Cours PHP

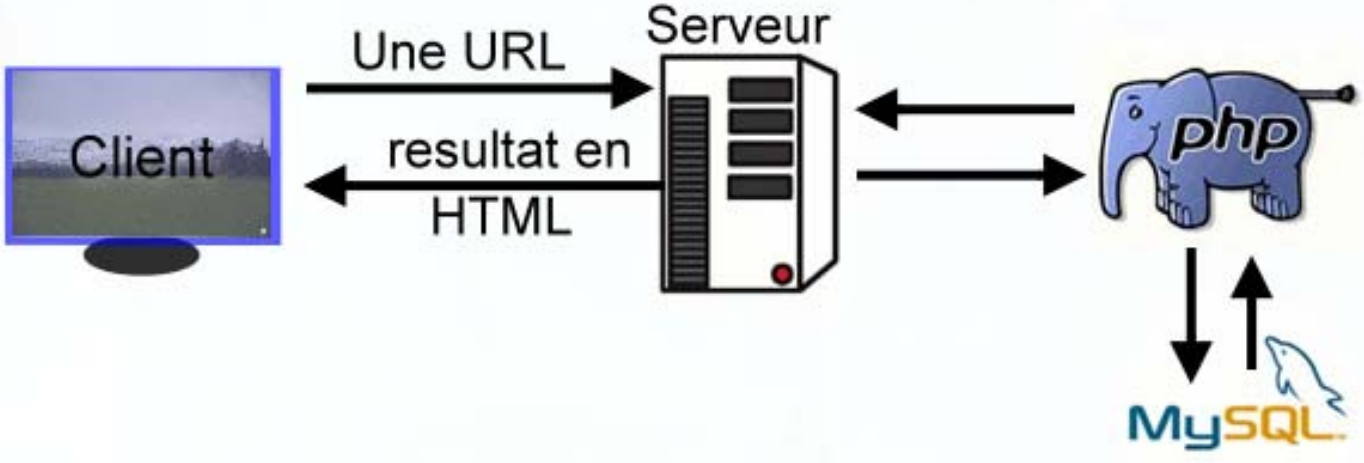
MAHMOUDI Khattab

Commencer

Dans ce Tutoriel on s'intitule à la découverte du language PHP. Nous allons découvert :

- Qu'est-ce que PHP ?
- Avantage et inconvénient?
- Quel outil pour faire du PHP ?
- Codage PHP
- Les constantes et les variables
- Les structures conditionelles
- Les structures itératives
- Les fonctions
- Les fonctions prédéfinie
- PHP et les formulaires
- PHP et base de données

Peut-on créer une page web qui change de contenu avec HTML et JavaScript seulement ?
Pour avoir des pages web qui changent du contenu on a besoin d'une base de données.
HTML et JavaScript ne permettent pas de questionner un serveur de gestion de base de données.
On a besoin d'un mécanisme permettant la communication avec une base de données.
Le langage PHP(**PHP: Hypertext Preprocessor**) peut etre une solution parmi plusieurs.



Mode de fonctionnement

Avantage:

- Générer dynamiquement des pages web.
- C'est gratuit.
- C'est simple
- La plus grande qualité et le plus important avantage du langage PHP est le support d'un grand nombre de bases de données.

Inconvénient:

- Ce n'est pas un langage compilé c'est un langage interprété ::> le code source n'est pas protégé

Il faut trouver un hébergeur mettant à disposition un interpréteur PhP. Parmi ces hebergeurs on peut citer EasyPHP, Xampp, ...

EasyPHP:

- Telechargement: **http://easyphp.softonic.fr/telecharger**
- Deux dossiers sont importants: le dossier **www** et le dossier **mySql**. Le premier contient le dossier des sites web, et le deuxieme contient les bases de donnees.

XAMPP:

- Telechargement: **http://www.apachefriends.org**
- Deux dossiers sont importants: le dossier **htdocs** et le dossier **mySql**. Le premier contient le dossier des sites web, et le deuxieme contient les bases de donnees.

