

有没有new区别很大

数组(Array)

数组进一步探究

小结

匿名函数

chrome的console里面如何换行

构造函数Function例子

一般到MDN查找js的api相关;

[https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Array)

这里先讲标准库:

<https://wangdoc.com/javascript/stdlib/index.html>

# 标准库

---

Object对象

Array对象

包装对象

Number对象

String对象

Math对象

Date对象

RegExp对象

JSON对象

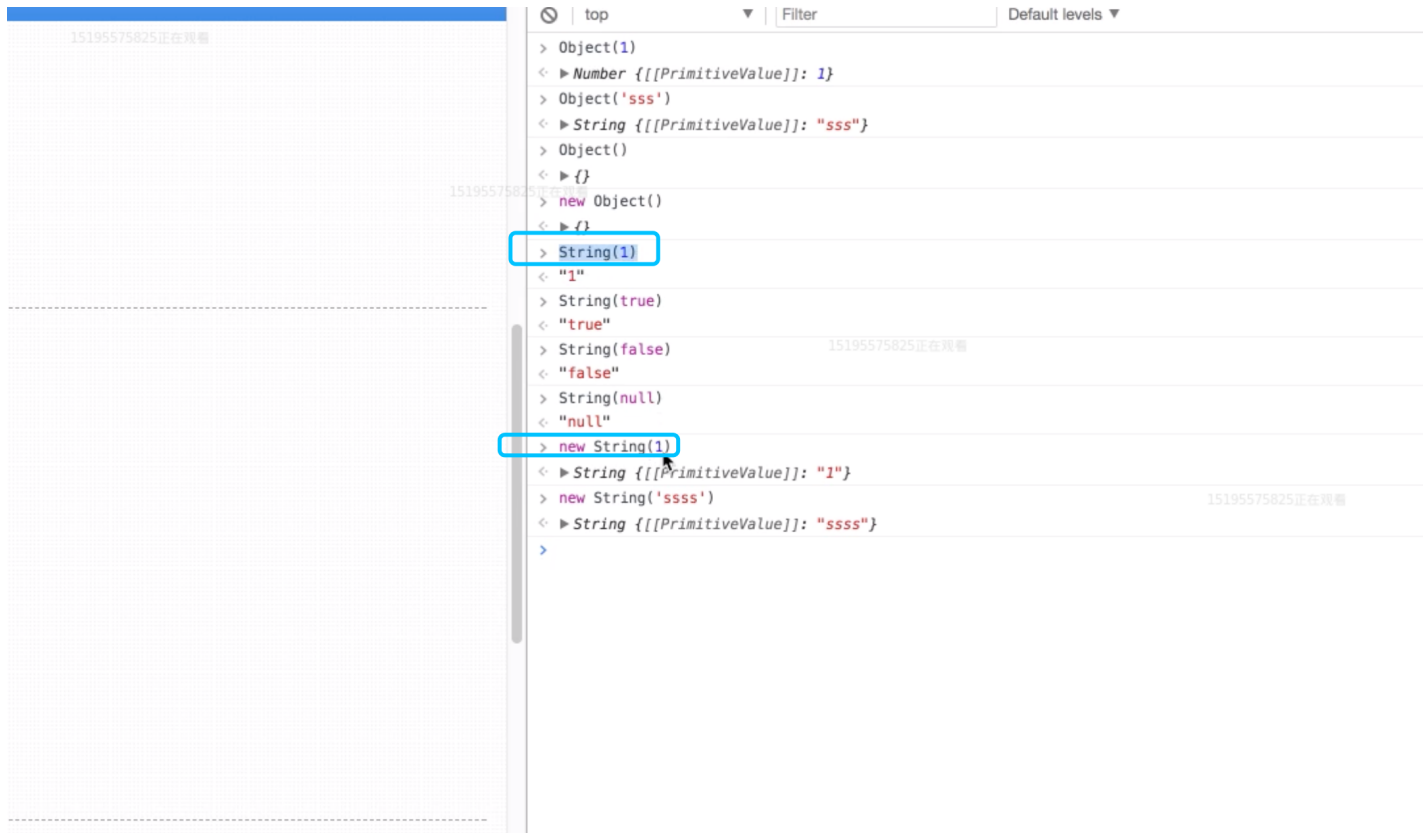
console对象

属性描述对象

# 有没有new区别很大

普通类型没有new就是基本类型;  
有new就是对象;

```
1 String(1)
2 "1"
3 new String(1)
4 String {"1"}0: "1"length: 1__proto__: String[[PrimitiveValue]]: "1"
```



