

TP SIN

Programmation PHP7

(Document ressource)

<http://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql/introduction-a-php>

Pré requis (l'élève doit savoir):

- Savoir utiliser un ordinateur
- Connaître le html et css

Programme

Objectif terminale :

L'élève doit être capable de réaliser un programme PHP

1. Introduction

PHP: Hypertext Preprocessor, plus connu sous son sigle PHP (acronyme récursif), est un langage de programmation libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet comme C++.

C'est un langage que seuls les serveurs comprennent. Le rôle du serveur PHP est de générer du code HTML (on peut aussi générer du CSS, mais c'est plus rare), code qui est ensuite envoyé au client de la même manière qu'un site statique.

Par rapport au JavaScript qui fonctionne directement sur le client, la page PHP doit être réactualisée à chaque fois.

2. Matériel nécessaire

Pour tester votre programme en mode local, votre ordinateur doit pouvoir lire du PHP, de ce fait il faut qu'il se comporte comme un serveur. Pour cela il suffit simplement d'installer les mêmes programmes que ceux que l'on trouve sur les serveurs qui délivrent les sites web aux internautes.

Les programmes nécessaires :

- **Apache** : c'est ce qu'on appelle un serveur web. Il s'agit du plus important de tous les programmes, car c'est lui qui est chargé de délivrer les pages web aux visiteurs. Cependant, Apache ne gère que les sites web statiques (il ne peut traiter que des pages HTML). Il faut donc le compléter avec d'autres programmes.
- **PHP** : c'est un plug-in pour Apache qui le rend capable de traiter des pages web dynamiques en PHP. En clair, en combinant Apache et PHP, notre ordinateur sera capable de lire des pages web en PHP.
- **MySQL** : c'est le logiciel de gestion de bases de données. Il permet d'enregistrer des données de manière organisée.

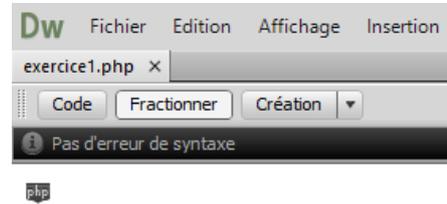
Il existe plusieurs paquets tout prêts pour Windows :

- Wamp : <http://www.wampserver.com/>
- Xampp : <https://www.apachefriends.org/fr/index.html>
- EasyPHP : <http://www.easyphp.org/>

3. La balise PHP (< ?php ?>)

Pour créer un document PHP, il faut utiliser un fichier php dans lequel on va insérer une balise php de la forme :

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Document sans titre</title>
6 </head>
7
8 <body>
9 <?php
10 //texte php
11 ?>
12 </body>
13 </html>
```



4. Utilisation des variables

Pour définir une variable, on va utiliser le symbole « \$ »

Exemple :

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Document sans titre</title>
6 </head>
7
8 <body>
9 <?php
10 $resultat=2;
11 ?>
12 </body>
13 </html>
```

Maintenant si on veut afficher la variable on va utiliser la syntaxe « echo »

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Document sans titre</title>
6 </head>
7
8 <body>
9 <?php
10 $resultat=2;
11 echo 'le résultat est égal à : ' ;
12 echo $resultat;
13 ?>
14 </body>
15 </html>
```



Autre possibilité

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Document sans titre</title>
6 </head>
7
8 <body>
9 <?php
10 $resultat=2;
11 ?>
12 <h3>
13 Le résultat est égal à :<?php echo $resultat| ?>;
14 </h3>
15 </body>
16 </html>
```

Attention :

Lorsqu'on veut ajouter du texte suivi d'une variable, on doit utiliser une concaténation grâce à des points

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Document sans titre</title>
6 </head>
7 <?php
8 $prenom='pierre';
9 echo "mon prénom est ".$prenom."";
10 ?>
11 <body>
12 </body>
13 </html>
```

On obtient :



a. Récupération d'une variable depuis une balise <Form

On veut transmettre une variable d'un fichier à un autre. Pour cela on va d'abord utiliser une balise form pour transmettre une variable avec deux possibilités :

i. La méthode GET

Exemple : Choisir un nom dans une liste et la transmettre un autre fichier





bonjour pierre:

Programme :

