

Les bases de PHP - débiter en PHP

Date de dernière mise à jour : 27/06/2007 à 19:36

Source : <http://www.vulgarisation-informatique.com/bases-php.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)

Les bases de PHP

Après avoir vu comment installer un serveur qui va vous permettre d'interpréter PHP sur votre ordinateur, nous allons maintenant voir avec quels outils on peut coder en PHP, et comment on dit à telle page "il y a du code PHP dedans, interprète-le".

Ce qu'il faut savoir avant tout, c'est que les fichiers PHP ne sont que des fichiers textes. Ils ne sont donc pas compilés (en binaire). Lorsque vous demandez une page, PHP lit le code et l'interprète au fur et à mesure. On parle de code interprété. Comme ce sont des fichiers textes, vous allez pouvoir utiliser n'importe quel éditeur de texte pour coder en PHP. Personnellement j'utilise le bloc-notes intégré à Windows, mais il existe des centaines d'éditeurs de ce genre. A vous d'utiliser celui qui vous convient le mieux :)



Les tags (ou balises) PHP

Lorsque vous enregistrez votre fichier texte avec l'extension .php, vous pouvez avoir plusieurs codes dedans : du (X)HTML, du javascript, du CSS ... mais aussi du PHP. Comment l'analyseur syntaxique de PHP fait-il la différence ? il sait en fait que le code est délimité par des balises appelées **tags**. Ces tags sont un peu spéciaux, ils se présentent sous cette forme :

```
<?php
//code PHP ici//encore du code ci nécessaire, sur plusieurs lignes
?>
```

Le code PHP est donc situé entre les tags **<?php** et **?>**. Avant de vous parler du reste, je fais le point sur l'autre syntaxe existante et que vous rencontrerez parfois si vous téléchargez certaines sources mal codées. Il s'agit des tags **<? et ?>**. N'utilisez pas ces tags qui pourront ne pas être compatibles avec toutes les configurations de serveurs existantes. Pour faire simple, PHP dispose d'un réglage appelé **short_open_tags**, celui-ci peut prendre deux valeurs (On ou Off). Si il est sur On, cette syntaxe sera acceptée et comprise par l'analyseur de code, dans le cas contraire vos scripts ne pourront être interprétés, tandis que si vous utilisez tout le temps **<?php** et **?>**, quelle que soit la valeur de cette option, vos scripts fonctionneront parfaitement.

Quand utiliser PHP ?

Vous allez sûrement en apprenant PHP vouloir convertir tout votre site en PHP et donc remplacer le code (X)HTML par des echo() en rafale. Ce n'est pas une bonne solution, on ne doit pas utiliser PHP pour générer du contenu statique. Si votre page n'est pas destinée à changer dans le temps dynamiquement, laissez-la en HTML. Vous gagnerez en performances et soulagerez votre serveur. Vous pourrez également lorsque vous aurez beaucoup de fichiers dans votre répertoire savoir quels sont ceux qui sont statiques des autres.

Afficher du texte

Opération de base en PHP, l'affichage du texte peut se faire de différentes façons. Nous allons voir la façon la plus simple et la plus rapide (au niveau

du temps de traitement serveur) d'afficher du texte. Il s'agit de la "fonction" **echo()** (il s'agit d'une structure du langage en réalité et non d'une fonction, en effet il n'est pas obligatoire de l'utiliser avec des parenthèses).

Voici l'exemple d'affichage de texte utilisant echo :

```
<?php
echo
'bonjour !'
;
//affiche le texte "bonjour !"
?>
```

Tout ça c'est bien beau mais si vous souhaitez afficher une apostrophe comment fait-on ? Et bien il suffit d'ajouter le caractère \ (antislash) devant l'apostrophe à ajouter. Un petit exemple vaut mieux qu'un long discours :

```
<?php
echo
'\apostrophe au milieu d'une chaîne'
;
?>
```

Vous auriez pu écrire également ça comme ceci (avec des guillemets doubles) :

```
<?php
echo
"l'apostrophe au milieu d'une chaîne"
;
?>
```

Le premier exemple étant plus rapide à traiter par PHP (on reparlera de la rapidité d'exécution dans la section traitant de l'optimisation de PHP).

Chaque instruction se termine par un point virgule en PHP, ne l'oubliez pas car vous obtiendrez des erreurs parfois insolubles si votre code devient long.

Les commentaires :

Les commentaires sont très utiles pour donner des informations utiles sur ce que va faire une portion de code. Ils ne seront pas affichés ni analysés par PHP. Vous pouvez mettre ce que vous voulez dedans (même du code PHP, (X)HTML, etc...). Il existe deux sortes de commentaires que nous allons voir maintenant.

-Les commentaires sur une ligne : ils permettent de commenter sur une seule ligne, on les place généralement à droite ou au dessus de (ou des) ligne(s) concernée(s) :<?php

```
echo
'bonjour !'
;
//affiche le texte "bonjour !"
?>
```

Comme vous le constatez, le commentaire qui ne sera pas analysé se situe après les //. Ceci marque le début du commentaire.

-Les commentaires sur plusieurs lignes : ils peuvent occuper une ou plusieurs lignes et sont délimités par les tags /*- et */-. Voici comment on les utilise :<?php

```
/* Nous allons maintenant afficher du texte On peut commenter sur plusieurs lignes et mettre du code <?php echo 'test'; ?>
dans les commentaires */
```

echo

'\l'apostrophe au milieu d'une chaîne'

;

?>

Chapitre suivant



Les variables

Définition et présentation

Types de variables

Les opérateurs de calcul

Les variables de formulaires et adresses Web

La concaténation

Source : <http://www.vulgarisation-informatique.com/bases-php.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)