

PHP – COURS 01 - Introduction à la programmation

INTRODUCTION

PHP (Hypertext Preprocessor) est avant tout un langage dédié à la génération de contenu pour le Web. Le client (un navigateur par exemple) demande l'exécution d'une "page PHP" (en fournissant d'éventuels paramètres) à un serveur, ce dernier sous-traite la demande à un processeur PHP qui interprète le code et retourne un résultat au serveur qui le renvoie au client.

Mais dans ce cours, nous l'utiliserons de manière à faire des programmes autonomes, dans le but d'avoir une initiation aux principes de base de la programmation. Les programmes seront exécutés depuis une console sous Linux ou invite de commande sous WINDOWS.

OUTILS NÉCESSAIRES

Pour écrire et exécuter un programme, vous aurez besoin d'un éditeur de texte, sous Windows vous pouvez utiliser **NotePad++**, un éditeur de texte disponible gratuitement. Vous avez également besoin d'installer l'interpréteur PHP, qui servira à exécuter vos programmes.

UN EXEMPLE DE PROGRAMME

bonjour.php

```
<?php
    echo ("Bonjour tout le monde"); // affiche Bonjour tout le monde
?>
```

Pour exécuter un script PHP, il faut ouvrir une console (terminal) puis déclencher l'interpréteur PHP en lui passant en paramètre le nom du script. Exemple: php bonjour.php

```
stephane@tristan:~$ php bonjour.php
Bonjour tout le monde stephane@tristan:~$
```

bonjour2.php

```
<?php
    echo ("Bonjour tout le monde /n");
    // affiche Bonjour tout le monde avec un retour à la ligne
?>
```

```
stephane@tristan:~$ php bonjour2.php
Bonjour tout le monde
stephane@tristan:~$
```

Si votre programme ne s'exécute pas vous devez :

- vérifier que l'interpréteur PHP est installé : php -version
- vérifier que le programme se situe bien à l'endroit où vous souhaitez l'exécuter
- vérifier la syntaxe du programme

QUELQUES RÈGLES DE BASE

1- Un script PHP doit être placé à l'intérieur d'une paire de balises, il existe plusieurs styles de balises, celles que nous utiliserons s'écrivent ainsi :

```
<?php  
instructions PHP;  
?>
```

2 – Les instructions PHP doivent se terminer par un point virgule.

3 - Pour afficher une information, vous pouvez utiliser 2 instructions: print ou echo.

Syntaxe :

```
print (expression );  
echo (expression );
```

4 – Un fichier contenant des instructions PHP doit avoir l'extension « .php »

LES VARIABLES

Le rôle d'une variable est de stocker une information temporairement dans la mémoire vive le temps de l'exécution d'un programme. Les noms des variables doivent obligatoirement commencer par le caractère dollar (\$). Ils sont sensibles à la case : \$prix et \$Prix ne désignent pas la même variable. Par convention le nom d'une variable commence par une minuscule et une majuscule au début de chaque nouveau terme.

exemple :

- \$prix=100
- \$nombreArticle=5

RÉCUPÉRER UNE INFORMATION SAISIE AU CLAVIER

PHP n'étant pas conçu pour des programmes autonomes, les données sont normalement saisies dans des formulaires WEB. Cependant, il existe une fonction appelée readline() qui permet à un programme PHP de récupérer les données saisies au clavier, malheureusement celle-ci n'existe pas sous Windows.

Pour interagir avec l'utilisateur, voici le code PHP :

```
<?php  
echo 'enter votre reponse ' ;  
$reponse = trim(fgets(STDIN));  
?>
```

Lorsque PHP atteindra cette ligne, le script se "bloquera". L'utilisateur pourra alors entrer du texte dans la console, et pourra l'envoyer dans \$reponse à l'aide de la touche **Entrée**. le script se "débloquera" alors.

Pour un plus grand confort de travail, nous définirons nous même une fonction lire() dans un fichier nommé lireClavier.php.

```
<?php
function lire( )
{ return trim(fgets(STDIN)); }
?>
```

Ainsi à chaque fois que nous aurons besoin d'utiliser cette fonction, il faudra l'inclure dans les programmes.

Exemple :

```
< ?php
include("lireClavier.php"); // on indique au script d'aller chercher le fichier
echo("Entrez votre prénom : \n") ;
$prenom= lire( ) ; // on utilise la fonction lire du fichier lireClavier.php
echo ("Bonjour ".$prenom." ! \n") ;
?>
```

INITIALISER OU METTRE À JOUR UNE VARIABLE

l'instruction :

```
$prix = 10 ;
```

se lit de **droite à gauche** : la valeur 10 est affectée à la variable \$prix.

Modifions légèrement le programme :

```
<?php
$prix=10;
$cout=$prix;
echo($cout);
echo("\n");
?>
```

La deuxième instruction se lit donc : la valeur de la variable \$prix est affectée à la variable \$cout. La variable \$prix est de type numérique entier. D'autres types de données sont disponibles en PHP. Le type d'une variable est déterminé automatiquement par PHP lors de l'affectation.

Attention comme beaucoup de langages de programmation, PHP note l'affectation par le symbole traditionnellement réservé à l'égalité (=). A ne pas confondre avec le symbole de test d'égalité (==).

MANIPULER DES CHAÎNES DE CARACTÈRES

Exemple :

```
<?php
$prenom ="tristan" ;
$bonjour = "bonjour ".$prenom ;
echo ( $bonjour. " \n" ) ;
?>
```

L'opération qui consiste à construire une chaîne de caractère (on affecte à la variable \$bonjour la chaîne de caractère bonjour + le contenu de la variable \$prenom) s'appelle la **concaténation**.

On remarquera que PHP utilise le point (.) et non le (+) comme opérateur de **concaténation** : cette opération permet de regrouper plusieurs valeurs en une seule.

CALCULER - COMPARER

Opérateurs mathématiques

\$a+\$b	addition
\$a-\$b	soustraction
\$a/\$b	division
\$a*\$b	multiplication
\$a%\$b	modulo ->reste entier de la division

Opérateurs de comparaison

\$a>\$b	supérieur
\$a>=\$b	supérieur ou égal
\$a<\$b	inférieur
\$a<=\$b	inférieur ou égal
\$a==\$b	égal
\$a!=\$b	différent

LISTE DES CARACTÈRES SPÉCIAUX

L'antislash (\) est un caractère spécial, on peut l'utiliser dans les cas suivants :

Un saut de ligne	\n
Une tabulation horizontale	\t
Un antislash \	\\
Un caractère \$	\\$

REMARQUE

Lorsque l'on écrit des programmes les erreurs sont fréquentes (erreur de syntaxe, erreur sur le nom ou l'emplacement du programme, oubli du « ; » à la fin d'une instruction , etc...). Pour des raisons de sécurité, lorsque l'interpréteur PHP est configuré en mode production il n'affiche pas les erreurs et interrompt l'exécution du script.

Il est possible de modifier ce comportement dans les scripts PHP, en indiquant au début de chaque programme :

```
<?php
error_reporting(E_ALL);
ini_set('display_errors','On');
```