```
6 </head>
7
8 <body>
9 <form action="recu_get.php" method="get" name="form1" id="form1">
10 <p>
11 <label for="select">Sélectionner un nom :</label>
12 <select name="nom" id="nom">
13 <option value="pierre">pierre</option>
14 <option value="paul">paul</option>
15 <option value="jean">jean</option>
16 </select>
17 </p>
18 <p>
19 <input type="submit" name="submit" id="submit" value="Envoyer">
20 </p>
21 </form>
22 </body>
```

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Document sans titre</title>
6 </head>
7 <?php
8 $nom=$_GET['nom'];
9 ?>
10 <body>
11 bonjour <?php echo $nom ?>;
12 </body>
13 </html>
```

ii. La méthode POST

Il suffit de remplacer GET par POST dans les programmes

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Document sans nom</title>
6 </head>
7
8 <body>
9 <form action="recu_post.php" method="post" name="form1" id="form1">
10 <p>
11 <label for="select">Sélectionner un nom :</label>
12 <select name="nom" id="nom">
13 <option value="pierre">pierre</option>
14 <option value="paul">paul</option>
15 <option value="jean">jean</option>
16 </select>
17 </p>
18 <p>
19 <input type="submit" name="submit" id="submit" value="Envoyer">
20 </p>
21 </form>
22 </body>
23 </html>
```

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Document sans titre</title>
6 </head>
7 <?php
8 $nom=$_POST['nom'];
9 ?>
10 <body>
11 bonjour <?php echo $nom ?>
12 </body>
13 </html>
```

Nouveau dans PHP7

Nouvel opérateur null coalescing : ??

?? retourne le premier opérande s'il existe et s'il n'est **pas NULL**, sinon il retourne le deuxième.

```
// combinaison ?? en PHP 7
```

```
$_GET["user"] ?? $_POST["user"] ?? "anonyme";
```

b. Variable de type tableau :

Pour définir des variables de type tableau nous allons utiliser la fonction Array()

Exemple : tableau d'éléments électroniques :

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Document sans titre</title>
6 </head>
7 <?php
8 $element=Array();
9 $element[0]="résistance";
10 $element[1]="condensateur";
11 $element[2]="diode";
12 ?>
13 <body>
14 </body>
15 </html>
```

Maintenant au lieu d'utiliser des chiffres pour les indices, on peut aussi utiliser du texte

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Document sans titre</title>
</head>
<?php
$element=Array();
$element["résistance"]=2;
$element["condensateur"]=3;
$element["diode"]=5;
?>
<body>
</body>
</html>
```

Exercice2 :

On veut réaliser la somme de deux listes de chiffres (0 à 9) et afficher le résultat sur la même page en actionnant un bouton:

Première valeur sélectionnée:

Deuxième valeur sélectionnée:

La somme des deux valeurs est égale à :0

Correction :<!doctype html>

```

<html>
<head>
<meta charset="utf-8">
<title>Document sans titre</title>
</head>

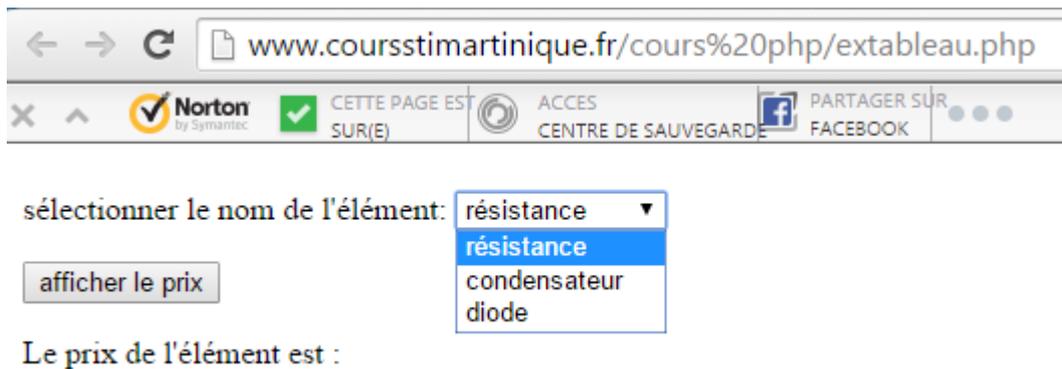
<body>
<?php
$chiffre1=$_GET['valeur1'];
$chiffre2=$_GET['valeur2'];
$resultat=$chiffre1+$chiffre2;
?>
<div align="center">Sélectionner un chiffre sur les deux listes et appuyer sur calcul pour afficher le résultat </div>
<form action="exercice2.php" method="get" name="form1" id="form1">
<p>
<label for="select">Première valeur sélectionnée:</label>
<select name="valeur1" id="valeur1">
<option value="0">0</option>
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>
<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
</select>
</p>
<p>
<label for="select2">Deuxième valeur sélectionnée:</label>
<select name="valeur2" id="valeur2">
<option value="0">0</option>
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>
<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
</select>
</p>
<p>
<input type="submit" name="submit" id="submit" value="Calcul">
</p>
<p>La somme des deux valeurs est égale à :<?php echo $resultat?></p>
</form>

</body>
</html>

