LES BASES DE PHP

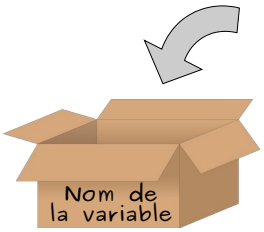
3 – Les variables: Introduction et chaîne de caractères

Table des matières

Introduction

Qu'est ce qu'une variable ?

Dès l'instant où je vais vouloir manipuler une donnée, c'est à dire afficher un contenu dynamique comme le nom de quelqu'un, le panier d'un site marchand ou ce que vous voulez, je vais devoir utiliser des variables.



Donnée :
- Chaîne de caractères
- Entiers (1, 2, 3 …)
- Décimaux (8,551)
- Valeur booléenne (vrai/faux)
- Tableau (clé,valeur)
- Objet

Une variable est comme un carton de déménagement. On écrit ce qu'il y a dessus, puis ensuite on le rempli. Un variable c'est exactement la même chose. Nous devons lui donner un nom, puis ensuite mettre quelque chose dedans … au harsard … une donnée.

Pour créer une variable, il vous suffit de mettre le caractère $ devant votre nom de variable suivi du signe = puis ce que vous voulez y mettre dedans.

```
<?php
$nom = "Denis" ;
?>
```

Il est possible d'utiliser des nom plus compliqué pour vos noms de variable comme par exemple $nomPersonne8, mais vous ne pouvez pas utiliser un nombre comme premier caractère pour votre variable comme $8nomPersonne. PHP interdit cet usage.

Notez que le contenu de la variable vient après le signe = . Le caractère = est reconnu par PHP comme un opérateur d'affectation. PHP prend en charge le mécanisme de réservation et d'affection de la mémoire (la RAM) afin stocker vos données quelqu'en soit le type.

Retenez qu'une variable n'est pas enregistrée sur le disque dur, mais bien en RAM. De ce fait, une variable à une durée de vie limitée ; Cela implique que pour conserver une donnée, il faudra passer par un autre mécanisme comme un enregistrement dans une base de données par exemple.

Travaux Pratiques (exo1.php)

Recopiez et **testez** l'ensemble des exemples proposés depuis l'interpréteur PHP :

- Pour ceux qui utilisent LINUX utilisez `nano` comme éditeur de texte :
- Pour ceux qui utilisent Windows, utilisez VisualCode pour créer et modifier votre fichiers exo1.php.
- Vérifiez l'emplacement du votre répertoire où sont enregistré vos fichiers php.

Pour cet exercice, allez créer un fichier exo1.php dans un répertoire dédié à ce TP. Dans ce fichier, vous recopirez les exemples un par un afin de tester les différents codes proposés. Pour tester votrce code, procédez comme ceci :

1. Modifiez votre fichier PHP
2. Enregistrez vos modifications
3. Retournez dans l'invite de commande Windows ou dans un terminal si vous utilisez Linux
4. Tapez ceci :
```
php exo1.php
```
5. Votre code s'exécutera si votre répertoire de travail est bien positionné.

Afficher une variable

Pour afficher le contenu d'une variable, nous allons utiliser l'instruction echo.

```
<?php
$nom = "Denis" ;
echo $nom ;
?>
```

C'est pas très jolie, mais ça marche. Arrangeons un peu ce code.

```
<?php
$nom = "Denis" ;
print "Nom : ",$nom ;
?>
```

Comme vu précédent, l'instruction echo permet de mentionner plusieurs paramètres en les séparant d'une virgule.

Si l'on veut réaliser une fiche d'information d'un client par exemple, on peut ainsi rajouter le prénom, l'adresse, le téléphone et l'émail par exemple.

Commençons par ajouter le prénom.

```
<?php
$nom = "Denis" ;
echo "Nom : ",$nom ;
$prenom = "Roger" ;
echo "Prénom: ",$prenom;
?>
```

Travaux Pratiques (exo2.php)

Finalisez le code PHP présentez précédemment pour obtenir :
Nom : **nom1**
Prénom : **prenom1**
Adresse : **adresse1**
Email : **email1**

nom1, prenom1, adresse1 et email1 doivent être des variables.

Différent type de donnée PHP

Les données que l'on sera amené à manipuler peuvent être de différente nature. PHP prend en charge les types de données suivants :

- Chaîne de caractères ("Ce est une chaîne de caractères")
- Entier (Un entier est un nombre sans virgule … c'est pour cela qu'il est entier)
- Float (nombres à virgule comme 1,851 ou sous forme exponentielle)
- Booléen (TRUE ou FALSE pour vrai ou faux. Utilisé pour les tests)
- Array (stocke des informations sous forme de tableau)
- Objet (Les notions de programmation objet feront l'objet d'un chapitre)
- NULL (variable vide)
- Ressource (Ne sera pas abordé)

Nous allons voir dans la partie suivante que le type de donnée manipulé permet de faire des opérations qu'un autre type de données ne permettra pas. En effet additionner des caractères ne produit pas du tout le même résultat qu'additionner des nombres.

