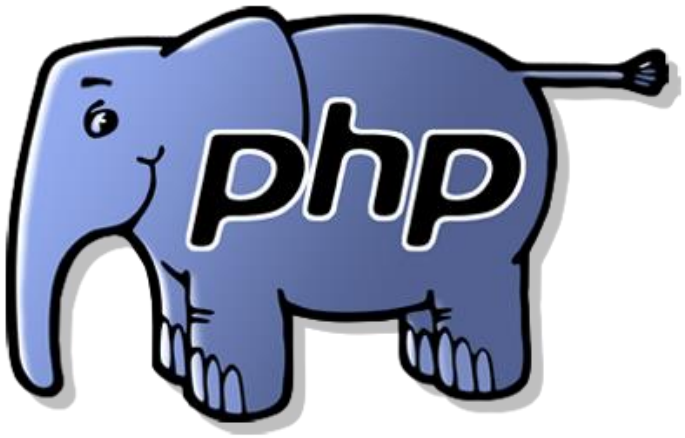

LES BASES DU PHP

Table des matières

I. Structures et principes 1

II. Variables, types, opérateurs, commentaires et caractères spéciaux 1

1. Variables et types 1
2. Opérateurs 1
3. Les commentaires 2
4. Les délimitateurs et caractères spéciaux 2

III. Les structures de programmation 3

1. Les structures conditionnellles 3
2. Les structures itératives 3

IV. La transmission de données 3

1. La méthode POST 3
2. La méthode GET (à réserver à la phase de mise au point) 4

V. Connexion et manipulation d'une base de données 4

1. La connexion à la base MySQL avec mysqli 4
2. La requête 4
3. Récupération et traitements des résultats 4
4. Fermeture et libération des ressources 4

Merci à Manuel ALVES et Frédéric BAURAND

I. Structures et principes

Un script est un ensemble de commandes destinées au serveur web qui les interprète et renvoie le résultat sous forme de page html au navigateur. Un script Php est interprété par le serveur web. Les utilisateurs ne voient pas le code PHP en consultant le code source. Les fichiers Php sont stockés sur le serveur.
Tout fichier contenant des scripts Php doit avoir l'extension .php.
Le code Php doit être délimité par les balises <?php et ?>. Chaque instruction se termine par un point-virgule (;).

```
<html>
        <head>
                <Title>Bonjour</Title>
        </head>
        <body>
                <?php
                  echo "<h1>Mon premier script</h1>";
                ?>
        </body>
</html>
```

II. Variables, types, opérateurs, commentaires et caractères spéciaux

1. Variables et types

Les identificateurs de variable sont précédés du symbole « $ » (dollars).

Le typage des variables est implicite en php. Il n'est donc pas nécessaire de déclarer leur type au préalable ni même de les initialiser avant leur utilisation. Les variables peuvent être de type entier (***integer***), réel (***double***), chaîne de caractères (***string***), tableau (***array***), objet (***object***), booléen (***boolean***).

Exemple :

```
$str = "12";            // $str vaut la chaîne "12"
$nbr = $str;            // $nbr vaut également la chaîne "12"
```

Quelques fonctions utiles avec les variables :

empty($var) : renvoie vrai si la variable est vide
isset($var) : renvoie vrai si la variable existe
unset($var) : détruit une variable
gettype($var) : retourne le type de la variable

2. Opérateurs

- **Opérateurs sur les chaînes de caractères**

L'opérateur de concaténation est le point : **.**

```
$a = "Hello ";
$b = $a . "World!"; // Maintenant $b = "Hello World!"
```

Opérateur d'affectation

L'opérateur d'affectation est le signe égal : **=**

```
$a = 3;
$a = $a+5; // affecte la valeur 8 à la variable $a.
$b = "Hello ";
$b = $b."There!"; // affecte la valeur "Hello There!" à la variable $b
```

Opérateurs arithmétiques

Opération	Opérateur	Exemple
Addition	+	$Nb1 + $Nb2
Soustraction	-	$Nb1 - $Nb2
Multiplication	*	$Nb1 * $Nb2
Division	/	$Nb1 / $Nb2
Modulo	%	$Nb1 % $Nb2

Opérateurs logiques

Opération	Opérateur	Exemple
ET	and	$a and $b
OU	or	$a or $b
OU Exclusif	xor	$a xor $b
NON	!	! $a

Opérateurs de comparaison

Opération	Opérateur	Exemple
Egal à	==	$a == $b
Différent de	!=	$a != $b
Inférieur à	<	$a < $b
Supérieur à	>	$a > $b
Inférieur ou égal à	<=	$a <= $b
Supérieur ou égal à	>=	$a >= $b

3. Les commentaires

```
<?php
// commentaire sur une ligne

/* commentaire
sur plusieurs
lignes */
?>
```

4. Les délimitateurs et caractères spéciaux

Il existe 2 types de délimiteurs : Les doubles et simple quotes.