```

Exercice 3 :

On veut afficher le prix d'éléments depuis un tableau php



The screenshot shows a web browser window with the address bar containing the URL `www.coursstimartinique.fr/cours%20php/extableau.php`. Below the address bar, there are several security and utility icons: Norton by Symantec, a green checkmark indicating the page is secure, a lock icon for the save center, and a Facebook share button. The main content area of the browser displays a form with the text "sélectionner le nom de l'élément:" followed by a dropdown menu. The dropdown menu is open, showing three options: "résistance", "résistance", "condensateur", and "diode". Below the dropdown is a button labeled "afficher le prix". Underneath the button, the text "Le prix de l'élément est :" is visible.

Correction :

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Document sans titre</title>
</head>
<?php
$element=$_GET['element'];
$element=Array();
$element["résistance"]=2;
$element["condensateur"]=3;
$element["diode"]=5;
?>
<body>
<form action="extableau.php" method="get" name="form1" id="form1">
<p>
<label for="select">sélectionner le nom de l'élément:</label>
<select name="element" id="element">
<option value="résistance">résistance</option>
```

```

<option value="condensateur">condensateur</option>
<option value="diode">diode</option>
</select>
</p>
<p>
<input type="submit" name="submit" id="submit" value="afficher le prix">
</p>
</form>
<p>Le prix de l'élément est :<?php echo $element[$element1] ?></p>
</body>
</html>

```

5. Racines prédéfinies

Il existe en php des variables prédéfinies :

| Variable | Description |
|--|--|
| <code>\$_SERVER['DOCUMENT_ROOT']</code> | Racine du serveur |
| <code>\$_SERVER['HTTP_ACCEPT_LANGUAGE']</code> | Langage accepté par le navigateur |
| <code>\$_SERVER['HTTP_HOST']</code> | Nom de domaine du serveur |
| <code>\$_SERVER['HTTP_USER_AGENT']</code> | Type de navigateur |
| <code>\$_SERVER['PATH_INFO']</code> | Chemin WEB du script |
| <code>\$_SERVER['PATH_TRANSLATED']</code> | Chemin complet du script |
| <code>\$_SERVER['REQUEST_URI']</code> | Chemin du script |
| <code>\$_SERVER['REMOTE_ADDR']</code> | Adresse IP du client |
| <code>\$_SERVER['REMOTE_PORT']</code> | Port de la requête HTTP |
| <code>\$_SERVER['QUERY_STRING']</code> | Liste des paramètres passés au script |
| <code>\$_SERVER['SERVER_ADDR']</code> | Adresse IP du serveur |
| <code>\$_SERVER['SERVER_ADMIN']</code> | Adresse de l'administrateur du serveur |
| <code>\$_SERVER['SERVER_NAME']</code> | Nom local du serveur |
| <code>\$_SERVER['SERVER_SIGNATURE']</code> | Type de serveur |
| <code>\$_SERVER['REQUEST_METHOD']</code> | Méthode d'appel du script |

6. Les fonctions et procédures

Une procédure est un groupe d'instructions qui ne renvoie pas de valeur après leur exécution. Une fonction, quant à elle, est en réalité une procédure qui retourne une valeur.

Il existe deux possibilités. Soit on intègre la fonction sur un fichier à part, qu'on appelle après, soit on intègre la fonction directement dans le programme.

a. Fonction dans le programme

On veut calculer le volume d'une sphère en rentrant le rayon.

Calcul du volume d'une sphère

Rentrer le rayon de la sphère:

Le volume est égal à :

i. Fonction mathématique de base

<http://php.net/manual/fr/ref.math.php>

ii. Définir la fonction

```
<?php
function volume($rayon)
{
    $volume=(pow($rayon,3))*4/3*3.14;
    return $volume; // indique la valeur à renvoyer ici le volume
}
?>
```

iii. Appel de la fonction

