



Introduction au langage de programmation PHP

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

Introduction

Nous allons retrouver en PHP les [opérateurs](#) que nous connaissons déjà, à savoir :

- [les opérateurs arithmétiques](#)
- [les opérateurs de comparaison](#)
- [les opérateurs logiques](#)

En plus de ces opérateurs nous verrons [les opérateurs d'assignation](#) et [les opérateurs de concaténation de chaînes](#).

Tous les opérateurs ne seront pas évoqués ici, nous en verrons d'autres lors de notre progression.

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](#)



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;
```

□

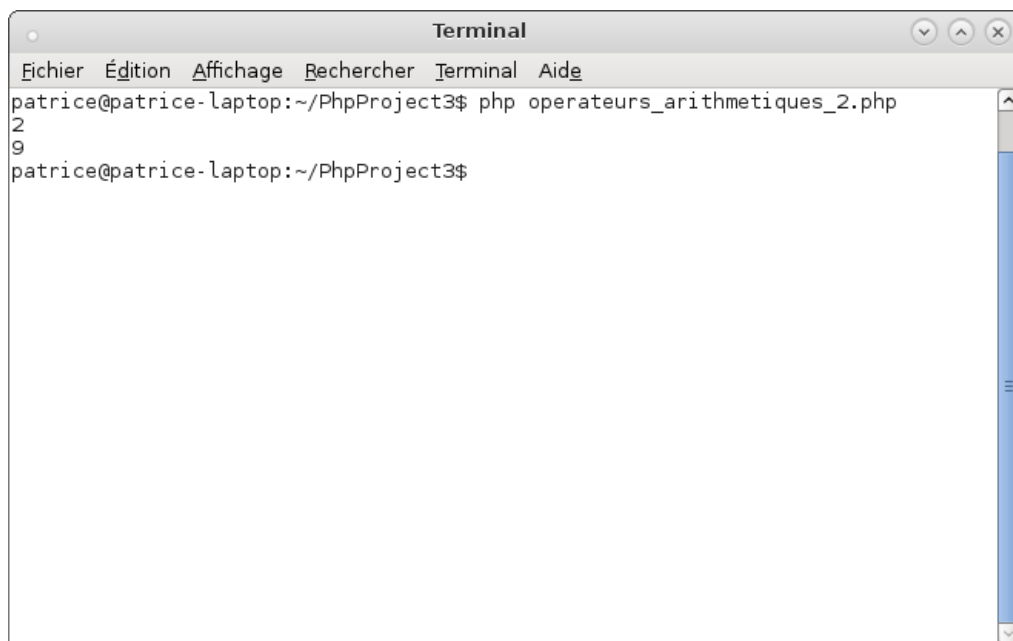
Lorsque j'écris dans mon programme `echo 2 + 10 / 5;` j'effectue un calcul arithmétique à partir de valeurs littérales, c'est-à-dire de valeurs qui sont données explicitement dans le code. En programmation, l'utilisation de valeurs littérales dans les traitements est une mauvaise pratique, il sera donc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

À ces opérateurs nous allons ajouter les opérateurs d'incrément 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 :

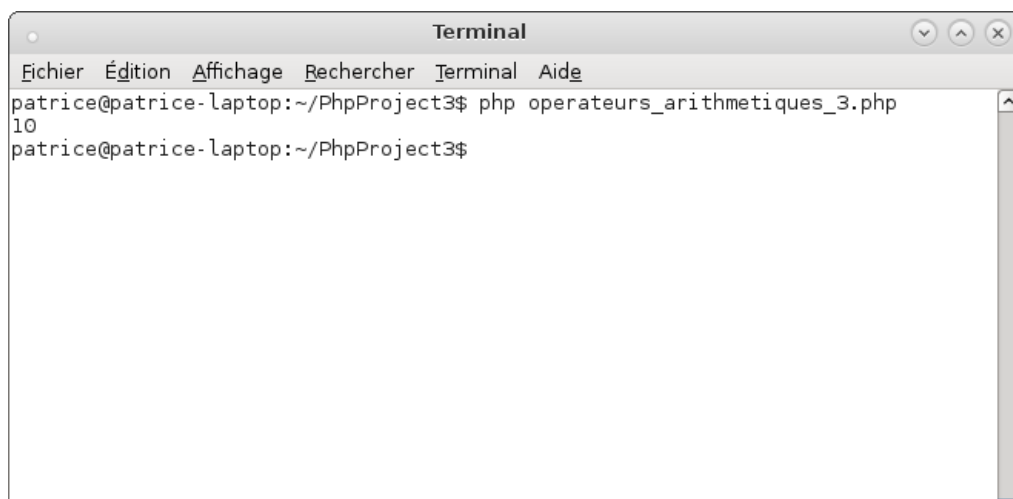