## 数组(Array)

js里面一共七种类型:  
Number,String,Boolean,Symbol,Null,Undefined,Object;  
下面两种等价

```
1 let f = ['a','b']
2 undefined
3 let f2 =new Array('a','b')
4 undefined
5 f
6 (2) ["a", "b"]
7 f2
8 (2) ["a", "b"]
```

## 数组进一步探究

```
1 var a = Array(3)
2 undefined
3 a.length
4 3
5 a.push
6 f push() { [native code] }
7 console.dir(a)
8 VM1484:1 Array(3)length: 3__proto__: Array(0)concat: f concat()constructor: f Array()c
9 undefined
10 `0` in a
11 false
12 `1` in a
13 false
14 `2` in a
15 false
```

```
> var a = Array(3)
< undefined
> a.length
< 3
> a.push
< f push() { [native code] }
> console.dir(a)
  ▼Array(3) ⓘ
    length: 3
    ▶__proto__: Array(0)
< undefined
> `0` in a
< false
> `1` in a
< false
> `2` in a
< false
>
```

```
< f push() { [native code] }  
> console.dir(a)
```

VM1484:1

```
▼ Array(3) ⓘ  
  length: 3  
  ▼ __proto__: Array(0)  
    ▶ concat: f concat()  
    ▶ constructor: f Array()  
    ▶ copyWithin: f copyWithin()  
    ▶ entries: f entries()  
    ▶ every: f every()  
    ▶ fill: f fill()  
    ▶ filter: f filter()  
    ▶ find: f find()  
    ▶ findIndex: f findIndex()  
    ▶ flat: f flat()  
    ▶ flatMap: f flatMap()  
    ▶ forEach: f forEach()  
    ▶ includes: f includes()  
    ▶ indexOf: f indexOf()  
    ▶ join: f join()  
    ▼ keys: f keys()  
      arguments: (...)  
      caller: (...)  
      length: 0  
      name: "keys"  
      ▶ __proto__: f ()  
      ▶ [[Scopes]]: Scopes[0]  
    ▶ lastIndexOf: f lastIndexOf()  
      length: 0  
    ▶ map: f map()  
    ▶ pop: f pop()  
    ▶ push: f push()  
    ▶ reduce: f reduce()  
    ▶ reduceRight: f reduceRight()  
    ▶ reverse: f reverse()  
    ▶ shift: f shift()  
    ▶ slice: f slice()  
    ▶ some: f some()  
    ▶ sort: f sort()  
    ▶ splice: f splice()  
    ▶ toLocaleString: f toLocaleString()  
    ▶ toString: f toString()  
    ▶ unshift: f unshift()  
    ▶ values: f values()  
    ▶ Symbol(Symbol.iterator): f values()  
    ▶ Symbol(Symbol.unscopables): {copyWithin: true, entries: true, fill: true, find: true, ...  
    ▶ __proto__: Object  
< undefined
```

### js里面的不一致性

```
> var a = Array(3)
< undefined
> a.length
< 3
> a.push
< f push() { [native code] }
> console.dir(a)
  ▶ Array(3)
< undefined
> `0` in a
< false
> `1` in a
< false
> `2` in a
< false
> var a = Array(3,3)
< undefined
> a
< ▼ (2) [3, 3] ⓘ
  0: 3
  1: 3
  length: 2
  ▶ __proto__: Array(0)
>
```

var a = Array(3)

15195575825正在观看

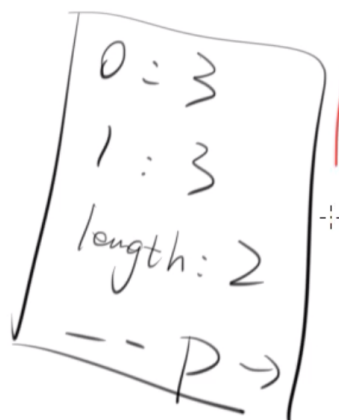
length



15195575825正在观看

var a = Array(3, 3)

不是 length



15195575825正在观看

不一致

致

性

垃圾



number string boolean

15195575825正在观看

Number ( ) => 基本类型

new Number ( ) => 对象

15195575825正在观看

不一样

object array  
function

Object ( ) => 对象

new Object ( ) => 对象

译

15195575825正在观看

加不加一样

1519557582

function和Function的区别:

```

1 function 关键字 if else var function
2 Function
3
4
5
6 var 声明一个变量
7 var a = 1
8 function 声明一个函数
9 function f(){
10
11 window.Object
12 window.Function
  
```

15195575825正在观看

15195575825正在

```
1 function 关键字 if else var function
2 Function 全局对象
3
4
```

三种形式:(一般不用第三种!)

