

Développement web en PHP

- Variables et Constantes -

Groupe des étudiants : CIR2

Ayoub KARINE

E-Mail : ayoub.karine@isen-ouest.yncrea.fr

Numéro de bureau : A3-78



Nom des variables

- Est toujours précédé du signe \$;
- Commence par une lettre ou un underscore ;
- Ne contient que des lettres, des chiffres et des underscore ;
- Est sensible à la casse.

```
1 $var = 5;
```

Type des variables

- Le type de la variable est déterminé automatiquement en fonction de l'initialisation

```
1 $var = 5;
```

Entier

```
1 $var = 87.56;
```

Flottant

```
1 $var1 = true;  
2 $var2 = false;
```

Booléen

Catégorie	Type	Description
scalaires	boolean	Vérité : True ou false
	integer	Entier négatifs et positifs
	float	Décimal (1.2 ; 1.2e3 ; 2E-10)
	string	Chaîne de caractères
composés	array	Tableau (association de valeurs et de clés) → un tableau, une liste, une table de hashage, un dictionnaire, une collection, une pile, une file d'attente,...
	object	Programmation objet : instanciation à partir d'une classe
spéciaux	resource	Référence vers une référence ou une ressource externe
	NULL	Variable sans valeur : valeur unique null

Chaînes de caractères

- **Syntaxe**

- **Entre guillemet double** : dans ce cas, PHP interpole les variables et les caractères d'échappement à l'intérieur de la chaîne de caractères.

```
1 $res = 42;  
2 echo "Result is $res."; // Affiche : Result is 42.
```

- **Entre guillemet simple (apostrophes)** : aucune interpolation de variable ou de caractère d'échappement n'est possible;

```
1 $res = 42;  
2 echo 'Result is $res.'; // Affiche : Result is $res.
```

Chaînes de caractères

- **Comparaison entre deux chaînes de caractères :**
 - opérateur d'égalité (même valeur) : ==
 - opérateur d'identité (même valeur et même type) : ===
- **Caractères spéciaux échappés :**
 - `\n` (fin de ligne : LF),
 - `\r` (retour à la ligne : CR),
 - `\t` (tabulation : HT),
 - `\v` (tabulation verticale : VT),
 - `\f` (saut de page : FF),

Chaînes de caractères

- **Opération sur les chaînes de caractères**
 - Concaténation

```
1 echo 'Hello ' . 'World!'; // Affiche : Hello World!
```

- Affectation combiné à une affectation

```
1 $var = 'Hello ';  
2 $var .= 'World!';  
3 echo $var; // Affiche : Hello World!
```

- Opérateur d'accès (lecture/écriture) à un caractère []

```
1 $var = 'Helbo.';  
2 $var[3] = 'l';  
3 echo $var; // Affiche : Hello.
```

Variables dynamiques

- En PHP, il est possible de transformer une chaîne de caractères en nom de variable. Pour cela on double le signe \$ devant la variable.

- Exemple 1 :

```
1 $a = 'variable';  
2 $variable = 2;  
3 echo $a; // Affiche : variable.  
4 echo $$a; // Affiche : 2.
```

- Exemple 2 :

```
1 $a = 'variable';  
2 $$a = 5;  
3 echo $a; // Affiche : variable.  
4 echo $variable; // Affiche : 5.
```

Variables pré-définies

- En PHP, il existe des variables prédéfinies (en majuscules) permettant de récupérer les informations suivantes :
 - `$_SERVER` : variables fournies par le serveur Web ;
 - `$_GET` : variables fournies par HTTP en méthode GET ;
 - `$_POST` : variables fournies par HTTP en méthode POST ;
 - `$_SESSION` : variables enregistrées dans la session ;
 - `$_COOKIE` : variables fournies par HTTP dans les cookies ;
 - `$_REQUEST` : équivalent à `$_GET` + `$_POST` + `$_COOKIE` ;
 - `$GLOBALS` : contient les noms (i.e. les clés) et les valeurs des variables globales ;
 - `$_ENV` : variables fournies par l'environnement ;
 - `$_FILES` : fichiers envoyés en méthode POST.

Constantes

- **Définition** : une constante possède une valeur qui ne peut pas être modifiée
- **Déclaration** :

```
define("CONSTANTE ", Valeur);  
Ou  
const CONSTANTE = Valeur;
```

- **Caractéristiques** :
 - Un nom qui ne commence pas par le signe \$;
 - Un nom, par convention, toujours en majuscules ;
 - Une définition et un accès partout dans le code (portée globale) ;
 - Par défaut, les constantes sont insensible à la casse.
- **Exemple** :

```
1 define('SERVER_PORT', 80);  
2 echo SERVER_PORT; // Affiche : 80.
```

Constantes pré-définies

- Plusieurs constantes sont déjà pré-définies en PHP

<https://www.php.net/manual/fr/reserved.constants.php>

PHP_VERSION (chaîne de caractères)

La version courante de PHP sous la forme d'une chaîne de caractères dans la notation "major.minor.release[extra]".

PHP_MAJOR_VERSION (entier)

La version majeure courante de PHP sous la forme d'une chaîne de caractères (i.e., int(5) depuis la version "5.2.7-extra"). Disponible à partir de PHP 5.2.7.

PHP_MINOR_VERSION (entier)

La version mineure courante de PHP sous la forme d'une chaîne de caractères (i.e., int(5) depuis la version "5.2.7-extra"). Disponible à partir de PHP 5.2.7.

PHP_RELEASE_VERSION (entier)

La version stable courante de PHP sous la forme d'une chaîne de caractères (i.e., int(5) depuis la version "5.2.7-extra"). Disponible à partir de PHP 5.2.7.

PHP_VERSION_ID (entier)

La version courante de PHP sous la forme d'un entier, utile pour la comparaison de versions (i.e., int(50207) depuis la version "5.2.7-extra"). Disponible à partir de PHP 5.2.7.

PHP_EXTRA_VERSION (chaîne de caractères)

La version "extra" courante de PHP sous la forme d'une chaîne de caractères (i.e., '-extra' depuis la version "5.2.7-extra"). Habituellement utilisée par les packagers des distributions pour indiquer une version de paquet. Disponible à partir de PHP 5.2.7.

PHP_ZTS (entier)

.....

Contrôle de l'état d'une variable

Définition des fonctions	Rôle
<code>bool isset(mixte \$v)</code>	Cette fonction indique si une variable est définie ou non. Elle retourne : <ul style="list-style-type: none">la valeur <code>FALSE</code> si la variable <code>\$v</code> n'a pas été initialisée ou si elle contient la valeur <code>NULL</code>;la valeur <code>TRUE</code> si elle contient une valeur quelconque.
<code>bool empty(mixte \$v)</code>	Cette fonction vérifie si une variable est vide. Elle retourne : <ul style="list-style-type: none">la valeur <code>TRUE</code> si la variable <code>\$v</code> n'a pas été initialisée, ou contient la valeur <code>NULL</code> ou <code>0</code> ou une chaîne égale à <code>" "</code> ou <code>"0"</code>;la valeur <code>FALSE</code> si elle contient une autre valeur.
<code>bool unset(mixte \$v)</code>	Supprime la variable <code>\$v</code> .
<code>void var_dump(mixte \$v)</code>	Affiche le type et la valeur sur la variable <code>\$v</code> passée en paramètre.