# 数组：

## Array.prototype.splice()

**非纯函数splice**

**splice()**方法通过删除现有元素和/或添加新元素来更改一个数组的内容（被称为非纯函数的原因，**改变了原来的数组**，**返回的是由删除元素组成的数组**）。

var myFish = ['angel', 'clown', 'mandarin', 'sturgeon'];

myFish.**splice**(2, 0, 'drum'); *// 在索引为2的位置插入'drum'*

*// myFish 变为 ["angel", "clown", "drum", "mandarin", "sturgeon"]*

myFish.**splice**(2, 1); *// 从索引为2的位置删除一项（也就是'drum'这一项）*

*// myFish 变为 ["angel", "clown", "mandarin", "sturgeon"]*

## Array.prototype.slice()

**纯函数slice**

**slice()** 方法返回一个从开始到结束（***不包括结束***）选择的数组的一部分**浅拷贝**到一个新数组对象。且原始数组不会被修改（被称为纯函数原因，不会改变原数组，只是**单纯地返回一个新的数组**）。

var animals = ['ant', 'bison', 'camel', 'duck', 'elephant'];

**console**.**log**(animals.**slice**(2));

*// expected output: Array ["camel", "duck", "elephant"]*

**console**.**log**(animals.**slice**(2, 4));

*// expected output: Array ["camel", "duck"]*

**console**.**log**(animals.**slice**(1, 5));

*// expected output: Array ["bison", "camel", "duck", "elephant"]*

***“加了一个p就不纯了”—— slice=>splice***

## Array.prototype.map(function(currentValue,index,array){})

**map()** 方法创建一个新数组，其结果是该数组中的每个元素都调用一个提供的函数后返回的结果（***返回一个新数组***）。

**currentValue**

callback 的第一个参数，数组中正在处理的当前元素。

**index**

callback 的第二个参数，数组中正在处理的当前元素的索引。

**array**

callback 的第三个参数，map 方法被调用的数组。

var numbers = [1, 4, 9];

var doubles = numbers.**map**(function(num) {

return num \* 2;

});

## Array.prototype.forEach(function(currentValue,index,array){})

**forEach()** 方法对数组的每个元素执行一次提供的函数。

注意: 没有返回一个新数组! & 没有返回值!

const arr = ['a', 'b', 'c'];

arr.**forEach**(function(element) {

**console**.**log**(element);

});

arr.**forEach**( element => **console**.**log**(element));

forEach和map的区别：

map返回一个新数组，foreach不返回任何值

## for...of

**for...of语句**在[可迭代对象](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Guide/iterable)（包括 [Array](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Array)，[Map](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Map)，[Set](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Set)，[String](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/String)，[TypedArray](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/TypedArray)，[arguments](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions_and_function_scope/arguments) 对象等等）上创建一个迭代循环，调用自定义迭代钩子，并为每个不同属性的值执行语句。

### 迭代[Array](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Array)

let iterable = [10, 20, 30];

for (let value of iterable) {

value += 1;

**console**.**log**(value);

}

*// 11*

*// 21*

*// 31*

### 迭代[String](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/String)

let iterable = "boo";

for (let value of iterable) {

**console**.**log**(value);

}

*// "b"*

*// "o"*

*// "o"*

## Array.prototype.indexOf()

注意：对于String方法，请参阅 [String.prototype.indexOf()](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/String/indexOf)。

**indexOf()**方法返回在数组中可以找到一个给定元素的第一个索引，如果不存在，则返回-1。

arr.**indexOf**(searchElement[, fromIndex = 0])

searchElement

要查找的元素

fromIndex

开始查找的位置。

let a = [2, 9, 7, 8, 9];

a.**indexOf**(2); *// 0*

a.**indexOf**(6); *// -1*

a.**indexOf**(7, 3); *// -1*

# 对象：

## for...in

**for...in语句**以任意顺序遍历一个对象的[可枚举属性](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Enumerability_and_ownership_of_properties)。对于每个不同的属性，语句都会被执行。

语法

for (variable in object) {...}

var obj = {a:1, b:2, c:3};

for (var prop in obj) {

**console**.**log**("obj." + prop + " = " + obj[prop]);

}

*// Output:*

*// "obj.a = 1"*

*// "obj.b = 2"*

*// "obj.c = 3"*

### 仅迭代自身的属性（for…in一般和hasOwnProperty()配合使用）

执行 [hasOwnProperty()](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Object/hasOwnProperty) 来确定某属性是否是对象本身的属性

### 数组迭代和 for...in

for...in不应该用于迭代一个 [Array](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Array)，其中索引顺序很重要。

因此当迭代访问顺序很重要的数组时，最好用整数索引去进行[for](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Statements/for)循环（或者使用 [Array.prototype.forEach()](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach) 或 [for...of](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Statements/for...of) 循环）。

# 字符串

## String.prototype.indexOf()

**indexOf()** 方法返回调用  [String](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/String) 对象中第一次出现的指定值的索引，开始在 fromIndex进行搜索。

如果未找到该值，则返回-1。

"Blue Whale".**indexOf**("Blue"); *// returns 0*

"Blue Whale".**indexOf**("Blute"); *// returns -1*

"Blue Whale".**indexOf**("Whale", 0); *// returns 5*

"Blue Whale".**indexOf**("Whale", 5); *// returns 5*

"Blue Whale".**indexOf**("", 9); *// returns 9*

"Blue Whale".**indexOf**("", 10); *// returns 10*

"Blue Whale".**indexOf**("", 11); *// returns 10*