```
<?php
$volume2=volume($_GET['rayon1']);
?>
```

iv. Affichage du résultat

```
<p>Le volume est égal à : <?php echo $volume2 ?></p>
```

v. Ensemble

```
<body>

<?php
function volume($rayon)
{
    $volume=(pow($rayon,3))*4/3*3.14;
    return $volume; // indique la valeur à renvoyer ici le volume
}
?>
<h2 align="center">Calcul du volume d'une sphère </h2>
<form action="exercice3.php" method="get">
    <p>
        <label for="textfield">Rentrer le rayon de la sphère:</label>
        <input name="rayon1" type="text" id="rayon1" value="1">
    </p>
    <p>
        <input type="submit" name="submit" id="submit" value="Calculer">
    </p>
</form>
<?php
$volume2=volume($_GET['rayon1']);
?>
<p>Le volume est égal à : <?php echo $volume2 ?></p>
</body>
```

b. Fonction fichier à part

On va réutiliser le même cahier des charges.

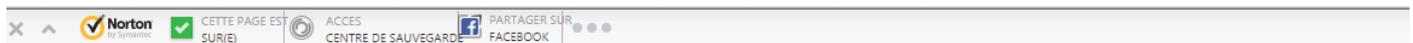
i. Réalisation du fichier fonction.php dans lequel se trouvera la fonction

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Document sans titre</title>
6 </head>
7 <?php
8 function volume($rayon)
9 {
10     $volume=(pow($rayon,3))*4/3*3.14;
11     return $volume; // indique la valeur à renvoyer ici le volume
12 }
13 ?>
14 <body>
15 </body>
16 </html>
```

ii. Réalisation du fichier principal avec appel de la fonction

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Document sans titre</title>
6 </head>
7 <?php
8 include ('fonction.php');
9 ?>
10 <body>
11 <h2 align="center">Calcul du volume d'une sphère </h2>
12 <form action="exercice3.php" method="get">
13     <p>
14         <label for="textfield">Rentrer le rayon de la sphère:</label>
15         <input name="rayon1" type="text" id="rayon1" value="1">
16     </p>
17     <p>
18         <input type="submit" name="submit" id="submit" value="Calculer">
19     </p>
20 </form>
21 <?php
22 $volume2=volume($_GET['rayon1']);
23 ?>
24 <p>Le volume est égal à : <?php echo $volume2 ?></p>
25 </body>
26 </html>
```

Le résultat est le même :



Calcul du volume d'une sphère

Rentrer le rayon de la sphère:

Le volume est égal à : 0

Avec PHP7, il est possible de définir le type de valeur que doit retourner la fonction.

```
function foo(int $foo, int $bar)
{
    echo $foo + $bar;
}

foo(1, 1); // affiche 2
foo(1, '1'); // affiche 2 (conversion car mode coercitif)
foo('a', 1); // Uncaught TypeError
```

7. Structures de contrôle

Elles fonctionnent comme le JavaScript :

| Instruction | Signification |
|--------------------------|-----------------------|
| if | Si |
| else | Sinon |
| elseif | Sinon si |
| switch | Selon |
| for | Pour chaque (boucle) |
| while | Tant que (boucle) |
| == | Strictement égal |
| != | Différent |
| < | Strictement inférieur |
| > | Strictement supérieur |
| <= | Inférieur ou égal |
| >= | Supérieur ou égal |
| and ou && | ET logique |
| or ou | OU logique |

Nouveautés PHP7 :

Nouvel opérateur de comparaison <=>

Cet opérateur, sur le modèle déjà utilisé pour des fonctions de comparaison comme strcmp(), retournerait trois valeurs :

- 0 si les deux opérandes sont égaux,
- 1 si l'opérande de gauche est plus grand que celui de droite
- -1 si l'opérande de gauche est plus petit que celui de droite.

```

// Integers
echo 1 <=> 1; // 0
echo 1 <=> 2; // -1
echo 2 <=> 1; // 1

// Floats
echo 1.5 <=> 1.5; // 0
echo 1.5 <=> 2.5; // -1
echo 2.5 <=> 1.5; // 1

// Strings
echo "a" <=> "a"; // 0
echo "a" <=> "b"; // -1
echo "b" <=> "a"; // 1

echo "a" <=> "aa"; // -1
echo "zz" <=> "aa"; // 1

// Arrays
echo [] <=> []; // 0
echo [1, 2, 3] <=> [1, 2, 3]; // 0
echo [1, 2, 3] <=> []; // 1
echo [1, 2, 3] <=> [1, 2, 1]; // 1
echo [1, 2, 3] <=> [1, 2, 4]; // -1

