



# JavaScript - Les tableaux

[Mise à jour le 25/9/2020]

- **Sources** et compléments sur **MDN Web Docs**
  - [Les tableaux](#)
- **Lectures connexes**
  - Wikis WebPEM : ["Préparer un projet de site Web avec l'IDE VSCode"](#)

Adresse	Valeur
1600	10
1601	23
1602	505
1603	8



En JavaScript, les tableaux sont des **listes** qui contiennent des valeurs de **différents types de données**.

Ils permettent de rassembler **différentes informations** sous **une seule variable**.

## 1. Créer un tableau



Pour **initialiser** un tableau, il faut insérer des crochets **[ ]**.

Exemples

\*.js

```
// Tableau vide
// Première forme de déclaration
var messages = new Array();
// Deuxième forme de déclaration
var smartphones=[];

// Tableau initialisé
// Avec deux éléments
var messages = new Array("Bonjour","Hello");

// Avec trois éléments.
```

```
var smartphones=["GalaxyS8", "iPhone7", "Honor9"];  
// Forme d'écriture lorsque la liste est longue  
var smartphones=[  
  "GalaxyS8",  
  "iPhone7",  
  "Honor9"  
];  
  
smartphones; // résultat dans la console : Array(3) ["iPhone8",  
  "iPhone7", "Honor9"]
```

## 2. Dimensionner un tableau



Il n'est **pas nécessaire de préciser la taille** d'un tableau lors de sa déclaration.

Si on souhaite le faire on écrira :

Exemples

\*.js

```
var messages = new Array(5);  
var smartphones=[5];
```

## 3. Accéder aux éléments d'un tableau



Pour accéder à un élément d'un tableau, il suffit de placer son **indice** (sa position dans le tableau) entre crochets à la suite du nom de la variable. **Le premier élément a pour indice 0.**

Exemples

\*.js

```
var smartphones=["GalaxyS8", "iPhone7", "Honor9"];  
smartphone[0]; // résultat dans la console: "GalaxyS8"
```

## 4. Attribuer des valeurs aux éléments d'un tableau

## Exemples

\*.js

```
var smartphones = [];  
smartphones[1] = "iPhone4"; // résultat dans la console: "iPhone4"  
// remplace "iPhone7"  
smartphones[33] = "iphone6";  
smartphones; // résultat dans la console: Array(34) [undefined × 1,  
"iPhone4", undefined × 31, "iphone6"]
```

## 5. Mélanger des éléments de type différent dans un tableau

Tous les éléments d'un tableau ne sont pas obligatoirement de même type (**un tableau est une liste en JavaScript**). Il est également possible de placer un tableau dans un autre.

### Exemples

\*.js

```
var voitures = [ // Tableau à une dimension  
  "Peugeot", 10,  
  "Renault", 5,  
  "Ford", 15  
];  
  
var voitures = [ // Tableau à deux dimensions  
  ["Peugeot", "3008", 10],  
  ["Renault", "Captur", 5],  
  ["Ford", "C-Max", 15]  
];  
  
voitures[2]; // résultat dans la console : (3) ["Ford", "C-Max", 15]  
  
voitures[2][1];  
// résultat dans la console: "C-Max" (l'accès à un élément nécessite  
// une deuxième paire de crochets)
```

## 6. Manipuler des tableaux

En JavaScript, il existe des **propriétés** et des **méthodes** pour manipuler facilement des tableaux. Les propriétés fournissent des informations au sujet des tableaux et les méthodes permettent de les modifier ou de les renvoyer.

## 6.1 Connaître la longueur d'un tableau



Il suffit de **lier** la propriété **length** à la variable du tableau.

### Exemples

\*.js

```
var voitures = [           // Tableau à une dimension
  "Peugeot", 10,
  "Renault", 5,
  "Ford", 15
];

voitures.length; // résultat dans la console : 6
```



## 6.2 Insérer des éléments dans un tableau

- À la fin du tableau



On utilise la méthode **push(éléments(s))** pour ajouter un ou plusieurs éléments **à la fin d'un tableau**.

### Exemples

\*.js

```
var voitures = [           // Tableau à une dimension
  "Peugeot", 10,
  "Renault", 5,
  "Ford", 15
];
voitures.push("Citroën", 3);

voitures; // résultat dans la console : Array(8) ["Peugeot", 10,
"Renault", 5, "Ford", 15, "Citroën", 3]
```

- **Au début** du tableau



On utilise la méthode **unshift**(éléments(s)) pour ajouter un ou plusieurs éléments **au début** d'un tableau.