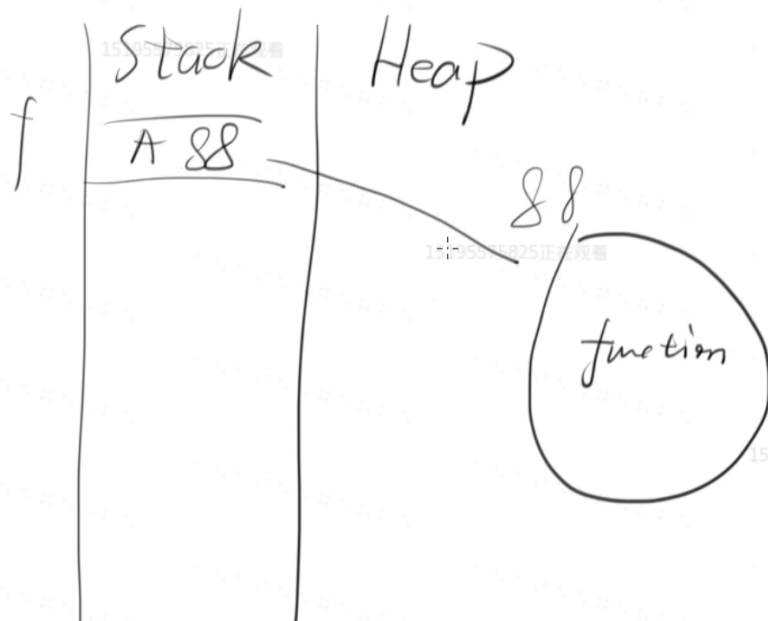
```
1 具名函数
2 function f(){
3     return undefined
4 }
5 匿名函数
6 function (){
7
8 }
9 Function
10 new Function('x','y', 'x+y')
```

151955

匿名函数

var f = function() {}

var f  
f = function() {}



其实 f 和匿名函数没有任何关系;

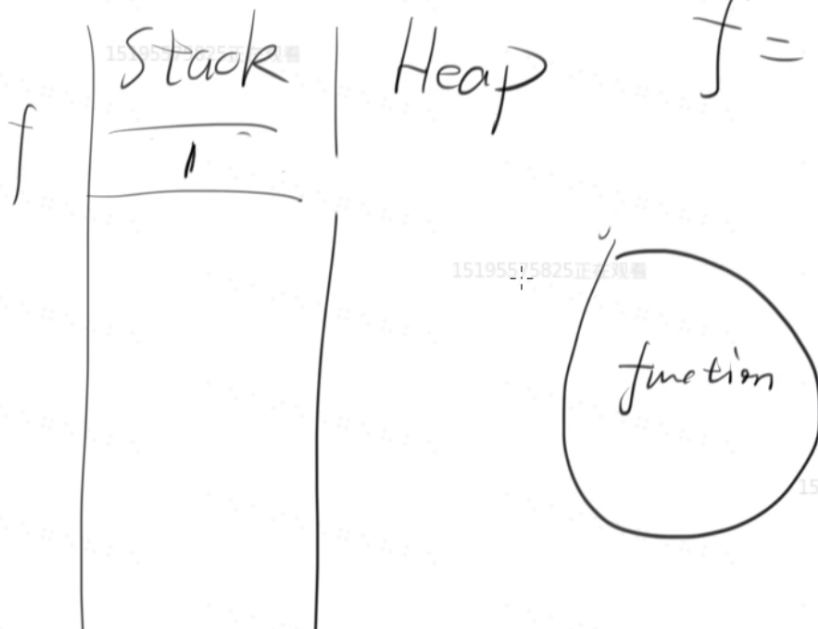
如 f=1, 则: function 和 f 就没有任何联系了;

var f = function() {}, 其实是分两步的:

```
1 1. var f ;
2 2. f = function() {}
```

var f = function() {}

var f  
f = function() {}  
f = 1





shift+enter键

## 构造函数Function例子

```
1 var f = new Function('a','b','return a+b')
2 f(1,2)
3 3
4 f(3,4)
5 7
```

```
> var a = new Array(1,2,3)
var a = [1,2,3]
```

```
< undefined
```

```
> var f = function(a,b){
  return a+b
}
```

```
< undefined
```

```
> var f = new Function('a','b','return a+b')
```

```
< undefined
```

```
>
```

