

Développement PHP

Partie 3 : Transmettre des données

DENIS LOEUILLET – IFA - 2017

Partie 3 : Transmettre des données

- Variables superglobales
- Transmettre des données avec l'URL
- Transmettre des données avec les formulaires
- Sessions & Cookies
- Lire et écrire dans un fichier

Partie 3 : Transmettre des données

Lire et écrire dans un fichier

1. Autoriser l'écriture de fichiers (chmod)
2. Ouvrir et fermer un fichier
3. Lire et écrire dans un fichier

Partie 3 : Transmettre des données

Lire et écrire dans un fichier

- Les variables :
 - sont simples à utiliser,
 - mais elles ne contiennent que des informations temporaires.
 - La durée de vie d'une variable n'est jamais très longue.
 - On a besoin parfois de stocker des informations définitivement.
- Il faut écrire ces informations sur le disque dur.
 - Quoi de plus logique pour cela que de créer des fichiers ?
- PHP permet justement d'enregistrer des données dans des fichiers sur le disque dur du serveur.

Partie 3 : Transmettre des données

Lire et écrire dans un fichier : autoriser l'écriture

- Pour que PHP puisse créer des fichiers :
 - il doit avoir accès à un dossier qui lui en autorise la création.
 - Il faut donner le droit à PHP de créer et modifier les fichiers, sinon celui-ci ne pourra rien faire.
- Pour créer ces droits, on dit qu'on doit modifier le CHMOD du fichier ou du dossier.
 - C'est le nom de la commande qui permet de modifier les droits sous Linux.
 - Le CHMOD est un nombre à trois chiffres que l'on attribue à un fichier (par exemple 777).
 - ✓ Selon la valeur de ce nombre, Linux autorisera (ou non) la modification du fichier.
 - Linux n'autorise généralement pas les modifications de fichiers par un script PHP.
 - ✓ Il faut donc modifier le CHMOD

Partie 3 : Transmettre des données

Lire et écrire dans un fichier : autoriser l'écriture

- Il y a trois types de personnes qui ont le droit de lire/modifier des fichiers.
 - Le propriétaire :
 - ✓ C'est l'utilisateur sous Linux qui a créé le fichier.
 - ✓ Il a en général tous les droits : lire, écrire, exécuter.
 - ✓ Selon les droits qu'il possède, le premier chiffre du CHMOD change.
 - ❖ 7 : ça veut dire qu'il a tous les droits.
 - Le groupe :
 - ✓ Ce sont les droits du groupe d'utilisateurs auquel appartient le fichier.
 - ✓ Cela correspond au deuxième chiffre du CHMOD :
 - ❖ 7 : ça veut dire qu'il a tous les droits
 - Permissions publiques :
 - ✓ Elles concernent tout le monde, c'est-à-dire même vos fichiers PHP.
 - ✓ C'est le troisième chiffre du CHMOD (par défaut 5, il faut mettre cette valeur à 7).

Partie 3 : Transmettre des données

Lire et écrire dans un fichier : autoriser l'écriture

- Si on règle le CHMOD à 777 :
 - Tous les programmes du serveur ont le droit de modifier le fichier, notamment PHP
- Il faut donc régler le CHMOD à 777 pour que PHP puisse modifier un fichier
- On peut aussi modifier le CHMOD d'un dossier.
 - Cela déterminera si on a le droit de lire/écrire dans ce dossier.
 - Cela vous sera utile si on a besoin de créer des fichiers dans un dossier en PHP.
- L'instruction PHP est : « chmod »
<http://php.net/manual/fr/function.chmod.php>
 - 2 paramètres :
 - ✓ Le chemin vers le fichier concerné
 - ✓ Le mode : les droits en lecture / écriture
 - Exemple : `chmod('fichier.php', 0777)`
 - ✓ Donne les droits en écriture et lecture aux trois types de personnes

Partie 3 : Transmettre des données

Lire et écrire dans un fichier : ouvrir / fermer

- Avant de lire/écrire dans un fichier, il faut d'abord l'ouvrir :

➤ « fopen » :

<http://php.net/manual/fr/function.fopen.php>

✓ `$monfichier = fopen (chemin_fichier, mode)`

✓ Mode : indique à PHP comment on veut ouvrir le fichier

Mode	Explication
r	Ouvre le fichier en lecture seule. Cela signifie que vous pourrez seulement lire le fichier.
r+	Ouvre le fichier en lecture et écriture. Vous pourrez non seulement lire le fichier, mais aussi y écrire (on l'utilisera assez souvent en pratique).
a	Ouvre le fichier en écriture seule. Mais il y a un avantage : si le fichier n'existe pas, il est automatiquement créé.
a+	Ouvre le fichier en lecture et écriture. Si le fichier n'existe pas, il est créé automatiquement. Attention : le répertoire doit avoir un CHMOD à 777 dans ce cas ! À noter que si le fichier existe déjà, le texte sera rajouté à la fin.

Partie 3 : Transmettre des données

Lire et écrire dans un fichier : ouvrir / fermer

- quand on a fini d'utiliser le fichier, on fait un « fclose » pour le fermer :
 - « fclose » :
<http://php.net/manual/fr/function.fclose.php>
 - ✓ fclose (variable_fichier)
 - ✓ On doit indiquer quel fichier fermer : on passe en paramètre la variable de retour créée lors du « fopen »
 - ✓ Cette fonction retourne TRUE en cas de succès ou FALSE si une erreur survient.

