



La gestion des fichiers dans PHP

Langage PHP avancé

El hadji Mamadou NGUER Enseignant chercheur en Informatique à l'UVS

1



Chapitre 2 : La gestion des fichiers dans PHP



Objectifs spécifiques : A la suite de ce chapitre, l'apprenant doit être capable :

1. d'inclure un fichier dans une page PHP
2. d'ouvrir et fermer un fichier dans une page PHP,
3. de décrire les modes d'ouverture de fichier,
4. de créer un fichier à partir d'une page PHP,
5. d'écrire dans un fichier,
6. et de charger un fichier dans un serveur web.

Chapitre 2 : La gestion des fichiers dans PHP



Plan de la séquence :

1. Introduction
2. L'inclusion de fichiers
3. L'ouverture et fermeture d'un fichier
4. Les modes d'ouverture d'un fichier
5. La lecture d'un fichier
6. La création d'un fichier
7. L'écriture dans un fichier
8. Le chargement de fichiers
9. Conclusion

Introduction

- La gestion des fichiers est une partie importante de toute application Web. On doit souvent inclure, ouvrir et traiter des fichiers pour différentes tâches.
- PHP contient plusieurs fonctions permettant d'inclure, créer, lire, télécharger et éditer des fichiers.
- Par exemple la fonction **readfile ()** lit un fichier et l'écrit dans le tampon de sortie. Elle renvoie le nombre d'octets lus en cas de succès.

Exemple

```
<?php  
    echo readfile("nomdufichier.txt");  
?>
```

Inclusion de fichier

- La déclaration **include** (ou **require**) copie le contenu (texte/code/ balise) d'un fichier spécifié, dans le fichier appelant l'instruction include.
- L'inclusion de fichiers est très utile lorsqu'on souhaite inclure le même code PHP/HTML/texte dans plusieurs pages d'un site Web.
- Les instructions include et require sont identiques, sauf en cas d'échec:
 - require provoque une erreur fatale (E_COMPILE_ERROR) et arrête le script
 - include ne produit qu'un avertissement (E_WARNING) sans arrêter le script

Inclusion de fichier

- Donc, si on souhaite que l'exécution continue et affiche les résultats, même si le fichier d'inclusion est manquant, utilisez l'instruction `include`.
- Sinon, dans le cas des Framework, des CMS ou d'une application PHP avec un codage complexe, on utilise toujours l'instruction `require` pour inclure un fichier sur le flux d'exécution.

Cela aidera à éviter de compromettre la sécurité et l'intégrité de votre application, au cas où un fichier clé manquerait accidentellement.

- L'inclusion de fichiers permet aussi d'éviter aux développeurs de répéter inutilement un même travail. C'est le cas des fichiers entête (header, footer et menu) inclus dans les autres pages d'un site Web.

Inclusion de fichier

Syntaxe : `include 'nomdefichier';` ou `require 'nomdefichier';`

Exemple 1: Supposons que nous disposions d'un fichier standard de bas de page appelé "footer.php", qui ressemble à ceci:

```
<?php
echo "<p>Copyright &copy; 1999-" . date("Y") . " ugb.sn</p>";
?>
```

Le fichier footer.php est inclus dans la page suivante comme suit :

```
<html>
<body>

<h1>Bienvenue dans la page de l'UGB!</h1>
<p>Du texte.</p>
<p>Encore du texte.</p>
<?php include 'footer.php';?>

</body>
</html>
```

Inclusion de fichier

Exemple 2: Inclusion d'un fichier "menu.php":

```
<?php
echo ' <a href="/default.asp">Accueil</a> -
      <a href="/html/default.asp">HTML</a> -
      <a href="/css/default.asp">CSS</a> -
      <a href="/js/default.asp">JavaScript</a> -
      <a href="default.asp">PHP</a>';
?>
```

