

Initiation au PHP

Cédric Leclinche



Qu'est-ce que PHP ?

- PHP: Hypertext Processor
- Langage de programmation libre
- Utilisé principalement pour créer des pages web dynamiques

Historique

- Créé par Rasmus Lerdorf en 1994
- Sortie de la version 3 de PHP en 1998 par Andi Gutmans et Zeev Suraski
- Réécriture du moteur interne de PHP baptisé Zend Engine
- Sortie de la version 4 de PHP en 2000
- Sortie de la version 5 de PHP en 2004
- Prochaine version : PHP 7

Qui utilise PHP ?

- 75% de parts de marché
- Plus de 200 millions de sites en PHP
- Ils utilisent PHP :



YAHOO!



Qu'est-ce que ça peut faire ?

- Création de sites web dynamiques
- Création de scripts

- Exemples d'applications :

Dolibarr ERP/CRM

MEDIAWIKI


WORDPRESS


Joomla!


Drupal

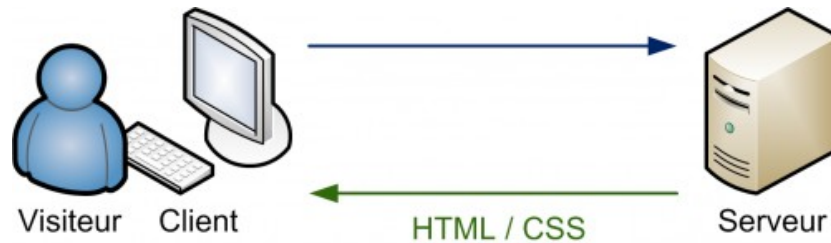

PRESTASHOP

Comment ça marche ?

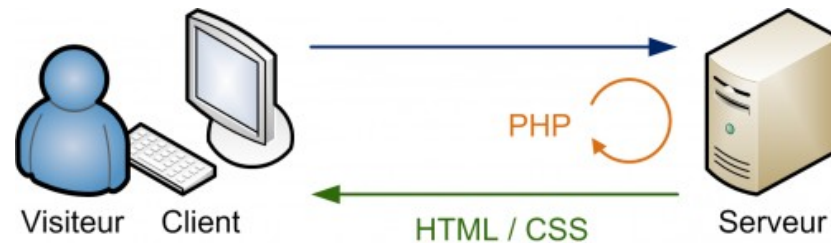
- Architecture client/serveur
- Langage interprété côté serveur
- Génère du code HTML

Comment ça marche ?

- Cas site statique



- Cas site dynamique



De quoi a-t-on besoin ?

- Serveur HTTP (Apache, Nginx, Lighttpd...)
- Serveur PHP
- Serveur de base de données (MySQL, PostgreSQL, Oracle...)

De quoi a-t-on besoin ?

- Sous Windows : WAMP, EasyPHP
- Sous MAC : MAMP
- Sous Linux : XAMPP

Installer

- Télécharger wampserver sur le site <http://www.wampserver.com/>
- Installer en double-cliquant dessus et se laisser guider
- Lancer le logiciel et aller à la page <http://localhost> dans votre navigateur

Remarques

- Les fichiers sont à mettre dans le dossier wamp/www/
- Apache fonctionne sur le port 80 – attention de ne pas avoir une application qui l'utilise

Tester sans installer

- Serveur web portable (usbwebserver, server2go...)
- Application en ligne (ideone, writecodeonline)

Les bases de PHP

- Fichiers .php
- Commencent par `<?php` et finissent par `?>`
- Afficher des informations avec « `echo` »

Exemple de page avec PHP

```
<!DOCTYPE html>
<html>
<head>
  <title>Page de test</title>
  <meta charset="utf-8" />
</head>
<body>
  <h2>Page de test</h2>
  <?php echo 'Hello world !'; ?>
</body>
</html>
```

Inclure des fichiers

- Inclure fichier PHP dans un autre fichier PHP
- Syntaxe : `include('fichier.php')` ou `require('fichier.php')`
- Différence : `require` arrête le script si le fichier est introuvable

