

Opérateurs JavaScript

- L'opérateur d' **affectation** (=) affecte une valeur à une variable:

Exemple : let x = 10;

- L'opérateur d' **addition** (+) ajoute des nombres:

Exemple :

```
let x = 5;  
let y = 7;  
let z = x + y;
```

- L'opérateur de **multiplication** (*) multiplie des nombres:

Exemple :

```
let x = 5;  
let y = 2;  
let z = x * y;
```

- L'opérateur de **division** (/) divise des nombres:

Exemple :

```
let x = 5;  
let y = 2;  
let z = x / y;
```

- L'opérateur de **module** (%) renvoi le reste de la division:

Exemple :

```
let x = 5;  
let y = 2;  
let z = x % y
```

- L'opérateur d' **incrément** (++) incrémente des nombres :

Exemple:

```
let x = 5;  
x++;  
let z = x;
```

- L'opérateur de **décrément** (--) incrémente des nombres
- L'opérateur d' **exponentiation** (**) élève le premier opérande à la puissance du deuxième opérande:

Exemple:

```
let x = 5;  
let z = x ** 2;           // résultat : 25
```

Opérateurs arithmétiques

Opérateurs	Description
+	Addition
-	Soustraction
*	Multiplication
/	Division
%	Modulo
++	Incrémentation
--	Décrémentation

Opérateurs d'affectation

Opérateur	Exemple	Egal à
=	$x = y$	$x = y$
+=	$x += y$	$x = x + y$
-=	$x -= y$	$x = x - y$
*=	$x *= y$	$x = x * y$
/=	$x /= y$	$x = x / y$
%=	$x \% = y$	$x = x \% y$
**=	$x ** = y$	$x = x ** y$

Exemples d'affectation

L'opérateur d'affectation `=` affecte une valeur à une variable.

Exemple : `let x = 10;`

L'opérateur d'affectation `+=` ajoute une valeur à une variable.

Exemple: `let x = 10;`
`x += 5;`

L'opérateur d'affectation `-=` soustrait une valeur d'une variable.

`let x = 10;`
`x -= 5;`

L'opérateur d'affectation `*=` multiplie une variable

`let x = 10;`
`x *= 5;`

L'opérateur d'affectation `/=` divise une variable

`let x = 10;`
`x /= 5;`

L'opérateur d'affectation `%=` affecte un reste à une variable.

`let x = 10;`
`x %= 5;`

Opérateur sur les chaînes

L'**opérateur** **+** peut également être utilisé pour ajouter (**concaténer**) des chaînes.

```
<script>
let text1 = "John";
let text2 = "Doe";
let text3 = text1 + " " + text2;
document.getElementById("demo").innerHTML = text3;
</script>
```

L'opérateur d'affectation **+=** peut également être utilisé pour ajouter (concaténer) des chaînes

```
<script>
let text1 = "Quelle belle ";
text1 += "journée";
document.getElementById("demo").innerHTML = text1;
</script>
```

Opérateurs de comparaison

Opérateurs	Description
==	Égal à
===	Égal à valeur et type
!=	Différent
!==	Différent valeur et type
>	Supérieur à
>=	Supérieur ou égal à
<	Inférieur à
<=	Inférieur ou égal à
?	Opérateur ternaire

Opérateurs logiques

Opérateur	Description
&&	Et logique
	Ou logique
!	Not logique

Opérateurs de types

Opérateur	Description
Typeof	Retourne le type de variable
instanceof	Renvoie vrai si un objet est une instance d'un type d'objet

Priorité de l'opérateur

La priorité des opérateurs décrit l'ordre dans lequel les opérations sont effectuées dans une expression arithmétique.

Exemple:

*let x = 100 + 50 * 3;*

Le résultat de l'exemple ci-dessus est-il le même que $150 * 3$, ou est-ce le même que $100 + 150$?

L'addition ou la multiplication est-elle faite en premier ?

Comme dans les mathématiques scolaires traditionnelles, la multiplication se fait en premier.

La multiplication (*) et la division (/) ont une priorité plus élevée que l'addition (+) et la soustraction (-).

Et (comme dans les mathématiques scolaires) la priorité peut être modifiée en utilisant des parenthèses :

Exemple:

*let x = (100 + 50) * 3;*

Lorsque vous utilisez des parenthèses, les opérations à l'intérieur des parenthèses sont calculées en premier.

Lorsque de nombreuses opérations ont la même priorité (comme l'addition et la soustraction), elles sont calculées de gauche à droite :

Exemple:

*let x = 100 + 50 * 3;*