



Variables et constantes

Technologies web



Les variables

Variables et constantes

Les variables

- L'utilisation de **variables** est une notion de **base** en **programmation**
- Elle servent notamment à rendre nos **programmes dynamiques**
- Une **variable** est une **structure de donnée** pouvant correspondre à **2 types** :
 - Le **type primitif** permettant de représenter une valeur **entière, réelle, caractère, chaîne de caractères, booléenne** ou **nulle**
 - Le **type structuré** pour représenter des **tableaux** ou des **objets** afin de pouvoir **stocker plus d'une valeur** à la fois
- Une valeur d'une **variable** est susceptible d'être remplacée par une autre au cours de l'exécution du **programme**

Les variables PHP

- En programmation, une **variable** est définie selon **3 caractéristiques** principales :
 - Un **nom**
 - Un **type**
 - Une ou plusieurs **valeurs**
- Contrairement à des **langages fortement typés** comme le **C**, le **C++**, le **Java** ou encore le **C#**, **PHP** ne porte pas d'importance au **typage** des **variables**
- Il nous est donc possible de **déclarer** une **variable entière** avant de remplacer sa valeur par une **chaîne de caractères** plus tard dans l'exécution du **programme**

Les variables PHP

- Le **faible typage** de **PHP** rend son utilisation **plus souple mais peu assidue**
- Pour adopter de bonnes pratiques, nous affecterons toutes nos **variables** à un **type** ainsi qu'à une **valeur par défaut**
- Pensez également à **nommer** correctement vos **variables** - Un nom **concis** et **évocateur** facilitera la **lecture**, le **débogage** et la **maintenance** de vos **programmes**
- En **PHP**, toutes les **variables** doivent être **déclarées** à l'aide du caractère **\$** obligatoirement suivi d'une lettre MAJUSCULE ou minuscule ou d'un caractère **underscore** (**_**)

Déclarer une variable PHP

Correct	Incorrect	Explications
<code>\$variable</code>	<code>variable</code>	Une variable doit commencer par \$
<code>\$Variable1</code>	<code>\$Variable 1</code>	Les espaces sont interdits
<code>\$variable_suite</code>	<code>\$variable-suite</code>	Le tiret est interdit
<code>\$_variable</code>	<code>\$-variable</code>	Le tiret est interdit
<code>\$variable2</code>	<code>\$2variable</code>	Il ne peut y avoir de chiffre après le \$

- Notez également que **PHP** est un des rares langages acceptant les **accents** dans le nom de ses **variables**
- Pour autant et même si **PHP** nous offre ces différentes possibilités, nous déclarerons toutes nos variables au format **camel-case**, par convention (Exemple : `$maVariable`)

Les variables PHP

- Voici un court **script PHP** déclarant 7 **variables** de **types** différents

```
10  <?php
11      $prenom = 'Chris'; // Type string (chaîne de caractères)
12      $nom = "Chevalier"; // Type string (chaîne de caractères)
13      $age = 28; // Type entier
14      $estFormateur = true; // Type booléen
15      $cours = array("PHP", "Symfony", "P00"); // Tableau de strings
16      $autresCours = ["JavaScript", "React", "React Native"]; // Tableau de strings
17      $etudiant = new Etudiant(); // Objet de type 'Etudiant'
18  ?>
```


Explications

- Une **chaîne de caractères** doit être encadrée par des **guillemets simples ou doubles**
- Les **entiers**, **flottants** et **réels** prennent leur **valeur** en tant que telle
- Un **booléen** prend la valeur `true` ou `false` directement
- Un **tableau** peut être déclaré à l'aide de la **fonction** `array()` ou directement avec des **crochets** `[]`
- Si on ne veut pas typer une **variable**, on peut lui affecter la valeur `null`