Pour tester un serveur Appache Il suffit, le serveur étant démarré,de taper **http://localhost** ou **http://127.0.0.1/**

- Le code PHP s'exécute dans le coté serveur pour accélérer le temps de chargement des pages web il faut utiliser PHP que lorsqu'on a besoin de se connecter a une base de donnés ou lorsqu'on a besoin d'une valeur de l'environnement serveur.

- Si un fichier contient même une seule instruction en PHP il faut l'enregistrer avec l'extension **.php**

- Un script php commence par un balise d'ouverture **< ?php** et se termine par une balise de fermeture **?>** de cette façon **<?php** xxxxscriptxxxx **?>** ou encore **<?** xxxxscriptxxxx **?>**

Activité 1 :

Créer un dossier nommé activités dans le dossier www du dossier d'installation de easyPHP. Dans ce dossier créer un fichier nommé activite1.php contenant le code suivant :

```
<HTML>
<Head><Title>Premier code php</Title></Head>
<Body>
< ?php
Echo "<B><center>Hellow word !!</center></B>" ;
Echo "Chaine de caracteres";
Echo (1+2)*7;
?>
</body>
</HTML>
```

Question: Quel est le résultat affiché par ce fichier**?** Quel est le rôle de la fonction **echo?**
Remarque: pour voir le résultat de la page ecrire l'URL suivante:
http://localhost/activités/activite1.php
Réponse:

Les variables:

Le nom des variables doit :

- commencer par le symbole **$**
- comporter que des lettres **A-Z**, **a-z** et les chiffres de **0 à 9 ,** l'underscore _ (la longueur n'estpas limitée)
- ne pas contenir de caractère espace ou autres "hors liste"
- ne pas commencer par un chiffre

La déclaration est optionnelle, car php décide lui-même du type de lavariable lors de sa première affectation entre string (chaîne) integer et double(numérique)

La fonction **isset**(variable) returne vrai si la variable contient une valeur et faux si non

La fonction **gettype**(variable) retourne le type de la variable: integer,double,string,array..

Exemple: Pour les chaines de caracteres: $nom = "Dupont" ou $nom='Dupont'

Pours les variables numeriques:

```
<?php
$quantite=6;
echo $quantite // affiche 6
echo isset( $quantite) // affiche 1 c'est a dire vrai
?>
```

Les constantes:

Les constantes se comportent comme des variables, à l'exception du fait que leur valeur est définie grâce à la fonction define(), et qu'elle ne peut pas être modifiée par la suite

define("NomConst","Valeur"); Exemple: define("**PI**",3.14) echo PI; affiche 3.14

NB: les constantes ne peuvent avoir un nom commençant par **$** (on lesconfondrait avec des variables)

Exemple:

```
<?php
define("CONSTANT", "Bonjour le monde.");
echo CONSTANT; // affiche "Bonjour le monde."
?>
```

Les opérateurs

Arithmétiques :

n'ont de sens que s'ils sont utilisés sur des variables de type Numérique

l'addition (**+**) la multiplication (*) la soustraction(**-**) la division (**/**) le modulo (**%**) : reste de la division entière d'un entier par un autre

L'affectation:

L'opérateur d'affectation le plus simple est le signe "**=**".

"certains fanatiques" peuvent utiliser les conventions suivantes

Opération	Signification	Exemple
x+=a	x=x+a	**$a=3;**
x-=a	x=x-a	**$a += 5; // affecte la valeur 8 à la variable $a.**
x*=a	x=x*a	**(correspond à'$a = $a + 5');**
x/=a	x=x/a	**$b = "bonjour ";**
x++ ou ++x	x=x+1	**$b .= "ici!"; // affecte la valeur "Bonjour ici!" à**
x-- ou --x	x=x-1	**la variable $b**

Les opérateurs de comparaison :

== égalite **!=** différence **<** **>** **>=** **<=**

Les opérateurs logiques:

|| L'opérateurs ou logique **&&** L'opérateurs ET logique **!** L'opérateurs non logique

Les tableaux

PHP supporte les tableaux scalaires et les tableaux associatifs. En fait, il n'y a aucune différence entre les deux. Vous pouvez créer un tableau en utilisant la fonction array(), ou bien en affectant explicitement chacune des valeurs.

