

Programmation PHP (1)

En ligne : http://2igc.cours.free.fr/Cours_Web_Reference/php_cours1.php

1 Prérequis

1.1 Installation

Il faut installer (windows) wampServer, easyPHP ou (linux) tout autre pack AMP qui contiennent :

- Le serveur web apache,
- L'interpréteur de scripts PHP
- Le SGBDr mySQL
- L'application de gestion des bases de données PhpMyAdmin

Avec WampServer, les fichiers sont stockés dans le répertoire de travail C:/wamp/www/.

Sous Linux, ce sera soit dans /home/userId/public_html, soit dans /var/www/.

L'accès aux fichiers se fait avec le navigateur, à l'adresse "localhost".

La modification des fichiers se fera avec (windows) Notepad++ ou (linux) gedit.

1.2 Connaissances

Obligatoire : HTML (PHP produit du HTML)

Conseillées : CSS de base, SQL.

2 Base de syntaxe

Le PHP est un langage de programmation qui sert à produire du texte html envoyé au navigateur du client.

Extension : un script php est écrit dans un fichier **.php**, ils peuvent aussi contenir du HTML pur.

Le php est écrit entre les pseudo balises **<?php** et **?>**

Le code php peut être mis sur une seule ligne, n'importe où dans le code html.

Chaque instruction se termine par un ; (point-virgule).

Les chaînes de caractères sont entourées de " ou "" (simples ou doubles guillemets).

Un commentaire sera précédé de // ou encadré par /* et */

Les {} (accolades) permettent de grouper des instructions en un bloc d'instruction compris comme une seule instruction. Très utile pour les alternatives, les boucles, les fonctions, etc...

Exemple (fichier **test.php**) :

```
ICI, on est en HTML. Et je coupe le html ...
<?php
echo "J'écris en html avec du <b>php</b>"; // envoi d'un texte html au navigateur
?>
Et je remet le html ...
```

3 Les notions de base

3.1 Variables et affectations

<code>\$var</code>	Toute variable commence toujours par \$ suivi d'au moins une lettre ou _	<code>\$var, \$_REQUEST</code>
<code>=</code>	Opérateur d'affectation. L'ancienne valeur est écrasée	<code>\$abc=3;</code>
<code>array</code>	Opérateur de définition de tableau	<code>\$tab=array(...);</code>

Une valeur peut être de n'importe quel **type** : un nombre entier, réel, une chaîne de caractères, une valeur booléenne (VRAI/FAUX), un tableau, un objet ou autre. Le PHP ne tient pas vraiment compte du type de la variable (il donne automatiquement le type de la valeur à la variable).

3.2 Instructions de base

<code>echo</code> <code>print</code>	Envoyer des chaînes de caractères au navigateur. Elles contiennent le code html à afficher chez le client.
<code>include</code> <code>require</code>	Importer des codes sources externes (html ou php)
<code>+</code> , <code>-</code> , <code>*</code> , <code>/</code>	Opérateurs mathématiques classiques
<code>Round</code>	Arrondi
<code>Pow</code>	Puissance
<code>Trim</code>	Supprime les espaces et tabulations devant et derrière une chaîne de caractères
<code>Date</code>	Fonction qui permet d'obtenir la date et/ou l'heure courante. On ajoute le format entre parenthèses : <code>\$aujourd'hui=date("Y-m-d"); \$maintenant=date("Y-m-d H:i:s");</code>

3.3 Exemples

```
echo "Voici du <i>code html</i> généré par <b>php</b>";
include "fichier_a_inclure.php";
```

Les variables et les types

```
<?php // Début du script
    echo "<h4>Exercices avec les variables : </h4>";

    $a=5; $b=3; $c=0;
    $t='Lorem ipsum';
    echo "a=$a, b=$b, c=$c et t=$t";
    echo "<h4>Echanger les valeurs de a et b</h4>";

    $c=$a;
    $a=$b;
    $b=$c;
    echo "<p>C'est fini : <b>a=$a, b=$b (et c=$c)</b></p>";
    echo "<h4>Additionner a+b dans c</h4>";

    $c=$a+$b;
    echo "<p>C'est fini : <b>a=$a, b=$b et c=$c</b></p>";
    echo "<h4>Concaténer a et b (mettre les valeurs bout à bout) dans c</h4>";

    $c=$a.$b;
    echo "<p>C'est fini : <b>a=$a, b=$b et c=$c</b></p>";
    echo "<h4>On ajoute le texte 'lorem ipsum' à c ...</h4>";

    $c=$c." lorem ipsum";
    echo "<p>C'est fini : <b>c=$c</b>. (noter qu'il faut gérer les espaces soi même)</p>";
    // FIN du script
?>
```