Le typage et la casse

- Remarquez que le type d'une variable **PHP** n'est pas déclaré explicitement comme pour les langages **C**, **Java** ou encore **C++** mais **implicitement** en lui affectant une **valeur** d'un certain **type**
- Ce mécanisme est rendu possible grâce à **l'inférence de types** permettant à **l'interpréteur** du code source de rechercher automatiquement le **type** associé à une **expression** sans avoir à l'indiquer explicitement
- Notez également que les noms des variables sont **sensibles à la casse** - `$myVar` et `$MyVar` seront donc considérés comme **2 variables distinctes** par **PHP**

Affecter une valeur à une variable PHP

- L'**affectation** permet d'attribuer/fixer une **valeur** à une **variable** donnée
- Une **affectation** se fait à l'aide de l'**opérateur** `=`
- Dans le **script** précédent, nous avons donc déclaré 7 **variables** et nous leur avons affecté une ou plusieurs **valeurs par défaut**

Affecter une valeur à une variable PHP

- Voyons maintenant comment mettre à jour/changer la valeur d'une **variable**

```
10  <?php
11      $prenom = 'Chris';
12      $age = 28;
13
14      echo $prenom; // Affiche 'Chris'
15      echo '<br>'; // Affiche un saut de ligne
16      echo $age; // Affiche '28'
17
18      $prenom = 'Thomas';
19      $age = 34;
20
21      echo $prenom; // Affiche 'Thomas'
22      echo '<br>'; // Affiche un saut de ligne
23      echo $age; // Affiche '34'
24  ?>
```

La concaténation

- La **concaténation** permet d'assembler 2 ou davantage d'informations au sein d'une même **variable**
- En **PHP**, la **concaténation** se fait à l'aide du caractère point (.)

```
10  <?php
11      $prenom = "Chris";
12      $nom = "Chevalier";
13      $identite = $prenom . ' ' . $nom;
14
15      echo $identite // Affiche 'Chris Chevalier'
16  ?>
```



Les commentaires

Variables et constantes

Les commentaires

- Tous les langages de programmation proposent une syntaxe pour **commenter** des portions de code
- Il s'agit de bouts de code passés **sous silence** et qui ne seront donc **pas interprétés** par le navigateur
- Les **commentaires** servent essentiellement à **décrire** un morceau de code dans le cadre d'un travail collaboratif ou afin de faciliter la **relecture** et la **maintenance** d'un programme
- Ils doivent néanmoins être utilisés avec **parcimonie** - Il n'est pas utile de commenter chacune de ses lignes de code

Les commentaires PHP

- PHP propose **2 types** de commentaires :

- Le **commentaire linéaire** est un commentaire **sur une seule ligne**

```
10  <?php
11      // Voici un commentaire sur une ligne
12      echo "Hello ";
13
14      # Voici un autre commentaire sur une ligne
15      echo "World!";
16  ?>
```

- Le **commentaire multi-lignes** permet d'écrire un commentaire sur **plusieurs lignes**

```
10  <?php
11      /*
12      Voici un commentaire écrit
13      sur plusieurs
14      lignes
15      */
16      echo "Hello World!";
17  ?>
```




Déboggage

Variables et constantes

Débogguer le PHP (1/2)

- Pour rendre cette configuration permanente sous MAMP, WAMP ou XAMPP, rendez-vous dans le fichier `php.ini` de la version de **PHP** installée sur votre serveur local
- Sous **Windows**, ce fichier est disponible en suivant le chemin `C:\MAMP\conf\php8.0.8` -
⚠ Le dossier `php8.0.8` dépend du **numéro de version de PHP** utilisé
- Sous **Mac OS**, ce fichier est disponible en suivant le chemin `/Applications/MAMP/conf/php8.0.8` - ⚠ Le dossier `php8.0.8` dépend du **numéro de version de PHP** utilisé


Débogguer le PHP (2/2)

- Ouvrez le fichier `php.ini` dans votre éditeur de texte favoris
- À l'aide du raccourci `Ctrl + f` ou `Cmd + f`, recherchez la ligne `error_reporting = E_ALL`
- **Décommentez-la** en retirant le point-virgule `;` placé devant

```
455 error_reporting = E_ALL
```
- Une vingtaine de lignes plus bas, vous devriez voir une ligne `display_errors = off`
- Remplacez la valeur `off` par `On`

```
472 display_errors = On
```
- Redémarrez votre serveur MAMP pour que les changements soient pris en considération

