Cours de PHP

Qu'est-ce que PHP ?

PHP (ou Hypertext Preprocessor) est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web. Il p
intégré facilement au HTML.

PHP est un language de programmation interprété, cela signifie que le code source (celui que vous écrivez) est interprété, par un logiciel qu'on appelle inter
Celui-ci va utiliser le code source et les données d'entrée pour calculer les données de sortie. L'interprétation du code source est un processus « pas à pas »
l'interpréteur va exécuter les lignes du code une par une, en décidant à chaque étape ce qu'il va faire ensuite.

Que faut-il installer ?

Pour pouvoir développer vos applications web en php, vous allez devoir mettre en place un serveur web local sur votre machine. Un serveur web PHP comp
éléments suivants:

--> Serveur Apache (http://www.apache.org)

--> Interpréteur de code PHP (http://www.php.net)
--> Base de données MySQL (http://www.mysql.com)
--> Utilitaire phpMyAdmin, qui permet de créer et de gérer bases et tables de données MySQL (http://www.phpmyadmin.net)

Installation

Windows:
Nous allons pouvoir installer tous ces éléments en une seule opération, grâce à une solution appelée WAMP Server, évitant du même coup les problèmes de configuration. WAMP server est disponible à l'adresse suivante:

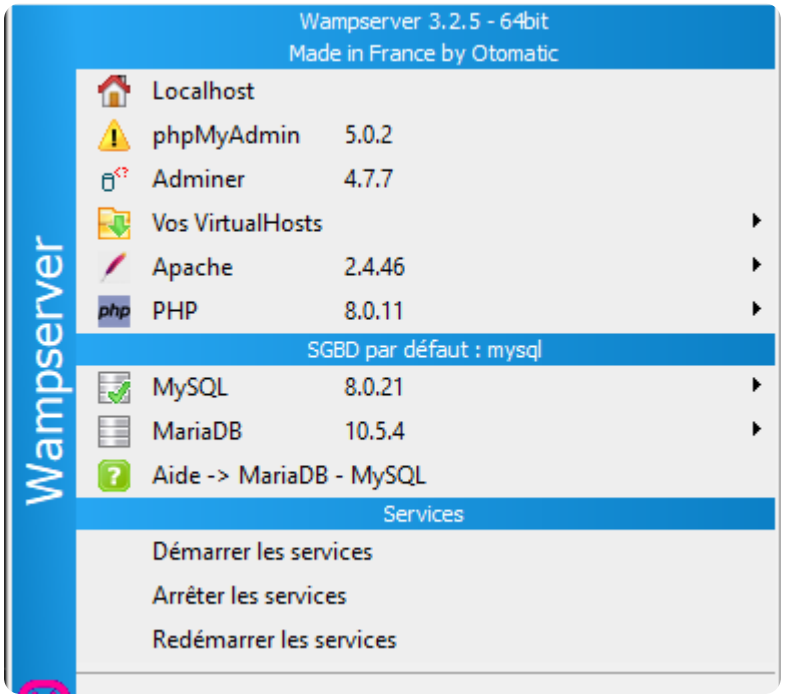
http://www.wampserver.com

Mac et Linux:
Pour les utilisateurs de Mac OS, il existe une solution appelée MAMP et pour les distributions linux, il existe une solution nommée LAMP:

http://www.mamp.info/en/index.php
http://doc.ubuntu-fr.org/lamp

Premier programme

Dans, votre serveur web, vos programmes PHP prendrons place dans le dosser 'www', pour ce faire, vous pouvez lancer WAMP, puis cliquer sur l'icône WAM barre des tâches et sélectionner 'Répertoire www'.