// Objects
$a = (object) ["a" => "b"];
$b = (object) ["a" => "b"];
echo $a <=> $b; // 0

```

Exemple :

Contrôler qu'un chiffre est bien rentré et inférieur à une valeur

ctype_alpha() vérifie si tous les caractères de la chaîne text sont des lettres, et FALSE sinon.



Rentrer une valeur strictement inférieure à 5 et cliquer sur envoyer

Valeur :

la valeur est bonne

```

6 </head>
7 <?php
8 $valeur1=$ GET['valeur'];
9 $test=cctype_alpha($valeur1); Test si la valeur est du texte
10 if ($test==true)
11 {
12 $resultat='la valeur n\' |est pas un chiffre, recommencer';
13 }
14 else
15 {
16 if ($valeur1<5)
17 {
18 $resultat='la valeur est bonne';
19 }
20 else
21 {
22 $resultat='la valeur est mauvaise';
23 }
24 }
25 ?>
26 <body>
27 <p>Rentrer une valeur strictement inférieure à 5 et cliquer sur envoyer
28 </p>
29 <form action="exercice5.php" method="get" name="form1" id="form1">
30 <p>
31 <label for="textfield">Valeur :</label>
32 <input name="valeur" type="text" id="valeur" value="0">
33 </p>
34 <p>
35 <input type="submit" name="submit" id="submit" value="Envoyer">
36 </p>
37 </form>
38 <p>&nbsp;&nbsp;&nbsp;</p>
39 <h3><?php echo $resultat ?></h3>
40 </body>

```

8. Lire et écrire dans un fichier .Txt

Il y a des moments où l'on veut sauvegarder des informations dans un fichier. Le souci des fichiers Txt c'est qu'une seule personne peut y accéder à la fois.

Pour lire ou écrire, il faut d'abord l'ouvrir par l'instruction fopen

```
$fp=fopen("donne.txt","r");
```

r indique le fichier est en lecture seule

- r : Ouverture en lecture seule, le pointeur sera au début
- r+ : Ouverture en lecture et écriture, le pointeur sera au début
- w : Ouverture en écriture seule, crée le fichier ou le vide avant écriture, pointeur au début
- w+ : Ouverture en rw, crée le fichier ou le vide avant écriture, pointeur au début
- a : Ouverture en écriture seule, crée le fichier si n'existe pas, pointeur à la fin si existe
- a+ : Ouverture en lecture et écriture, crée le fichier si n'existe pas, pointeur à la fin (écrit à la suite du fichier)
- x : Ouverture en lecture seule, crée le fichier et ne fonctionne pas si le fichier existe déjà.
- x+ : Ouverture en lecture et écriture, crée le fichier et ne fonctionne pas si le fichier existe déjà.
- c : Ouverture en écriture seule, crée le fichier si il n'existe pas, ne vide pas le fichier comme w, pointeur au début (écrit par au-dessus de ce qui existe déjà)

- c+: Ouverture en lecture et écriture, pareil que pour c

Pour lire le fichier on va utiliser l'instruction « fgets » pour lire toutes les valeurs de la ligne courante du fichier. On peut aussi lire en entier en une seule fois avec l'instruction « readfile » ou alors le lire ligne par ligne avec « file ». Pour écrire, on utilisera la fonction fwrite

Exemple :



Rentrer une valeur puis appuyer sur envoyer pour enregistrer:

Valeur enregistrée :1bbcd