Déboguer

 Notice: Undefined variable: myVar in /Applications/MAMP/htdocs/tmp copie.php on line 14				
Call Stack				
#	Time	Memory	Function	Location
1	0.0005	402808	{main}()	.../tmp copie.php:0



Un peu de maths

Variables et constantes

Des opérations mathématiques sur des variables (1/2)

- Étant donné que les **variables** peuvent contenir des **nombres**, il est normal et logique de pouvoir réaliser des **opérations mathématiques** sur ces dernières
- **PHP** propose justement plusieurs **opérateurs mathématiques** pour répondre à cette nécessité
- Il nous est donc possible d'**additionner**, **soustraire**, **multiplier** ou encore **diviser** les valeurs de nos **variables PHP**

Quelques opérateurs supplémentaires

- L'**opérateur modulo** (%) renvoie le **reste** de la division de 2 nombres - Ainsi, `4 % 2` donne `0` car le reste de l'opération `4 / 2` est égal à `0`
- L'**opérateur d'incrément** (++) **augmente de 1** la valeur d'une variable - Ainsi, `$a++` revient à écrire `$a += 1` ou encore `$a = $a + 1`
- L'**opérateur de décrémentation** (--) **diminue de 1** la valeur d'une variable - Ainsi, `$a--` revient à écrire `$a -= 1` ou encore `$a = $a - 1`

Des opérations mathématiques sur des variables (2/2)

```
10  <?php
11      $a = 10;
12      $b = 2;
13
14      $c = $a + $b; // $c vaut 10 + 2 = 12
15
16      $c = $a - $b; // $c vaut 10 - 2 = 8
17
18      $c = $a * $b; // $c vaut 10 * 2 = 20
19
20      $c = $a / $b; // $c vaut 10 / 2 = 5
21
22      $c = $a % $b; // $c vaut 10 % 2 = 0
23
24      $a++; // $a vaut 10 + 1 = 11
25      $b--; // $b vaut 2 - 1 = 1
26  ?>
```



Les constantes

Variables et constantes

Les constantes

- En programmation, l'utilisation de **constantes** peut être utile pour stocker des valeurs qui ne sont pas censées être modifiées au cours de l'**exécution** d'un **programme**
- L'une des **constantes** mathématiques les plus populaires est **le nombre Pi** dont la valeur correspond approximativement à `3.1415926535898`
- En **PHP**, on utilise la **fonction** `define()` pour **déclarer une constante**
- Cette **fonction** prend **2 arguments** obligatoires :
 - Le **nom** de la **constante** (**chaîne de caractères**)
 - La **valeur** de la **constante** (**chaîne de caractères, entier, booléen, etc.**)

Déclarer une constante PHP

- Déclarons puis affichons une **constante** représentant la température de l'eau en ébullition

```
10  <?php
11      // Déclaration de la constante 'TEMPERATURE_EAU_EBULLITION'
12      define("TEMPERATURE_EAU_EBULLITION", 100);
13
14      // Affichage de sa valeur : 100
15      echo "L'eau bout à " . TEMPERATURE_EAU_EBULLITION . " °C";
16  ?>
```

- Représentons maintenant la formule chimique de l'eau correspondant donc à la **chaîne de caractères** « H2O »

```
10  <?php
11      // Déclaration de la constante 'FORMULE_EAU'
12      define("FORMULE_EAU", "H2O");
13
14      // Affichage de sa valeur : 'H2O'
15      echo "La formule chimique de l'eau : " . FORMULE_EAU;
16  ?>
```

Redéfinir une constante PHP

- Comme son nom l'indique, une **constante** est censée contenir une **valeur fixe**
- Si vous tentez de redéfinir une **constante** existante, **PHP** vous renverra une **erreur**
- Rappelons également que l'**opérateur** `=` permet d'affecter une valeur à une **variable** et non pas à une **constante**
- Par **convention**, nous déclarons le **nom d'une constante** en `MAJUSCULE` et en `SNAKE_CASE`



TP1 - Hello World!

