

## Conception de Sites Web dynamiques

Cours 5

Patrick Reuter

## Programme

- Programmation côté serveur

## Conception de Site Webs Interactifs

Déroulement

- Pages web statiques (**HTML/XHTML**)
- Mise en forme avec feuilles de styles (**CSS**)
- Programmation côté serveur
  - Pages web dynamiques (**PHP**)
  - avec connexion à une base de données (**MySQL**)
- Programmation côté client
  - JavaScript
- Référencement Internet (moteur de recherche)

## Automatisation

- Exemple :
  - Une punition genre ...



## HTML

Exemp

```
<!DOCTYPE...>
<html xmlns=http://www.w3.org/1999/xhtml xml:lang="fr">
  <head>
    <title>ma premiere page</title>
    <meta http-equiv="Content-Type"
content="text/html" />
  </head>
  <body>
    <p>J'aime les lundi matin</p>
    <p>J'aime les lundi matin</p>
    <p>J'aime les lundi matin</p>
    <p>J'aime les lundi matin</p>
    <p>J'aime les lundi matin</p>
    <p>J'aime les lundi matin</p>
    <p>J'aime les lundi matin</p>
    <p>J'aime les lundi matin</p>
    <p>J'aime les lundi matin</p>
    <p>J'aime les lundi matin</p>
  </body>
</html>
```

## En PHP

```
<!DOCTYPE...>
<html xmlns=http://www.w3.org/1999/xhtml xml:lang="fr">
  <head>
    <title>ma premiere page</title>
    <meta http-equiv="Content-Type"
content="text/html" />
  </head>
  <body>
    <?php
    $i = 1;
    while ($i <= 10)
    {
      echo "<p>J'aime les lundi matins </p>";
      $i = $i + 1;
    }
    ?>
  </body>
</html>
```





## Qu'est-ce que le PHP ?

- PHP : **P**ersonal **H**ome **P**ages / **P**eople **H**ate **P**erl / **H**ypertext **P**re**P**rocessor
- PHP est un langage de programmation interprété côté serveur
- Inventé par Rasmus LEDOORF (v1 : 1994, v2 : 1996, ... v5 : 2004)
- Il est dédié au Web : traitement des formulaires, communication avec des bases de données (souvent couplé à MySQL)

**Avantages**

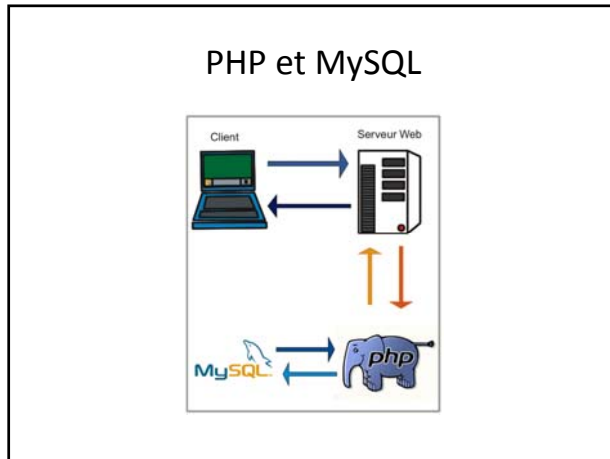
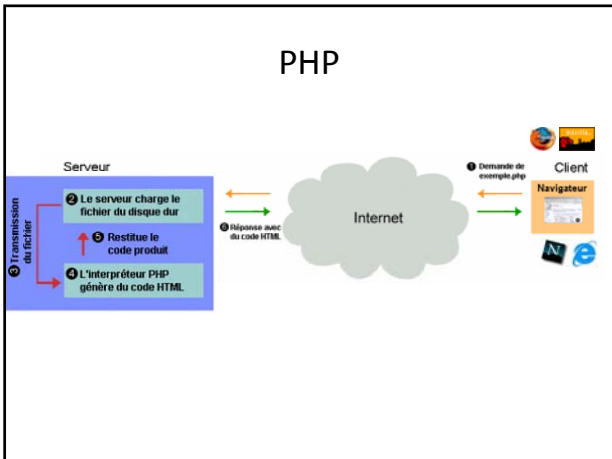
- Il est simple (pas type, interprète, ...)
- Il est gratuit (?)
- Il est fait pour le Web
- Il est TRES répandu





**Inconvénient**

- Code difficilement maintenable



### PHP

<http://fr2.php.net/manual/fr/>

### Premier exemple

```

<?php
echo "Hello World";

?>
```

## Remarques

- Extension du fichier doit être .php
- Le navigateur ne comprend que l'HTML
- Le code "php" inclus dans la page HTML est transformé par le serveur php
- Le code "php" est invisible dans la page HTML affichée car il est interprété

## Un exemple pratique : Hello World

The diagram shows two code snippets on the left and two browser screenshots on the right. The top snippet is for 'helloworld.htm' and contains pure HTML: 

```
<html>
<head>
  <title>Ma premiere page</title>
</head>
<body>
  <h1>Hello World</h1>
</body>
</html>
```

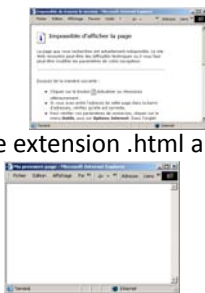
. The bottom snippet is for 'helloworld.php' and contains PHP code: 

```
<html>
<head>
  <title>Ma premiere page</title>
</head>
<body>
  <?php
    echo "<h1>Hello World</h1>";
  ?>
</body>
</html>
```

. Arrows point from the HTML code to a browser window showing 'Hello World', and from the PHP code to another browser window also showing 'Hello World'.

