How to use array sort in javascript

Posted on March 4, 2019 | by Prashant Yadav

Posted in Arrays, Javascript, Sorting

Learn how to use array sort method in javascript.

Sorting a list of data is one of the most common task we face while programming. But every time implementing the advance sorting algorithm is not feasible. So we have inbuilt method to help us out.

Javascript's `sort()` method can be used to sort an array of data.

Example

```
let arr = [1, 5, 4, 6, 7, 3 , 2];
arr.sort();
console.log(arr);
//[1, 2, 3, 4, 5, 6, 7]
```

`sort()` methods sorts the array inplace and return's the sorted array. By default it sorts in the ascending order. It sorts the original array no copy is made.

The time and space complexity of the `sort()` methods varies for different browsers and their implementation in javascript engines.

For example chrome uses two different sorting algorithms depending upon the input size.
`InsertionSort` if the array length is short (length <= 10) otherwise `QuickSort`.

Syntax

```
Array.prototype.sort([compareFunction]);
```

`compareFunction`: Optional

A function that defines the sort order. If not provided then the array will be sorted according to the each character's Unicode code point value or according to string conversion of each character.

The `compareFunction` takes two parameters. `firstElement` and `secondElement`.

```
Array.sort(function(a, b){
   return a - b;
});
```

How it works?

The `compareFunction` sort the array depending upon the return value it gets.
If we are comparing `firstElement` with the `secondElement` then

- If value less than `0` is returned then `firstElement` will come before the `secondElement`.
- If value greater than `0` is returned then `secondElement` will come before the `firstElement`.
- If `0` is returned then `firstElement` and `secondElement` will be unchanged. But they are sorted with other elements.
- It should not return inconsistent value for different pairs with similar element else the sorting order will be undefined.

```
let arr = [1, 3, 5, 2, 9, 11, 8, 4];

//Sort in ascending order
arr.sort((a, b) => {
   if(a < b){
      return -1;
   }else if(a > b){
      return 1;
   }else {
      return 0;
   }
});

console.log(arr);
//[1, 2, 3, 4, 5, 8, 9, 11]
```

Sorting array in ascending order

```
let arr = [1, 3, 5, 2, 9, 11, 8, 4];

//Sort in ascending order
arr.sort((a, b) => {
   if(a < b){
      return -1;
   }else if(a > b){
      return 1;
   }else {
      return 0;
   }
});
```

OR

```
arr.sort((a,b) => {
   return a - b;
});

//     OR
arr.sort((a, b) => a - b);
```

```
console.log(arr);
//[1, 2, 3, 4, 5, 8, 9, 11]
```

Sorting array in descending order

```
let arr = [1, 3, 5, 2, 9, 11, 8, 4];

//Sort in descending order
arr.sort((a, b) => {
   if(a < b){
      return 1;
   }else if(a > b){
      return -1;
   }else {
      return 0;
   }
});
```

OR

```
arr.sort((a,b) => {
   return b - a;
});

//     OR
arr.sort((a, b) => b - a);
```

```
console.log(arr);
//[11, 9, 8, 5, 4, 3, 2, 1]
```

Sorting a string

We can also sort the strings. It’s comparison will be done on the basis of sum of the ASCII characters of each words.

Sort string in ascending order

```
let arr = ['prashant', 'aman', 'yogesh', 'sachin', 'pranav'];

//sorting in ascending order
arr.sort((a, b) => {
   if(a < b){
      return -1;
   }else if(a > b){
      return 1;
   }else {
      return 0;
   }
});
```

```
console.log(arr);
//["aman", "pranav", "prashant", "sachin", "yogesh"]
```

Sort string in descending order

```
let arr = ['prashant', 'aman', 'yogesh', 'sachin', 'pranav'];

//sorting in descending order
arr.sort((a, b) => {
   if(a < b){
      return 1;
   }else if(a > b){
      return -1;
   }else {
      return 0;
   }
});
```

```
console.log(arr);
//["yogesh", "sachin", "prashant", "pranav", "aman"]
```

Sorting non-ASCII strings

we can use String's localCompare method to sort strings with accented characters(e, é, è, a, ä, etc).

```
let arr = ['réservé', 'premier', 'cliché', 'communiqué', 'café', 'adieu'];
arr.sort((a, b) => {
  return a.localeCompare(b);
});

console.log(arr);
//['adieu', 'café', 'cliché', 'communiqué', 'premier', 'réservé']
```

Sorting array of objects

We can also sort the array of objects with the `sort()` method in javascript.

Copy

```
let arr = [
  {
    name: "prashant",
    age: 23
  },
  {
    name: "aman",
    age: 24
  },
  {
    name: "yogesh",
    age: 24
  },
  {
    name: "sachin",
    age: 25
  },
  {
    name: "pranav",
    age: 22
  },
];

//Sort based on the age in ascending order
arr.sort((a, b) => {
  if(a.age < b.age){
    return -1;
  }else if(a.age > b.age){
    return 1;
  }else{
    return 0;
  }
});
```

```
console.log(arr);
/*
[
 {
    age: 22,
    name: "pranav"
 },
 {
    age: 23,
    name: "prashant"
 },
 {
    age: 24,
    name: "aman"

 },
 {
    age: 24,
    name: "yogesh"
 },
 {
    age: 25,
    name: "sachin"
 }
]

*/
```

Complex Array sorting

We can also perform complex or nested sorting with the `sort()` method.

We will sort the array of object in ascending order based on their age. If age is similar then we will sort it in descending order based on their names.

Copy

```
let arr = [
  {
    name: "prashant",
    age: 23
  },
  {
    name: "aman",
    age: 24
  },
  {
    name: "yogesh",
    age: 24
  },
  {
    name: "sachin",
    age: 25
  },
  {
    name: "pranav",
    age: 22
  },
];

//Sort in ascending based on their age
//If age are same then sort it in descending based on their names
arr.sort((a, b) => {
  if(a.age < b.age){
    return -1;
  }else if(a.age > b.age){
    return 1;
  }else{
    if(a.name < b.name){
      return 1;
    }else if(a.name > b.name){
      return -1;
    }else{
      return 0;
    }
  }
});
```

```
console.log(arr);
/*
[
 {
    age: 22,
    name: "pranav"
 },
 {
    age: 23,
    name: "prashant"
 },
  {
    age: 24,
    name: "yogesh"
 },
 {
    age: 24,
    name: "aman"

 },
 {
    age: 25,
    name: "sachin"
 }
]
*/
```

As `aman` and `yogesh` are having same age. we sort them in descending order based on their names. So `yogesh` came before `aman`.

Recommended Posts:

Convert entity relation array to object in Javascript

What is throttling in javascript?

Javascript function that returns sum of the previous values

Share data between two browser window with JavaScript

Memoize a function in JavaScript

Convert string to array in javascript

Polyfill for getElementByClassName()

Check if function is called with new keyword

Print all the quadruplets with given | 4 sum algorithm

How to check if given object is array in javascript

Prev Next

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Start typing...

Name*

Name

Email*

Email

POST COMMENT

Handcrafted with ♥somewhere in **Mumbai**