Développement Web

Le duo magique: PHP et MySql

Halim Djerroud <hdd@ai.univ-paris8.fr>

LIASD - Université Paris 8

2018

Plan de cours général

8 semaines de cours :

1. Introduction au web dynamique
2. Gestion de rendu et décoration
3. **Le duo magique : PHP et MySql**
4. Programmation OO et les design patterns
5. TP noté pour faire une pause et un peu de JS avec ses saveurs (JQuery).
6. Les framworks et Symfony4
7. Symfony4 et ses rouages
8. Correction de projets

Plan de cours

- 1h30 Cours
- 2h00 Travaux pratiques.

1. PHP
2. MySql
3. PDO

Support de cours

- http://djerroud.halim.info/index.php/cours/programmation-web/

Introduction PHP

- PHP pour **PHP Hypertext Preprocessor**
- PHP s'exécuter côté serveur
- PHP permet créer des sites dynamiques et interactifs
- PHP permet l'interface avec des Bases de données
- PHP est un langage Orienté Objet

Emplacement du code PHP

- Insertion des scripts PHP au sein de nos pages HTML

```
<html>
<head> ... </head>
<body> ...
<?php
  // code php
?>
...<body>
</html>
```

Types de variables

- Conteneur servant à stocker des informations
- Type de variables automatique
- Les variables commencent par le symbole $

```
...
<?php
  $text = "ceci est un text";
  $pi   = 3.14;
  $i    = 10;
  $b    = true ;
  $obj  = new Class();
?>
...
```

Variables

Type de variables :

- Entiers, long ex : $\$a = -100;\ \$b = 10;$
- Flottants, double ex : $\$pi = 3.14159265358979323 8$
- Strings ex : $\$t1 = "texte";\ \$t2 = 'texte'$
- Caractère ex : $\$c = 'a';$
- Boolean ex : $\$b = false;$
- Objets ex : $\$obj = new\ class();$
- `NULL` : la valeur null correspond à l'absence de valeur

Consulter le type d'une variable :

- La fonction `gettype()` permet d'obtenir le type d'une variable

Afficher le contenu d'une variables

- La fonction echo permet d'afficher le contenu d'une variable

Code

```
...
<?php
  $text = "ceci est un text";
  echo $text ;
?>
...
```

Résultat

```
ceci est un text
```

Les opérations sur les variables

Opération	**Opérateur**
Addition	+
Soustraction	-
Multiplication	*
Division	/
Modulo	%
Exponentiel	**
Concaténation	.

Les opérations sur les variables

Opération	**Opérateur**
Additionne puis affecte	+=
Soustrait puis affecte	-=
Multiplie puis affecte	*=
Divise puis affecte	=
Calcule le modulo puis affecte	%=

Les opérations sur les variables

Opération	**Opérateur**
Incrémente $x, et retourne la valeur incrémentée	++$x
Retourne $x, et incrémente par la suite	$x++
Décrémente $x, et retourne la valeur décrémente	- -$x
Retourne $x, et décrémente par la suite	$x- -

Les condition

- La condition `if` (si) ;
- La condition `if... else` (si... sinon) ;
- La condition `if... elseif... else` (si... sinon si... sinon).

```
...
<?php
  if($a % 2 == 0){
    echo "impaire";
  else{
    echo "paire";
  }
?>
...
```

Les opérateurs de comparaison

Signification	Opérateur
Est égal (en valeur) à	==
Est égal en type et en valeur à	===
Est différent (en valeur) de	!=
Est différent en valeur ou en type de	!==
Est strictement supérieur à	>
Est strictement inférieur à	<
Est supérieur ou égal à	>=
Est inférieur ou égal à	<=

Les opérateurs logiques

Signification	**Opérateur**
seulement si toutes les comparaisons sont vrai	AND (ou &&)
L'une des comparaisons est vrai	OR (ou ||)
Uniquement une des comparaisons est vrai	XOR
Inverse la valeur logique d'un test	!