Vous pourrez ensuite créer un nouveau dossier et le nommer comme vous le souhaitez. La prochaine étape sera de déclarer ce dossier en tant que « virtual l le déclarer en tant que virtual host:

1. Cliquer sur l'icône WAMP dans la barre des tâches

2. Sélectionner 'Localhost', cela ouvrira vôtre navigateur par défaut à l'adresse 'http://localhost'

3. Vous pourrez voir une page d'accueil de vôtre serveur web, il faudra alors cliquer 'ajouter un Virtual Host' dans la section « Outils » et suivre les instructio

Une fois toutes ces étapes remplies vous pouvez créer un fichier avec l'extension .php, et y accéder dans vôtre navigateur de la manière suivante: "http:localhost/mon_repertoire_virtual_host/mon_fichier.php"

Nous pouvons créer un premier programme pour tester que tout fonctionne bien. Pour cela créer un fichier nommé 'index,php' dans votre tout nouveau Vir ajoutez y le code suivant:

```php
<?php
    echo "hello world";
?>
```

Une fois le fichier sauvegardé, vous constatez que lorsque vous visitez l'adresse "localhost/mon_repertoire_virtual_host/mon_fichier/index.php", Votre navig affiche une page blanche avec le mot « Bonjour! »
Félicitation, vous venez de créer vôtre tout premier programme PHP !

Structure du langage

```php
// La chaine de caractère aussi appelée "string".
$prenom = "Bob";

// Le nombre entier naturel ou relatif, appelé "Integer".
$age = 24;
```

```php
// Le nombre flottant naturel ou relatif, appelé "Float".
$poids = 72.5;

// Le Booléen ou "Boolean" (ne peut prendre que les valeurs true ou false).
$est_un_adulte = true;
$est_une_femme = false;

// Le type "null" est un type spécial qui caractérise une variable sans valeur.
$emploi = null;

// Le Tableau, aussi appelé "Array" permet de stocker une liste de valeurs.
// Il s'écrit entre crochets et chaque élément est séparé par une virgule.
$liste_de_courses = ["Tomates", "Oeufs", "Champignons"];
// on peut ainsi accéder à une valeur via son index ex: $liste_course[0]
// renvoie ''Tomates''.

// Le Tableau associatif, permet d'associer une clé à une valeur
$liste_de_courses = ["Tomates" => 2, "Oeufs" => 6, ''Champignons'' => 20];
// on peut ainsi accéder à la valeur via la clé
// ex: $liste_course[''Tomates''] renvoie '2'.
```

Conditions

En PHP, il est possible d'exécuter du code seulement dans une ou plusieurs conditions précises, définies arbitrairement. Pour coder ces conditions, plusieurs existent:

--> La structure « if … elseif … else »
--> Condition ternaire
--> La structure « switch case »

Structure « if … elseif … else »

```php
$age = 24;
if ($age > 18) // Après le mot clé « if », dans les parenthèses,
// nous écrivons la condition à respecter.
{
// Si la condition est respectée, le corps de la condition s'exécutera.
// Dans notre cas, le programme affichera « vous êtes majeur !»
echo "Vous êtes majeur!";
}
```

```php
$age = 24;
if ($age < 0)
{
echo "Vous ne pouvez pas avoir un âge négatif!";
}
elseif ($age > 18)
{
echo "Vous êtes majeur!";
// Dans notre cas, le programme affichera uniquement « vous êtes majeur !»
}
else
{
echo "Vous êtes mineur!";
}
```

Condition ternaire

La condition ternaire est une façon plus concise d'écrire la structure «if … else» vue précédemment, elle s'utilise avec l'opérateur « ? », de la manière suivant

```
$age = 24;
$majeur = $age >= 18 ? True : False; //  ?  :  ;
```

Dans ce cas si la variable « majeur » vaut « True » car la variable « age » est superieur à 18, rendant ainsi la condition vraie.

