Cours Web n°5
Introduction au langage PHP

Pierre Senellart (pierre@senellart.com)
Pierre Yger (yger@unic.cnrs-gif.fr)

Semaine du 15 octobre 2007

Plan du cours

1 PHP : un langage de programmation
2 Introduction au langage
3 Opérations diverses
4 Structures de contrôle
5 Validation
6 Références
7 Applications

- Script (ou programme) PHP : un fichier texte (généralement avec l'extension *.php*) qui sera interprété par le serveur Web.
- Utilisé pour fournir un comportement dynamique côté serveur.
- Langage de programmation impératif (variables, expressions, instructions, fonctions...)
- Un client Web n'a (normalement) jamais accès au code source du programme PHP.

Avantages par rapport aux langages classiques : Spécialement conçu pour faciliter la réalisation de pages Web.

Alternatives : ASP, Servlets Java, JSP, scripts CGI Perl...

- Script PHP : document XHTML (par exemple), dans lequel est incorporé du code PHP.
- Le code PHP est à l'intérieur d'une pseudo-balise `<?php ... ?>`.

Exemple

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/ xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  ...
  <body>
  <h1><?php echo 'Hello world!'; ?></h1>
</body>
</html>
```


Plan du cours

1 PHP : un langage de programmation
2 Introduction au langage
3 Opérations diverses
4 Structures de contrôle
5 Validation
6 Références
7 Applications

- Un script PHP est une suite d'instructions terminées par des points-virgules.

Exemple (Écrire une phrase avec l'instruction echo)

```
echo 'Ceci apparaîtra dans la page générée.';
```

- Ces instructions contiennent des éléments variables ou constants, peuvent être conditionnées ou encore itérées plusieurs fois.

- Il existe trois manières d'inclure des commentaires au sein d'un code PHP :
  1. tout ce qui suit `//` sur une ligne ;
  2. tout ce qui suit `#` sur une ligne ;
  3. tout ce qui est à l'intérieur des signes `/* ... */`.

Exemple

```
ligne PAS commentée
# ligne commentée
ligne PAS commentée
// ligne commentée
ligne PAS commentée
/* ligne commentée
  ligne commentée
  ligne commentée */
ligne PAS commentée
```


- Une variable est le nom d'un récipient destiné à contenir une valeur. Dans un ordinateur, ce récipient est en fait une zone mémoire.
- L'affectation `$var=valeur` est une instruction qui permet de changer la valeur d'une variable `$var` :
  - La valeur de la variable à gauche de `=` est remplacée par la valeur à droite de `=`.
- L'affectation est une instruction dite destructive : l'ancienne valeur de la variable est détruite, écrasée, effacée par la nouvelle valeur !
- Une valeur peut être :
  - un nombre entier ou réel ;
  - une chaîne de caractères définie entre deux guillemets simple (`'`) ou entre deux guillemets double (`"`).

Exemple

```
$Somme = 0; // se lit « la variable $Somme reçoit la valeur 0. »
$SommeEntiere = 0;
$SommeReelle = 0.0;
$Chaine1 = 'PHP, c\'est facile!';
$Chaine2 = "Pince mi et Pince moi";
```


- Dans les chaînes de caractères, il existe des caractères particuliers précédés par un `\` :

Caractère	Description
`\n`	Saut de ligne
`\t`	Tabulation
`\\`	Le caractère \
`\$`	Le caractère $
`\'`	Le guillemet simple
`\"`	Le guillemet double

- Dans les chaînes de caractères entre guillemets simples (`'`), seuls `\\` et `\'` sont disponibles.

- La valeur affectée à une variable peut également être le résultat d'une expression.
- Attention, à droite d'une affectation `=`, les variables utilisées dans l'expression désignent les valeurs qu'elles contiennent.

Exemple

`$Compteur= $Compteur + 1;` a pour effet de mettre le résultat de la somme de la valeur de `$Compteur` avec la valeur 1 dans le récipient `$Compteur` .

Plan du cours

1 PHP : un langage de programmation
2 Introduction au langage
3 Opérations diverses
4 Structures de contrôle
5 Validation
6 Références
7 Applications

- L'opérateur le plus courant est la concaténation de chaîne, simplement représentée par le point (`.`).

Exemple

