

JavaScript Concepts →

[Documents](#) / [Javascript](#) / [Tableaux](#)

Tableaux

Les tableaux sont des listes de données ordonnées et stockées. Ils peuvent contenir des éléments de n'importe quel type de données.

Syntaxe

Les tableaux peuvent être créés en utilisant des crochets, avec des éléments individuels séparés par des virgules.

```
// Un tableau contenant des nombres
const nombreTableau = [ 0 , 1 , 2 , 3 ] ;

// Un tableau contenant différents types de données
const tableaumixte = [ 1 , 'poulet' , faux ] ;
```

Une autre façon de créer une nouvelle instance de la `Array` classe et de transmettre les éléments sous la forme d'une liste d'arguments séparés par des virgules :

```
const classArray = new Array ( 0 , 1 , 2 , 3 ) ;
```

Création d'un tableau

La façon la plus simple de créer un tableau consiste à utiliser un littéral de tableau comme dans l'exemple ci-dessus.

```
const fruit = [ 'Pomme' , 'Orange' , 'Banane' ] ;
```

Les espaces blancs sont ignorés et les déclarations peuvent s'étendre sur plusieurs lignes.

Les tableaux peuvent également être créés avec la `Array` classe en utilisant le `new` mot-clé et en

passant les éléments comme arguments :

```
const fruit = new Array ( 'Pomme' , 'Orange' , 'Banane' ) ;
```

Cependant, comme l'effet est exactement le même, il est préférable d'utiliser la méthode littérale pour plus de simplicité et de rapidité d'exécution.

Accéder aux éléments d'un tableau

Les éléments du tableau sont classés par valeurs d'index, en commençant à 0 :

- L'index 0 a le premier élément.
- L'indice 1 a le deuxième élément.
- L'indice 2 contient le troisième élément.
- L'indice n-1 a le nième élément.

Les éléments individuels du tableau sont accessibles à l'aide du nom du tableau et de l'index de l'élément entouré de crochets.

Ce code accède à la valeur du premier élément du `fruit` tableau :

```
const fruit = [ 'Pomme' , 'Orange' , 'Banane' ] ;

consoler . bûche ( fruit [ 0 ] ) ;
// Sortie : Apple
```

Les éléments du tableau peuvent être modifiés en accédant à l'élément et en lui attribuant une nouvelle valeur.

```
const fruit = [ 'Pomme' , 'Orange' , 'Banane' ] ;

fruit [ 1 ] = 'Mangue' ;

consoler . bûche ( fruit [ 1 ] ) ;
// Sortie : Mangue
```

L'accès à un tableau à l'aide d'un index inutilisé renverra `undefined` . Cependant, une nouvelle valeur peut toujours être affectée à un index inutilisé d'un tableau. Ce faisant, tout écart dans les indices attribués restera `undefined` .

```
const fruit = [ 'Pomme' , 'Orange' , 'Banane' ] ;

fruit [ 4 ] = 'Mangue' ; // une affectation valide

consoler . bûche ( fruit [ 3 ] ) ;
// Sortie : non définie
```

Tableaux imbriqués

Tout objet peut être un élément d'un tableau, y compris d'autres tableaux. Les tableaux avec un ou plusieurs tableaux comme éléments sont appelés « tableaux imbriqués ». Semblable à l'accès aux éléments d'un tableau normal, l'accès aux éléments dans les tableaux imbriqués nécessite des indices supplémentaires pour référencer les éléments du tableau interne.

```
// Crée un tableau imbriqué
const nourriture = [
  [ 'Pomme' , 'Orange' , 'Banane' ] ,
  [ 'Fraise' , 'Myrtille' , 'Framboise' ] ,
  [ 'Pomme de terre' , 'Carotte' , 'Brocoli' ] ,
] ;

// Récupère la chaîne alimentaire à l'index 2 du tableau à l'index 1
consoler . bûche ( nourriture [ 1 ] [ 2 ] ) ;
// Sortie : Framboise

// Récupère la chaîne alimentaire à l'index 1 du tableau à l'index 0
consoler . log ( nourriture [ 0 ] [ 1 ] ) ;
// Sortie : Orange
```

Le même processus s'applique aux tableaux imbriqués qui contiennent eux-mêmes des tableaux imbriqués. Plus le tableau est "imbriqué", plus il [] faut d'indices ou de paires de crochets pour accéder à leurs éléments.