Le fichier incluant le fichier menu.php

```
<html>
<body>

<div class="menu">
<?php include 'menu.php';?>
</div>

<h1>Bienvenue !</h1>
<p>Du texte.</p>
<p>S Encore du texte.</p>
</body>
</html>
```


Inclusion de fichier

Exemple 3: Inclusion de variables:

Supposons avoir le fichier vars.php suivant :

```
<?php  
$login='man';  
$mdp='yaw';  
?>
```

Son inclusion dans le fichier suivant permet d'utiliser les variables \$login et \$mdp.

```
<html>  
<body>  
  
<h1>Bienvenue!</h1>  
<?php  
include 'vars.php';  
echo "Login = $login et mdp = $mdp.";  
?>  
  
</body>  
</html>
```

Ouverture et fermeture d'un fichier

- La meilleure manière pour ouvrir les fichiers est avec d'utiliser la fonction `fopen ()`.

Syntaxe : `fopen(fichier, mode)`

où

fichier est le nom du fichier à ouvrir

mode est le mode d'ouverture du fichier.

- La fonction **`fclose ()`** est utilisée pour fermer un fichier ouvert.

Exemple :

```
<?php
    $myfile = fopen("fichier.txt", "r") or die("Ouverture impossible!");
    ...
    fclose($myfile);
?>
```

Les modes d'ouverture d'un fichier

- Un fichier peut être ouvert dans l'un des modes suivants:

Mode	Description
r	Ouverture en lecture seule. Le pointeur de fichier commence au début du fichier
w	Ouverture en écriture seule. Efface le contenu du fichier ou crée un nouveau fichier s'il n'existe pas. Le pointeur de fichier commence au début du fichier
a	Ouverture en écriture seule. Les données existantes dans le fichier sont préservées. Le pointeur de fichier commence à la fin du fichier. Crée un nouveau fichier si le fichier n'existe pas
x	Crée un nouveau fichier en écriture seule. Renvoie FALSE et une erreur si le fichier existe déjà
r+	Ouverture en lecture/écriture. Le pointeur de fichier commence au début du fichier
w+	Ouverture en lecture/écriture. Efface le contenu du fichier ou crée un nouveau fichier s'il n'existe pas. Le pointeur de fichier commence au début du fichier
a+	Ouverture en lecture/écriture. Les données existantes dans le fichier sont préservées. Le pointeur de fichier commence à la fin du fichier. Crée un nouveau fichier si le fichier n'existe pas
x+	Crée un nouveau fichier en lecture/écriture. Renvoie FALSE et une erreur si le fichier existe déjà

Lecture d'un fichier

Les fonctions fread()

- La fonction fread () permet de lire un fichier ouvert.
- **Syntaxe : fread(\$fichier, \$size)**

où

\$fichier est le nom du fichier à lire

\$size est le nombre maximum d'octets à lire.

Exemple :

```
<?php
```

```
$myfile = fopen("fichier.txt", "r") or die("Ouverture impossible!");  
echo fread($myfile, filesize("fichier.txt"));  
fclose($myfile);
```

```
?>
```

Lecture d'un fichier

Les fonctions **fgets()** et **eof()**

- La fonction **fgets ()** est utilisée pour lire une seule ligne à partir d'un fichier.
- La fonction **feof ()** vérifie si la "fin de fichier" (EOF) est atteinte.

Exemple :

```
<?php
```

```
$myfile = fopen("fichier.txt", "r") or die("Ouverture impossible!");
```

```
// Affichage d'une ligne jusqu'à la fin du fichier
```

```
while(!feof($myfile)) {  
    echo fgets($myfile) . "<br>";  
}
```

```
fclose($myfile);
```

```
?>
```

Lecture d'un fichier

Les fonctions **fgetc()**

- La fonction **fgetc ()** est utilisée pour lire un seul caractère à partir d'un fichier.