```
$titre = "Mr le Président";
$prenom = "Nicolas";
$nom = "Sarkozy";
$president = $titre . " " . $prenom . " " . $nom;
$age = 52;
echo "$president a $age ans"; // Concaténation implicite
```

(Le dernier exemple ne marche pas avec des guillemets simples).

Opérateur	Description
`$a + $b`	Addition de `$a` et `$b`
`$a - $b`	Soustraction de `$b` à `$a`
`$a * $b`	Multiplication de `$a` par `$b`
`$a / $b`	Division de `$a` par `$b`
`$a % $b`	`$a` modulo `$b` (reste de la division de `$a` par `$b`)

Exemple

```
$a = 3;
$b = 5 + $a;
$c = $a + $b;
```


- Le résultat d'une comparaison est une valeur booléenne i.e. soit `true` (vrai) soit `false` (faux)

Opérateur	Description
`$a == $b`	Vrai si `$a` est égal à `$b`
`$a != $b`	Vrai si `$a` est différent de `$b`
`$a < $b`	Vrai si `$a` est strictement inférieur à `$b`
`$a > $b`	Vrai si `$a` est strictement supérieur à `$b`
`$a <= $b`	Vrai si `$a` est inférieur ou égal à `$b`
`$a >= $b`	Vrai si `$a` est supérieur ou égal à `$b`

- Les opérateurs logiques travaillent sur les valeurs booléennes `true` et `false` .
- Cependant, une valeur non booléenne peut être interprétée comme étant `true` si elle est différente de 0 ou de la chaîne vide et `false` sinon.

Opérateur	Description
`$a && $b`	Vrai si `$a` ET `$b` sont vrais
`$a || $b`	Vrai si `$a` OU `$b` est vrai
`!$a`	Vrai si `$a` est faux et Faux si `$a` est vrai

- Propriétés du `||`
  - `$b || $a == $a || $b`
  - `false || $a == $a`
  - `true || $a == true`
  - `$a || !$a == true`
- Propriétés du `&&`
  - `$b && $a == $a && $b`
  - `true && $a == $a`
  - `false && $a == false`
  - `$a && !$a == false`

Plan du cours

1 PHP : un langage de programmation
2 Introduction au langage
3 Opérations diverses
4 Structures de contrôle
5 Validation
6 Références
7 Applications

- Une instruction conditionnelle est de la forme :
  Si *la condition C est vraie* alors *faire liste d'instructions T*
  sinon *faire liste d'instructions E*
- où une condition est une expression et T (resp. E) est un bloc d'instructions.
- En PHP, on utilise la syntaxe suivante :

```
if (C) { T } else { E }
```

Exemple

```
if ($sexe=='f') {
  echo 'Madame';
} else {
  echo 'Monsieur';
}
```


- La condition est généralement un test de comparaison, ou plusieurs tests reliés par des opérateurs logiques.
- Le `else` et son bloc associé sont optionnels.
- Si un bloc ne contient qu'une seule instruction, on pourra omettre les accolades.

Exemple

```
if ($sortie=='q') { echo 'Merci et au revoir!'; }

if ($a > $b) { echo "$a est plus grand que $b"; }
else {
  if ($a==$b) { echo "$a est égal a $b"; }
  else { echo "$a est plus petit que $b"; }
}
```


- Une itération correspond à la répétition d'une séquence de calcul
- Il existe différentes façons d'itérer un calcul :
  - Tant que *la condition est vraie*
    faire *le bloc d'instructions*
  - Pour *un index* allant de *une borne inférieure* à *une borne supérieure*
    faire *le bloc d'instructions*

- En PHP, les boucles *Tant que ... faire ...* se notent :

