

L'objet Array

Les propriétés de l'objet *Array*

Propriété	description
length	Contient le nombre d'éléments du tableau.
input	Cette propriété permet de faire une recherche dans le tableau à l'aide d'une expression régulière
prototype	Cette propriété permet d'ajouter des propriétés personnalisées à l'objet.

Les méthodes standards de l'objet *Array*

Méthode	description
concat (tab1, tab2[, tab3, ...])	Permet de concaténer plusieurs tableaux, c'est-à-dire de créer un tableau à partir de plusieurs tableaux.
Tableau.join (",")	Renvoie une chaîne de caractères contenant tous les éléments du tableau séparés par le séparateur mis en paramètre.
Tableau.pop ()	Supprime le dernier élément du tableau et retourne sa valeur.
Tableau.shift ()	Supprime le premier élément du tableau.
Tableau.push (valeur1, valeur2, ...)	Ajoute un ou plusieurs éléments en fin de tableau et retourne la nouvelle taille du tableau..

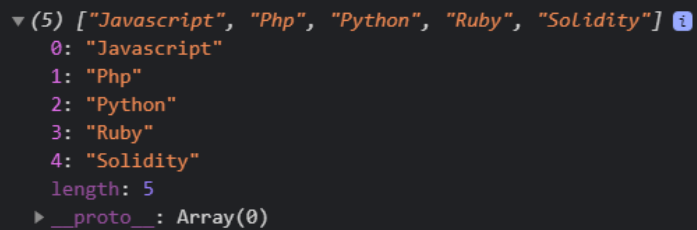
Tableau.unshift(valeur1, valeur2, ...)	Ajoute un ou plusieurs éléments en début de tableau et retourne la nouvelle taille du tableau.
Tableau.splice(debut,nbr_elements,"new_1","new_2")	Ajoute/retire des éléments d'un tableau en fonction de la position indiqué.
Tableau.slice(debut , fin)	Renvoie un tableau contenant une partie (extraction) des éléments d'un tableau. Le paramètre fin est exclu
Tableau.sort()	trie les éléments d'un tableau.
Tableau.includes(valeur recherché,position)	Permet de savoir si un élément se trouve dans un le tableau elle recois en parametre l'element recherché et la position dans le tableau a partir de laquelle la recherche commence, elle retourne true ou false.
Tableau.toString()	Renvoie la chaîne de caractères correspond à l'instruction qui a permis de créer l'objet Array.
Tableau.indexOf(valeur,position)	Retourne la position de l'élément mis en paramètre la recherche commence en fonction du deuxième paramètre si ce deuxième paramètre est omis la recherche commence en début de tableau si l'élément n'est pas trouvé elle retourne -1.
Tableau.lastIndexOf(valeur,position)	similaire a indexOf la recherche se faisant a partir de la dernière position.
Tableau.valueOf	Retourne tout simplement la valeur de l'objet Array auquel elle fait référence.
Tableau.reverse()	Inverse l'ordre des éléments du tableau

Exemples

```
let array3 = ["Javascript", "Php", "Python"];
let array4 = ["Ruby", "Solidity"];
```

concat()

```
let newArray = array3.concat(array4);
```

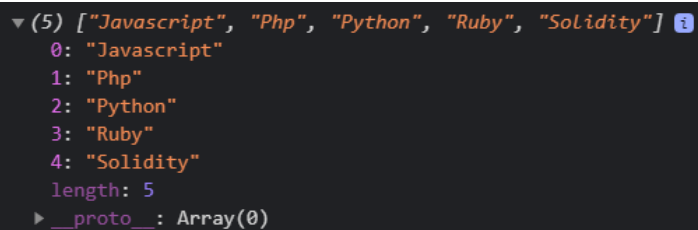


```
▼ (5) ["Javascript", "Php", "Python", "Ruby", "Solidity"] ⓘ
  0: "Javascript"
  1: "Php"
  2: "Python"
  3: "Ruby"
  4: "Solidity"
  length: 5
  ▶ __proto__: Array(0)
```

Equivalent à concat()

```
let newArray = [...array3, ...array4];
```

... permet de casser le tableau et fusionner avec les autres tableaux en paramètres.



```
▼ (5) ["Javascript", "Php", "Python", "Ruby", "Solidity"] ⓘ
  0: "Javascript"
  1: "Php"
  2: "Python"
  3: "Ruby"
  4: "Solidity"
  length: 5
  ▶ __proto__: Array(0)
```

.join()

```
console.log(array3.join(' -> '));
```

```
Javascript -> Php -> Python
```

```
console.log(array3.join(' '));
```

```
Javascript Php Python
```

.slice()

```
console.log(array3);  
console.log(newArray);
```

```
▶ (3) ["Javascript", "Php", "Python"]
```

```
▶ (5) ["Javascript", "Php", "Python", "Ruby", "Solidity"]
```

```
console.log(array3.slice(1));
```

