

PHP

Les fondamentaux

Introduction

PHP: Hypertext Preprocessor

Langage de programmation libre principalement utilisé pour produire des pages web dynamiques ou des scripts de gestion de données.

Syntaxe: `<?php /* Code */ ?>`

Interpréter le PHP

Dès lors qu'une balise PHP est ouverte dans un document HTML, celui-ci devient de fait un fichier PHP et il faut donc l'enregistrer avec l'extension ".php".

Il faut un interpréteur PHP installé en local.

Chaque ligne de code se termine par un point-virgule ";".

Afficher du texte sous PHP

“echo” est utilisé pour afficher des données à l’écran.

Exemple: `echo 'Hello world!';`

La concaténation en PHP se fait à l’aide du point “.”

Variables

Elles sont toujours précédées du symbole “\$”.

En PHP, il n'est pas nécessaire de déclarer une variable avant de pouvoir lui affecter une valeur.

- Pas d'espace, ni d'accent
- Ne pas commencer par un chiffre
- On peut mettre un underscore “_”

Variables

```
$var = '0'
```

Constantes

```
define('MIN_VALUE', '0');
```


On ne fait pas précéder le nom de la variable par un type, le typage est dit automatique.

String	Chaîne de caractères
Integer	Nombre entier
Float	Nombre décimal
Boolean	Booléen
Array	Tableau
Object	Objet
NULL	NULL

Opérateurs logiques

&& : AND

|| : OR

! : NOT

Opérateurs de comparaison

== Si les valeurs sont identiques

=== Type et valeur sont identiques

!= or <> or !== Inverse des deux lignes précédentes

> Plus grand que

< Plus petit que

>= Plus grand ou égal à

<= Plus petit ou égal à

Tableaux

Il existe trois types de tableaux différents en PHP :

- Les tableaux numérotés ou indexés
- Les tableaux associatifs
- Les tableaux multidimensionnels

`var_dump($nomTab)` ou `print_r($nomTab)` permettent d'afficher le tableau.


```
1 <?php
2 $array = array(
3     "php" => "foreach",
4     "js" => "map",
5 );
6
7 $array = [
8     "php" => "foreach",
9     "js" => "map",
10 ];
11
12 $array = array("hello", "world");
13
14 var_dump($array);
15 ?>
```

Tableaux


```
1 <?php
2 $a = 1;
3 for ($a = 1; $a < 10; $a++) {
4     echo "a is now $a <br>";
5 }
6
7 foreach (array(1, 2, 3, 4, 5) as $v) {
8     echo "$v <br>";
9 }
10
11 while($a < 10) {
12     echo "a is now $a <br>";
13     $a++;
14 }
15
16 do {
17     echo "a is now $a <br>";
18     $a++;
19 } while($a < 20);
20 ?>
```

Boucles

Fonctions

Une fonction est constituée:

- d'un nom
- de paramètres éventuels
- d'un bloc de code qui représente une série d'instructions qui vont être exécutées dès l'appel de la fonction.

Fonctions - Déclaration

```
1 // Déclaration de la fonction
2 function nameFunction()
3 {
4   // Instructions
5 }
6 // Appel de la fonction
7 nameFunction();
```


Fonctions - Exemple

```
1 // Déclaration de la fonction
2 function hello()
3 {
4     echo 'Hello world';
5 }
6 // Appel de la fonction
7 hello();
```


Fonctions - Retour

```
1 function name($param1, $param2, ...)  
2 {  
3     //Bloc d'instructions  
4     return val; //Pour retourner une valeur  
5 }
```

```
1 // Typage des paramètres (int) et de la valeur de  
  retour (int)  
2 function multiplication(int $nb1, int $nb2): int  
3 {  
4     return $nb1 * $nb2;  
5 }
```


Références

```
1 <?php
2 function tmp($x)
3 {
4     $x += 5;
5     return $x;
6 }
7
8 $x = 10;
9 echo $x; // Affiche 10
10 echo tmp($x); // Affiche 15
11 echo $x // Affiche 10
12 ?>
```

```
1 <?php
2 function tmp(&$x)
3 {
4     $x += 5;
5     return $x;
6 }
7
8 $x = 10;
9 echo $x; // Affiche 10
10 echo tmp($x); // Affiche 15
11 echo $x // Affiche 15
12 ?>
```


Superglobales

<code>\$_GET</code>	Contient les données envoyées en paramètres dans l'URL
<code>\$_POST</code>	Contient les informations envoyées par un formulaire
<code>\$_FILES</code>	Contient la liste des fichiers envoyés par un formulaire
<code>\$_SERVER</code>	Correspond aux valeurs renvoyées par le serveur
<code>\$_ENV</code>	Correspond aux variables d'environnement
<code>\$_SESSION</code>	Correspond aux variables de session. Ce sont des variables qui restent stockées sur le serveur le temps de la présence d'un visiteur.
<code>\$_COOKIE</code>	Correspond aux valeurs des cookies enregistrés sur l'ordinateur du visiteur.

Inclusion de fichier

Il est possible d'inclure dans un fichier PHP d'autres fichiers PHP.

```
1 <?php
2 require('nom_fichier.php');
3 require_once('nom_fichier.php');
4 include('nom_fichier.php');
5 ?>
```


Formulaires

```
<form method="POST" action="verify.php">  
    // Champs du formulaire  
</form>
```

Deux attributs très importants:

- la méthode: method
- le fichier cible: action

Formulaires - Method

GET: Cette méthode est assez peu utilisée car on ne peut pas envoyer beaucoup d'informations dans l'URL (≤ 255 caractères). On récupère les données dans la superglobale `$_GET`.

POST: Cette méthode permet d'envoyer autant de données que l'on veut. Néanmoins, les données ne sont pas plus sécurisées qu'avec la méthode GET et il faudra toujours vérifier si tous les paramètres sont bien présents et valides. On récupère les données dans la superglobale `$_POST`.

La fonction `isset($nomVariable)` permet de tester si une variable existe.

Fonctions PCRE

- preg_grep
- preg_match
- preg_match_all
- preg_replace
- preg_split

Caractère spécial	Signification
.	représente un caractère quelconque
?	ce qui précède doit apparaître 0 ou 1 fois
*	ce qui précède peut apparaître 0, 1 ou plusieurs fois
+	ce qui précède doit apparaître au moins 1 fois
^	ce qui suit correspond au début de la chaîne
\$	ce qui précède correspond à la fin de la chaîne
[]	définit un ensemble de caractère possibles
()	définit une séquence de caractères
\	le caractère spécial suivant doit être considéré comme normal
	opérateur logique 'OU'
{n}	ce qui précède doit apparaître exactement n fois
{n, m}	ce qui précède doit apparaître entre n et m fois

Exemple d'ER

```
1 preg_match('/[a-z]/', "Cette phrase contient une lettre", $matches);  
2 var_dump($matches); // Affiche array(0 => 'e')  
3  
4 preg_match('/[A-Z0-9]/', "cette phrase ne comporte ni majuscule ni chiffre", $matches);  
5 var_dump($matches); // Affiche array()
```


Fin.