Par exemple :

- Si l'on considère 2 chaînes de caractère « 3 » et « 5 » et qu'on les additionne. On obtiendra « 35 ». On a bien additionné les caractères « 3 » et « 5 » entre eux. Une chaîne de caractère faisant uniquement référence à un code ASCII, les additionner revient à les accoler.
- Si l'on considère la valeur des nombres 3 et 5 et qu'on les additionne, ce n'est plus du tout la même chose. On obtiendra la valeur 8.

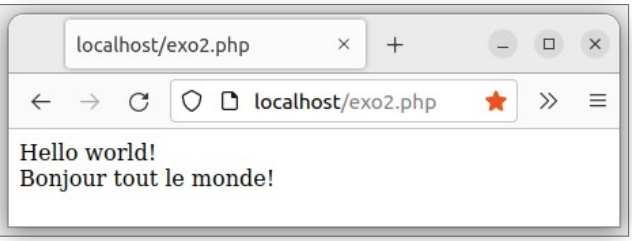
Variable de type « Chaîne de caractères »

Une chaîne de caractère est une suite de lettre ou de n'importe quel caractères, comme par exemple **`!?ferf@erfa.1234é`**.

PHP identifie une chaîne de caractère lorsque vos caractères sont encadrés par des guillemets. Vous pouvez utiliser pour cela des guillemets simples ou doubles :

Exemple

```
<?php
 $x = "Hello world!";
 $y = 'Bonjour tout le monde!';
 echo $x;
 echo "<br>";
 echo $y;
?>
```



PHP possède des fonctions permettant de réaliser des traitements dédiés aux chaînes de caractères.

La fonction `strlen()` retourne la longueur (le nombre de caractères) d'une chaîne de caractères :

```
<?php
 echo strlen("salut tout le monde!"); // affiche 20
?>
```

La fonction `str_word_count()` compte le nombre de mots dans une chaîne de caractères.

```
<?php
 echo str_word_count("Salut tout le monde!"); // Affiche 4
?>
```

Le fonction `strrev()` inverse les caractères d'une chaîne de caractères.

```
<?php
 echo strrev("Salut!"); // affiche !tulaS
?>
```

La fonction PHP `strpos()` recherche un texte dans une chaîne de caractères. Si une correspondance est trouvée, la fonction renvoie la position du caractère de la première correspondance. Si aucune correspondance n'est trouvée, il retournera `FALSE`.

```
<?php
 echo strpos("Hello world!", "world"); // affiche 6
 echo strpos("Hello world!", "Bonjour"); // affiche FALSE
?>
```

Note : La première position commence toujours à 0 et non 1.

La fonction `str_replace()` permet de remplacer des caractères par d'autres caractères.

```
<?php
 echo str_replace("a", "b", "abc abc abc"); // affiche bbc bbc bbc
?>
```

Ici tous les « a » de la chaîne « abc abc abc» seront remplacé par des « b ».

Variable de type « entier »

Une variable de type entier (c'est à dire un nombre sans virgule) compris entre -2 147 483 648 et 2 147 483 647.

Règles pour les entiers :

- Un entier doit posséder au moins un chiffre
- Un entier ne doit pas avoir de point décimal, c'est à dire une virgule
- Un entier peut être positif ou négatif
- Les nombres entiers peuvent être spécifiés avec une notation en décimale (base 10), en hexadécimale (base 16), en octale (base 8) ou en binaire (base 2)

Dans l'exemple suivant, $x est un entier. La fonction PHP var_dump() renvoie le type de données et sa valeur entre parenthèses :

```
<?php
 $x = 268;
 var_dump($x);
?>
```

Variable de type « float »

Une variable de type float est un nombre décimal (c'est à dire avec une virgule) ou un nombre sous forme exponentielle. **PHP utilise le point et non la virgule pour les nombres décimaux**.

Dans l'exemple suivant, la variable $x est un type float. La fonction PHP var_dump() renvoie le type de données et la valeur :

```
<?php
 $x = 61.875;
 var_dump($x);
?>
```

Variable de type booléens

Une variable de type booléen ne peut prendre que deux états : VRAI ou FAUX soit TRUE ou FALSE.

```
<?php
 $x = TRUE;
 $y = FALSE;
 var_dump($x);
```

```
 echo "<br>";
 var_dump($y);
?>
```

Les booléens sont utilisés pour vérifier si le résultat d'un test est vrai ou faux. Par exemple si l'on souhaite tester l'affirmation « Il pleut aujourd'hui », la réponse ne peut prendre que 2 formes c'est vrai (`true`) ou c'est faux (`false`). Ce point sera abordé ultérieurement dans la partie des tests conditionnels.