



JavaScript, les tableaux

Alvin Berthelot

Version 1.0.0



Ce(tte) œuvre est mise à disposition selon les termes de la Licence Creative Commons.

Attribution - Partage dans les Mêmes Conditions 3.0 non transposé.



La licence, ses explications ainsi que les moyens de contribution et réappropriation sont détaillés à la fin.

Les tableaux

Définition d'un tableau

Rappel

En JavaScript, un tableau n'est pas un type de valeur en soit.

Même si il s'agit d'un objet, la déclaration courante s'effectuera avec une notation concise ("array litteral") avec des `[]`.

```
var myAnimals = ['Grumpy', 'Droopy'];  
console.log(myAnimals[1]); // Droopy
```



L'index d'un tableau commence à 0.



Un tableau est dynamique, il n'est donc pas nécessaire de spécifier le nombre d'éléments.

Le contenu des tableaux

Les tableaux sont des objets particuliers héritant de `Array.prototype` et donc disposant de propriétés et méthodes spécifiques.

Rappel

Les objets en JavaScript sont des ensembles de propriétés, elles-mêmes définies comme des paires clé/valeur où la clé est une chaîne de caractère et la valeur, toute valeur possible en JavaScript.

Il est donc tout à fait possible qu'un tableau contienne des fonctions.

```
var helloAnimals = [  
  function() {  
    console.log('Hello Grumpy');  
  },  
  function() {  
    console.log('Hello Droopy');  
  }  
];  
helloAnimals[0](); // Hello Grumpy
```

Les propriétés et méthodes

La propriété `length`

Contrairement aux autres langages Il ne s'agit pas de la limite supérieure. La propriété `length` retourne la valeur du plus grand positionnement +1 au sein du tableau.

```
var animals = [];  
  
console.log(animals.length); // 0  
  
animals[41] = 'Grumpy';  
console.log(animals.length); // 42
```

La propriété `length` peut être écrasée. La spécifier peut tronquer le tableau si la nouvelle valeur est inférieure à la valeur courante.

```
var animals = ['Grumpy', 'Droopy', 'Nemo'];  
animals.length = 2;  
console.log(animals); // ['Grumpy', 'Droopy']
```

Méthodes mutateurs

Les tableaux disposent d'un jeu de méthodes pour manipuler son contenu, voici les plus intéressantes :

- `push()` : ajoute un élément au tableau (et renvoie sa nouvelle longueur).
- `shift()` : supprime et renvoie le premier élément d'un tableau.
- `sort()` : trie le contenu d'un tableau.

```
var animals = ['Grumpy', 'Droopy', 'Nemo'];  
var cat = animals.shift();  
console.log(animals); // ['Droopy', 'Nemo']
```



Ces méthodes modifient l'état du tableau.

Méthodes accesseurs

Les tableaux disposent d'un jeu de méthodes pour retourner une représentation de son contenu, voici les plus intéressantes :

- `indexOf()` : retourne le premier index d'un élément.
- `includes()` : détermine si élément est présent.
- `concat()` : concatène des tableaux.
- `join()` : crée une chaîne de caractères à partir d'un tableau.
- `slice()` : crée une copie d'une portion d'un tableau.

```
var animals = ['Grumpy', 'Droopy', 'Nemo'];  
var index = animals.indexOf('Nemo');  
console.log(index); // 2
```

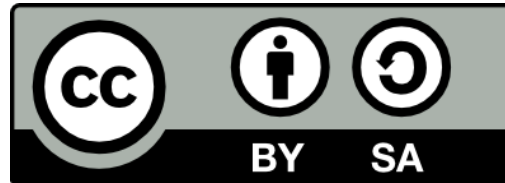
Méthodes itératives

Les tableaux disposent d'un jeu de méthodes pour itérer sur ses éléments et les transformer, voici les plus intéressantes :

- `map()` : applique une transformation sur chaque élément.
- `reduce()` : applique une fonction sur un accumulateur et sur chaque élément.
- `filter()` : filtre les éléments d'un tableau selon un prédicat.
- `forEach()` : appelle une fonction sur chaque élément.

```
var animals = ['Grumpy', 'Droopy', 'Poppy'];
animals.forEach(function(animal) {
  console.log(animal);
});
```

Licence



CC BY-SA 3.0

Ce(tte) œuvre est mise à disposition selon les termes de la Licence Creative Commons. Attribution - Partage dans les Mêmes Conditions 3.0 non transposé.

Copyright © 2017 [Alvin Berthelot](#).

Pour toutes questions, réclamations ou remarques, merci d'envoyer un message à alvin.berthelot@webyousoon.com.

Explications licence CC BY-SA 3.0

Cette licence permet aux autres de remixer, arranger, et adapter votre œuvre, même à des fins commerciales, tant qu'on vous accorde le mérite en citant votre nom et qu'on diffuse les nouvelles créations selon des conditions identiques.

Cette licence est souvent comparée aux licences de logiciels libres, "open source" ou "copyleft".

Toutes les nouvelles œuvres basées sur les vôtres auront la même licence, et toute œuvre dérivée pourra être utilisée même à des fins commerciales.

C'est la licence utilisée par Wikipédia ; elle est recommandée pour des œuvres qui pourraient bénéficier de l'incorporation de contenu depuis Wikipédia et d'autres projets sous licence similaire.

Contribution et réappropriation

Ce fichier PDF est généré avec [Asciidoctor](#) à partir d'un dépôt Git se trouvant sous GitHub.

<https://github.com/alvinberthelot/slides-js>

Cela signifie que vous n'avez pas besoin de vous battre avec un fichier binaire (le PDF) pour **contribuer**, **vous réapproprier le contenu** ou **modifier le thème** de présentation.



Contribution

Vous voulez **contribuer au contenu** car :

- Il y a une erreur (ça arrive à tout le monde), de typographie, de compréhension, ou tout autre chose.
- Vous souhaitez apporter une précision.

Il vous suffit de [contribuer au projet via Git](#) par le moyen d'une "pull request" sur le [dépôt Git](#).



Réappropriation



N'oubliez pas les conditions de la licence.

Vous voulez vous **réapproprier le contenu** car :

- Vous souhaitez donner un style différent.
- Vous souhaitez enlever/ajouter/modifier des sections dans votre contexte.

Il vous suffit de "forker" le [dépôt Git](#) et d'y apporter vos propres modifications, puis de générer par vous même le nouveau PDF.