```

1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Document sans titre</title>
6 </head>
7 <?php
8 $valeur=$_GET['donne'];
9 $fp=fopen("donne.txt","a+");
10 fwrite($fp,$valeur);
11 fclose ($fp);
12 $fp=fopen("donne.txt","a+");
13 $contenu= fgets ($fp, 255);
14 fclose ($fp);
15 ?>
16 <body>
17 <form action="fichier1.php" method="get" name="form1" id="form1">
18 <p>
19 <label for="textfield">Rentrer une valeur puis appuyer sur envoyer pour enregistrer:</label>
20 <input type="text" name="donne" id="donne">
21 </p>
22 <p>
23 <input type="submit" name="submit" id="submit" value="Envoyer">
24 </p>
25 </form>
26 <p>Valeurs enregistrées :<?php echo $contenu ?></p>
27 </body>
28 </html>

```

- Liste de fonctions

- basename — Retourne le nom du fichier dans un chemin
- chgrp — Change le groupe d'un fichier
- chmod — Change le mode du fichier
- chown — Change le propriétaire du fichier
- clearstatcache — Efface le cache de stat
- copy — Copie un fichier
- delete — Voir unlink ou unset
- dirname — Renvoie le chemin du dossier parent
- disk_free_space — Renvoie l'espace disque disponible sur le système de fichiers ou la partition
- disk_total_space — Retourne la taille d'un dossier ou d'une partition
- diskfreespace — Alias de disk_free_space
- fclose — Ferme un fichier
- feof — Teste la fin du fichier
- fflush — Envoie tout le contenu généré dans un fichier
- fgetc — Lit un caractère dans un fichier
- fgetcsv — Obtient une ligne depuis un pointeur de fichier et l'analyse pour des champs CSV
- fgets — Récupère la ligne courante sur laquelle se trouve le pointeur du fichier
- fgetss — Renvoie la ligne courante du fichier et élimine les balises HTML
- file_exists — Vérifie si un fichier ou un dossier existe
- file_get_contents — Lit tout un fichier dans une chaîne
- file_put_contents — Écrit un contenu dans un fichier
- file — Lit le fichier et renvoie le résultat dans un tableau
- fileatime — Renvoie la date à laquelle le fichier a été accédé pour la dernière fois
- filectime — Renvoie la date de dernière modification de l'inode d'un fichier
- filegroup — Lire le nom du groupe
- fileinode — Lit le numéro d'inode du fichier
- filemtime — Lit la date de dernière modification du fichier
- fileowner — Lit l'identifiant du propriétaire d'un fichier
- fileperms — Lit les droits d'un fichier
- filesize — Lit la taille d'un fichier
- filetype — Retourne le type de fichier
- flock — Verrouille le fichier
- fnmatch — Teste un nom de fichier au moyen d'un masque de recherche
- feof — Teste 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...
- fopen — Ouvre un fichier ou une URL
- fpassthru — Affiche le reste du fichier
- fputcsv — Formate une ligne en CSV et l'écrit dans un fichier
- fputs — Écrit une chaîne de caractères dans le fichier. Alias de fwrite ex : fputs(\$fp, "\n"); //passe à la ligne suivante
- fread — Lecture du fichier en mode binaire
- fscanf — Analyse un fichier en fonction d'un format
- fseek — Modifie la position du pointeur de fichier
- fstat — Lit les informations sur un fichier à partir d'un pointeur de fichier
- ftell — Renvoie la position courant du pointeur de fichier
- ftruncate — Tronque un fichier
- fwrite — Écrit un fichier en mode binaire
- glob — Recherche des chemins qui vérifient un masque
- is_dir — Indique si le fichier est un dossier
- is_executable — Indique si le fichier est exécutable
- is_file — Indique si le fichier est un véritable fichier
- is_link — Indique si le fichier est un lien symbolique
- is_readable — Indique si un fichier existe et est accessible en lecture
- is_uploaded_file — Indique si le fichier a été téléchargé par HTTP POST
- is_writable — Indique si un fichier est accessible en écriture

- is_writeable — Alias de is_writable
- lchgrp — Change l'appartenance du groupe d'un lien symbolique
- lchown — Change le propriétaire d'un lien symbolique
- link — Crée un lien
- linkinfo — Renvoie les informations d'un lien
- lstat — Retourne les informations sur un fichier ou un lien symbolique
- mkdir — Crée un dossier
- move_uploaded_file — Déplace un fichier téléchargé
- parse_ini_file — Analyse un fichier de configuration
- parse_ini_string — Analyse une chaîne de configuration
- pathinfo — Retourne des informations sur un chemin système
- pclose — Ferme un processus de pointeur de fichier
- popen — Crée un processus de pointeur de fichier
- readfile — Affiche un fichier
- readlink — Renvoie le contenu d'un lien symbolique
- realpath_cache_get — Récupère les entrées du cache realpath
- realpath_cache_size — Récupère la taille du cache realpath
- realpath — Retourne le chemin canonique absolu
- rename — Renomme un fichier ou un dossier
- rewind — Remplace le pointeur de fichier au début
- rmdir — Efface un dossier
- set_file_buffer — Alias de stream_set_write_buffer
- stat — Renvoie les informations à propos d'un fichier
- symlink — Crée un lien symbolique
- tempnam — Crée un fichier avec un nom unique
- tmpfile — Crée un fichier temporaire
- touch — Modifie la date de modification et de dernier accès d'un fichier
- umask — Change le "umask" courant
- unlink — Efface un fichier