existe pas, il sera créé, s'il existe, il n'est pas tronqué (contrairement à 'w') et l'appel à la fonction n'échoue pas (comme dans le cas de 'x'). Le pointeur du fichier est positionné au début. Ce mode peut être utile pour obtenir un verrou (voyez <a href="#">flock()</a> ) avant de tenter de modifier le fichier, utiliser 'w' pourrait tronquer le fichier avant d'obtenir le verrou (vous pouvez toujours tronquer grâce à <a href="#">ftruncate()</a> ).
'c+'	Ouvre le fichier pour lecture et écriture, le comportement est le même que pour le mode 'c'.
'e'	Définit l'indicateur close-on-exec sur le descripteur de fichier ouvert. Disponible uniquement en PHP compilé sur les systèmes conforme POSIX.1-2008.

- La fonction **fopen()** renvoie **true** si l'ouverture est possible et **false** si elle ne l'est pas.
- Afficher le contenu d'un fichier dans un navigateur
- `fpassthru($fichier)`  
`<?php`  
`$sourceI = "fichierI.txt";`  
`$fichierI = fopen($sourceI, "r");`  
`fpassthru($fichierI); // Permet d'afficher le`  
`contenu du fichier dans un navigateur`  
`?>`

# Lecture de fichiers

- **fgets(\$fichier, longueur)**
- La fonction fgets() permet d'extraire une chaîne d'une certaine longueur.
- Il faut aussi savoir que la fonction extrait la chaîne de la longueur précise définie en argument, plus un caractère et que la fonction s'arrête aux sauts de lignes

- fgets(string pointeur, string longueur)

```
<?php
```

```
$source4="fichier.txt";
```

```
$fichier4=fopen($source4,"r+");
```

```
echo fgets($fichier4,4);
```

```
?>
```



# Lecture de fichiers

- **fgetc(\$fichier,longueur)**
- fgetc() permet d'extraire le premier caractère du fichier
- fgetc(string pointeur)  
< ?php  
\$fichier = fopen("fichier.txt","r");  
\$premier = fgetc(\$fichier);  
echo "Premier Caractère : " . \$premier;  
fclose(\$fichier);  
?>

# Lecture de fichiers

- **fgetss(\$fichier,longueur)**
- La fonction fgetss() a quasiment la même utilité que la fonction fgets() sauf **qu'elle n'extrait ni les balises HTML, ni les balises PHP**
- Exemple :  
Pour un fichier contenant : '< b>Bonjour !< /b>< br>Test de la fonction fgetss', la fonction ne renverra que les caractères, la mise en forme HTML sera ignorée : 'Bonjour !Test de la fonction fgetss'



# Lecture de fichiers

- **fread(\$fichier,longueur)**
- La fonction fread() permet de lire une chaîne de caractère dans un fichier ouvert, jusqu'à la longueur indiquée en argument
- fread(string pointeur, string longueur)

```
<?php
```

```
$fichier = fopen("essai.txt","r");
```

```
$premier = fread($fichier, 10);
```

```
echo "Dix Premiers Caractères : " . $premier;
```

```
fclose($fichier);
```

```
?>
```

# Ecrire dans un fichier

- Pour écrire dans un fichier, on peut utiliser au choix la fonction **fwrite()** ou **fputs()**
- **fwrite(\$fichier, 'Texte a ajouter')**
- La fonction retourne TRUE si l'écriture se passe normalement, sinon elle retourne FALSE.  
Attention toutefois : le fichier doit bien être ouvert en mode écriture : w, a ou r+.
- fwrite(string pointeur, string chaîne, (string longueur))
- fputs(string pointeur, string chaîne, (string longueur))

# Exemple

- ```
<?php
$fichier = fopen("essai.txt","w");
if(fwrite($fichier, "TEXTE A ECRIRE"))
{
    echo "OK !";
}
else echo "Erreur";
fclose($fichier);
?>
```

