



Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [Les opérateurs](#) ► [Les opérateurs arithmétiques](#)

Les opérateurs arithmétiques

Nous retrouvons en PHP les mêmes [opérateurs arithmétiques](#) que ceux rencontrés précédemment :

Opérateur	Exemple	Opérations	Résultat
-	-a	négation	opposé de a
+	a + b	addition	somme de a et b
-	a - b	soustraction	différence de a et b
*	a * b	multiplication	produit de a et b
/	a / b	division	quotient de a et b
%	a % b	modulo	reste de a divisé par b

Les mêmes priorités s'appliquent à ces opérateurs :

Priorité	Opérateurs
1	parenthèses
2	+ unaire, -unaire
3	multiplication, division, modulo
4	addition, soustraction

Voici quelques exemples :

```
1 <?php
2
3 // operateurs_arithmetiques_1.php
4 echo 12 + 5;
5 echo PHP_EOL;
6 $nb = 18;
7 $nb1 = 3;
8 $nb3 = $nb % $nb1;
9 echo "$nb mod $nb1 est égal à $nb3 ";
10 echo PHP_EOL;
11 echo 2 + 10 / 5;
12 echo PHP_EOL;
13 // notez l'utilisation des parenthèses...
14 echo (2 + 10) / 5;
15 echo PHP_EOL;
```

□

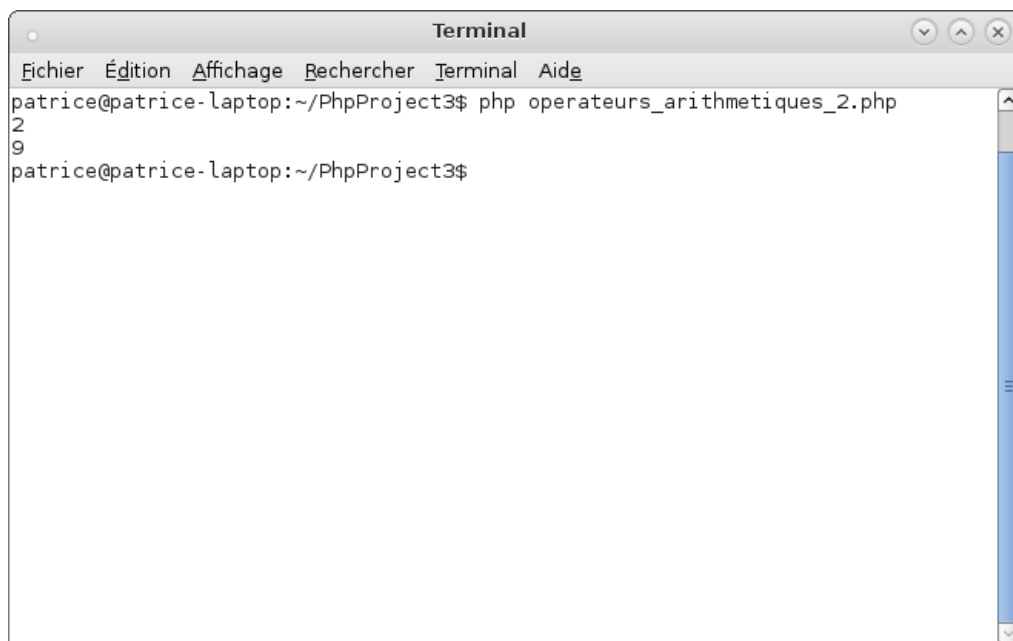
Lorsque j'écris dans mon programme `echo 2 + 10 / 5;` j'effectue un calcul arithmétique à partir de valeurs littérales, c'est-à-dire de valeurs qui sont données explicitement dans le code. En programmation, l'utilisation de valeurs littérales dans les traitements est une mauvaise pratique, il sera donc indispensable d'utiliser des variables dans nos calculs. Par exemple : `echo $nb4 + $nb5 / $nb6;`.

À ces opérateurs nous allons ajouter les opérateurs d'incrémentation et de décrémentation. Comme leurs noms

l'indiquent, ces opérateurs servent à incrémenter (ajouter 1 à) ou à décrémenter (soustraire 1 à) une valeur numérique ou une chaîne de caractères. L'opérateur d'incrémentation est ++, celui de décrémentation --.

```
1 <?php
2
3 // operateurs_arithmetiques_2.php
4 $nb = 1;
5 $nb++; // équivaut à $nb = $nb + 1;
6 echo $nb;
7 echo PHP_EOL;
8 $nb2 = 10;
9 $nb2--; // équivaut à $nb2 = $nb2 - 1;
10 echo $nb2;
11 echo PHP_EOL;
```

Cela affichera :