```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
patrice@patrice-laptop:~/PhpProject3$ php operateurs_arithmetiques_2.php
2
9
patrice@patrice-laptop:~/PhpProject3$
```

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 :

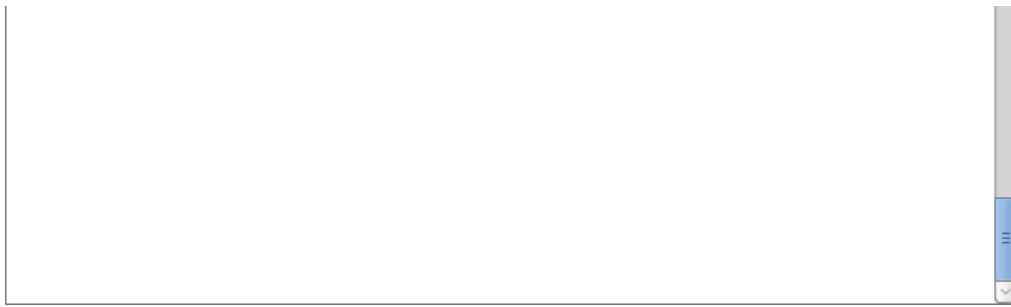
```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
patrice@patrice-laptop:~/PhpProject3$ php operateurs_arithmetiques_4.php
10
11
patrice@patrice-laptop:~/PhpProject3$
```

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 :

```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
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](#)



Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [Les opérateurs](#) ► [Les opérateurs d'assignation](#)

Les opérateurs d'assignation

Pour assigner une valeur à une variable nous avons, pour l'instant, utilisé l'opérateur `=`. L'opérateur `=` peut aussi être combiné à un opérateur arithmétique pour assigner une valeur après la réalisation d'un calcul. Par exemple :

```
1 <?php
2
3 // les_operateurs_d_assignation_1.php
4 $nb = 10; // déclare une variable nb et lui assigne la valeur 10
5 $nb += 20; // ajoute 20 à la valeur de nb et affecte à nb cette nouvelle valeur
6 echo $nb; // affiche la valeur de nb
7 echo PHP_EOL;
```

Cela affiche :

```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
patrice@patrice-laptop:~/PhpProject3$ php les_operateurs_d_assignation_1.php
30
patrice@patrice-laptop:~/PhpProject3$
```

En fait,

```
$nb += 20;
```

équivalent à :

```
$nb = $nb + 20;
```

Voici les [opérateurs d'assignation](#) liés aux opérateurs arithmétiques :

Opérateur	Exemple	Signification
<code>+=</code>	<code>\$a += 10;</code>	<code>\$a = \$a + 10;</code>
<code>-=</code>	<code>\$a -= 10;</code>	<code>\$a = \$a - 10;</code>
<code>*=</code>	<code>\$a *= 10;</code>	<code>\$a = \$a * 10;</code>
<code>/=</code>	<code>\$a /= 10;</code>	<code>\$a = \$a / 10;</code>
<code>%=</code>	<code>\$a %= 10</code>	<code>\$a = \$a % 10;</code>

[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](#)

Introduction au langage de programmation PHP

Accueil

►

Mes cours

►

Développement logiciel

►

Intro PHP

►

Les opérateurs

►

Les opérateurs de comparaison

Les opérateurs de comparaison

Voici les [opérateurs de comparaison](#) que nous avons vus précédemment :