-> L'extraction commence à l'index 1 jusqu'au dernier index inclus

```
▶ (2) ["Php", "Python"]
```

```
console.log(newArray.slice(1, 4));
```

-> L'extraction commence à l'index 1 et s'arrete à l'index 4 (index 4 exclu)

```
▶ (3) ["Php", "Python", "Ruby"]
```

.indexOf() et .forEach()

```
console.log(array3.indexOf("Python"));
```

-> donne l'index de « python » soit 2

```
array3.forEach((languages) => console.log(languages));
```

-> appliques un forEach au tableau 3 et renvoi les valeurs de chaque index

```
2
```

```
Javascript
```

```
Php
```

```
Python
```

.every() et .some()

```
console.log(array3.every((language) => language == "Php"));  
-> renvoi false car tous les index ne sont pas « php »
```

```
console.log(array3.some((language) => language == "Php"));  
-> renvoi true car aux moins un index contient « php »
```

.shift()

```
let shift = array3.shift();  
-> supprime le premier élément du tableau  
console.log(array3);
```

```
▶ (2) ["php", "Python"]
```

.pop()

```
console.log(array3.pop());  
-> supprime le dernier élément du tableau et renvoi sa valeur
```

```
Python
```

.splice()

```
.splice(index de depart, nb à supprimer, élément à ajouter);
```

```
var mesPoissons = ["scalaire", "clown", "mandarin", "chirurgien"];
```

```
// supprime 0 élément à partir de l'index 2, et insère "tambour"  
var enleves = mesPoissons.splice(2, 0, "tambour");  
// mesPoissons est ["scalaire", "clown", "tambour", "mandarin", "chirurgien"]  
// enleves est 0, aucun élément supprimé
```

```
// supprime 1 élément à partir de l'index 3  
enleves = mesPoissons.splice(3, 1);  
// mesPoissons est ["scalaire", "clown", "tambour", "chirurgien"]  
// enleves est ["mandarin"]
```

```
// supprime 1 élément à partir de l'index 2, et insère "trompette"  
enleves = mesPoissons.splice(2, 1, "trompette");
```

```
// mesPoissons est ["scalaire", "clown", "trompette", "chirurgien"]
// enleves est ["tambour"]

// supprime 2 éléments à partir de l'index 0, et insère "perroquet", "anémone" et "bleu"
enleves = mesPoissons.splice(0, 2, "perroquet", "anémone", "bleu");
// mesPoissons est ["perroquet", "anémone", "bleu", "trompette", "chirurgien"]
// enleves est ["scalaire", "clown"]

// supprime 2 éléments à partir de l'indice 2
enleves = mesPoissons.splice(mesPoissons.length - 3, 2);
// mesPoissons est ["perroquet", "anémone", "chirurgien"]
// enleves est ["bleu", "trompette"]

var mesPoissons = ["perroquet", "anémone", "bleu", "trompette", "chirurgien"];
// on retire les éléments à partir de l'indice 2
enleves = mesPoissons.splice(2);
// mesPoissons vaut ["perroquet", "anémone"]
// enleves vaut ["bleu", "trompette", "chirurgien"]

var mesAnimaux = ["cheval", "chien", "chat", "dauphin"];
enleves = mesAnimaux.splice(-2, 1);
// mesAnimaux vaut ["cheval", "chien", "dauphin"]
// enleves vaut ["chat"]
```

.reduce()

Reduce permet de calculer toutes les valeurs du tableau

```
let arrayNumber = [4, 74, 28, 12, 1];
```

```
arrayNumber.reduce((x, y) => x + y);
```

donne 119

```
arrayNumber.push(17);
```

-> ajoute 17 dans le tableau

```
► (6) [4, 74, 28, 12, 1, 17]
```

filter()

permet de filtrer les éléments d'un tableau

```
let arrayNumber = [4, 74, 28, 12, 1, 17, 8];
```

```
arrayNumber.filter((number) => number > 10);  
donne (4) [74, 28, 12, 17]
```

sort()

permet de trier les éléments d'un tableau

```
arrayNumber.sort();  
sans autre précision la méthode tri selon la dizaine ce qui donne :  
(7) [1, 12, 17, 28, 4, 74, 8]
```

```
arrayNumber.sort((a, b) => a - b);  
en passant deux paramètres a et b et en indiquant a - b , le tri se fera  
par ordre croissant  
(7) [1, 4, 8, 12, 17, 28, 74]
```

```
arrayNumber.sort((a, b) => b - a);  
en passant deux paramètres a et b et en indiquant b - a , le tri se fera  
par ordre décroissant  
(7) [74, 28, 17, 12, 8, 4, 1]
```

map()