- **Si la chaîne est délimitée par des double-quotes ("), les variables à l'intérieur de la chaîne seront remplacées.**
- **Dans la chaîne limitée par des simple-quotes (’), les variables ne seront pas substituées.**

```
$texte="Bonjour" ;
echo "$texte" ;   //Affiche "Bonjour"
echo ‘$texte’ ;    //Affiche "$texte"
```

III. Les structures de programmation

1. Les structures conditionnellles

```
if (conditions pour rentrer dans le IF)   {
        //Instructions
}
else   {
        //Instructions
}
```

```
if ($Nb ==2) {
        $Nb = $NB + 1;
}
else {
        $Nb = $Nb - 1 ;
}
```

Remarques : on peut imbriquer les If-Else entre eux (programmation plus propre), et on peut également avoir des If sans Else.

2. Les structures itératives

- **La boucle for**

```
for (Valeur d'entrée pour la varaible dans la boucle ; Condition pour continuer la boucle ; Incrémentation)   {
        //Instructions
}
```

```
for ($Nb=1; $Nb <= 10; $Nb++) {
        echo $Nb;
}
```

- **La boucle do while**

Le Do-While permet de faire une boucle lorsque l'on est sur de rentrer au moins une fois dans la boucle.

```
do   {
        //Instructions
}   while (Contitions pour continuer la boucle);
```

```
do {
        echo $Nb;    $Nb = $Nb + 1;
} while ($Nb <= 10);
```

- **La boucle while**

A l'inverse, le While permet de faire une boucle où l'on n'est pas forcé de rentrer au moins une fois.

```
while (Contitions pour entrer dans la boucle)  {
        //Instructions
}
```

```
while ($Nb <= 10) {
        echo $Nb;    $Nb = $Nb + 1;
}
```

Attention : Il faut bien veiller à faire évoluer les valeurs des variables utilisées dans les conditions, sinon vous aller faire une boucle infinie !

IV. La transmission de données

1. La méthode POST

La méthode POST s'utilise avec les formulaires HTML. Le formulaire va envoyer ses données vers une page PHP.
Balise : <Form Action="page.php" Method="Post">

Les champs du formulaire sont normalement nommés avec la balise Input : <Input Type="Text" Name= "LeNomDuChamps">

En php, dans la page pointée par l'attribut Action de la balise <Form>, il suffit de faire **$_POST["LeNomDuChamps"]** pour accéder aux informations saisies dans le formulaire précédent par l'utilisateur.

```
<form action="Affiche.php" method="post">
        Votre nom : <input type="text" name="nom" />
        <input type="submit" value="valider" />
</form>
```

Fichier Affiche.php

```
<?php
        echo "Votre nom est : " . $_POST['nom'];
?>
```

2. La méthode GET (à réserver à la phase de mise au point)

Il est également possible de passer des paramètres dans l'adresse Internet.
Dans la page PHP, pour accéder aux valeurs de ces variables, il suffit de faire : **$_GET["NomDuParamètre"]**

Attention : Les valeurs de ces variables sont directement affichées dans la barre de liens, veillez à ne pas y mettre des informations confidentielles (Ex : Un login et son password).

V. Connexion et manipulation d'une base de données

1. La connexion à la base MySQL avec mysqli

```
$host = "Adresse IP du serveur MySQL";
$user = "Nom de l'utilisateur";                 //   root par exemple
$passwd  = "Mot de passe";
$bdd =  "Nom de la base de données";

$mysqli = mysqli_init();
if (!$mysqli)
{
   die('mysqli_init ne fonctionne pas');
}

if (!$mysqli->real_connect($host, $user, $passwd, $bdd))
{
   die('Connect Error (' . mysqli_connect_errno() . ')'. mysqli_connect_error());
}
```

2. La requête

```
$requete= "SELECT NumCli, NomCli, PrenomCli FROM CLIENT";          // Création de la requête
$resultat = $mysqli->query($requete);                              // Envoi de la requête
```

3. Récupération et traitements des résultats

```
while($row = $resultat->fetch_array())
{
            echo "Numero du client : ".$row["Numcli"]."<br/>";
            echo "Nom du client : ".$row["Nomcli"]."<br/>";
            echo "Prenom du client : ".$row["Prenomcli"]."<br/>";
}
```

4. Fermeture et libération des ressources

```
$resultat->free();                  /* Libération des résultats dans l'espace mémoire */

$mysqli->close();                   // Deconnexion de la base de données
```


php