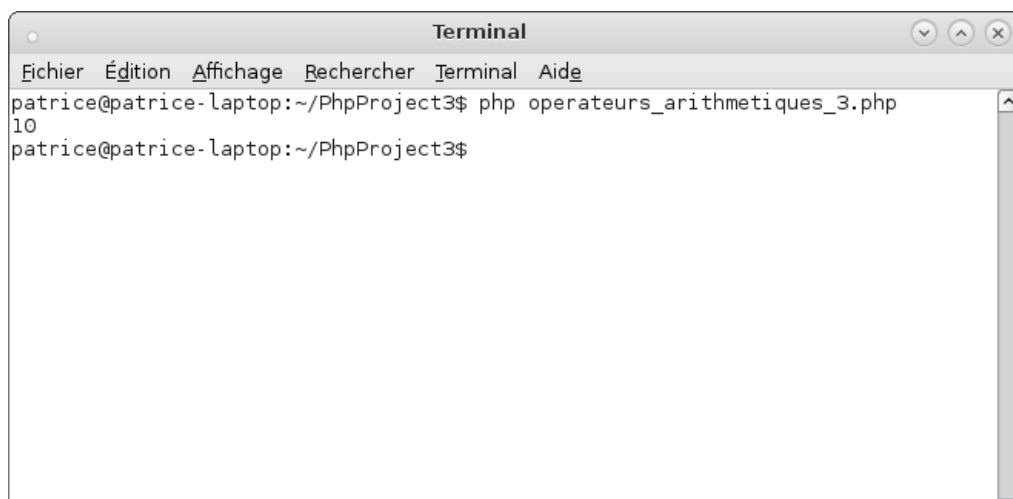


```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
patrice@patrice-laptop:~/PhpProject3$ php operateurs_arithmetiques_2.php
2
9
patrice@patrice-laptop:~/PhpProject3$
```

Maintenant si l'on souhaite déclarer une variable nb, lui assigner la valeur 10, incrémenter cette valeur avec l'opérateur ++ et afficher la valeur de nb, nous pourrions écrire :

```
1 <?php
2
3 // operateurs_arithmetiques_3.php
4 $nb = 10;
5 echo $nb++;
6 echo PHP_EOL;
```

Cela affichera :



```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
patrice@patrice-laptop:~/PhpProject3$ php operateurs_arithmetiques_3.php
10
patrice@patrice-laptop:~/PhpProject3$
```



En fait je m'attendais plutôt à ce que l'affichage soit 11 puisque j'ai augmenté la valeur de \$nb de 1. Il semble que la valeur n'ait pas changé, pourtant, si j'affiche \$nb sur la ligne suivante :

```
1 <?php
2
3 // operateurs_arithmetiques_4.php
4 $nb = 10;
5 echo $nb++;
6 echo PHP_EOL;
7 echo $nb;
8 echo PHP_EOL;
```

Cela affichera cette fois :

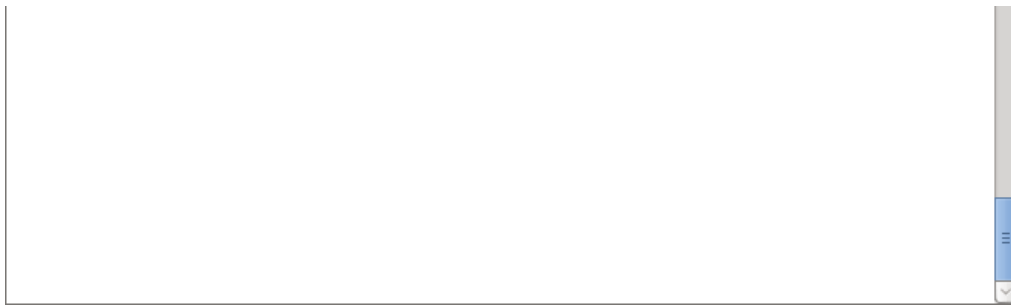
```
patrice@patrice-laptop:~/PhpProject3$ php operateurs_arithmetiques_4.php
10
11
patrice@patrice-laptop:~/PhpProject3$
```

La valeur de \$nb a donc bien été incrémentée mais après l'affichage. Dans l'expression `echo $nb++` ; PHP a d'abord affiché la valeur de \$nb, c'est-à-dire 10, puis a augmenté cette valeur de 1. Si je modifie la position de l'opérateur ++ pour le mettre devant la variable, j'obtiens :

```
1 <?php
2
3 // operateurs_arithmetiques_5.php
4 $nb = 10;
5 echo ++$nb;
6 echo PHP_EOL;
7 echo $nb;
8 echo PHP_EOL;
```

Ce qui affiche :

```
patrice@patrice-laptop:~/PhpProject3$ php operateurs_arithmetiques_5.php
11
11
patrice@patrice-laptop:~/PhpProject3$
```



Ici la valeur de `$nb` a été incrémentée avant l'affichage. Dans l'expression `echo ++$nb`; PHP a d'abord augmenté `$nb` de 1 puis a affiché sa valeur, c'est-à-dire 11.

Si l'opérateur est préfixé, c'est-à-dire placé avant la variable, l'incrémentation ou la décrémentation a lieu avant l'évaluation de l'expression. Si l'opérateur est postfixé, c'est-à-dire placé après la variable, l'incrémentation ou la décrémentation a lieu après l'évaluation de l'expression.

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

 Introduction

Les opérateurs arithmétiques

 Les opérateurs d'assignation

 Les opérateurs de comparaison

 Les opérateurs logiques

 Les opérateurs de concaténation de chaînes

 La priorité des opérateurs

 Types, variables et opérateurs

Les structures de contrôle

Les structure de données

Les fonctions

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](#)