## Un exemple pratique : remarques

- Si on "éteint" le serveur web (EasyPHP)
- Si on met une extension .html au fichier php



## Balise

Les scripts php étant destinés à être utilisés à l'intérieur de pages HTML, il faut une balise spéciale permettant au "parser / interpréteur" php de savoir à quel endroit se trouve le code.

Un code php est toujours compris entre les balises "<?php" d'une part et "?>" d'autre part.

The diagram shows a code snippet on the left: 

```
<?php
  echo "<h1>Hello World</h1>";
?>
```

 and a browser window on the right displaying 'Hello World'. This illustrates how the PHP code is embedded within HTML tags and rendered by the browser.

## Commentaires

- Comme en C ou en Java

```
<?php
// commentaire de fin de ligne

/* commentaire
sur plusieurs
lignes */

?>
```

- Tout ce qui se trouve dans un commentaire est ignoré par le serveur php

## Ecrire dans le fichier HTML

```
<?php

echo "hello world";

?>
```

## Concatenation 1

```
<?
echo "Salut"." Martin";
?>
```

## Variables

- php est interprété et faiblement typé.
- Il n'est pas nécessaire de déclarer le type d'une variable.
- Les identificateurs de variable sont précédés du symbole « \$ » (dollar).

```
<?php
$a = 5;
echo $a;
?>
```

### Types :

- entier (**integer**)
- réel (**double**)
- tableau (**array**)
- objet (**object**)
- booléen (**boolean**)
- chaîne de caractères (**string**)

## Variables

```
<?php
$a = 0;
$nom = "Lisa";
echo $a;

?>

<?php $a = 0; ?>
<?php $nom = "Lisa"; ?>
<?php echo $a; ?>
```

## Concatenation 2

```
<?
$nom = "Martin";

echo "Salut ".$nom;

?>
```

- Afficher le texte suivant à l'écran après l'affectation :

```
x = 10;
```

Résultat :

Le contenu de x est 10.

```
echo "Résultat"
echo "Le contenu de x est ".$x."
```

```
afficher ("Résultat:");
afficher ("Le contenu de x est",)
afficher (x,);
afficher (".");
```

```
$x = 10
echo "Résultat"
echo "Le contenu de x est ",
echo $x
echo "."
```

## Quotes

```
echo "<a href='index.php'>Accueil</a>";

echo '<a href="index.php">Accueil</a>';

// equivalent à :
```

```
echo "<a href=\"index.php\">Accueil</a>";
```

## Variables

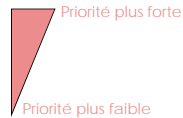
### •Portée

Limitée au bloc dans lequel elle a été créée.  
Une variable créée dans un bloc n'est pas connue dans les autres.  
Les variables de fonctions ne sont connues que dans la fonction.

### •Opérateurs

Par ordre de priorité

- ++, --, !
- +, -, \*, /, %
- ==, <, >, <=, >=, !=
- &&, ||, and, or
- =



## Condition

Permet d'exécuter ou non un bloc d'instructions.

```
if (condition1)
{
    /* bloc d'instructions lorsque
    condition1 est vraie */
}
else if (condition2)
{
    /* autre bloc d'instructions, lorsque
    condition2 est vraie */
}
else
{
    /* bloc d'instructions lorsque les
    conditions antécédentes sont fausses */
}
```

Rmq : la condition de son résultat rend vrai ou faux)

## Condition SI..SINON

```
<?php
$nombre = 512;

if ($nombre<500)
{
    echo $nombre." est compris entre 0 et 499";
}
else if($nombre>=500 && $nombre<1000)
{
    echo $nombre." est compris entre 500 et 999";
}
else
{
    echo $nombre." est plus grand que 999";
}
?>
```

## Boucle while

Permet d'exécuter plusieurs fois un bloc d'instructions donné (nombre d'itérations inconnu)

```
while (condition_pour_continuer)
{
    /* bloc d'instructions lorsque la condition
    est vérifiée */
}
```

Rmq : la condition est une expression booléenne (l'évaluation de son résultat rend vrai ou faux)  
- si vrai : le bloc d'instructions est exécuté  
- si faux : la boucle n'est plus itérée, et l'exécution continue à la suite de la boucle

## Boucle TANT QUE

```
<?php
$i = 1;
while ($i<=5)
{
    echo "Ligne No. ";
    echo $i;
    echo "<br />";
    $i++;
    // $i = $i + 1;
}
?>
```

## Boucle TANT QUE

```
<?php
$i = 1;
while ($i<=5)
{
    ?>
    Salut<br />
    <?
    $i++;
}
?>
```

```
<?php
$i = 1;
while ($i<=5)
{
    echo "Salut<br />";
    $i++;
}
?>
```

## Passer des paramètres dynamiques

- Par l'URL :  
→ méthode `$_GET`
- Par des formulaires HTML  
→ aussi méthode `$_POST`

## Méthode `_GET`

```
<?php
    $a = $_GET['param'];
    echo $a;
?>
```

URL :  
http://.../index.php?param=5

## Fonction `isset()`

```
<?php
    if isset($_GET['param'])
        $param = $_GET['param'];
    else
    {
        echo "Paramètre non donné";
        $param = 1;
        // valeur par défaut
    }
?>
```