

Apprendre JavaScript - Les opérateurs

Les opérateurs sont des symboles qui permettent d'effectuer des opérations sur des variables et des valeurs. Il existe plusieurs types d'opérateurs en JavaScript. Dans cette section, nous allons voir les opérateurs arithmétiques, d'assignation, de comparaison et logiques.

Opérateurs arithmétiques

Les opérateurs arithmétiques sont utilisés pour effectuer des opérations mathématiques sur les nombres.

- `+` : Addition
- `-` : Soustraction
- `*` : Multiplication
- `/` : Division
- `%` : Modulo. Celui-ci retourne le reste de la division entière de deux nombres. Pratique pour savoir si un nombre est pair ou impair ou pour créer une boucle qui se répète tous les x tours.

```
let a = 10;
let b = 3;

console.log(a + b); // 13
console.log(a - b); // 7
console.log(a * b); // 30
console.log(a / b); // 3.3333333333333335
console.log(a % b); // 1
```

Exercice: Demander à l'utilisateur de saisir un nombre et afficher si ce nombre est pair ou impair. Exercice2: Demander à l'utilisateur de saisir un nombre et afficher si ce nombre est divisible par 3.

Opérateurs d'assignation

Les opérateurs d'assignation sont utilisés pour assigner des valeurs à des variables.

- `=` : Assignation. Il assigne la valeur de la droite à la variable de gauche.
- `+=` : Addition. Ajoute la valeur de la droite à la variable de gauche.
- `-=` : Soustraction. Soustrait la valeur de la droite à la variable de gauche.
- `*=` : Multiplication. Multiplie la valeur de la droite à la variable de gauche.
- `/=` : Division. Divise la valeur de la droite à la variable de gauche.

```
let a = 10; // La variable a contient la valeur 10
let b = 3; // La variable b contient la valeur 3

a += b; // a = a + b. La variable a contient la valeur précédente de a à laquelle on
```

```
a ajouté la valeur de b
a -= b; // a = a - b. La variable a contient la valeur précédente de a à laquelle on
a soustrait la valeur de b
a *= b; // a = a * b. La variable a contient la valeur précédente de a à laquelle on
a multiplié la valeur de b
a /= b; // a = a / b. La variable a contient la valeur précédente de a à laquelle on
a divisé la valeur de b
```

Opérateurs de comparaison

Les opérateurs de comparaison sont utilisés pour comparer deux valeurs.

- `==` : Égal à. Vérifie si les deux valeurs sont égales.
- `===` : Égal à et du même type. Vérifie si les deux valeurs sont égales et du même type (Voir la section sur les types de données).
- `!=` : Différent de. Vérifie si les deux valeurs ne sont pas égales.
- `>` : Supérieur à. Vérifie si la valeur de gauche est supérieure à la valeur de droite.
- `<` : Inférieur à. Vérifie si la valeur de gauche est inférieure à la valeur de droite.
- `>=` : Supérieur ou égal à. Vérifie si la valeur de gauche est supérieure ou égale à la valeur de droite.
- `<=` : Inférieur ou égal à. Vérifie si la valeur de gauche est inférieure ou égale à la valeur de droite.

```
let a = 10;
let b = "3";

console.log(a == b); // false
console.log(a === b); // false
console.log(a != b); // true
console.log(a > b); // true
console.log(a < b); // false
console.log(a >= b); // true
console.log(a <= b); // false
```

La comparaison et opérations de chaînes et de nombres en JavaScript

En JavaScript, il est possible de comparer des chaînes et des nombres. Lorsque vous comparez une chaîne et un nombre, JavaScript convertit automatiquement la chaîne en nombre. Par exemple, `"10" > 5` retourne `true` car JavaScript convertit la chaîne `"10"` en nombre `10` avant de faire la comparaison.

Cependant, l'addition de chaînes et de nombres n'est pas possible. Par exemple, `"10" + 5` retourne `"105"` car JavaScript concatène (fusionne) les deux valeurs en une seule chaîne. Pour additionner une chaîne et un nombre, vous devez convertir la chaîne en nombre.

Convertir une chaîne en nombre

Pour convertir une chaîne en nombre, vous pouvez utiliser la fonction `parseInt()` ou `parseFloat()`.

Il existe depuis peu une syntaxe courte pour convertir une chaîne en nombre, en utilisant le signe `+` devant la chaîne sans espace. Par exemple, `+a` convertit la chaîne `a` en nombre

```
let a = "10";
let b = 5;

let somme = parseInt(a) + b; // 15
sommeSyntaxeCourte = +a + b; // 15
```

Opérateurs logiques

Les opérateurs logiques sont utilisés pour combiner des expressions logiques.

- `&&` : ET. Retourne `true` si les deux expressions sont `true`.
- `||` : OU. Retourne `true` si au moins une des deux expressions est `true`.
- `!` : NON. Inverse la valeur de l'expression.

```
let a = 10;
let b = 3;

console.log(a > 5 && b > 5); // false
console.log(a > 5 || b > 5); // true
console.log(!(a > 5)); // false
```

Meilleures pratiques

Il est recommandé d'utiliser l'opérateur `===` pour comparer deux valeurs. En effet, l'opérateur `==` ne vérifie pas le type des valeurs. Par exemple, `1 == '1'` retourne `true` alors que `1 === '1'` retourne `false`. L'opérateur `===` est plus strict et permet d'éviter des erreurs de comparaison.

De plus, il est recommandé de toujours utiliser des parenthèses pour améliorer la lisibilité du code. Par exemple, `a + b * c` est moins lisible que `(a + b) * c`.

Finalement, il est recommandé de convertir variables contenant des chaînes en nombres avant de faire des opérations mathématiques. Cela permet d'éviter des erreurs de calcul.