

Cours De Programmation web 2

PHP mysql

Nom de l'enseignant : Cisse Adianga Boubacar
Ingénieur informaticien

contact : adiangab@gmail.com
78 18 18 44


Classe : IG

Horaire : 12h 14H

PLAN DU COURS

▶ Voir le syllabus

Chapitre 1

- 
- 1.Qu'est-ce que PHP
 - 2.Structure de base d'une page PHP
 - 3.Configuration de PHP
 - 4.Utilisation de PHP en ligne de commande
 - 5.Les bases du langage PHP

Définition :

PHP : Php hypertext processor

Le php s'exécute coté serveur et permet de dynamiser un serveur

Il existe deux types de site web : nous avons le site web dynamique et le site web statique

➤ Site web statique: le site web statique doit être actualisé par quelqu'un d'autre

il est réalisé à partir de HTML et css

Le site web dynamique : le site web dynamique change tout seul en fonction de son

interaction avec le visiteur ou tous autres paramètres (heure , saison etc..)

Chaque fois que sur un site il y a une interaction entre le visiteur et le site web c'est un site

web dynamique et il est réalisé à partir de PHP PAR EXEMPLE forum de discussion ou fiche

d'inscription etc..

- 
- Il est indispensable de connaître la notion client serveur

Client : ordinateur qui demande une page web

- Serveur : mega-ordinateur qui ne s'arrête jamais et qui va envoyer la page web demandé par le client
- Pour un site web dynamique le serveur va générer une page différent pour chaque client
- indisponible d'accéder au serveur ?

Configuration de PHP




Pour faire notre travail nous allons travailler en mode local ce qui veut dire, transformer notre ordinateur en serveur.

Pour cela , il suffit d'installer wamp pour windows



Mon premier code php et les base du langage php



```
<!DOCTYPE html>
<html>
  <head>
    <title>  ma premiere page </title>
    <meta charset="utf-8/">

  </head>
  <body>
    <p>  Accueil </p>
    <?php
    echo " welcome dans la classe";
    ?>
    <?php
    echo " papy";
    ?>
  </body>
</html>
```

➤ Notre code PHP sera écrit à l'intérieur du code html et n'importe


Ou dans le code html

➤ pour coder en PHP, il faudra écrire la structure minimal du html

➤ faire la balise ouvrante comme celle-ci `<?php` Et saisir son code php et fermer la balise ainsi `?>`

➤ Comme en langage c++ chaque instruction est suivie de ;

➤ Les commentaires en php, comme en c++, nous avons les commentaires monolignes et les commentaires multilignes: `//` ou `/* */`



```
<html><body>

<?php

echo "Hello world!";

?>

</body></html>
```

« echo » sert à produire un affichage

Pour que « echo » affiche le texte tel quel, ce texte doit être entouré de guillemets « " ». Sinon « echo » essaie de comprendre. Par exemple « echo 1+2; » provoque l’affichage de « 3 ». Tandis que « echo "1+2"; » provoque l’affichage de « 1+2 ».

Rappels (?) d'algorithmique

1. La structure de base d'un programme est la séquence :
les instructions sont exécutées les unes après les autres, dans l'ordre où elles sont écrites.

Les instructions sont toujours terminées par un point-virgule (« ; »). Ce n'est pas un séparateur d'instructions.

2. Il existe des structures qui permettent de rompre la séquence. Vous trouverez ci-dessous les principales.

Il y en a d'autres...

3. L'alternative: un test est exécuté ; selon que son résultat soit OUI ou NON, une partie du programme ou une autre est exécutée.

Sa syntaxe en PHP est :

```
if (test) {  
instructions à exécuter quand la réponse est OUI  
}  
else {  
instructions à exécuter quand la réponse est NON  
}
```

4. L'itération (TANT QUE) : une partie du programme est exécutée

tant que le résultat d'un test est OUI

(le calcul est fait AVANT chaque entrée dans la boucle).

Sa syntaxe en PHP est :

```
while (test) {
```

```
instructions à exécuter tant que la réponse est OUI
```

```
}
```


5. La boucle POUR : ce n'est qu'un cas particulier de la boucle TANT QUE.

Sa forme générale est :

```
for (avant d'entrer ; test ; action de fin de boucle) {  
instructions à exécuter quand la réponse au test est  
OUI  
}
```

où « avant d'entrer » et « action de fin de boucle » sont des actions quelconques.

Cette forme est totalement équivalente à celle-ci :

avant d'entrer ;

while (test) {

instructions à exécuter quand la réponse au test est OUI ;

action de fin de boucle ;

}

Mais on l'utilisera le plus souvent sous cette forme :

```
for (initialiser compteur; test compteur; incrémenter compteur) {  
instructions à exécuter quand la réponse au test est OUI  
}
```

comme dans l'exemple suivant qui provoque l'affichage : «

357911131517 » :

```
for ($i=3 ; $i<19 ; $i=$i+2) {  
echo $i;  
}
```



Les constantes, les variables et les types de données

Variables et expressions

1. Les noms des variables sont préfixés par \$. Il n'est pas nécessaire de déclarer les variables avant de les utiliser. Le contrôle des types est plutôt souple...

2. L'affectation est notée « = ».

3. Les opérateurs classiques sur les nombres sont disponibles :

« + » (addition), « - » (soustraction), « * »

(multiplication), « / » (division).

4. Le seul opérateur sur les chaînes est la concaténation,

symbolisée par un point (« . »).



5. Les opérateurs de comparaison sont eux aussi (presque tous)

classiques : « < » (inférieur), « <= »

(inférieur ou égal), « > » (supérieur), « >= » (supérieur ou égal), « == »

(égal), « != » (différent).

6. Les opérateurs logiques de base sont « || » (ou) et « && » (et).

Travaux pratiques

1. Écrire un programme PHP qui affiche tous les nombres impairs entre 0 et 15000, par ordre croissant :

« 1 3 5 7 ... 14995 14997 14999 ». Exemple : impair.php

2. Écrire un programme qui écrit 500 fois « Je dois faire des sauvegardes régulières de mes fichiers. »

Exemple : punition.php

3. Écrire un programme qui affiche la table de multiplication

par 13. Exemple : `multiplication.php` ou mieux :

`multiplicationbis.php`

4. Écrire un programme qui calcule $30!$.

Exemple : `factorielle.php`

5. Écrire un programme qui affiche :

12345678910111213

12345678910111213

12345678910111213

12345678910111213

Exemple : 4lignes.php13

6. Écrire un programme qui affiche la table de multiplication totale de $\{1, \dots, 12\}$ par $\{1, \dots, 12\}$. Exemple : multiplicationtotale.php

Écrire une fonction qui renvoie $n!$. Puis utilisez cette fonction pour construire une table des factorielles.

Exemple : tablefact.php15.

7. Partie1 Écrire une fonction qui affiche un triangle et qui admet comme paramètre le nombre de lignes du triangle :

*

**

partie2 Utilisez cette fonction pour dessiner un demi-sapin (de 2 en 2) :

*

**

*

**

*

**

Exemple : demisapin.php16