

# Les variables et constantes

# Sommaire

Introduction.....	3
I. Déclarer une variable .....	4
II. Nommer une variable.....	5
III. Utiliser une variable.....	6
IV. Concaténer une variable.....	6
V. Modifier la valeur d'une variable .....	7
VI. Définir une constante .....	8
VII. Créer une page simple .....	9
A. Demande (ou besoin).....	9
B. Préparation.....	9
C. Conception.....	10
D. Mise en œuvre.....	10
E. Intégration HTML.....	11
F. Code complet.....	11
G. Rendu de la page.....	12

Crédits des illustrations :  
© Fotolia, DR

## Les repères de lecture



Retour au chapitre



Définition



Objectif(s)



Espace Élèves



Vidéo /  
Audio



Point important /  
À retenir



Remarque(s)



Pour aller  
plus loin



Normes et lois



Quiz



## Introduction

Les **variables** sont un élément fondamental de tout langage de programmation. PHP permet de manipuler des informations qui peuvent être de natures très diverses. Les informations sont également appelées des données. Elles peuvent être des nombres, des chaînes de caractères, des tableaux, des résultats mathématiques, etc.

Les variables sont considérées comme des boîtes pratiques dans lesquelles nous rangeons ces données que nous pourrions ensuite utiliser ou modifier quand nous le voulons.

Les variables en PHP sont écrites avec :

- le **symbole dollar \$** qui permet d'identifier une variable ;
- le **nom de la variable**.

Elles sont donc sous la forme : **\$nomDeLaVariable**.

Elles peuvent contenir des informations très variées. Par exemple, nous pouvons enregistrer dans une variable le jour de la semaine (prenons lundi). Cette variable que nous pourrions appeler *\$jour* sera utilisable en permanence tout au long de notre script et sa valeur pourra être modifiée lorsque le jour de la semaine changera.

Cette variable pourra donc contenir « lundi » ou « mardi » ou n'importe quel jour de notre choix. Nous disons que « lundi » ou « mardi » est la valeur contenue dans la variable.

## I. Déclarer une variable

Une variable se déclare simplement avec le signe égal = que nous appelons l'**opérateur d'assignation** et qui assigne (ou affecte ou attribue) une valeur à la variable.



Faites très attention, le signe *égal* = sert à associer une valeur. Pour tester une égalité, vous devrez utiliser un *double égal* ==.

Le fait de déclarer une variable revient à :

- **créer cette variable** ;
- **lui assigner une valeur**.

La syntaxe permettant la déclaration d'une variable est la suivante :

`$nomDeLaVariable = valeur`



Cette déclaration ne doit pas être lue de la gauche vers la droite mais **de la droite vers la gauche**. Nous ne devons pas dire ou penser « `$nomDeLaVariable égal valeur` » mais « **nous assignons la valeur à \$nomDeLaVariable** ».

Étudions cette syntaxe. Sur la même ligne, nous avons :

- le nom que nous donnons à la variable ;
- le signe *égal* qui est l'**opérateur d'assignation** (ou d'affectation) ;
- la **valeur** que nous **assignons** à la variable.

Nous utilisons de manière synonyme les verbes « assigner » ou « affecter ». Ils signifient que nous attribuons une valeur à une variable. Autrement dit, nous déclarons que la variable contient cette valeur.

Si nous reprenons l'exemple de jour de la semaine et si nous souhaitons assigner à la variable la valeur « lundi » nous aurons le code suivant :

```
1 $variable = 'lundi';
```

Étudions cette instruction :

- Elle comporte le signe *égal* puis le nom que nous avons choisi pour cette variable que nous avons appelée *\$variable*. Le choix du nom de la variable est libre, mais pour plus de cohérence avec la valeur qu'elle contient nous devrions nommer la variable « jour » ;
- Le signe *égal* qui permet d'assigner une valeur à la variable ;
- La valeur que nous souhaitons attribuer à la variable. Cette valeur est ici une chaîne de caractères, il est donc nécessaire de l'entourer avec des *quotes* ;
- Le point-virgule est obligatoire et indique la fin de l'instruction.

```
1 $jour = 'lundi';
```

Autre exemple de création d'une variable :

```
1 $saison = 'été';  
2 echo $saison; //affiche été
```

Soulignons ici quatre points importants :

