小凤凤之JS基础篇

加油哦.....

数据类型介绍

Js 数据类型分为：简单数据类型 和 复杂数据类型

简单（值类型）： String，Number ，Boolean，undefined，Null，Symbol

复杂（引用类型）：Array，Object，Function

Symbol 用于 表示独一无二的值；

```
> var a = Symbol()
undefined
> var b = Symbol()
undefined
> a == b
false
>
```

```
> var a = Symbol("a")
undefined
> var b = Symbol("a")
undefined
> a == b
false
>
```

```
> let s1 = Symbol.for('foo');
  let s2 = Symbol.for('foo');
undefined
> s1 === s2
true
>
```

Object.getOwnPropertySymbols 可以获取指定对象的所有 Symbol 属性名

值类型判断

- ***简单方法 typeof instanceof***
- ***instanceof***

```
> var A = function (a){this.a = a}
< undefined
> var newa = new A(1)
< undefined
> newa.a
< 1
> newa instanceof A
< true
> |
```

- ***常用方法：***
- ***Object.prototype.toString.call('') ; // [object String]***
- ***Object.prototype.toString.call([]) ; // [object Array]***
- ***Object.prototype.toString.call(1) ; // [object Number]***

NaN 判断

isNaN(NaN) or typeof NaN === “number” && String(NaN) === “NaN”

值的互相转换

```
> parseFloat("111.1")
< 111.1
> parseInt("111.111")
< 111
> Number("111.11")
< 111.11
> parseFloat("11a1.1")
< 11
> parseInt("11a1.111")
< 11
> Number("11a1.11")
< NaN
> parseFloat("111.a1")
< 111
> parseInt("111.a111")
< 111
>
```

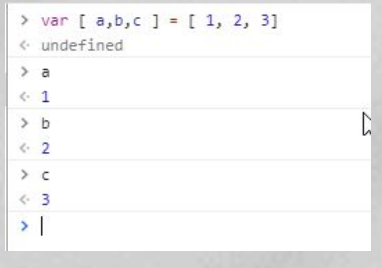
```
> String(11.11)
< "11.11"
> String(NaN)
< "NaN"
```

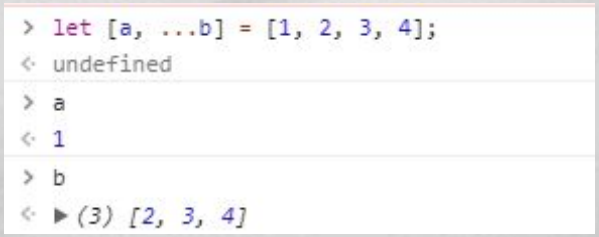
变量声明

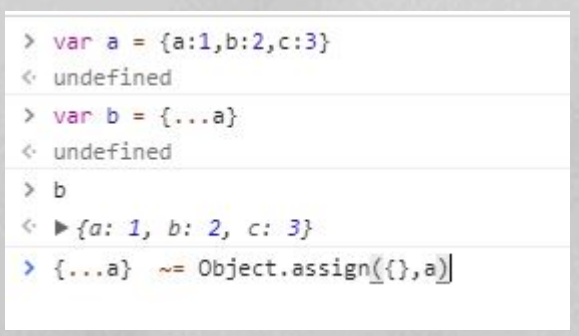
- ***var let const***

var 会变量提升，let const 不会提升

ES6 中的常见变量赋值







函数声明

- ***ES5 function test(){} or var test = function(){}***

- ***ES6 var test = ()=>{}***

自执行函数

(function test(){})() or (()=>{})()

函数参数

- 引用类型 和值类型
- 值类型为复制，引用类型为地址传递
- 参数传递

```
function test(a=1,b=2){
    console.log(a)
    console.log(b)
}
test()
//  1
//  2
```

```
function test(a,...b){
    console.log(a)
    console.log(b)
}
test(1,2,3,4)
//  1
//  [2,3,4]
```

```
function test(a,b){
    a = 1
    b.a = 1
    b = {
        b:2
    }
}
var a = 0,
    b = {
        a:0
    };
test(a,b)
console.log(a,b)
// 0 {a:1}
```

函数表达式 与 函数返回值

无参数　()=>{ ... }

一个参数　a => { ... }

多个参数　(a,b)=>{... }

返回值：

()=>1　===　()=>{　return 1　}

()=>({a:1,b:2})　=== ()=>{　return {a:1,b:2}}

回调

阮一峰大大的 js异步编程4种方法

http://www.ruanyifeng.com/blog/2012/12/asynchronous%EF%BC%BFjavascript.html