Opérateur	Exemple	Comparaisons	Résultat
==	a == b	égalité	vrai si a à la même valeur que b
!=	a != b	différence	vrai si a est différent de b
<	a < b	infériorité	vrai si a est inférieur à b
>	a > b	supériorité	vrai si a est supérieur à b
<=	a <= b	infériorité ou égalité	vrai si a est inférieur ou égal à b
>=	a >= b	supériorité ou égalité	vrai si a est supérieur ou égal à b

Nous allons en ajouter deux :

Opérateur	Exemple	Comparaisons	Résultat
===	a === b	égalité	vrai si a à la même valeur que b et est du même type
!==	a !== b	différence	vrai si a est différent de b ou si le type de a est différent de celui de b

```
1 <?php
2
3 // operateurs_de_comparaison_1.php
4 $nb1 = 10;
5 $nb2 = 10.0;
6 echo $nb1 == $nb2; // affiche 1
7 echo $nb1 === $nb2; // n'affiche rien, $nb1 et $nb2 ne sont pas du même type
8 echo 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

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](#)



Introduction au langage de programmation PHP

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

Les opérateurs logiques

Nous retrouverons en PHP les opérateurs logiques classiques : ET, OU, NON, auxquels viendra s'ajouter le OU exclusif. De plus PHP propose deux opérateurs ET ainsi que deux opérateurs OU qui ne diffèrent que par leur priorité. Voici les opérateurs du plus prioritaire au moins prioritaire ainsi que leurs significations :

Opérateur	Exemple	Signification
not (NON)	! \$a	vrai si \$a est faux
and (ET)	\$a && \$b	vrai si \$a et \$b sont vrais
or (OU)	\$a \$b	vrai si \$a est vrai ou si \$b est vrai ou si \$a et \$b sont vrais
and (ET)	\$a and \$b	vrai si \$a et \$b sont vrais
xor (OU exclusif)	\$a xor \$b	vrai si \$a est vrai ou si \$b est vrai mais pas les deux en même temps
or (OU)	\$a or \$b	vrai si \$a est vrai ou si \$b est vrai ou si \$a et \$b sont vrais

Quelques exemples :

```
1 <?php
2
3 // operateurs_logiques_1.php
4 $value1 = false;
5 $value2 = true;
6 $value3 = true;
7 // n'affiche rien puisque la valeur de cette expression est false
8 echo $value1 && $value2;
9 echo PHP_EOL;
10 // affiche 1
11 echo $value1 || $value2;
12 echo PHP_EOL;
13 // affiche 1
14 echo $value2 && $value3;
15 echo PHP_EOL;
16 // affiche 1
17 echo !$value1;
18 echo PHP_EOL;
```

□

Vous vous souvenez sûrement qu'en PHP des valeurs d'autres types peuvent être considérées comme des booléens.

D'autres exemples :

```
1 <?php
2
3 // operateurs_logiques_2.php
4 $nb = 456;
5 $nb1 = 0;
6 // affiche 1
7 echo !$nb;
8 // affiche 1
9 echo $nb || $nb1;
10 echo PHP_EOL;
```

```
11 // n'affiche rien
12 echo $nb && $nbl;
13 echo PHP_EOL;
14 $str = "";
15 $str1 = "azerty";
16 // affiche 1
17 echo !$str;
18 echo PHP_EOL;
19 // affiche 1
20 echo $str || $str1;
21 echo PHP_EOL;
22 // n'affiche rien
23 echo $str && $str1;
24 echo 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](#) [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](#)



Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [Les opérateurs](#) ► [Les opérateurs de concaténation de chaînes](#)

Les opérateurs de concaténation de chaînes

Pour réunir deux chaînes en une seule, PHP possède deux [opérateurs](#) : le point (.) et la virgule (,). Par exemple :

```
1 <?php
2
3 // operateurs_de_concatenation_1.php
4 $str1 = 'Hello';
5 $str2 = 'world';
6 echo $str1 . ' ' . $str2 . ' !';
7 echo PHP_EOL;
```