# Vérifier l'existence d'un fichier

- **file\_exists("fichier.txt")**
- Pour vérifier si un fichier existe, on utilise la fonction **file\_exists()**
- **file\_exists(string fichier)**  

```
<?php  
if(file_exists("fichier.txt")) {  
    echo "Fichier existant";  
}  
else echo "Introuvable !";  
?>
```

# Copier coller des fichiers

- Pour copier un fichier, il faut utiliser la fonction **copy()**  
**copy(fichier\_depart,fichier\_destination)**
- La fonction retourne TRUE si tout se passe correctement et sinon FALSE.
- ```
<?php  
if(@copy("essai.txt","copy_fichier.txt"))  
{  
Echo "Le fichier essai.txt a été copié dans le répertoire  
complet/copy_fichier.txt."  
}  
else echo "Erreur";  
?>
```

# Renommer un fichier

- il faut utiliser la fonction **rename()**
- **rename(nom\_depart, nom\_nouveau)**
- `rename(string nom_depart, string nom_nouveau)`

```
<?php
if(@rename("essai.txt","nouveau_fichier.txt")) {
    Echo "Le fichier essai.txt a été renommé en
nouveau_fichier.txt.";
}
else echo "Erreur";
?>
```



# Supprimer un fichier

- C'est la fonction **unlink()** que l'on utilise
- **unlink(fichier)**

- unlink(string fichier)

```
<?php
```

```
if(@unlink("essai.txt"))
```

```
{
```

```
Echo "Le fichier essai.txt a été supprimé.";
```

```
}
```

```
else echo "Erreur";
```

```
?>
```

# 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

# exemple

- <?php
- 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
- `is_file` — Indique si le fichier est un véritable fichier
- 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

# exemple

- <?php
- \$file = 'wamp.exe';
- if (is\_executable(\$file)) {
- echo \$file.' est exécutable';
- } else {
- echo \$file.' n\'est pas exécutable';
- }
- ?>

# 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.

```
1  <?php
2  $Fichier = "fichier.txt";
3  if (is_file($Fichier)) {
4      if ($TabFich = file($Fichier)) {
5          for($i = 0; $i < count($TabFich); $i++)
6              echo $TabFich[$i];
7      }
8      else {
9          echo "Le fichier ne peut être lu...<br>";
10     }
11 }
12
13 else {
14     echo "Désolé le fichier n'est pas valide<br>";
15 }
16 ?>
```

# Les fonctions de système de fichiers

**\$chaine = basename(chemin\_fichier);**

retourne le nom du fichier à partir de l'adresse du fichier spécifiée.

**true | false = chgrp(nom\_fichier, groupe\_propriétaire);**

modifie le groupe propriétaire du fichier.

**true | false = chmod(nom\_fichier, \$mode);**

modifie le mode exprimé en nombre octal, du fichier.

**true | false = chown(nom\_fichier, propriétaire);**

modifie le groupe propriétaire du fichier.

**clearstatcache();**

efface la mémoire cache remplie par les fonctions lsat et stat.

**true | false = copy(fichier, nouveau\_fichier);**

copie un fichier vers une nouvelle destination.

# Les fonctions de système de fichiers

**delete(fichier);**

efface le fichier.

**\$chaine = dirname(chemin);**

retourne le nom du dossier parent.

**\$nombre = disk\_free\_space(dossier);**

retourne l'espace disponible sur le disque sur lequel est le dossier.

**\$nombre = diskfreespace(dossier);**

identique à disk\_free\_space.

**\$nombre = disk\_total\_space(dossier);**

retourne la taille totale d'un dossier.

**true | false = fclose(ID\_fichier);**

ferme un fichier indiqué par un identificateur retourné par fopen ou fsockopen.

**true | false = feof(ID\_fichier);**

teste la fin du fichier.

# FIN

- Merci pour votre attention
- <http://www.conseil-webmaster.com/formation/php/02-bases-php.php>
- [http://www.atelierphp.net/commons/show\\_all\\_exercises.php?chap=12](http://www.atelierphp.net/commons/show_all_exercises.php?chap=12)