Exemple :

```
<?php
```

```
$myfile = fopen("fichier.txt", "r") or die("Ouverture impossible!");
```

```
// Affichage d'un caractère jusqu'à la fin du fichier
```

```
while(!feof($myfile)) {  
    echo fgetc($myfile) . "<br>";  
}
```

```
fclose($myfile);
```

```
?>
```

Création d'un fichier

- La fonction `fopen ()` est également utilisée pour créer un fichier.
- Si on utilise `fopen ()` sur un fichier qui n'existe pas, elle le créera, étant donné que le fichier est ouvert en écriture (w) ou en ajout (a).
- **Syntaxe : `fopen($fichier, "w")`**

où

\$fichier est le nom du fichier à créer

Exemple :

```
<?php
    $myfile = fopen("fichier.txt", "w");
?>
```

Ecriture dans un fichier

- La fonction **fwrite ()** est utilisée pour écrire dans un fichier.
- **Syntaxe : fwrite(\$fichier, \$chaine)** où
 \$fichier est le nom du fichier à créer
 \$chaine est la chaîne à écrire dans le fichier

Exemple :

```
<?php
$myfile = fopen("newfile.txt", "w") or die("Ouverture impossible!");
$txt = "Bonjour chers amis\n";
fwrite($myfile, $txt);
$txt = "Merci\n";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

Pour une référence complète sur les fichiers, voir :
https://www.w3schools.com/php/php_ref_filesystem.asp

Chargement de fichiers

- Le téléchargement de fichiers sur le serveur se fait en suivant les étapes suivants. Cependant, soyez prudent en autorisant le téléchargement de fichiers!
- Etape 1:** Configurer le fichier "**php.ini**" pour permettre le chargement de fichier en activant la directive file_uploads : **file_uploads = On**
- Etape 2: Créer le formulaire de chargement** permettant à l'utilisateur de choisir le fichier à télécharger:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<form action="upload.php" method="post" enctype="multipart/form-data">
```

Choisir un fichier :

```
<input type="file" name="fileToUpload" id="fileToUpload">
```

```
<input type="submit" value="Envoyer" name="submit">
```

```
</form>
```

```
</body>
```

```
</html>
```



Chargement de fichiers

Quelques règles nécessaires à observer :

Le téléchargement de fichier ne fonctionnera que si :

- Le formulaire utilise la **method = "post"**
- L'attribut **enctype = "multipart/form-data"**. Il spécifie le type de contenu à utiliser lorsque le formulaire sera soumis.
- L'attribut type de la balise <input> doit être à "file" : **type = "file"**. Il permet de montrer le champ de saisie comme un sélecteur de fichier, avec un bouton "Parcourir" à côté.
- La valeur de l'attribut action du formulaire doit être un fichier php que le formulaire enverra les données :
Action = "upload.php"

Chargement de fichiers

Etape 3 : Créer le script de téléchargement de fichier PHP

Le fichier "upload.php" contient le code de téléchargement d'un fichier:

<?php

```
// création d'une variable contenant toutes les infos utiles
$file = $_FILES["mon_fichier"];
// si un fichier a bel et bien été envoyé :
if($file || ($file != "none")) {
    // vérification de la taille et du type MIME
    if ($file["size"] <= 1024000){
        // nouveau nom, emplacement et extension du fichier :
        $name = "./images/".basename($file["name"]);
        // copie du fichier temporaire dans son nouvel emplacement :
        copy(($file["tmp_name"], $name);
    }
}
//...
```

?>

Pour plus de détails : https://www.w3schools.com/php/php_ref_filesystem.asp

Conclusion

- Au cours de cette séquence, nous avons vu comment :
 - inclure un fichier dans une page PHP
 - ouvrir et fermer un fichier dans une page PHP,
 - les modes d'ouverture de fichier,
 - créer un fichier à partir d'une page PHP,
 - écrire dans un fichier,
 - et charger un fichier dans un serveur web.
- Pour consolider toutes ses connaissances acquises, veuillez effectuer :
 - les tests de connaissance de la séquence
 - et les exercices de la fiche de TP