- le signe *égal* **n'est pas utilisé en PHP pour établir une égalité arithmétique**. Le signe *égal* est utilisé uniquement pour assigner une valeur à une variable ;
- **assigner une valeur à une variable équivaut à créer cette variable**. On dit que la variable est **déclarée** ;
- **choisir un nom pertinent pour la variable**. Il est pertinent d'envisager un nom en rapport avec le type d'information qu'il contient. Ainsi, une variable à laquelle on attribue des prénoms aura tout intérêt à être nommée *\$firstnames* (prénoms en anglais) et non pas *\$city* ou *\$paysArray* ;
- **le nom de la variable est libre**. Cependant il doit respecter les règles de nommage détaillées ci-après et doit être explicite (on doit pouvoir comprendre ce qu'elle contient).

## II. Nommer une variable

Les règles de nommage des variables qui suivent doivent obligatoirement être respectées :

- Un nom de variable **ne contient pas d'espace** ;
- Un nom de variable **ne contient pas de caractères spéciaux ni accentués** (é, ô, ù, \*, #, etc.) ;
- Un nom de variable **doit commencer par une lettre** ;
- Un nom de variable **doit être explicite**, et de préférence en **anglais** (langue internationale) ;
- Un nom de variable s'il contient plusieurs mots, **doit se noter en lower-CamelCase** : la première lettre du premier mot en minuscule et la première lettre des mots suivants en majuscule (exemple : *\$maVariableEstLongue*) ;
- **Le nom est sensible à la casse** (majuscule/minuscule). Ainsi *\$abc* est une variable différente de *\$ABC*, *\$Fruit* est une variable différente de *\$fruit*.



Pour aller plus loin et en savoir plus sur les conventions de nommage, reportez-vous aux publications de l'organisation PHP-Fig <http://www.php-fig.org/psr/>

### III. Utiliser une variable

Nous venons donc de voir comment assigner une valeur à une variable.

L'exemple suivant nous montre :

- la déclaration d'une variable à laquelle nous assignons un nom de saison (automne). Cette variable sera donc nommée `$saison` et l'affichage de ce que contient la variable avec la fonction `echo` ou `var_dump` pour le débogage ;
- l'affectation d'une longue chaîne de caractères (plusieurs phrases) à une variable ;
- la déclaration d'une variable contenant un nombre entier (pour stocker un nombre dans une variable, les *quotes* ne sont pas nécessaires) ;
- l'affectation à une variable `$saved` d'une autre variable `$number`.

```
1 <?php
2 $saison = 'automne';
3 echo $saison; // affiche automne
4
5 // on affecte une longue chaîne de caractères
6 $loremIpsum = 'Lorem ipsum dolor sit amet, consectetur adipiscing elit.
7     Nam sed libero at urna sagittis malesuada. Nam in mattis nisi.
8     Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae;
9     Praesent laoreet malesuada enim. Maecenas enim urna, dignissim ac tortor sit amet, consequat auctor lacus.
10    Vestibulum non gravida lacus, id euismod est.';
11
12 $number = 3; // on affecte un nombre entier à une variable
13
14 $saved = $number; // on affecte la valeur contenue dans $number (soit 3) à $saved
```

Fig. 1

### IV. Concaténer une variable

Il est souvent nécessaire d'afficher un texte qui doit contenir une variable.

Par exemple, nous pouvons afficher un texte contenant le nom d'une saison et une description de cette saison. Nous avons vu que les chaînes de caractères (en anglais : *string*) nécessitent des *quotes* pour être affichées tandis que les variables n'en ont pas besoin.

Pour afficher en même temps une chaîne de caractères et une variable, il suffit de placer ces deux éléments l'un après l'autre **en les séparant par un point**.

Le point permet de juxtaposer les deux éléments. On parle d'**opérateur de concaténation**. Il est recommandé de mettre un espace avant et après le point pour une meilleure lisibilité.

#### Exemples :

```
1 <?php
2 /* concaténation string / variable */
3 $sport = 'basket';
4 echo 'Nous pratiquons le ' . $sport;
5
6 /* concaténation string / variable / string */
7 $saison = 'automne';
8 echo 'les feuilles tombent en <strong>' . $saison . '</strong>';
9
10 /* concaténation variable / variable */
11 $first = 'Hello ';
12 $second = 'World!';
13 echo $first . $second;
```

Fig. 2

**La concaténation est une syntaxe fondamentale** de PHP et il importe de bien vérifier la succession des *quotes* (pour les chaînes de caractères) et des points (pour la concaténation).

