String , Array,Number

2019年11月22日

14:49

一 . String 的重要方法(以下均不改变原字符串)

1. *string*.charAt(*index*) , 根据下标获取字符串的字符
2. *string*.concat(*string1*, *string2*, ..., *stringX*) , 连接字符串,返回一个新的字符串,不改变原字符串
3. *string*.indexOf(*searchvalue*,*start*) , 获取字符串某个字段的下标
4. *string*.lastIndexOf(*searchvalue*,*start*), 获取字符串某个字段的下标
5. *string*.match(*regexp*), 返回字符串数组
6. *string*.replace(*searchvalue, newvalue*), 用新的字段代替旧字段, 不改变原字符串
7. *string*.search(*searchvalue*) , 获取某个字段的下标
8. *string*.slice(*start*,*end*) , 获取字符串的切片 , 不改变原字符串
9. *string*.split(*separator*,*limit*) , 分解字符串 , 返回新的数组,limit 表示数组元素的个数,多于limit的部分将被舍弃
10. *string*.substr(*start*,*length*) , 截取字符串,不改变原字符串
11. *string*.toLowerCase()
12. *string*.toUpperCase()
13. *string*.trim()
14. 遍历字符串

for (let i of text) {

console.log(i);

}

1. 模板字符串,使用反引号 ` 包括字符串 , 使用${}作为占位符,插入字符串,可以通过字符串变量或者函数插入

$('#result').append(`

There are <b>${basket.count}</b> items

in your basket, <em>${basket.onSale}</em>

are on sale!

`);

二 . Array 的重要方法(改变原数组的有push,reverse,splice)

1. concat() , 拼接数组,返回新的数组

var children = hege.concat(stale);

1. every() , 判断数组是否都满足条件,返回true或false

var boolean = ages.every(val => {

return val >= 18;

});

1. filter() , 遍历数组,返回符合条件的元素,返回新的数组

var ages = [32, 33, 16, 40];

var bol = ages.filter(val => {

return val >= 32;

})

1. find() , 查找数组中符合条件的第一个元素,返回其值

var ages = [3, 10, 18, 20];

var age = ages.find(val => {

return val >= 10

})

1. findIndex() , 查找数组中符合条件的第一个元素,返回其下标

var ages = [3, 10, 18, 20];

var age = ages.find(val => {

return val >= 10

})

1. *array*.forEach(*function(currentValue, index, arr),thisValue*), 遍历数组 , 可以带1 个或者两个参数,不能改变数组,thisValue可以传递function中this的值

var fruits = ["apple", "orange", "cherry"];

fruits.forEach((item,index) => {

console.log(item + "" + index);

});

1. *array*.indexOf(*item*,*start*), 查找元素,返回其下标, start 表示开始查找的下标,负值表示从后面往前查找

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

var a = fruits.indexOf("Apple");

1. Array.isArray*(obj*), 判断变量是否是数组

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

var boolean = Array.isArray(fruits);

1. *array*.join(*separator*), 将数组拼接成字符串,如果带参,则分隔符为参数,否则为逗号

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

var energy = fruits.join();

1. *array*.lastIndexOf(*item*,*start*) , 类似于 indexOf ,查找最后一个相等的元素,返回下标
2. *array*.push(*item1*, *item2*, ..., *itemX*) , 将新的元素添加到数组, 可以同时添加多个,改变数组

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

fruits.push("Kiwi","Banana", "Orange");

1. array.reverse() , 将数组反转 , 改变数组

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

fruits.reverse();

1. array.slice(start, end) , 获取数组的切片 , 不改变数组

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

var citrus = fruits.slice(1, 3);

1. array.splice(index, howmany, item1, ....., itemX) , 拼接数组, index表示起始下标, howmany 表示取代元素个数, item 表示插入的元素,返回值为剪切的数组.

当howmany 为 0 时, 往数组中插入元素,

当 没有item并且howmany 不为空时, 剪切数组

**改变**原数组

1. array.includes(object),判断数组是否含有指定值Object,返回布尔值
2. array.map(function(curr,index,arr)),根据当前数组和处理函数,获取新的数组

三、Number

1. 生成随机数

(Math.random()\*268).toFixed(0)

Math.random() 生成 0 到 1 的随机数,toFixed(num) 表示四舍五入保留 num位小数

1. Math.floor(num)

舍去所有小数

1. Math.pow(num1,num2)

表示num1的num2次方

1. Math.log(num)

表示num的自然对数

1. Math.log10(num)

表示num以10为底的对数