

Apprendre JavaScript - Trouver et changer le type d'une variable

Table des matières

- [Trouver le type d'une variable](#)
- [Changer le type d'une variable](#)
 - [Changer un nombre en chaîne de caractères](#)
 - [Changer une chaîne de caractères en nombre](#)
 - [Changer une chaîne de caractères en tableau](#)
 - [Changer un tableau en chaîne de caractères](#)
 - [Convertir une chaîne ou un nombre en booléen](#)

Parfois, il peut être pratique lorsqu'il y a des bogues dans le code de vérifier le type d'une variable. Parfois, il faut pouvoir changer le type d'une variable pour pouvoir effectuer des opérations dessus.

<https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Operators/typeof>

Trouver le type d'une variable

Pour trouver le type d'une variable, on utilise l'opérateur `typeof`. Ce n'est pas une fonction, mais un opérateur, donc on n'a pas besoin de mettre des parenthèses autour de la variable :

```
let nombre = 42;
let texte = "Bonjour";

console.log(typeof nombre); // Affiche 'number'

console.log(typeof texte); // Affiche 'string'
```

Changer le type d'une variable

Changer un nombre en chaîne de caractères

Pour changer un nombre en chaîne de caractères, on utilise la méthode `toString` :

```
let nombre = 42;

let texte = nombre.toString();

console.log(typeof texte); // Affiche 'string'
```

Changer une chaîne de caractères en nombre

Les informations saisies dans un formulaire sont toujours des chaînes de caractères. Pour changer une chaîne de caractères en nombre, on utilise la fonction `parseInt` ou `parseFloat`.

`parseInt` permet de transformer une chaîne de caractères en entier, tandis que `parseFloat` permet de transformer une chaîne de caractères en nombre à virgule flottante.

```
let texte = "42";

let nombre = parseInt(texte);

console.log(typeof nombre); // Affiche 'number'
```

Il est également possible de changer une chaîne de caractères en nombre en utilisant l'opérateur `+` :

```
let texte = "42";

let nombre = +texte;

console.log(typeof nombre); // Affiche 'number'
```

Changer une chaîne de caractères en tableau

Pour changer une chaîne de caractères en tableau, on utilise la méthode `split`. Il faut définir le séparateur qui sera utilisé pour découper la chaîne de caractères en tableau.

```
let texte = "pomme,banane,orange";

let tableau = texte.split(",");

console.log(tableau); // Affiche ['pomme', 'banane', 'orange']
```

Changer un tableau en chaîne de caractères

Pour changer un tableau en chaîne de caractères, on utilise la méthode `join`. Il faut définir le séparateur qui sera utilisé pour joindre les éléments du tableau en chaîne de caractères.

```
let tableau = ["pomme", "banane", "orange"];
```

Convertir un élément en booléen

Pour convertir un élément en booléen, on utilise la fonction `Boolean`. Cette fonction retourne `true` si l'élément est vrai, et `false` si l'élément est faux. Cependant, en Javascript, tous les éléments vrais et faux peuvent être un peu déroutants.

```
let texte = "Patate";

let bool = Boolean(texte);

console.log(typeof bool); // Affiche 'boolean'
```

Les éléments suivants retournent `false` avec la méthode `Boolean` :

- `false` - La valeur booléenne "false"
- `""` - chaîne vide
- `0` - Le nombre zéro
- `NaN` - La valeur NaN qui veut dire "Not a Number". C'est une valeur spéciale qui est retournée quand une opération mathématique ne peut pas être effectuée.
- `null`
- `undefined`

Tous les autres éléments retournent `true`.

- **TOUTES LES CHAINES DE CARACTERES RETOURNENT `true` SAUF `""`, incluant `"0"` et `"false"`**

Donc pour convertir la chaîne `"false"` en `false` ou la chaîne `"true"` en `true`, on utilise simplement une comparaison.

```
let texte = "false";
let bool = texte === "false"; //

console.log(bool); // bool est true car la comparaison est vraie
```