

# Tableaux

## Création de tableaux

Pour créer un tableau en JavaScript, vous pouvez utiliser l'une des deux notations suivantes :

```
// Notation avec crochets []
const fruits = ["pomme", "banane", "orange"];

const fruits = [];
// Notation avec le constructeur Array()
const numbers = new Array(1, 2, 3, 4, 5);
```

## Accès aux éléments du tableau

Les éléments d'un tableau sont accessibles via leur index, qui commence toujours à 0 pour le premier élément.

```
const fruits = ["pomme", "banane", "orange"];
console.log(fruits[0]); // Affiche "pomme"
console.log(fruits[1]); // Affiche "banane"
```

## Propriétés et méthodes des tableaux

JavaScript propose plusieurs propriétés et méthodes pour travailler avec des tableaux :

- `length` : La propriété `length` renvoie la longueur du tableau, c'est-à-dire le nombre d'éléments qu'il contient.

```
const fruits = ["pomme", "banane", "orange"];
console.log(fruits.length); // Affiche 3
```

- `push()` : La méthode `push()` ajoute un élément à la fin du tableau.

```
fruits.push("fraise");
console.log(fruits); // Affiche ["pomme", "banane", "orange", "fraise"]
```

- `pop()` : La méthode `pop()` supprime le dernier élément du tableau.

```
fruits.pop();
console.log(fruits); // Affiche ["pomme", "banane", "orange"]
```

- `unshift()` : La méthode `unshift()` ajoute un élément au début du tableau.

```
fruits.unshift("cerise");
console.log(fruits); // Affiche ["cerise", "pomme", "banane", "orange"]
```

- `shift()` : La méthode `shift()` supprime le premier élément du tableau.

```
fruits.shift();
console.log(fruits); // Affiche ["pomme", "banane", "orange"]
```

- `splice()` : La méthode `splice()` permet d'ajouter, de supprimer ou de remplacer des éléments du tableau à des positions spécifiques.

```
array.splice(start, deleteCount, item1, item2, ...)
```

`start` : L'indice où commencer la modification du tableau.

`deleteCount` : Le nombre d'éléments à supprimer à partir de `start`.

`item1, item2, ...` : Les éléments à ajouter au tableau, commençant à l'indice `start`.

```
fruits.splice(1, 1); // Supprime l'élément à l'indice 1 (banane)
console.log(fruits); // Affiche ["pomme", "orange"]
fruits.splice(1, 0, "cerise") // Ajoute l'élément à l'indice 1
console.log(fruits) // Affiche ["pomme", "cerise", "orange"]
```

- `concat()` : La méthode `concat()` fusionne deux tableaux en un nouveau tableau.

```
const moreFruits = ["fraise", "cerise"];
const allFruits = fruits.concat(moreFruits);
console.log(allFruits);
```

- `slice()` : La méthode `slice()` renvoie une copie superficielle (shallow copy) d'une partie du tableau.

```
const slicedFruits = fruits.slice(1, 2); // Extrait un sous-tableau [ "orange" ]
```

## Parcourir un tableau

Pour parcourir un tableau, vous pouvez utiliser des boucles `for`, `for...of`, ou les méthodes de tableau telles que `forEach`.

### Boucle `for`

```
for (let i = 0; i < fruits.length; i++) {
  console.log(fruits[i]);
}
```

### Boucle `for...of`

```
for (const fruit of fruits) {  
  console.log(fruit);  
}
```

## Tableaux multidimensionnels

Un tableau multidimensionnel est un tableau contenant d'autres tableaux. Cela permet de créer des structures de données plus complexes, comme des tableaux de tableaux.

```
const matrix = [  
  [1, 2, 3],  
  [4, 5, 6],  
  [7, 8, 9]  
];  
console.log(matrix[0][1]); // Accès à l'élément à la ligne 0, colonne 1 (2)
```

## Tableaux de chaînes de caractères

Les tableaux de chaînes de caractères peuvent être très utiles pour la manipulation de texte. Vous pouvez utiliser des méthodes de chaînes de caractères telles que `join()`, `split()`, `substring()`, `replace()`, etc., en combinaison avec des tableaux de chaînes de caractères pour effectuer des opérations de traitement de texte.

```
const sentence = "Ceci est une phrase."  
const words = sentence.split(" "); // Divise la phrase en mots  
console.log(words); // Affiche ["Ceci", "est", "une", "phrase."]  
  
const reversedSentence = words.reverse().join(" "); // Inverse les mots et les rejoint en une phrase  
console.log(reversedSentence); // Affiche "phrase. une est Ceci"
```

### ▼ Demo

```
// Créer un tableau de nombres  
const numbers = [1, 2, 3, 4, 5];  
  
// Afficher le tableau d'origine  
console.log("Tableau d'origine : " + numbers);  
  
// 1. Utiliser une boucle pour doubler chaque élément du tableau  
for (let i = 0; i < numbers.length; i++) {  
  numbers[i] *= 2;  
}
```

```
// Afficher le tableau après avoir doublé chaque élément
console.log("Tableau après avoir doublé chaque élément : " + numbers);

// 2. Utiliser une boucle pour calculer la somme de tous les éléments du tableau
let sum = 0;
for (let i = 0; i < numbers.length; i++) {
    sum += numbers[i];
}

// Afficher la somme des éléments
console.log("Somme des éléments : " + sum);

// 3. Utiliser une boucle pour trouver le plus grand élément du tableau
let max = numbers[0];
for (let i = 1; i < numbers.length; i++) {
    if (numbers[i] > max) {
        max = numbers[i];
    }
}

// Afficher le plus grand élément
console.log("Le plus grand élément : " + max);
```