Code	Output
<pre>1 // More levels of nesting 2 const nestedArray = [3 [4 [1, 2], 5 [3, 4], 6 [5, 6], 7], 8 [9 ['A', 'B', 'C'], 10 ['D', 'E', 'F'], 11], 12]; 13</pre>	
Run	

Tableaux

`.concat()`

Fusionne ou concatène deux tableaux ou plus.

`.filtre()`

Crée un nouveau tableau avec tous les éléments qui réussissent le test à partir de la fonction fournie.

`.trouver()`

Renvoie le premier élément du tableau qui satisfait la fonction donnée.

`.findIndex()`

Renvoie le premier index qui réussit le test de la fonction de rappel. Renvoie -1 si aucun élément ne passe le test.

`.pour chaque()`

Boucle sur le tableau, en passant chaque élément du tableau dans la fonction de rappel fournie.

`.Indice de()`

Renvoie le premier index auquel un élément peut être trouvé. Renvoie -1 si l'élément n'est pas trouvé.

`.isArray()`

Renvoie vrai pour les tableaux, sinon faux.

`.rejoindre()`

Les éléments d'un tableau sont convertis en chaînes et concaténés ensemble, renvoyant la chaîne résultante.

`.lastIndexOf()`

Renvoie le dernier index auquel un élément peut être trouvé. Renvoie -1 si l'élément n'est pas trouvé.

`.longueur`

Renvoie le nombre spécifique d'éléments dans le tableau.

`.carte()`

Crée un nouveau tableau avec les résultats de l'appel d'une fonction pour chaque élément du tableau.

`.pop()`

Supprime le dernier élément d'un tableau, décrémente la longueur du tableau et renvoie la valeur qu'il a supprimée.

`.pousser()`

Ajoute un ou plusieurs éléments à la fin du tableau et renvoie une nouvelle longueur.

`.réduire()`

Combine chaque élément d'un tableau, à l'aide d'une fonction de réduction que vous spécifiez, et renvoie une valeur unique.

`.inverser()`

Inverse l'ordre des éléments d'un tableau en place et renvoie le tableau inversé.

`.changement()`

Supprime et renvoie le premier élément du tableau. Tous les éléments suivants se déplaceront vers le bas d'une place.

`.tranche()`

Renvoie une copie superficielle d'une partie du tableau, tandis que le tableau d'origine n'est pas modifié.

`.sort()`

Trie les éléments d'un tableau en place et renvoie le tableau trié.

`.épissure()`

Modifie le tableau en insérant, supprimant et/ou remplaçant des éléments du tableau puis renvoie un tableau d'éléments supprimés.

`.toString()`

Renvoie une chaîne avec chacune des valeurs du tableau, séparées par des virgules. Ne modifie pas le tableau d'origine.

`.unshift()`

Ajoute un ou plusieurs éléments au début du tableau et renvoie une nouvelle longueur.

`.valeur de()`

Renvoie la valeur de tous les éléments du tableau d'origine.

[Edit this page on GitHub](#)

Contributors



@StevenSwiniarski



@Christine_Yang

Vous souhaitez participer à la création de Docs ? Lisez le [Guide de contribution](#) ou partagez vos [commentaires](#).

Apprendre JavaScript sur Codecademy

Cours


Apprendre JavaScript

○ ○ Convient aux débutants, 11 Cours

Cheminement De Carrière

Ingénieur Front-End

○○ Convient aux débutants, **130** Cours

 Retour au sommet