Trier un tableau Array en javascript

1. ## Objectifs
   -
2. ## Présentation
   - Les tableaux sont des éléments indispensables de la programmation dans tous les langages.
   - Un tableau est une structure ordonnée contenant des informations, accessibles et manipulables facilement.
   - La méthode "**sort**" est utilisable dans n'importe quel tableau crée avec Javascript. Elle permet tout simplement de trier le tableau en ordre croissant.
   - Il est possible de trier un tableau simple ou un tableau d'objets.
3. ## Trier un tableau Array à une dimensions en javascript
   - **Trie en ordre croissant**
     - L’ordre croissant est une disposition de nombres allant du **plus petit** au **plus grand**.

```
var etudiants =
['Mohamed','Adir','Salwa','Rotafi','Nejib','Salem','Tériza','Slah'];
etudiants.sort();
console.table(etudiants);
```

   - Ce qui affiche

(index)	Value
0	"Adir"
1	"Mohamed"
2	"Nejib"
3	"Rotafi"
4	"Salem"
5	"Salwa"
6	"Slah"
7	"Tériza"

apcpedagogie.com

4. **Trie en ordre décroissant**

   - L'ordre décroissant est une disposition de nombres allant du **plus grand** au **plus petit**.

```
var etudiants = ['Mohamed','Adir','Salwa','Rotafi','Nejib','Salem','Tériza','Slah'];
etudiants.reverse();
console.table(etudiants);
```

5. Utilisation des fonctions de comparaison pour trier

   - **La fonction `Array.sort(fonctionComparaison)`**

     - Le paramètre optionnel **`fonctionComparaison`** permet de spécifier une fonction définissant l'ordre de tri.
     - Si ce paramètre est absent, le tableau est trié selon la valeur de point de code Unicode de chaque caractère, d'après la conversion en chaine de caractères de chaque élément.
     - Par exemple, "banane" sera trié avant "cerise", mais "Cerise" arrivera avant "banane" à cause de la majuscule (les majuscules arrivent avant dans la liste).
     - Dans un tri numérique, 4 sera trié avant 30, mais comme ces nombres sont convertis en chaînes de caractères, "30" arrive avant "4" selon l'ordre des unités de code UTF-16.
     - Les éléments valant undefined sont placés à la fin du tableau.
     - Cette fonction prendra deux arguments : le premier élément à comparer et le deuxième élément à comparer.
     - Une fonction de comparaison aura généralement les formes suivantes :
     - **Pour les chaines de caractères**

```
function compare(a, b) {
  if (a est inférieur à b selon les critères de tri)
     return -1;
  if (a est supérieur à b selon les critères de tri)
     return 1;
  // a doit être égal à b
  return 0;
}
```

   - **Pour les nombres**

- Pour comparer des nombres plutôt que des chaînes, la fonction de comparaison peut simplement soustraire **b** à **a** (cela fonctionnera si le tableau ne contient pas **NaN** ou **Infinity**) :

```
function compareNombres(a, b) {
  return a - b;
}
}
//ou
nombres.sort((a, b) => a - b);
```

6. **Exemple:**

```
var stringArray = ["Mohamed", "Adir", "Salwa", "Rotafi", "Nejib", "Salem", "Tériza",
"Slah"];
var numericStringArray = ["30", "4", "500"];
var numberArray = [40, 1, 5, 200];
var mixedNumericArray = ["30", "4", "500", 40, 1, 5, 200];

function compareNombres(a, b) {
  return a - b;
}

console.log("Chaînes : " + stringArray.join() +"\n");
console.log("Triées : " + stringArray.sort() +"\n\n");

console.log("Nombres : " + numberArray.join() +"\n");
console.log("Triés sans fonction de comparaison : " + numberArray.sort() +"\n");
console.log("Triés avec compareNombres : " + numberArray.sort(compareNombres) +"\n\n");

console.log("Chaînes numériques : " + numericStringArray.join() +"\n");
console.log("Triées sans fonction de comparaison : " + numericStringArray.sort() +"\n");
console.log("Triées avec compareNombres : " + numericStringArray.sort(compareNombres)
+"\n\n");

console.log("Nombres et chaînes numériques : " + mixedNumericArray.join() +"\n");
console.log("Triés sans fonction de comparaison : " + mixedNumericArray.sort() +"\n");
console.log("Triés avec compareNombres : " + mixedNumericArray.sort(compareNombres)
+"\n\n");
```