La structure « switch … case »

```
$departement = 75;
switch ($departement) {
    case 75:
        echo "Paris";
        break;
    case 93:
        echo "Seine Saint Denis";
        break;
    case 92:
        echo "Haut de seine";
        break;
    default:
        echo "Autre département";
        break;
}
```

Répétitions ou boucles

Les structures de répétitions ou boucles, permettent d'exécuter une instruction ou un groupe d'instruction de manière répétée. Il existe plusieurs types de l
PHP:

--> Les boucles « for »
--> Les boucles « while » et « do … while »
--> Les boucles « foreach »

Les boucles répètent une instructions ou un groupe d'instructions tant qu'une condition est vraie, cette condition est définie par le développeur en suivant l
principes vus précédemment dans de la partie sur les structures conditionnelles.

Boucles « For »

```
for ($expression1 ; $expression2 ; $expression3)
{
// Mes instructions à répéter
}
```

--> $expression1 est l'initialisation d'une ou plusieurs variables servant de compteur pour la boucle.
--> $expression2 est ensuite évaluée avec une valeur booléenne : si elle vaut True, la boucle continue, sinon la boucle s'arrête.
--> $expression3 n'est exécutée qu'à la fin de chaque itération. Il s'agit le plus souvent d'une instruction d'incrémentation de la variable compteur ($express

Exemple

```php
for ($compteur = 0; $compteur < 10; $compteur = $compteur + 1)
{
echo "Bonjour";
}
// le code affichera "Bonjour" dix fois
```

boucles « while » et « do … while »

```php
while ($condition) {
// Mes instructions à répéter tant que la condition est vraie
}
```

$condition est toujours évaluée avec une valeur booléenne : si elle vaut True, la boucle continue et les instructions comprises entre accolades sont exécutée boucle s'arrête. Si elle est toujours vraie on obtient une boucle infinie (comme pour la boucle « for »).

Exemple

```php
$n = 1;
while ($n % 7 != 0)
{
$n = rand 1 100);
// cette instruction génère un nombre aléatoire entre 1 et 100,
// via la fonction «rand»
```

```
echo "$n / ";
}

// Vous obtenez, par exemple, la suite de nombres
// 72 / 79 / 50 / 95 / 11 / 43 / 18 / 49
// (cette suite varie évidemment à chaque tirage).
```

```
do {
// Mes instructions à répéter tant que la condition est vraie
} while ($condition);
```

Ici, les instructions sont exécutées avant l'évaluation de $condition qui est toujours évalué avec une valeur booléenne. Les instructions sont donc exécutées une fois.

Exemple

```
do {
$n = rand 1 100);
// cette instruction génère un nombre aléatoire
// entre 1 et 100, via la fonction « rand »
echo "$n / ";
} while ($n % 7 != 0);
// on obtiens le même type de résultat que l'exemple précédent, avec la boucle «while».
// À noter que l'on a pas eu besoin d'initialiser la variable «n»
// dans cette exemple. En effet dans l'exemple précédent c'était nécéssaire
// pour entrer dans la boucle, ce n'est pas nécéssaire dans une boucle de type «do…while»
```

Boucles «Foreach»

La boucle "foreach" permet de parcourir rapidement l'ensemble des éléments d'un tableau, ce que peut aussi faire la boucle "for" mais "foreach" se révèle efficace.

```php
$tableau = ["valeur1" => 10, "valeur2"=>20, "valeur3"=> 30];

foreach ($tableau as $cle=>$valeur){
echo "$cle $valeur/";
}
```

Concaténation

Il est possible de concaténer plusieurs chaînes de caractères en les séparant par un point ou en écrivant directement dans des guillemets doubles:

```php
$salut = "Bonjour";
$prenom = "Nathan";

echo $salut." ".$prenom;
echo "$salut $prenom";
```

Les fonctions

Une fonction est un bloc d'instructions réutilisable à volonté, une fois défini. Ce bloc d'instructions peut prendre des paramètres en entrée ounon et peut re valeur en sortie ou non.

```
function nomdemafonction($param1,$param2,...){
//instruction à exécuter
return $unevaleur;
}

nomdemafonction($param1,$param2,...);
```

Fin PHP

Blague de fin de séance:

```
!false
// c'est drôle parce que c'est vrai
```