

Développement web côté serveur avec PHP

les variables définies par l'utilisateur

[PHP: Essentiel - Manual](#)

En PHP, les variables sont représentées par un signe dollar “\$” suivi du nom de la variable. Le nom est sensible à la casse.

Les noms de variables suivent les mêmes règles de nommage que les autres entités PHP. Un nom de variable valide doit commencer par une lettre ou un souligné (`_`), suivi de lettres, chiffres ou soulignés. Exprimé sous la forme d'une expression régulière, cela donne : `'[a-zA-Z\u209f-\u209f][a-zA-Z0-9_\u209f-\u209f]*'`

En résumé : comme leur nom l'indique, les variables peuvent stocker des valeurs qui changent ou que l'on pourra changer soi-même. Les variables sont toujours précédées du signe \$ (dollar). Ensuite on peut utiliser les lettre de `A` à `Z`, en minuscule ou en majuscule et le signe `_` (souligné, underscore) puis les chiffres de `0` à `9`.

les types de données

nombre entier (integer, int)

[code/integer.php](#)

```
1 <?php
2 // affectation de la valeur 42 à la variable '$nombre'
3 $nombre = 42;
4
5 // affectation de la valeur -42 à la variable '$autre_nombre'
6 $autre_nombre = -42;
```

nombre à virgule flottante (float, flt)

[code/float.php](#)

```
1 <?php
2 // affectation de la valeur 3.14 à la variable '$pi'
3 $pi = 3.14;
4
5 // affectation de la valeur -3.14 à la variable '$moins_pi'
6 $moins_pi = -3.14;
```

chaîne de caractères (string, str)

[code/string1.php](#)

```
1<?php
2// affectation
3$nom = "Php";
4$salutation = "Bonjour";
5
6// concaténation et affichage
7echo $salutation . ' ' . $nom;
8echo PHP_EOL;
9
10// concaténation et affectation
11$phrase = $salutation . ' ' . $nom;
12
13// affichage
14echo $phrase;
15echo PHP_EOL;
16
17// interpolation et affichage
18echo "Bonjour $nom";
19echo PHP_EOL;
20
21// interpolation et affectation
22$phrase = "Salut $nom";
23
24// affichage
25echo $phrase;
26echo PHP_EOL;
27
28// affectation sans interpolation
29$phrase = 'Salut $nom';
30
31// affichage
32echo $phrase;
33echo PHP_EOL;
```

[code/string2.php](#)

```
1<?php
2// affectation
3$nom = "Php";
4
5// affichage
6// utilisation du signe d'échappement `\" pour afficher des guillemets
7echo 'Bonjour \"Php\"';
8echo PHP_EOL;
9
10// affichage
11// utilisation du signe d'échappement `\" pour afficher des doubles gu
12echo "Bonjour \"Php\"";
```

```

13 echo PHP_EOL;
14
15 // interpolation et affichage
16 // utilisation du signe d'échappement `\\` pour afficher des guillemets
17 echo 'Bonjour \'$nom\'';
18 echo PHP_EOL;
19
20 // interpolation et affichage
21 // utilisation du signe d'échappement `\\` pour afficher des doubles guillemets
22 echo "Bonjour \"$nom\"";
23 echo PHP_EOL;

```

[code/string3.php](#)

```

1 <?php
2 // affectation
3 $nom = "Php";
4
5 // interpolation et affichage
6 // utilisation du signe d'échappement `\\` pour afficher un retour à la ligne
7 echo "Bonjour Php\\n";
8
9 // affichage sans interpolation
10 // le retour à la ligne n'est pas affiché, la chaîne de caractère est lue telle quelle
11 echo 'Bonjour Php\\n';
12 echo PHP_EOL;
13
14 // attention : cette instruction n'affichera pas `Bonjour Php5.6`
15 // elle affichera `Bonjour .6` en générant une erreur `PHP Notice: Undefined variable: 5 in /code/string3.php on line 15`
16 echo "Bonjour $nom5.6";
17 echo PHP_EOL;
18
19 // pour afficher `Bonjour Php5.6` il faut utiliser les signes `{}` (accroches)
20 // limites du nom de la variable
21 echo "Bonjour {$nom}5.6";
22 echo PHP_EOL;
23
24 // mais on peut aussi utiliser la concaténation
25 echo "Bonjour $nom" . "5.6";
26 echo PHP_EOL;

```

booléen (boolean, bool)

[code/boolean.php](#)

```

1 <?php
2 // affectation de la valeur `true` (vrai) à la variable `$php_c_est_facile`
3 $php_facile = true;
4
5 // affectation de la valeur `false` (faux) à la variable `$php_c_est_son`

```

```
6 $php_c_est_soporifique = false;
```

tableau (array)

[code/array1.php](#)

```
1 <?php
2 // création d'un tableau simple vide
3 $tableau = array();
4
5 // ajout d'éléments dans le tableau
6 $tableau[] = 'foo';
7 $tableau[] = 'bar';
8 $tableau[] = 'baz';
9
10 // lecture des éléments du tableau
11 echo $tableau[0];
12 echo PHP_EOL;
13 echo $tableau[1];
14 echo PHP_EOL;
15 echo $tableau[2];
16 echo PHP_EOL;
```

[code/array2.php](#)

```
1 <?php
2 // création d'un tableau simple vide
3 $tableau = array();
4
5 // ajout d'éléments dans le tableau
6 $tableau[100] = 'foo';
7 $tableau[200] = 'bar';
8 $tableau[300] = 'baz';
9
10 // lecture des éléments du tableau
11 echo $tableau[100];
12 echo PHP_EOL;
13 echo $tableau[200];
14 echo PHP_EOL;
15 echo $tableau[300];
16 echo PHP_EOL;
```

[code/array3.php](#)

```
1 <?php
2 // création d'un tableau simple
3 $fibonacci = array(0, 1, 2, 3, 5, 8, 13, 21, 34, 55);
4
5 // lecture des éléments du tableau
6 echo $fibonacci[0];
```

```
7 echo PHP_EOL;
8 echo $fibonacci[1];
9 echo PHP_EOL;
10 echo $fibonacci[2];
11 echo PHP_EOL;
12 echo $fibonacci[3];
13 echo PHP_EOL;
14 echo $fibonacci[4];
15 echo PHP_EOL;
16 echo $fibonacci[5];
17 echo PHP_EOL;
18 echo $fibonacci[6];
19 echo PHP_EOL;
20 echo $fibonacci[7];
21 echo PHP_EOL;
22 echo $fibonacci[8];
23 echo PHP_EOL;
24 echo $fibonacci[9];
25 echo PHP_EOL;
```

[code/array4.php](#)

```
1 <?php
2 // création d'un tableau associatif (dictionnaire, hash array)
3 $legumes = array(
4     'a' => 'artichaut',
5     'b' => 'broccoli',
6     'c' => 'carotte',
7 );
8
9 // lecture des éléments du tableau
10 echo $legumes['a'];
11 echo PHP_EOL;
12
13 echo $legumes['b'];
14 echo PHP_EOL;
15
16 echo $legumes['c'];
17 echo PHP_EOL;
```