



# Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [Les fonctions](#) ► [Opérations sur les types et variables](#)

## Opérations sur les types et variables

Voici quelques fonctions qui permettent de travailler sur les types et [les variables](#), appelées par PHP [fonctions de gestion des variables](#). La fonction `gettype()` : renvoie une chaîne de caractères qui correspond au type de la variable passée en argument. La documentation officielle décrit la fonction de cette façon :

```
string gettype ( mixed $var )
```

Nous trouvons dans l'ordre :

- le type de retour de la fonction : string, cela signifie que la fonction renvoie une donnée de type string (chaîne de caractères)
- le nom de la fonction : `gettype`
- entre parenthèses les paramètres de la fonction avec
  - le type du paramètre : ici `mixed` signifie que le paramètre `var` accepte plusieurs types
  - le nom du paramètre : `var`

Nous trouvons ensuite dans la documentation la liste des paramètres et leurs descriptions. Puis la liste des valeurs de retour possibles suivie de quelques exemples d'utilisation. Quelques exemples :

```
1 <?php
2
3 // operations_types_variables_1.php
4 $nb = 123;
5 echo gettype($nb);
6 echo PHP_EOL;
7 echo gettype('azerty');
8 echo PHP_EOL;
9 $tab = [1, 2, 3];
10 $type = gettype($tab);
11 echo "le type de \$tab est $type" . PHP_EOL;
```

Autre fonction très intéressante, `var_dump()`, qui affiche le type et la valeur d'une variable.

```
void var_dump ( mixed $expression [, mixed $... ] )
```

Le type de retour est void, ce qui signifie que la fonction ne renvoie aucune valeur, c'est donc une procédure. Le paramètre `expression` peut être de n'importe quel type et peut être accompagné d'autres paramètres facultatifs. Les paramètres facultatifs sont toujours écrits entre crochets. Quelques exemples :

```
1 <?php
2
3 // operations_types_variables_2.php
4 $nb = 12;
5 var_dump($nb);
6 var_dump('azerty');
7 $tab = array(array(1.1, 2, 3), array('a', 56.8, 'c'));
8 var_dump($tab);
```

Trois autres fonctions à retenir :

1. `isset()` : vérifie si une variable a été définie (déclaration + assignation) et si elle est différente de null
2. `unset()` : détruit une variable
3. `empty()` : vérifie si une variable est vide, sont considérés comme vides "" (une chaîne vide), 0, 0.0, "0", NULL, FALSE, array() et [] (un tableau vide), var \$var; (une variable déclarée, mais sans valeur dans une classe)

Exemple :

```
1 <?php
2
3 // operations_types_variables_3.php
4 $message = '';
5 $var = 125;
6 if (isset($var)) {
7     $message = 'la variable $var a été définie et est différente de null, ';
8     if (empty($var)) {
9         $message .= 'mais est vide';
10    } else {
11        $message .= "et a pour valeur $var";
12    }
13 } else {
14     $message = 'la variable n\'a pas été définie ou est égale à null';
15 }
16 echo $message . PHP_EOL;
```

Fin

## NAVIGATION



[Accueil](#)

■ [Ma page](#)

[Pages du site](#)

[Mon profil](#)

[Cours actuel](#)

[Intro PHP](#)

[Participants](#)

[Le langage PHP : introduction](#)

[Les types et les variables](#)

[Les opérateurs](#)

[Les structures de contrôle](#)

[Les structure de données](#)

[Les fonctions](#)

[Les fonctions prédéfinies](#)

**[Opérations sur les types et variables](#)**

[Opérations sur les chaînes](#)

[T.P. voyelles et consonnes](#)

[Opérations sur les tableaux](#)

[T.P. occurrences](#)

[Les fonctions utilisateurs](#)

[T.P. fonction d'étoiles](#)

[Valeurs et références](#)

[T.P. fonction doubler](#)

[T.P. fonction calculer](#)

[Les erreurs](#)

[Les fichiers](#)

[Les expressions rationnelles](#)

[PHP et HTML](#)

[Petite application](#)

[Mes cours](#)

## ADMINISTRATION



[Administration du cours](#)

[Réglages de mon profil](#)

---

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))  
[Intro PHP](#)