Definition and Usage

The sort() method sorts the items of an array.

The sort order can be either alphabetic or numeric, and either ascending (up) or descending (down).

By default, the sort() method **sorts the values** as strings in alphabetical and ascending order.

This works well for strings ("Apple" comes before "Banana"). However, if numbers are sorted as strings, "25" is bigger than "100", because "2" is bigger than "1".

Because of this, the sort() method will produce an incorrect result when sorting numbers.

You can fix this by providing a "compare function" (See "Parameter Values" below).

**Note:** This method changes the original array.

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *compareFunction* | Optional. A function that defines an alternative sort order. The function should return a negative, zero, or positive value, depending on the arguments, like:   * function(a, b){return a-b}   When the sort() method compares two values, it sends the values to the compare function, and sorts the values according to the returned (negative, zero, positive) value.  **Example:**  When comparing 40 and 100, the sort() method calls the compare function(40,100).  The function calculates 40-100, and returns -60 (a negative value).  The sort function will sort 40 as a value lower than 100. |

## Technical Details

|  |  |
| --- | --- |
| **Return Value:** | The Array object, with the items sorted |

// 1st example - order numbers ASCENDINGLY

var points = [40, 100, 1, 5, 25, 10];

let res = points.sort(function(a, b){return a-b});

console.log(res) // [1, 5, 10, 25, 40, 100]

// 2nd example - order numbers DESCENDINGLY

var points = [40, 100, 1, 5, 25, 10];

let res = points.sort(function(a, b){return b-a});

console.log(res)  // [100, 40, 25, 10, 5, 1]

// 3rd example - get the maximum number of this array

var points = [40, 100, 1, 5, 25, 10];

let res = points.sort(function(a, b){return b-a});

console.log(res[0])  // 100

const companies = [

  { name: "Company One", category: "Finance", start: 1981, end: 2004 },

  { name: "Company Two", category: "Retail", start: 1992, end: 2008 },

  { name: "Company Three", category: "Auto", start: 1999, end: 2007 },

  { name: "Company Four", category: "Retail", start: 1989, end: 2010 },

  { name: "Company Five", category: "Technology", start: 2009, end: 2014 },

  { name: "Company Six", category: "Finance", start: 1987, end: 2010 },

  { name: "Company Seven", category: "Auto", start: 1986, end: 1996 },

  { name: "Company Eight", category: "Technology", start: 2011, end: 2016 },

  { name: "Company Nine", category: "Retail", start: 1981, end: 1989 }

];

const ages = [33, 12, 20, 16, 5, 54, 21, 44, 61, 13, 15, 45, 25, 64, 32];

// 4th example - sort companies by start date

let res = companies.sort((c1, c2) => (c1.start > c2.start ? 1 : -1));

console.log(res);

// 5th-Sort an array alphabetically, and then reverse the order of the sorted items (descending):

var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.sort();

fruits.reverse();

console.log(fruits)