\*.js

```
var voitures = [           // Tableau à une dimension
  "Peugeot", 10,
  "Renault", 5,
  "Ford", 15
];
voitures.unshift("Citroën",3);

voitures; // résultat dans la console : Array(8) ["Citroën", 3,
"Peugeot", 10, "Renault", 5, "Ford", 15]
```



### 6.3 Supprimer des éléments dans un tableau

- **A la fin** du tableau



On utilise la méthode **pop()** pour enlever un élément **à la fin** d'un tableau.

Exemples

\*.js

```
var voitures = [           // Tableau à une dimension
  "Peugeot", 10,
  "Renault", 5,
  "Ford", 15,
  "Citroën", 3
];

voitures.pop(); // résultat dans la console : Array(7) ["Peugeot", 10,
"Renault", 5, "Ford", 15, "Citroën"]
voitures.pop(); // résultat dans la console : Array(6) ["Peugeot", 10,
"Renault", 5, "Ford", 15]
```

- **Au début** du tableau



On utilise la méthode **shift()** pour enlever un élément **au début** d'un tableau.

### Exemples

\*.js

```
var voitures = [           // Tableau à une dimension
  "Citroën", 3,
  "Peugeot", 10,
  "Renault", 5,
  "Ford", 15,
];

voitures.shift(); // résultat dans la console : Array(7) [3, "Peugeot",
10, "Renault", 5, "Ford", 15]
voitures.shift(); // résultat dans la console : Array(6) ["Peugeot",
10, "Renault", 5, "Ford", 15]
```

## 6.4 Concaténer des tableaux



JavaScript donne la possibilité d'associer deux tableaux pour en faire un nouveau. On utilise la méthode **concat(tableau)**.

### Exemples

\*.js

```
var marquesv1 = [           // Tableau à une dimension
  "Citroën", 3,
  "Peugeot", 10
];
// Tableau à une dimension
var marquesv2 = [
  "Renault", 5,
  "Ford", 15
];

marquesv1.concat(marquesv2);
// résultat dans la console : Array(8) ["Citroën", 3, "Peugeot", 10,
"Renault", 5, "Ford", 15]
```



Il est possible de concaténer plusieurs tableaux en séparant leur nom par des virgules : `tableau1.concat(tableau2, tableau3, etc.)`

## 6.5 Trouver la position d'un élément particulier dans un tableau



On utilise la méthode **`indexOf(element)`**.

### Exemples

\*.js

```
var marquesv1 = [           // Tableau à une dimension
  "Citroën", 3,
  "Peugeot", 10
];

marquesv1.indexOf("Peugeot"); // résultat dans la console : 2
```

## 6.6 Convertir un tableau en une chaîne de caractères



Avec la méthode **`join()`**, on peut concaténer tous les éléments d'un tableau

### Exemples

\*.js

```
var prenom = ["Vincent", "Benoît", "Georges"];
prenom.join(); // résultat dans la console : "Vincent,Benoît,Georges"
```

## 6.7 Parcourir un tableau avec une boucle for

Pour parcourir un tableau, il faut faire autant d'**itérations** qu'il y a d'**items**. Le nombre d'items d'un tableau se récupère avec la propriété **`length`**, exactement comme pour le nombre de caractères d'une chaîne.

À chaque itération, on va avancer d'un item dans le tableau, en utilisant la variable de boucle `i`. Comme elle s'incrémente, cela permet d'avancer dans le tableau item par item.

### Exemples

\*.js

```
var smartphones=["GalaxyS8", "iPhone7", "Honor9"];

for (var i = 0; i < smartphones.length; i++) {

    console.log(smartphones[i]);
}
// résultat dans la console :
"GalaxyS8"
"iPhone7"
"Honor9"
```

## 7. Tableau associatif

- Lien à consulter sur XUL : [Tableau associatif en JavaScript](#)

## 8. pour aller plus loin...

[MDN web docs Mozilla](#)

From:  
<https://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:  
<https://webge.fr/dokuwiki/doku.php?id=web:javascript:fondamentaux:tableaux>

Last update: **2021/08/11 10:58**