Le switch

- Permet d'effectuer différentes actions en fonction de différents cas

```
...
switch($a){
  case 0:
    echo "zero";
    break;
  case 1:
    echo "Un";
    break;
  ...
  default:
    echo "je ne sais pas !";
}
```

L'opérateur ternaire

- Opérateur conditionnel sur en une seul ligne de code

```
...
echo ( $a == 10 ? "vrai" : "faux" );
```

```
...
if($a == 10){
  echo "vrai";
}
else {
  echo "faux";
}
```

La boucle *for*

```
...
for($i=0; $i < 10 ; $i++){
    ...
}
...
```

La boucle *while*

```
...
$i = 10;
while($i--){
    ...
}
...
```

```
...
$i = 10;
while($i){
    ...
    $i--;
}
...
```

La boucle *do while*

```
...
$i=10;
do{
    ...
} while($i--);
...
```

Les fonctions

```
...
fonction f1($p1, $p2){
  return $x;
}
...
$i = f1(1,2);
...
```

Les tableau

- On déclare un tableau avec `array()`
- Un tableau peut prendre des valeurs de différentes types
- Un tableau est indexé à partir de 0

```
...
$tab = array ("un", "deux", "trois");
...
```

Afficher le contenu d'un tableau

Pour afficher le contenu d'un tableau on peut utiliser :

- Parcourir et afficher les éléments un par un avec les boucles (for, while, do...while)
- Utiliser `foreach`
- Utiliser la fonction `var_dump()`
- Utiliser la fonction `print_r()`

Afficher le contenu d'un tableau avec for

```
...
$tab = array ("un", "deux", "trois");
for($i=0 ; $i < count($tab) ; $i++)
{
   echo $tab[$i];
}
...
```

Afficher le contenu d'un tableau avec print_r()

Code

```
...
$tab = array ("un", "deux", "trois");
print_r($tab);
...
```

Résultat

```
Array
(
    [0] => un
    [1] => deux
    [2] => trois
)
```

Afficher le contenu d'un tableau avec var_dump()

Code

```
...
$tab = array ("un", "deux", "trois");
var_dump($tab);
...
```

Résultat

```
array(3) {
    [0]=>string(2) "un"
    [1]=>string(4) "deux"
    [2]=>string(5) "trois"
}
```

Afficher le contenu d'un tableau avec foreach()

Code

```
...
$tab = array ("un", "deux", "trois");
foreach($tab as $line){
    echo $line;
}
...
```

Les tableaux associatifs

- Permet de choisir un index au-lieu d'utiliser l'index proposé par défaut

Code

```
...
$tab ["titre"]       = "PHP" ;
$tab ["Description"]= "Cours PHP" ;
$tab ["duree"]       = 30 ;
$tab ["note"]        = 4.5;
...
```

Afficher le contenu d'un tableau avec foreach()

Code

```
...
$tab = ... ;

foreach($tab as $key => $val){
    echo "Clé : " . $key . " Valeur : " . $val;
}
...
```

Introduction PDO

- Définit interface pour accéder à une base de données depuis PHP
- Chaque pilote de base de données implémenté dans l'interface PDO
- Pour notre cas nous allons utiliser MySql (Maria DB)
- PDO fournit une interface d'abstraction à l'accès de données, donc les mêmes fonctions pour n'importe quelle BD

connexion à la base de donnée

- Les connexions sont établies en créant des instances de la classe de base de PDO

Code

```
...
$user = "myuser";
$pass = "mypassword";
$dbh = new PDO('mysql:host=localhost;dbname=test',
                $user, $pass);
...
```

Exécuter une requête SQL

- On utilise l'objet retourné par PDO pour effectuer une requête via la méthode `query()`

Code

```
...
$dbh = new PDO('mysql:host=localhost;dbname=test',
                $user, $pass);
...
$sth = $dbh->query('SELECT * FROM foo');
...
```

Diviser le code en plusieurs fichiers

- La programmation Web impose un style de programmation particulier qui impose beaucoup de réutilisation de code

Code

```
...
<?php
// Importations avec require()
require('../dossier/fichier.php');
require 'fichier2.php';

// Importations avec include()
include('../dossier/fichier.php');
include 'fichier2.php';
...
```

Diviser le code en plusieurs fichiers

- Les fonctions `require_once()` et `require_once()` permettent d'importer une fois seulement un fichier même s'il y'a plusieurs tentatives d'importation du fichier dans la page.