Tableaux à indice numérique

```
$t=array('val0', 'val1', 'val2', 'val3', 'val4', 'val5', 'val6'); // déclaration
echo $tab[3]; // lecture
$tab[3]=3*5; // modification
echo $tab[3];
```

Une chaîne de caractère est aussi un tableau de caractères :

```
$t="Lorem ipsum";
echo $t[3]; // affiche la lettre "e"
```

Tableaux à indice associatifs/alphabétiques

```
$user=array('id'=>1, 'nom'=>'toto', 'age'=>33).
echo $user['nom']; // affiche 'toto'
```

4 Programmation structurée - intro

Un langage structuré intègre des structures de programmation dont : les **conditionnelles** et les **boucles**.

4.1 Les conditionnelles ou alternatives

Les **conditionnelles** permettent de faire un choix selon des conditions déterminées.

Exemple : si le temps est beau : je mets un T-shirt, sinon : je mets un pull.

On a bien une condition : "*le temps est beau*" et des instructions qui sont exécutées selon que la condition est vraie ou non (il n'y a pas d'intermédiaire possible)

Autre Exemple : si (totalHorsTaxe>1000) alors : transport=0, sinon : transport=15.

Exemples :

```
<?php
    // Si le temps est beau
    $temps='beau'; // initialisation
    if ($temps=='beau') echo "je met un t-shirt";
    else echo "je met un pull";

    // Si (totalHorsTaxe>1000)
    $totalHorsTaxe=0; // initialisation
    if (totalHorsTaxe>1000) $transport=0; else $transport=15;
?>
```

4.2 Les boucles

Les **boucles** sont faites pour répéter un traitement, un affichage, un calcul plusieurs fois.

Exemple :

```
POUR i DE 1 À 10 : afficher i.
```

On va compter i de 1 à 10 et afficher la valeur de i à chaque comptage.

On a bien une boucle "i de 1 à 10" et une instruction à exécuter à chaque passage de la boucle : "afficher la valeur de i".

Autre Exemple :

```
TANT_QUE (article=lirePanier)
    totalHorsTaxe=totalHorsTaxe+prix(article)*quantite(article).
```

Le code correspondant aux exemples :

```
<?php
    // Pour i de 1 à dix, afficher i
    For ($i=1;$i<=10;$i++) echo "i=$i<br>";

    // Pour chaque article du panier des achats, calculer le total HT du panier
    while ($article=lirePanier())
        $totalHorsTaxe = $totalHorsTaxe + prix($article)*quantite($article);
?>
```

5 Algorithmes, les alternatives

```
<, >, <=, >=, <>
```

Opérateurs de comparaison idem en maths

==	Opérateur de comparaison d'égalité (double égal) /!\ ne pas confondre avec l'opérateur d'affectation = (simple égal), c'est une erreur courante !
&&,	Opérateurs de conditions ET et OU (AND et OR)
!	Opérateur de négation NOT
isset	Fonction de vérification qu'une variable soit déclarée, retourne TRUE/FALSE
Empty	Fonction de vérification du contenu d'une variable, retourne TRUE si vide
is_array	Fonction de vérification que la variable soit ou non de type tableau
is_str	Idem pour les chaînes

<pre>if (...) {instructions1} else {instruction2}</pre>	<p>Instruction alternative classique, Le bloc else n'est pas obligatoire</p>
elseif	Mot unique pour enchaîner des alternatives
<pre>switch (...) { case (...): instructions break; default : instructions break; }</pre>	<p>Alternative SELON QUE, permet de faire des sélections de choix multiple de valeurs pour une variable.</p> <p>L'instruction break permet de terminer l'étude des cas, sans elle, php lit toutes les instructions, même celle pour les cas suivants.</p>

Exemple :

```
<?php
$age=17;
if ($age>=18) echo "La condition est vraie : Vous êtes majeur(e)";
else echo "La condition est fausse : Vous êtes mineur(e)";

if ($montant >100 && $montant<=1000) {...}
if ($statLivr ==true && $qteLiv>=$qteTotale) {...}
if ($temps =="beau" && $temperature>25) {...}

$age=17;
    if ($temps=='pluie') echo "Je prend le parapluie";
    elseif ($temps =="beau" && $temperature>25)
        echo "je mets un t-shirt";
    else
        echo "je mets un pull";
?>
```

Le code correspondant :

```
<?php
$temps=brouillard;
switch ($temps) {
    case ('pluie'): echo "je prend un parapluie";break;
    case ('beau'): echo "je mets un t-shirt";break;
    case ('brume'): echo "je mets un pull";break;
```

```

default: echo "je mets une cravate et une veste";break;
}
?>

