

JS ARRAY OBJECT : Les iterators (SeA 28/02/2019)

Code commun pour les tests :

```
<div> 1</div>
<div>10</div>
<div>18</div>
<div> 1</div>

<script>
  var liste = document.querySelectorAll('div'); //tableau de div
</script>
```

NOTE : Dans les exemples ci-dessous, on utilise la méthode compacte avec les *arrow functions* !

The `forEach()` method calls a function (a callback function) once for each array element.

Dans l'exemple ci-dessous, on associe à chaque `<div>` un évènement 'click' qui incrémente la valeur:

```
liste.forEach( (value) => { value.addEventListener('click', (e) => { e.target.innerHTML ++;}); });
// forEach connaît 3 arguments dans cet ordre : value, index, array
```

Dans l'exemple suivant, on crée un tableau (contenu) à partir du contenu des `<div>` :

```
var contenu = []; //tableau vide
liste.forEach( (value, index) => { contenu[index] = value.innerHTML ;} );
// contenu : 1, 10, 18, 1
```

The `filter()` method creates a new array with array elements that passes a test.

On crée un autre tableau (liste2) avec uniquement les valeurs `>3` du tableau « contenu » précédent :

```
var liste2 = contenu.filter( (value) => { return value > 3; } ); // liste2 : 10, 18
// filter connaît 3 arguments dans cet ordre : value, index, array
```

The `map()` method creates a new array by performing a function on each array element.

On crée un autre tableau (liste3) avec les valeurs `x2` du tableau « contenu » précédent :

```
var liste3 = contenu.map( (value) => { return value * 2; } ); // liste3 : 2, 20, 36, 2
// map connaît 3 arguments dans cet ordre : value, index, array
```

The `some()` method check if some array values pass a test.

```
var res = contenu.some( (value) => { return value > 5; } );
console.log(res); // Renvoie TRUE car des valeurs sont > 5
// some connaît 3 arguments dans cet ordre : value, index, array
```

The `every()` method check if all array values pass a test.

```
var res = contenu.every( (value) => { return value > 5; } );
console.log(res); // Renvoie FALSE car il y a des valeurs qui ne sont pas > 5
// every connaît 3 arguments dans cet ordre : value, index, array
```

The `reduce()` method runs a function on each array element to produce (reduce it to) a single value.

```
var res = contenu.reduce( (somme, value) => { return parseInt(somme) + parseInt(value); } );
console.log(res); // Affiche 30, c-à-d la somme de chaque valeur
// reduce connaît 4 arguments dans cet ordre : total (initial/précédente), value, index, array
```

Voir aussi : *reduceRight*

The `indexOf()` method searches an array for an element value and returns its position.

```
console.log( contenu.indexOf('18') ); // Affiche 2, car 18 est à l'index 2 (en 3e position)
```

Voir aussi : *lastIndexOf*

The `find()` method returns the value of the first array element that passes a test function.

```
var res = contenu.find( (value) => { return value > 5; } );
console.log(res); // Affiche 10 qui est la première valeur du tableau à être > 5
```

Voir aussi *findIndex*