## V. Modifier la valeur d'une variable

Une variable se modifie aisément. Pour ce faire, il suffit de simplement lui assigner une autre valeur.

#### Exemples

```
1 <?php
2 $jour = 'lundi';
3 echo 'nous sommes le ' . $jour; // nous sommes le lundi
4 $jour = 'mardi';
5 echo 'nous sommes le ' . $jour; // nous sommes le mardi
6
7 $number = 5;
8 $number = $number + 1;
9 echo $number; // 6
```

Fig. 3



Comme nous l'avons vu, le signe «égal» n'établit pas l'égalité mais il permet d'assigner une valeur. Le résultat de l'addition `$number + 1` est une valeur que nous pouvons tout à fait affecter à la variable `$number`.

## VI. Définir une constante

Les constantes sont très utiles pour manipuler une valeur qui n'est pas destinée à évoluer, ce qui permet d'optimiser votre code en prenant moins de mémoire.

Une constante est donc une valeur définie de façon permanente et dans certains cas il peut être envisagé d'utiliser une constante plutôt qu'une variable car nous venons de voir qu'une variable pouvait être aisément modifiée ce qui n'est pas le cas d'une constante.

Une constante aura un nom de constante et une valeur. Par convention, le nom de la constante est en lettres capitales (majuscule). Elle n'est précédée d'aucun signe.

Pour déclarer une constante, nous utilisons la fonction PHP *define*.

- nous ouvrons une parenthèse ;
- nous écrivons le nom de la constante entre deux *quotes* ;
- nous ajoutons une virgule pour séparer le nommage de la constante avec la définition de sa valeur ;
- nous écrivons ensuite la valeur de la constante (en utilisant les *quotes* si cette valeur est une chaîne de caractères) ;
- nous fermons la parenthèse ;
- le point-virgule indique la fin de l'instruction.

Ainsi si nous souhaitons déclarer puis afficher une constante contenant le nom d'un pays :

```
1 <?php
2 define('PAYS', 'France');
3 echo 'nous sommes en ' . PAYS; // nous sommes en France
```

Fig. 4

### **Autre exemple**

Si nous souhaitons déclarer puis afficher une constante contenant le nom de la page d'accueil :

```
1 <?php
2 define('HOME', 'index.php');
3 echo '<a href="' . HOME . '">Accueil</a>'; // <a href="index.php">Accueil</a>
```

Fig. 5

*Il est à retenir que les constantes ont un rôle très spécifique et sont beaucoup moins utilisées que les variables.*



## VII. Créer une page simple

Nous avons effectué nos premiers pas en PHP et il est temps de passer à la pratique. Nous allons donc créer un projet qui consiste à produire une page web simple permettant une mise en application des aspects étudiés jusqu'à présent dans ce cours.

### A. Demande (ou besoin)

Le besoin consiste à créer une page web qui présente une saison (le printemps) en utilisant nos connaissances PHP, notamment l'affichage des chaînes de caractères et l'utilisation des variables.

Nous souhaitons que cette page offre le contenu suivant :

- un titre ;
- une image ;
- un texte court.

Afin de réaliser cette page, nous allons récupérer le contenu provenant de la page Printemps du site Wikipédia et dont l'URL est : <http://fr.wikipedia.org/wiki/Printemps>.

- Le titre sera : « Le printemps » ;
- l'image sera fournie par le fichier image que nous allons renommer : « printemps.jpg » ;
- le texte court sera : « Le printemps est l'une des quatre saisons des zones tempérées, suivant l'hiver et précédant l'été. Le printemps se caractérise par un radoucissement du temps, la fonte des neiges, le bourgeonnement et la floraison des plantes. » ;

### B. Préparation

Afin de traiter au mieux la demande, nous devons préparer les éléments suivants :

- récupérer le contenu de la page (texte et image) et vérifier que nous avons l'ensemble du contenu (il est fourni ci-dessus pour être copié-collé) ;
- préparer les outils : démarrer votre éditeur de code, démarrer votre serveur ;
- créer un nouveau répertoire *saison* dans Wamp sous *www* ;
- créer un nouveau fichier *saison.php* et l'enregistrer dans le répertoire *www/saison*.