Les commentaires

- Ligne qui ne sera pas interprétée par le serveur

- Commentaire sur une seule ligne

```
<?php echo 'Hello world'; //Ceci est un commentaire ?>
```

- Commentaire multi-lignes

```
<?php  
echo 'Hello world';  
/* Ceci est  
un commentaire sur  
plusieurs lignes */  
?>
```


Les variables

- Espace de stockage pour une information temporaire
- Afficher des variables, les modifier, réaliser des opérations dessus
- Déclaration : `$variable`

Exemple

```
<?php
$var;                //Création de la variable
$var = 'Hello world'; //Affection de la valeur Hello world à la variable
echo $var;           //Affichage de la variable
?>
```

Cet exemple va afficher « Hello world »

Les types de variables

- Chaînes de caractères

Exemple : `$var = « Hello world »` ou `$var = 'Hello world';`

- Numériques

Exemple : `$var = 5;`

- Booléens

Exemple : `$var = true;` ou `$var = false;`

Opérations sur les variables

- Opérations classiques : +, -, *, /
- Reste de la division entière: %
- Concaténation : mettre bout à bout plusieurs chaînes de caractères
En PHP on utilise le « . »

Exemple

Affichage de la différence de deux variables

```
$a = 2;  
$b = 5;  
echo $a - $b;    //Affiche -3  
  
//On peut également stocker  
//le résultat dans une autre variable  
$c = $a - $b;  
echo $c;         // Affiche -3
```

Exemple

Concaténation de deux chaînes

```
$age = 18;  
echo 'J\'ai '.$age.' ans'; //Affiche j'ai 18 ans  
  
$a = 'Hello';  
$b = 'world';  
  
echo $a.' '.$b;           //Affiche Hello world
```

Remarques

- \ devant un ' dans une chaîne de caractère avec guillemets simples
- ' ' est une chaîne de caractère avec un espace

Les tableaux

- Enregistrer plusieurs informations dans une seule variable
- 2 types de tableau :
 - Tableau numéroté
 - Tableau associatif

Tableau numéroté

- Associe une valeur à un nombre (commence par l'index 0)
- Syntaxe : `array('valeur 1', 'valeur 2')`

Clé	Valeur
0	valeur 1
1	valeur 2

Tableau associatif

- Associe une valeur à une clé
- Syntaxe : `array('clé 1' => 'valeur 1', 'clé 2' => 'valeur 2')`

Clé	Valeur
clé 1	valeur 1
clé 2	valeur 2

Les tableaux

- Récupérer une information :
\$tableau['clé']

```
//Tableau ordonné
```

```
$fruits = array('banane', 'fraise', 'ananas');  
echo $fruits[1];    //Affiche fraise
```

```
//Tableau associatif
```

```
$user = array('username' => 'Alice', 'age' => 19);  
echo $user['age'];  //Affiche 19
```

Les conditions

- Définition : Exécute certaines instructions si la condition est vérifiée
- Syntaxe
 - «\$a == \$b» teste si les 2 variables sont égales
 - «\$a != \$b» teste si les 2 variables sont différentes
 - «\$a <= \$b» teste si a est plus petit ou égal à b

Exemple

```
<?php
if($age >= 18)
{
    echo 'Bienvenue';
}
else
{
    echo 'Vous devez être majeur pour accéder à ce site';
}
?>
```

Les boucles

- Définition : Exécute plusieurs fois certaines instructions si la condition est vérifiée
- 3 types de boucle :
 - ▢ while
 - ▢ for
 - ▢ do... while

Boucle while

- Syntaxe : `while(condition){instructions}`

```
$nombre = 100;  
$compteur = 0;                                //Compte le nombre de passages dans la boucle  
while($nombre > 0)  
{  
    $compteur = $compteur + 1;    //On ajoute 1 au compteur  
    $nombre = $nombre - 1;  
}  
echo $compteur.' passages';                //Affiche 100 passages
```