```

6 Les boucles

for (...;...;...) {instructions}	POUR (origine, fin, incrémentation) Le nombre de termes, la fin est connue par avance. Parcours de tableau.
foreach (\$tab AS \$key => \$val) {instructions}	Boucle permettant de parcourir des tableaux sans connaître leur longueur
while (condition) {instructions}	TANT QUE (condition) ... Les instructions sont exécutées tant que la condition est vérifiée Parcours de liste dont le nombre de ligne n'est pas connu à l'avance.
do {instructions} while (condition);	REPETER ... TANT QUE (condition de continuation) Les instructions sont exécutées une fois puis on fait le test : si il faut poursuivre le traitement. Forme d'instruction moins utilisé.
until	Instruction non disponible en PHP REPETER ... JUSQU'À (condition d'arrêt)

Exemple :

```

<?php
echo "Afficher la table de trois";
for ($i=1;$i<=10;$i++)
    echo "3*$i=".(3*$i)."<br>";
?>

```

Foreach

```

<?php
$fruits = array('pomme', 'orange', 'poire', 'cerise', 'carambole', 'banane');
$i=0;
foreach ($fruits as $key =>$val) {
    echo "$key : " .($val)."<br>";
}
?>

```

Tant que

```

<?php
$fruits = array('pomme', 'orange', 'poire', 'cerise', 'carambole', 'banane');
$i=0;
while (isset($fruits[$i]) && $unFruit=$fruits[$i]) {
    echo "$i : " .($unFruit)."<br>";
    $i++;
}
?>

```

Tant que (2)

```

$connexion_id=mysqli_connect($server, $user, $password); // 1) connexion
$db_id=mysqli_select_db($connexion_id, $databaseName); // 2) sélection de la base
$query="SELECT * FROM fruits "; // 3) préparation de la requête
$query_id=mysqli_query($connexion_id, $query); // 4) exécution de la requête
while ($record=mysqli_fetch_array($query_id)) { // 5) TANT QUE la lecture fonctionne

```

```

        echo $record['nom']."<br>"; // 6) traiter l'enregistrement
    }
    mysqli_close($connexion_id);

```

Répéter Tant Que

```

$i=0;
do { // répéter
echo "$i : ".$fruits[$i]."<br>";
$i++;
}while ($i<=sizeof($fruits)); // tant que i < longueur(tableau)

```

On utilise l'instruction sizeof() pour obtenir la longueur du tableau.

Exemple UNTIL en VB

```

'On est en Visual Basic !!
Dim fruits() as string = {'pomme', 'orange', 'poire', 'cerise', 'carambole',
'banane'};
Dim i as integer =1;
Dim message as string;
Do 'répéter
message= i & " : " & fruits(i) & vbCrLf
i=i+1
Loop Until ($i>UBound($fruits)) 'jusqu'à i>longueur(tableau)
MsgBox message

```

7 Bases de données

\$connect_Id=mysqli_connect(\$server, \$user, \$mdp);	Se connecter à un serveur mysql
Mysqli_db_connect(\$connect_Id, \$dbName);	Sélectionner une base de données
\$query_Id=mysqli_query(\$connexion, \$query);	Exécuter une requête
\$record=mysqli_fetch_array(\$query_Id);	Lire une ligne d'une requête vers un tableau mixte
\$record=mysqli_fetch_row(\$query_Id);	Lire une ligne d'une requête vers un tableau numérique
\$record=mysqli_fetch_assoc(\$query_Id);	Lire une ligne d'une requête vers un tableau associatif
\$record=mysqli_fetch_obj(\$query_Id);	Lire une ligne d'une requête vers un objet
Mysqli_close(\$connect_Id);	Fermer une connexion à un serveur mysql

Voir aussi les PDO

8 Liens utiles

Sur 2igc :

http://2igc.cours.free.fr/Cours_Web_Reference/php_liste_instruction.php

http://2igc.cours.free.fr/Cours_Web_Reference/

En cas d'erreur (gros cadre orange) : <http://www.phpdebutant.org/article112.php>

Pour commencer : <http://www.phpdebutant.org/article118.php>

Pour commencer (aussi) : <http://sylvie-vauthier.developpez.com/tutoriels/php/grand-debutant/?page=intro#LI>

Pour la référence : www.php.net

Ou [<nom>](http://www.php.net) instruction pour aller directement sur l'aide de l'instruction

Et plein d'infos sur alsacreations, openclassrooms, php fance, developpez.com, etc. ...