Cela affichera :

```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
patrice@patrice-laptop:~/PhpProject3$ php operateurs_de_concatenation_1.php
Hello world !
patrice@patrice-laptop:~/PhpProject3$
```

Nous obtenons le même résultat avec le code suivant :

```
1 <?php
2
3 // operateurs_de_concatenation_2.php
4 $str1 = 'Hello';
5 $str2 = 'world';
6 echo $str1, ' ', $str2, ' !';
7 echo PHP_EOL;
```

□

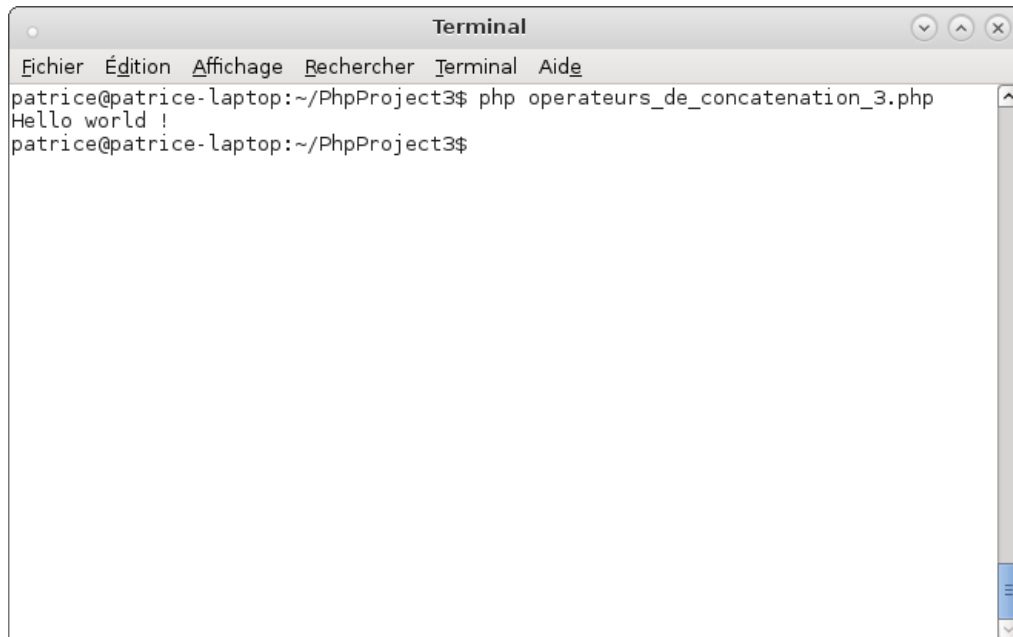
Mais s'il est possible de concaténer des chaînes de caractères avec l'opérateur ',', la documentation officielle ne fait mention que de l'opérateur '.'. Vous savez donc celui qu'il faut privilégier...

Il existe aussi un opérateur d'assignation/concaténation : .= .

```
1 <?php
2
3 // operateurs_de_concatenation_3.php
```

```
4 $str = 'Hello';  
5 $str .= ' world !';  
6 echo $str;  
7 echo PHP_EOL;
```

Cela affichera :



```
Terminal  
Fichier Édition Affichage Rechercher Terminal Aide  
patrice@patrice-laptop:~/PhpProject3$ php operateurs_de_concatenation_3.php  
Hello world !  
patrice@patrice-laptop:~/PhpProject3$
```

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](#)



Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [Les opérateurs](#) ► [La priorité des opérateurs](#)

La priorité des opérateurs

Voici [la priorité des opérateurs](#) (du plus prioritaire au moins prioritaire) telle que présentée dans la documentation officielle :

clone new
[]
++ -- ~ (int) (float) (string) (array) (object) (bool) @
instanceof
!
* / %
+ - .
<< >>
< <= > >=
== != === !== <>
&
^
&&
? :
= += -= *= /= .= %= &= = ^= <<= >>= =>
and
xor
or
,