Introduction à PHP

TP - Hello World!

- Téléchargez et installez **MAMP** sur votre ordinateur
- Dans votre répertoire `htdocs`, créez un nouveau dossier `Hello-World` contenant un fichier `hello-world.php`
- Créez la structure **HTML** de ce **fichier**
- Cette page doit **afficher un message** `"Hello World!"` écrit avec **PHP**
- Ce message doit être affiché en **gras** et dans une **couleur différente** du noir de base - Souvenez-vous, le texte placé après l'instruction `echo` est interprété comme du code **HTML**
- Testez votre premier **programme PHP** en démarrant votre serveur **MAMP** puis en y localisant votre fichier `hello-world.php`



TP2 - Les nombres et les opérateurs

Variables et constantes

1 - Nombre

- Dans la section `<?php ?>` d'un fichier `tp2.php`
- Créez une **variable** `$integer` et donnez lui la valeur `102`
- Créez une **variable** `$float` et donnez lui la valeur `13.9` (🔴 on utilise le **point** `.` pour les décimaux quand on code !)
- Affichez les **2 variables** dans le navigateur à l'aide de la **fonction** `var_dump($variableAAfficher)`

2 - Conversion

- Créez une **variable** `$basic` de valeur `34`
- Créez une **variable** `$stringified` qui contient la version **string** de `$basic` à l'aide de la **fonction** `strval()`
- Affichez `$stringified` dans le **navigateur**

3 - Arrondir

- Créez une **variable** `$num` de valeur `1.5`
- Créez une **variable** `$rounded` qui contient l'arrondi de `$num` à l'aide de la **fonction** `round()`
- Affichez `$rounded` dans le **navigateur**

4 - Arithmétique

- Créez une **variable** `$test` de valeur `12` et une **variable** `$bis` de valeur `5`
- Utilisez tous les **opérateurs arithmétiques** sur ces **variables** et affichez les résultats
- 💡 `$test + $bis; // affiche 17`



TP3 - Mise en pratique

Variables et constantes

TP3 - Mise en pratique (1/2)

- Suivez les instructions décrites aux diapositives 17 et 18 pour mettre en place le **débogage** de votre **code PHP** de manière persistante
- Dans un nouveau fichier `welcome.php`, stockez vos prénom, nom et âge dans des **variables** `$firstName`, `$lastName` et `$age` (Cf. diapo 7)
- Stockez votre genre (`M.` ou `Mme`) dans une **constante** (Cf. diapo 26)
- Affichez un message de bienvenue du type `"Bonjour M. Tony Stark !" ou "Bonjour Mme Peper Potts !"` en utilisant ces **variables** et cette **constante** (Cf. diapo 12 et 26)
- Incrémentez la valeur stockée dans la **variable** `$age` pour vous faire vieillir d'un an puis, affichez un second message du type `"Vous avez maintenant 30 ans"` (Cf. diapo 22 et 23)

TP3 - Mise en pratique (2/2)

- Voici une proposition d'affichage final attendu pour valider le TP :

```
Bonjour M. Chris Chevalier !  
Vous avez maintenant 29 ans.
```



★ Bonus

Introduction à PHP

★ Bonus

- 💡 Certains éléments demandés ici seront abordés dans le cours suivant - N'hésitez pas à réaliser des recherches pour trouver vos réponses et/ou à demander à votre formateur
- Utiliser l'instruction d'affichage `echo` pour afficher :
 - Une **chaîne de caractères**
 - Une **variable** contenant un **entier**
 - Une **chaîne de caractères** faisant apparaître une **variable** (dont on veut connaître le contenu)
 - Un texte différent selon qu'une condition est vraie ou fausse



Des questions ?