Exemples:

```
$a[0] = "abc";
$a[1] = "def";
$b["note"] = 13; // tableau "associatif"
```

Vous pouvez aussi créer un tableau en ajoutant simplement les valeurs à ce tableau.

```
$a[] = "hello"; // revient à $a[2] == "hello"
$a[] = "world"; // revient à $a[3] == "world"
```

On peut créer un tableau par l'instruction array

Exemple: `$tab = array(0,2,4,6);`

Vous pouvez compter le nombre d'éléments qu'il y a dans un tableau en utilisant la fonction count().

Un tableau peut être trié en utilisant la fonction sort() et rsort() pour inverser le tri.

Avançons un peu, voila un tableau à 2 dimensions 2x2

```
$a[0] [0] = "def";
$a[0] [1] = "def";
$a[1] [0] = "def";
$a[1] [1] = "def";
```

Quelques exemples :

- Pas de else, et une seule instruction à exécuter

```
if ($a > $b)
    print "a est plus grand que b";
```

- Avec un else, et plusieurs instructions prévues à exécuter

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

- Pas de else, et plusieurs instructions à exécuter

```
if ($a > $b) {
print "a est plus grand que b";
$b = $a;
}
```

Les opérateurs logiques Et (&&) et OU (||) restent disponibles pour écrire des tests plus complexes.

Application:

Ecrire le code php permettant d'initialiser deux valeurs entieres et d'afficher si la valeur 1 est superieus à la deuxieme ou le contraire ou qu'il sont egaux.

Réponse:

```
<?php
$val1=7; $val2=4;
if($val1==$val2) echo "Les 2 valeurs sont égales";
else
{
   if ($val1>$val2) echo "Valeur 1 superieure à valeur 2";
   else   echo "Valeur 2 superieure à valeur 1";
}
?>
```

Quelques exemples :

- **for** ([valeur initiale] ; [condition] ; [incrément])
{
code
}

- **while** (condition) {
code
}

Application:

Ecrire le code php permettant d'initiliser deux tableaux $noms et $moyennes par les noms et les notes de 5 eleves et de les afficher dans un tableaux HTML

Réponse:

```
<?php
$noms=array("Mohamed Tounsi","Rim Ben Salah","Ali abidi","Foued Nasri","Fatma Dhaouedi");
$notes=array(13.25,14.5,12.75,19.00,18.25);
echo "<table border=1><tr><td>Nom</td><td>Note</td></tr>";

for($i=0;$i<count($notes);$i++)
 echo "<tr><td>".$noms[$i]."</td><td>".$notes[$i]."</td></tr>";
 echo "</table>";
?>
```

- **La syntaxe** de déclaration d'une fonction la plus générique est :

```
function nomdelafonction($param1, $param2, $param3=1, $param4="chaine")
{
// instructions de la fonction disposant de $param1 $param2 $param3 et $param4
}
```

- Un paramètre peut être envoyé par variable ou pare valeur (on ajoute & devant tout paramètre pour l'envoyer par variable)

Exemple:

```
function Afficher($size,$color,$texte)
{
echo("<font face=Arial size=".$size." color=".$color.">".$texte."</font>");
}
```

- **Appel de le fonction**

```
Afficher("2","red","Ici le texte ..."); // Resultat  Ici le texte ...
Afficher("3","#0F74A3","Le second texte ..."); // Resultat Le second texte ...
```

Application

Ecrire le code d'une fonction qui prend en parametre un tableaux a deux dimensions contenant une liste des noms et des abreviations des pays et les affiche dans une zone de liste deroulante

Réponse:

```
<?php
function ListePays($pays){
echo "<select name=pays>";
for($i=0;$i<count($pays[0]);$i++) echo "<option value=".$pays[0][$i].">".$pays[1][$i]."</option>";
}
Appel
$pays[0]=array("Tun","Fra","Esp","Bra");  $pays[1]=array("Tunisie","France","Espagne","Brazil");
ListePays($pays);
?>
```