Quelques exemples :

```

1 <?php
2
3 // priorite_operateurs_1.php
4 $nb = 100;
5 $nb1 = 10;
6 $nb2 = 5;
7 $nb3 = 3;
8 /* dans l'ordre des priorités :
9  * l'addition entre parenthèses
10  * puis la division
11  * puis l'assignation
12  */
13 $nb += (10 + 5) / 3;
```



```
14 echo $nb;  
15 echo PHP_EOL;  
16 $value = false;  
17 $str = "azerty";  
18 $nb = 0;  
19 /* dans l'ordre des priorités :  
20  * le NOT logique  
21  * puis le ET logique  
22  * puis le OU logique  
23  * puis l'assignation  
24  */  
25 $result = !$value || $str && $nb;  
26 echo $result;  
27 echo 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](#) [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](#)



Introduction au langage de programmation PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [Les opérateurs](#) ► [Types, variables et opérateurs](#)

Commencé le	mardi 24 mars 2015, 17:11
État	Terminé
Terminé le	mardi 24 mars 2015, 17:15
Temps mis	3 min 50 s
Note	6,00 sur 7,00 (86%)

Question 1

Correct

Note de 1,00 sur 1,00

🚩 Marquer la question

Relier l'expression à son type

12 + 25	integer	✓
1.5 - \$nb	double	✓
\$a \$b	boolean	✓
"Hello " . \$str	string	✓
"12" . "25"	string	✓
\$a && \$b	boolean	✓

Votre réponse est correcte.

La réponse correcte est : 12 + 25 – integer, 1.5 - \$nb – double, \$a || \$b – boolean, "Hello " . \$str – string, "12" . "25" – string, \$a && \$b – boolean

Question 2

Correct

Note de 1,00 sur 1,00

🚩 Marquer la question

Qu'affichera

```
<?php
```

```
$nb = 10;  
$nb++;  
echo $nb;
```

Veuillez choisir une réponse :

- ☐ a. 1
- ☒ b. 11 ✓
- ☐ c. aucun affichage
- ☐ d. 10

Votre réponse est correcte.

La réponse correcte est : 11

Question 3

Correct

Note de 1,00 sur 1,00

🚩 Marquer la question

Qu'affichera

```
<?php
```

```
$nb = 456;  
echo $nb > 10;
```

Veuillez choisir une réponse :

- ☐ a. true
- ☐ b. aucun affichage
- ☒ c. 1 ✓
- ☐ d. 412

Votre réponse est correcte.

La réponse correcte est : 1

Question 4

Correct

Note de 1,00 sur 1,00

🚩 Marquer la question

Qu'affichera

```
<?php
```

```
$nb = 10;  
echo ++$nb;
```

Veuillez choisir une réponse :

- ☐ a. 10
- ☒ b. 11 ✓
- ☐ c. 12
- ☐ d. true
- ☐ e. Hello !

Votre réponse est correcte.

La réponse correcte est : 11

Question 5

Incorrect

Note de 0,00 sur 1,00

🚩 Marquer la question

Qu'affichera

```
<?php
```

```
$nb = 10;  
echo $nb === 10.0;
```

Veuillez choisir une réponse :

- ☐ a. aucun affichage
- ☐ b. 10
- ☐ c. 11
- ☐ d. true
- ☒ e. false ✖

Votre réponse est incorrecte.

La réponse correcte est : aucun affichage

Question 6

Correct

Note de 1,00 sur 1,00

🚩 Marquer la question

Qu'affichera

```
<?php
```

```
$nb = 10;  
echo $nb !== 10.0;
```

Veuillez choisir une réponse :

- ☒ a. 1 ✔
- ☐ b. aucun affichage
- ☐ c. 0
- ☐ d. 456
- ☐ e. Hello !

Votre réponse est correcte.

La réponse correcte est : 1

Question 7

Correct

Note de 1,00 sur 1,00

🚩 Marquer la question

Qu'affichera

```
<?php
```

```
$nb = 17;  
echo $nb % 3;
```

Veuillez choisir une réponse :

- ☐ a. 1

- ☒ b. 2 ✓
- ☐ c. 3
- ☐ d. true
- ☐ e. aucun affichage

Votre réponse est correcte.

La réponse correcte est : 2

[Terminer la relecture](#)

NAVIGATION DU TEST



[Terminer la relecture](#)

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