apcpedagogie.com

7. Cet exemple produit la sortie suivante. Comme on peut le voir, lorsqu'on utilise la fonction de comparaison, les nombres sont triés correctement qu'ils soient des nombres ou des chaînes numériques.

```
Chaînes : Mohamed,Adir,Salwa,Rotafi,Nejib,Salem,Tériza,Slah
Triées : Adir,Mohamed,Nejib,Rotafi,Salem,Salwa,Slah,Tériza

Nombres : 40,1,5,200
Triés sans fonction de comparaison : 1,200,40,5
Triés avec compareNombres : 1,5,40,200

Chaînes numériques : 30,4,500
Triées sans fonction de comparaison : 30,4,500
Triées avec compareNombres : 4,30,500
```

8. **La fonction `String.localeCompare`**

   - Pour des chaines de caractères contenant des caractères non ASCII, c'est à dire des chaines de caractères contenant par exemple des accents (é, è, a, ä, etc.) : utilisez `String.localeCompare`. Cette fonction peut comparer ces caractères afin qu'ils apparaissent dans le bon ordre.

9. ## Trier un tableau Array à deux dimensions en javascript

   - La méthode javascript `sort()` permet de trier un **array** simple, mais peut prendre une function en paramètre de tri (elle sera nommée **trierParTexte** dans l'exemple en dessous).
   - La méthode `sort()` trie les éléments d'un tableau, dans ce même tableau, et renvoie le tableau. Par défaut, le tri s'effectue sur les éléments du tableau convertis en chaînes de caractères et triées selon les valeurs des unités de code UTF-16 des caractères.
   - Pour trier le tableau, nous devons indiquer à JavaScript comment trier. Cela provient d'une fonction qui renvoie **-1**, **0** ou **+1** en tant que valeur. **-1** mettra la première valeur avant la seconde dans le tableau trié, **+1** mettra la seconde valeur avant la première et **0** signifie que les valeurs sont égales. Par exemple, si nous voulons trier par les prénoms, voici une fonction qui peut être utilisée:

```
function trierParTexte( a, b )
{
    var x = a[0][0].toLowerCase();
    var y = b[0][0].toLowerCase();
```

```
    return ((x < y) ? -1 : ((x > y) ? 1 : 0));
}

etudiants.sort(trierParTexte);
console.table(etudiants);
```

10. **Activité**

    - Trier la table etudiants (le dernier résultat) en ordre décroissant (L'ordre décroissant est une disposition de nombres allant du **plus grand** au **plus petit**.) selon les notes obtenues
    - Trier la table etudiants (le dernier résultat) en ordre croissant (L'ordre croissant est une disposition de nombres allant du **plus petit** au **plus grand**.) selon les notes obtenues
    - **Ordre décroissant**

```
etudiants.sort(function(a, b){
    if(a[1] == b[1]){
        return a[2] - b[2];
    }
    return b[1] - a[1];
});
console.table(etudiants);
```

11. **Ordre croissant**

```
etudiants.sort(function(a, b){
    if(a[2] == b[2]){
        return a[1] - b[1];
    }
    return b[2] - a[2];
});
console.table(etudiants);
```

Sommaire du cours JavaScript

Source : https://apcpedagogie.com/trier-un-tableau-array-en-javascript · téléchargé le 07/11/2026 · © APC Pédagogie · apcpedagogie.com