Partie 3 : Transmettre des données

Lire et écrire dans un fichier : lire et écrire

- Lire :
 - 2 possibilités :
 - ✓ lire caractère par caractère avec la fonction « fgetc »
<http://php.net/manual/fr/function.fgetc.php>
 - ✓ lire ligne par ligne avec « fgets »
<http://php.net/manual/fr/function.fgets.php>
 - On utilise en général « fgets »

Partie 3 : Transmettre des données

Lire et écrire dans un fichier : lire et écrire

- Lire :
 - Exemple avec un fichier ne contenant qu'une ligne

```
1 <?php
2 // 1 : on ouvre le fichier
3 $monfichier = fopen('compteur.txt', 'r+');
4
5 // 2 : on lit la première ligne du fichier
6 $ligne = fgets($monfichier);
7
8 // 3 : quand on a fini de l'utiliser, on ferme le fichier
9 fclose($monfichier);
10 ?>
```

Partie 3 : Transmettre des données

Lire et écrire dans un fichier : lire et écrire

- Lire :
 - Exemple avec un fichier ne contenant qu'une ligne
 - ✓ On ouvre le fichier `compteur.txt` en lecture/écriture (`r+`)
 - ✓ `fgets` renvoie toute la ligne (la fonction arrête la lecture au premier saut de ligne).
 - ❖ Donc notre variable `$ligne` devrait contenir la première ligne du fichier.
 - ✓ `fclose` ferme le fichier `compteur.txt`
 - Pour un fichier qui contient plus d'une ligne
 - ✓ On fait une boucle :
 - ❖ Un premier `fgets` vous donnera la première ligne.
 - ❖ Au second tour de boucle, le prochain appel à `fgets` renverra la deuxième ligne, et ainsi de suite.

Partie 3 : Transmettre des données

Lire et écrire dans un fichier : lire et écrire

- Écrire :
 - On utilise la fonction « fputs » :
<http://php.net/manual/fr/function.fputs.php>
 - ✓ Écrit une ligne dans le fichier

```
1 <?php fputs($monfichier, 'Texte à écrire'); ?>
```

- ✓ À quel endroit va-t-on écrire ?
 - ❖ Après un fgets, le curseur de PHP se trouve à la dernière ligne lue dans le fichier
 - ❖ Pour déplacer le curseur on utilise « fseek »
<http://php.net/manual/fr/function.fseek.php>
 - `fseek(variable_fichier, offset)`
 - `variable_fichier` : on passe en paramètre la variable de retour créée lors du « fopen »
 - `offset` : position désirée du curseur (int) : 0 = début du fichier

Partie 3 : Transmettre des données

Lire et écrire dans un fichier : lire et écrire

- exemple :

```
1  <?php
2  $monfichier = fopen('compteur.txt', 'r+');
3
4  $pages_vues = fgets($monfichier); // On lit la première ligne (nombre de pages vues)
5  $pages_vues += 1; // On augmente de 1 ce nombre de pages vues
6  fseek($monfichier, 0); // On remet le curseur au début du fichier
7  fputs($monfichier, $pages_vues); // On écrit le nouveau nombre de pages vues
8
9  fclose($monfichier);
10
11 echo '<p>Cette page a été vue ' . $pages_vues . ' fois !</p>';
12 ?>
```

Partie 3 : Transmettre des données

Lire et écrire dans un fichier : lire et écrire

- exemple :
 - `$monfichier = fopen('compteur.txt', 'r+');`
 - ✓ on ouvre le fichier en lecture / écriture
 - `$pages_vues = fgets($monfichier);`
 - ✓ on récupère la première ligne du fichier (contient un nombre)
 - ✓ On mémorise la ligne récupérée dans `$pages_vues`
 - `$pages_vues += 1;`
 - ✓ on ajoute 1 à la variable `$pages_vues`.
 - `fseek($monfichier, 0);`
 - ✓ on replace notre « curseur » au début du fichier (parce que sinon, il se trouvait à la fin de la première ligne et on aurait écrit à la suite)
 - `fputs($monfichier, $pages_vues);`
 - ✓ on écrit notre nouveau nombre de pages vues dans le fichier, en écrasant l'ancien nombre.
 - `fclose($monfichier);`
 - ✓ On referme le fichier

Partie 3 : Transmettre des données

Lire et écrire dans un fichier : lire et écrire

- remarque :
 - Le fichier doit exister et contenir un nombre (tel que 0) pour que ce code fonctionne.
 - De plus, si vous avez oublié de mettre un CHMOD à 777 sur le fichier compteur.txt, vous aurez l'erreur suivante :
« Warning : fopen(compteur.txt): failed to open stream: Permission denied »

Partie 3 : Transmettre des données

Lire et écrire dans un fichier

- Résumé :
 - PHP permet d'enregistrer des informations dans des fichiers sur le serveur.
 - Il faut au préalable s'assurer que les fichiers autorisent PHP à les modifier.
 - ✓ Pour cela, il faut changer les permissions du fichier (on parle de CHMOD)
 - La fonction « fopen » permet d'ouvrir le fichier
 - La fonction « fgets » permet de lire le fichier ligne par ligne
 - La fonction « fputs » permet d'écrire dans le fichier.