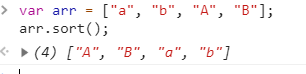
1.字符串[sort()]排序

sort(sortby) //sortby 可选。规定排序顺序。必须是函数。

**var arr = ["a", "b", "A", "B"];**

**arr.sort();**

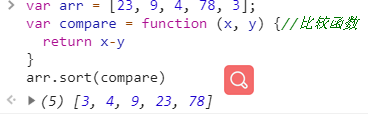


2.数字[函数]排序

sort(sortby) //sortby 可选。规定排序顺序。必须是函数。

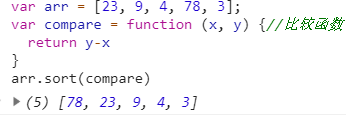
**//顺序**

**var arr = [23, 9, 4, 78, 3];  
var compare = function (x, y) {//比较函数  
 return x-y   
}  
arr.sort(compare)**



**//倒序**

**var arr = [23, 9, 4, 78, 3];  
var compare = function (x, y) {//比较函数  
 return y-x   
}  
arr.sort(compare)**



3.数组[对象属性]排序

**根据数组项的某个属性对数组进行排序**

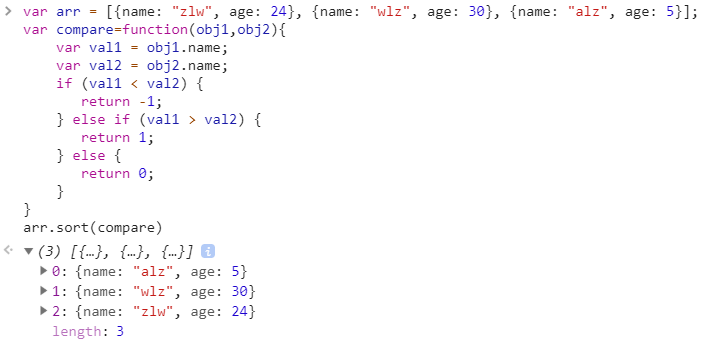
**var arr = [{name: "zlw", age: 24}, {name: "wlz", age: 30}, {name: "alz", age: 5}];**

**var compare=function(obj1,obj2){**

**var val1 = obj1.name;  
 var val2 = obj2.name;  
 if (val1 < val2) {  
 return -1;  
 } else if (val1 > val2) {  
 return 1;  
 } else {  
 return 0;  
 }**

**}**

**arr.sort(compare)**



**封装**

**var arr = [{name: "zlw", age: 24}, {name: "wlz", age: 30}, {name: "alz", age: 2}];****var compare=function(prop){**

**return function(obj1,obj2){**

**var val1 = obj1[prop];  
 var val2 = obj2[prop];  
 if (val1 < val2) {  
 return -1;  
 } else if (val1 > val2) {  
 return 1;  
 } else {  
 return 0;  
 }**

**}**

**}**

**arr.sort(compare("name"))**



**封装(数字字符串转化为数字)**

**var arr = [{name: "zlw", age: 24}, {name: "wlz", age: 30}, {name: "alz", age: 2}];  
var compare=function(prop){**

**return function(obj1,obj2){**

**var val1 = obj1[prop];  
 var val2 = obj2[prop];**

**if(!isNaN(val1) && !isNaN(val2)){**

**val1 = Number(val1);**

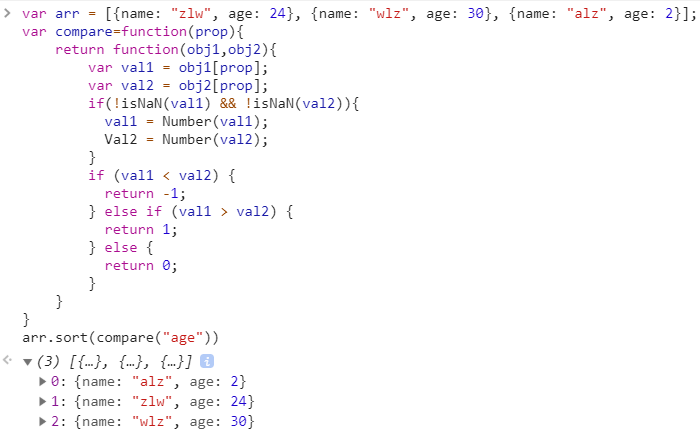
**val2 = Number(val2);**

**}  
 if (val1 < val2) {  
 return -1;  
 } else if (val1 > val2) {  
 return 1;  
 } else {  
 return 0;  
 }**

**}**

**}**

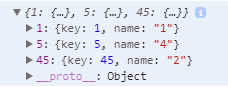
**arr.sort(compare("age"))**



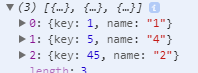
1. 数组对象按照[key值]排序

循环方法:

**let keyJson = { res: [{ key: 1, name: '1' }, { key: 45, name: '2' }, { key: 5, name: '4' }] }  
let keyObj = {} //以key属性为key的对象  
keyJson.res.map(res => {  
 keyObj[res.key] = { key: res.key,name:res.name}  
})  
console.log(keyObj)**

 **let keyArr = [] //key值集合  
for (var key in keyObj) {  
 keyArr.push(key)  
}  
console.log(keyArr)**

 **keyArr.sort(function(a,b){return a-b})  
let keySort=[] //排序后key属性索引对应值  
keyArr.map(res=>{  
 keySort.push(keyObj[res]) //obj[‘keyName’]对象寻址  
})  
console.log(keySort)**



keys()方法:获取key的方法 .keys()

|  |
| --- |
| const data = { 1001: 1, 1002: 3, 1003: 2, 1004: 2, 1005: 1 };  const keys=Object.keys(data);  keys.sort(**function**(a, b){  **return** data[b]-data[a];});  console.log(keys); |

|  |
| --- |
| [ '1002', '1003', '1004', '1001', '1005' ] |