Boucle do... while

- Syntaxe : `do{instructions}`
`while(condition);`
- Exécution au-moins une fois
- Même principe que la boucle while mais condition vérifiée à la fin et non au début

Boucle for

- Syntaxe : for(initialisation; condition; pas)

```
$compteur = 0; //Compte le nombre de passages dans la boucle
for($nombre = 0; $nombre < 100; $nombre++)
{
    $compteur = $compteur + 1; //On ajoute 1 au compteur
}
echo $compteur.' passages'; //Affiche 100 passages
```

Les fonctions

- Définition : Série d'instructions réutilisable
- Fonctions disponibles : count(), unset(), var_dump(), isset(), empty()... <https://php.net/manual/fr/funcref.php>

Créer fonction

- Syntaxe :

function nom(argument){instructions}

```
function hello($nom)
{
    echo 'hello ' . $nom;
}
hello('world'); //Affiche hello world
```

Transmettre des données

- En passant par l'URL
- En passant par un formulaire

Avec l'URL

- `http://www.adresse-du-site.com/hello.php?prenom=Jack&nom=Sparrow`
- Variable superglobale `$_GET` : tableau avec les paramètres dans l'URL

Clé	Valeur
prenom	Jack
age	Sparrow

Exemple d'utilisation

```
<!DOCTYPE html>
<html>
<head>
  <title>Transmission de données avec l'url</title>
  <meta charset="utf-8" />
  <link rel="stylesheet" type="text/css" href="form.css">
</head>
<body>
  <h2>Transmission de données avec l'url</h2>
  Bonjour <?php echo $_GET['prenom'].' '.$_GET['nom']; ?>
</body>
</html>
```

Avec un formulaire

- Utilisation d'un formulaire :

<input type="text" value="utilisateur"/>	<input type="text" value="mot de passe"/>	<input type="button" value="Envoyer"/>
--	---	--

- Variable superglobale \$_POST : tableau avec les paramètres envoyées par le formulaire

Clé	Valeur
utilisateur	admin
password	admin

Exemple d'utilisation

- Page avec le formulaire

```
<form action="site.php" method="post">  
  <input type="text" name="prenom" />  
  <input type="text" name="nom" />  
  <input type="submit" />  
</form>
```

- Page site.php :

```
<?php echo 'Bonjour ' . $_POST['prenom'] . ' ' . $_POST['nom']; ?>
```


Les sessions

- Sauvegarde de variables sur le serveur
- Permettent de conserver des données sur plusieurs pages
- Utilisées notamment pour les connexions utilisateurs

Comment les utiliser ?

- `session_start()` avant tout code html pour ouvrir la session
- `$_SESSION['cle'] = 'valeur'`
- `echo $_SESSION['cle']`
- `Session_destroy()` pour fermer la session

Les cookies

- Fichier enregistré sur l'ordinateur du visiteur
- Retenir informations sur le visiteur
- Utilisés par exemple par les « Se souvenir de moi »

Comment les utiliser ?

- setCookie('clé', 'valeur', duree) avant tout code html
- Exemple

```
<?php  
setcookie('pseudo', 'user', time() + 365*24*3600, null, null, false, true);  
setcookie('password', 'secret', time() + 365*24*3600, null, null, false, true);  
?>
```

- echo \$_COOKIE['clé']

Aller plus loin

- Lecture/Écriture dans des fichiers
- Connexion à des bases de données
- PHP CLI (Commande Line Interface)

Avez-vous des questions ?



Atelier – Création d'un formulaire de connexion

- Création de deux pages : un formulaire et une page avec le site
- Affichage du site avec le nom d'utilisateur si jamais le mot de passe est correct
- Si le mot de passe est faux on redirige vers le formulaire

Atelier – Aller plus loin

- Authentifier l'utilisateur
- Ajouter un message si le mot de passe est faux
- Ajouter un case à cocher « Se souvenir de moi »