```
while (C) { B }
```

- Pour utiliser correctement une telle boucle, il faut :
  - initialiser les variables utilisées dans la condition C
  - modifier dans le bloc B au moins une des variables utilisées dans C pour que la boucle puisse s'arrêter.

Exemple

```
while(true) { echo 'Au secours!'; }

$fact = 1;
$N = 13;
while($N > 0) {
  $fact = $fact*$N;
  $N = $N - 1;
}
```


- En PHP, les boucles *Pour ...* se notent :

```
for (I;C;P) { B }
```

- où I est une affectation permettant d'initialiser la variable itérée,
- C est la condition d'arrêt de la boucle (valeur limite que doit atteindre la variable itérée),
- P est le pas de la boucle : c'est une affectation qui permet de modifier à chaque itération la valeur de la variable itérée,
- B est le bloc d'instructions itéré.

Exemple

```
for( $i=0 ; $i<10 ; $i=$i+1 ) { echo 'Au secours!'; }

$fact=1;
for( $i=13 ; $i>0 ; $i=$i-1 ) { $fact = $fact*$i; }
```


Plan du cours

1 PHP : un langage de programmation
2 Introduction au langage
3 Opérations diverses
4 Structures de contrôle
5 Validation
6 Références
7 Applications

- Un script PHP avec du code PHP incorporé au sein de code XHTML n'est pas un document XHTML valide !
- Ce qu'il faut valider, c'est une (des) page(s) XHTML produites par le script.
- Possibilité d'indiquer une URL au validateur du W3C. Malheureusement pas utilisable quand l'URL est privée. Possibilité dans ce cas de sauvegarder le fichier XHTML produit, et de l'envoyer au validateur du W3C comme fichier local.
- Il n'y a pas de « validateur » PHP, mais les erreurs de syntaxe causeront des erreurs à l'exécution du script.
- Rien ne change pour les CSS, on peut les valider à part.

Plan du cours

1 PHP : un langage de programmation
2 Introduction au langage
3 Opérations diverses
4 Structures de contrôle
5 Validation
6 Références
7 Applications

- *Introduction à l'informatique*, de Fréderic Gruau et Philippe Dague, Université Paris-Sud, L1-MPI-S1
- `http://www.php.net/`
- *Pratique de MySQL et PHP*, Philippe Rigaux, O'Reilly

Plan du cours

1 PHP : un langage de programmation
2 Introduction au langage
3 Opérations diverses
4 Structures de contrôle
5 Validation
6 Références
7 Applications

- Afin de pouvoir tester les scripts qui suivent, placez ces lignes tout au début de votre page PHP. Ils permettent de récupérer les paramètres de la requête HTTP dans les variables `$M` et `$N`. On comprendra mieux au prochain cours.

```
<?php
  $M=$_REQUEST['M'];
  $N=$_REQUEST['N'];
?>
```

- Ainsi, on testera le script *toto.php* du répertoire *titi*, sur le serveur *ifips.senellart.com*, avec `$M=10` et `$N=20`, à l'URL : `http://ifips.senellart.com/titi/toto.php?M=10&N=20`

Penser à tester tous vos scripts en passant différentes valeurs aux variables dans l'URL.

- Écrire un script PHP qui teste si l'entier `$N` est le carré de l'entier `$M`.
- Écrire un script PHP qui calcule, par additions successives, le produit de deux variables `$M` et `$N` en utilisant une boucle `while`.
- Écrire un script PHP qui calcule, par additions successives, le produit de deux variables `$M` et `$N` en utilisant une boucle `for`.
- Écrire un script PHP qui, si l'année contenue dans la variable `$M` est bissextile, écrit un message correspondant.
  - Toute année divisible par 4 est bissextile.
  - Mais seuls les siècles divisibles par 400 sont bissextiles.
- Écrire un script PHP qui construit la table de multiplication correspondant à l'entier `$M`.