

Priorité des opérateurs

<code>**</code>	arithmétique
<code>++ -- ~ (int) (float) (string) (array) (object) (bool) @</code>	types et incrément/décrément
<code>instanceof</code>	types
<code>!</code>	logique
<code>* / %</code>	arithmétique
<code>+ - .</code>	arithmétique et chaîne de caractères
<code><< >></code>	bitwise
<code>< <= > >=</code>	comparaison
<code>== != === !== <></code>	comparaison
<code>&</code>	bitwise et références
<code>^</code>	bitwise
<code> </code>	bitwise
<code>&&</code>	logique
<code> </code>	logique
<code>? :</code>	ternaire
<code>= += -= *= **= /= .= %= &= = ^= <<= >>= ==</code>	affectation
<code>and</code>	logique
<code>xor</code>	logique
<code>or</code>	logique
<code>,</code>	plusieurs utilisations

```
<?php
```

```
$a = 45 * 3;  
var_dump($a);
```

```
($a = 45) * 3;  
var_dump($a);
```

```
$b = 45 * 3 > 100;  
var_dump($b);
```

```
$b = 45 * (3 > 100);  
var_dump($b);
```

```
$val = true || true && false; //true || (true && false);  
var_dump($val);
```

```
$val = true || true and false;  
//(true || true) and false;  
//$val = true and false;  
//($val = true) and false;  
//$val and false ;  
var_dump($val);
```

```
$test = !false || true;  
//$test = true || true => true  
$test = !(false || true);  
//$test= !(true);  
//$test = false;
```

```
?>
```