## C. Conception

Nous souhaitons donc produire une page web qui sera structurée en HTML.

Le titre sera intégré avec l'élément HTML *title* et *h 1*, le paragraphe avec *p* et l'insertion d'image avec *img*.

Dans ce cours, nous ne vous présentons pas le code HTML, pensez à toujours respecter la sémantique HTML complète pour que votre page soit correctement indexée et éviter des erreurs.

Cette structure pourra accueillir l'information que nous allons gérer avec PHP.

Nous notons que l'information principale est le nom de la saison et que celui-ci revient souvent dans le texte. Nous pouvons à raison penser à utiliser une variable pour afficher cette saison.

L'utilisation d'une constante pourrait être envisagée mais il faut penser que si nous réalisons une page sur le printemps, il est fort possible que nous ayons ensuite à réaliser une page sur l'été, l'hiver ou l'automne. De manière générale, et cet exercice est une bonne occasion de le constater, l'utilisation des constantes est très spécifique et est beaucoup plus rare que l'utilisation des variables.

Puisque nous allons gérer la saison avec une variable, et que cette saison peut changer, nous allons également gérer le paragraphe ainsi que l'image avec des variables.

Le nommage des variables pourra être celui-ci :

- *\$seasonName* : le nom de la saison ;
- *\$seasonDescription* : le paragraphe de description de la saison ;
- *\$seasonImage* : le nom de l'image de la saison.

À noter qu'en tant que développeur, nous sommes libres de gérer comme nous l'entendons cette demande. Il serait tout à fait possible par exemple de nommer différemment les variables ou de ne pas assigner le texte à une variable. Ceci est une première démonstration du fait que le développement autorise des conceptions variées.

## D. Mise en œuvre

Nous commençons par déclarer nos variables :

```
1 <?php
2 $seasonName = 'printemps';
3 $seasonDescription = 'Le printemps est \\une des quatre saisons des zones tempérées, suivant \\l'hiver et
4   précédant \\l'été. Le printemps se caractérise par un radoucissement du temps, la fonte des neiges,
5   le bourgeonnement et la floraison des plantes.';
6 $seasonImage = 'printemps.jpg';
```

Fig. 6

## E. Intégration HTML

Du code HTML peut être écrit dans un fichier PHP. Si le code HTML est placé en dehors des balises PHP, ce code HTML ne sera pas interprété par PHP mais **son rendu sera visible dans le navigateur**.

Notons ici en outre que **les balises PHP peuvent s'intégrer dans du HTML**.

```
9  <title><?php echo $seasonName; ?></title>
10 <h1><?php echo $seasonName; ?></h1>
```

Fig.7

La variable contenant le nom du fichier est intégrée dans l'attribut *src* de l'élément *img*.

Une image devant contenir un attribut *alt*, nous intégrons donc la variable contenant le nom de la saison dans cet attribut.

```
12 ">
```

Fig.8

La variable contenant le texte est intégrée dans l'élément *p*.

```
14 <p><?php echo $seasonDescription; ?></p>
```

Fig.9

## F. Code complet

```
1  <?php
2  $seasonName = 'printemps';
3  $seasonDescription = 'Le printemps est l\'une des quatre saisons des zones tempérées, suivant l\'hiver et
4      précédant l\'été. Le printemps se caractérise par un radoucissement du temps, la fonte des neiges,
5      le bourgeonnement et la floraison des plantes.';
6  $seasonImage = 'printemps.jpg';
7  ?>
8
9  <!DOCTYPE html>
10 <html lang="fr">
11   <head>
12     <meta charset="UTF-8">
13     <title><?php echo $seasonName; ?></title>
14   </head>
15   <body>
16     <h1><?php echo $seasonName; ?></h1>
17     ">
18     <p><?php echo $seasonDescription; ?></p>
19   </body>
20 </html>
```

Fig.10

## G. Rendu de la page

### printemps



Le printemps est l'une des quatre saisons des zones tempérées, suivant l'hiver et précédant l'été. Le printemps se caractérise par un radoucissement du temps, la fonte des neiges, le bourgeonnement et la floraison des plantes.

*Fig. 11*