Fonction	Role	Exemple
strlen()	Retourne la longueur de la chaîne	**strlen**("chainecaracteres");
strtolower()	Passe tous les caractères en minuscules.	**strtolower**("CHAINEcaRActERes");
strtoupper()	Passe tous les caractères en MAJUSCULES.	**strlen**("chainecaracteres")
str_replace()	Remlpace une chaine par une autre	**str_replace**("a","o","Lalala");
trim()	Efface les espaces au début et à la fin d'une chaîne	**trim**(" Salut le monde ");
strpos()	Recherche la position du premier caractères trouvé.	**strpos**("abcdef","e");
implode()	retourne une chaine qui contient les elements du tableau	**implode**(separateur,tableau);
explode()	Retourne un tableau qui contient les elements d'une chaine	**explode**(chaine,separateur);

Exemples:

```
<?php
$ch="12a34a16a9";
echo "La taille de la chaine est:".strlen($ch); // affiche La taille de la chaine est:10
$ch2=strtoupper($ch); // ch2 contient les caracteres de ch mais en majuscules
echo "<br>".$ch2; // affiche: 12A34A16A9
$ch3=str_replace("A","#",$ch2);
echo "<br>".$ch3; // affiche: 12#34#16#9
$t=explode("#",$ch3); // le tableau T contient les elements 12,34,16,9
$ch3=implode($t,"***"); // reconstruction de la chaine
echo "<br>".$ch3; // affiche 12***34***16***9
?>
```

Soit le formulaire suivant :

```
<html>
<body>
<form method="post" action="verif.php">
Nom : <input type="text" name="nom" size="12"><br>
Prénom : <input type="text" name="prenom" size="12">
<input type="submit" value="OK">
</form>
</body>
</html>
```

Method: **Post** ou **Get**
Action: c'est le fichier destination du formulaire.
Pour récupérer la valeur de ces champ
$_POST["nom"]
$_POST["prenom"]
Si la méthode est get on les récupère avec
$_GET["nom"]
$_GET["prenom"]

Options: ☐ Espagnole ☐ Almand ☐ Dessin ☐ Autre

Dans le cas de plusieurs choix avec les checkbox la récupération des choix se fait de la manière suivante:

Le nom de toutes les cases doit être le même comme suit: <input type="checkbox" name="**op[]**" value="esp">
lorsque on écrit **$options=$_POST['op']** ::> **$options** c'est un tableau php contenant les valeurs choisis.

Application:
Ecrire le code php permettant d'afficher le nom et le prenom du formulaire

Réponse:

```
<?php
// page verif.php
echo "Votre nom est ".$_POST['nom']."et votre prenom est".$_POST['prenom'];
?>
```

Connexion au serveur: **mysql_connect**("nom du serveur","login","Mot de passe") dans le cas d'un serveur local
mysql_connect("localhost","root",'');
on peu l'écrire comme suit: **mysql_connect("localhost","root",'') or die("Connexion impossible au serveur");**
Permet d'afficher le message dans le cas ou la connexion n'est pas effectuée.

Connexion a la base: **mysql_select_db**("nom de la base"); on peut l'écrire comme suit:
mysql_select_db("nom de la base") or die ("Base de donnees introuvable"); pour afficher le message dans le case ou la base est introuvable

Exécution des requêtes:
Requête d'**insertion**, de **mise a jour**, ou de **suppression** de données **$req="code de la requête"; mysql_query($req);**

Requête de **selection:**

```
$req="code de la requette de selection";
$res=mysql_query($req);
```

Parcours du résultat
Premiere methode:

```
while($row=mysql_fetch_row($res)){
//$row c'est un tableau contenant les différentes colonnes d'une ligne du résultat
echo $row[0].$row[1];  etc...
}
```

Deuxieme methode:

```
while($row=mysql_fetch_array($res)){
//$row c'est un tableau contenant les différentes colonnes d'une ligne du résultat
echo $row["nomcolonne1"].$row["nomcolonne2"] ;  etc...
}
```

mysql_num_rows($res); //retourne le nombre de lignes de la requête.

mysql_insert_id(); // retourne le dernier element auto increment d'une table