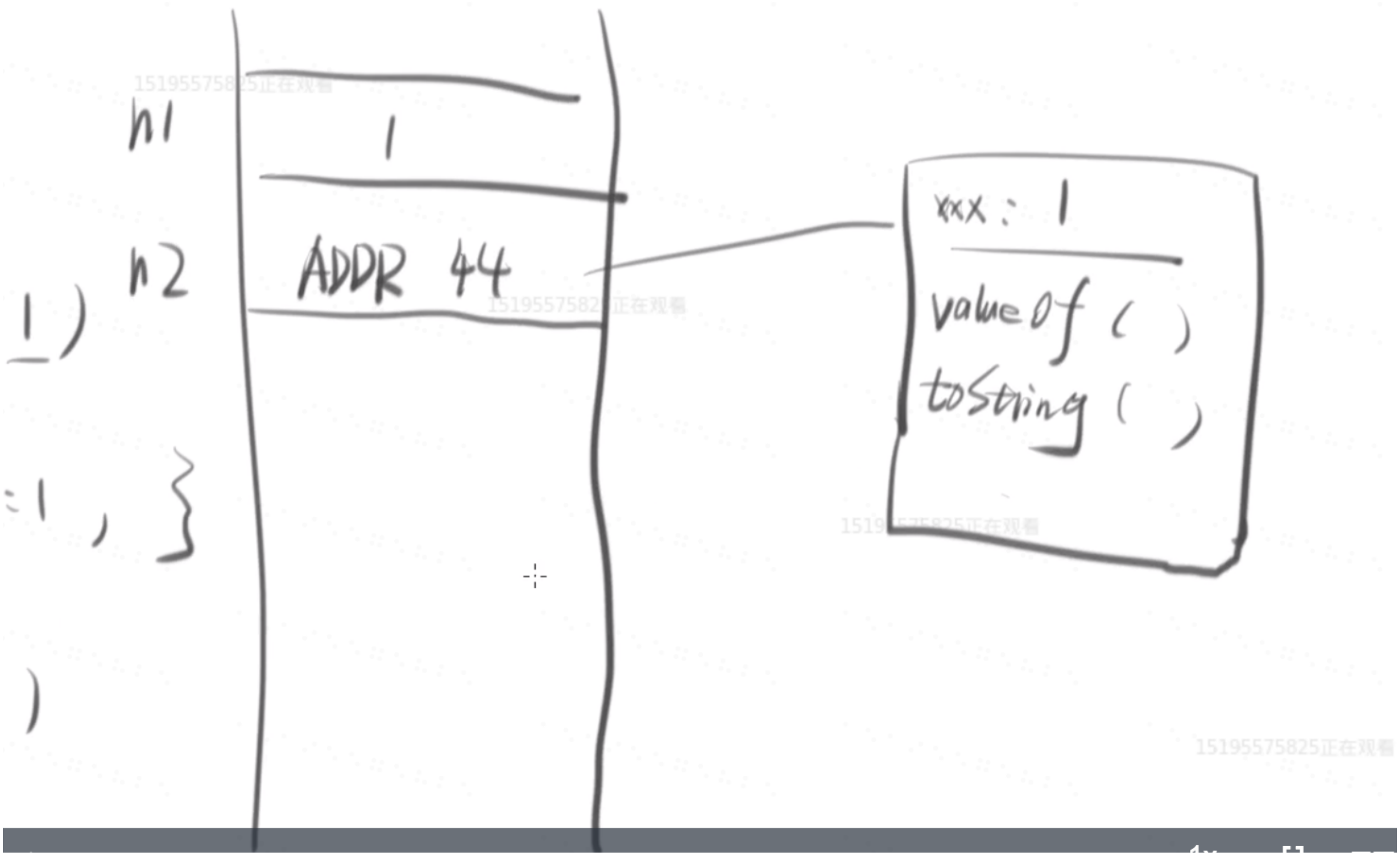
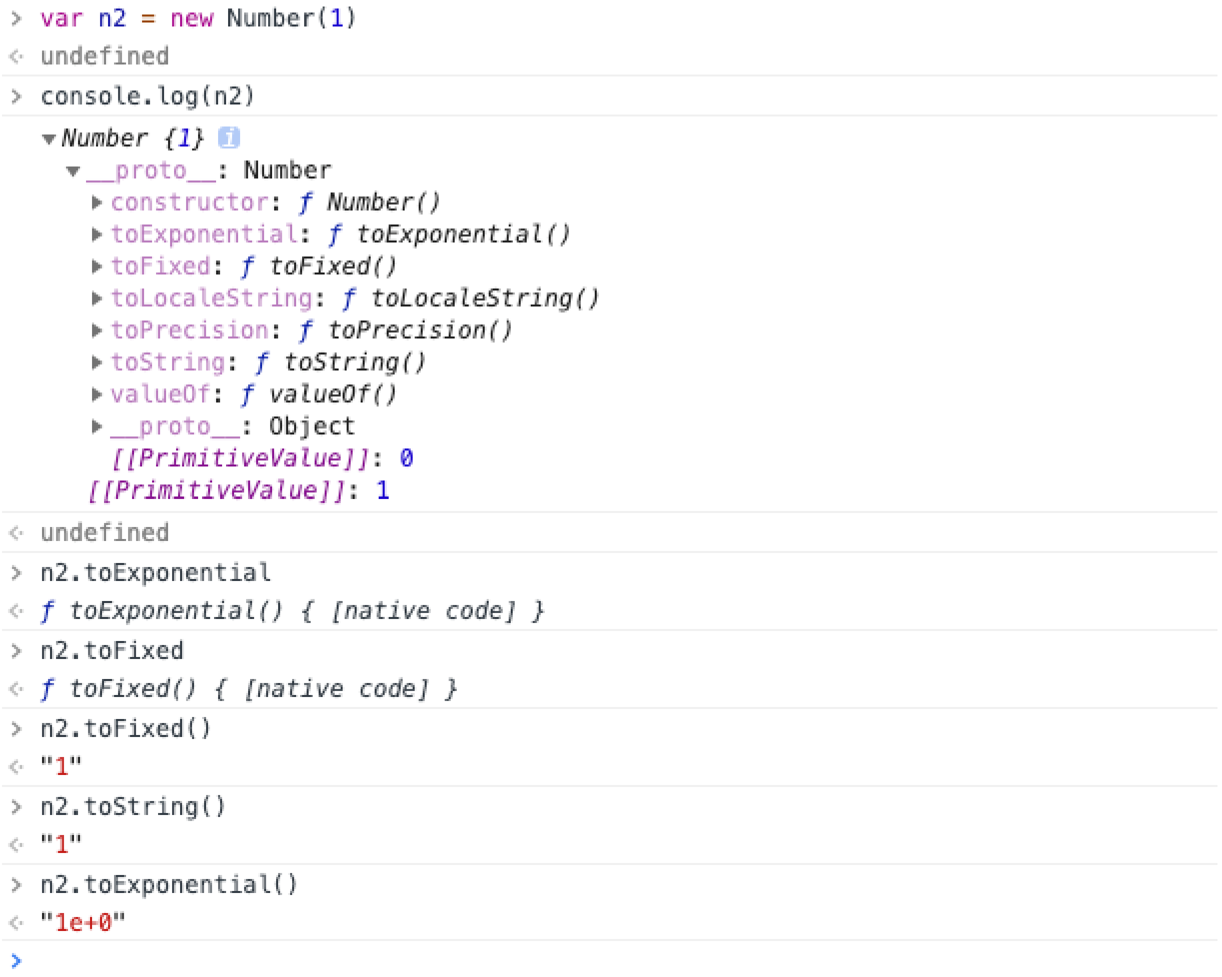
Number

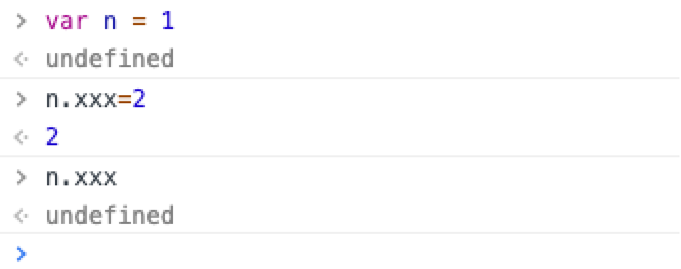
var n = new Number(1) 创建一个 Number 对象
1 与 new Number(1) 【这种形式是js创始人老板js像java的==】的区别是什么？看内存图,包装成对象,除了1,还有操作1的函数



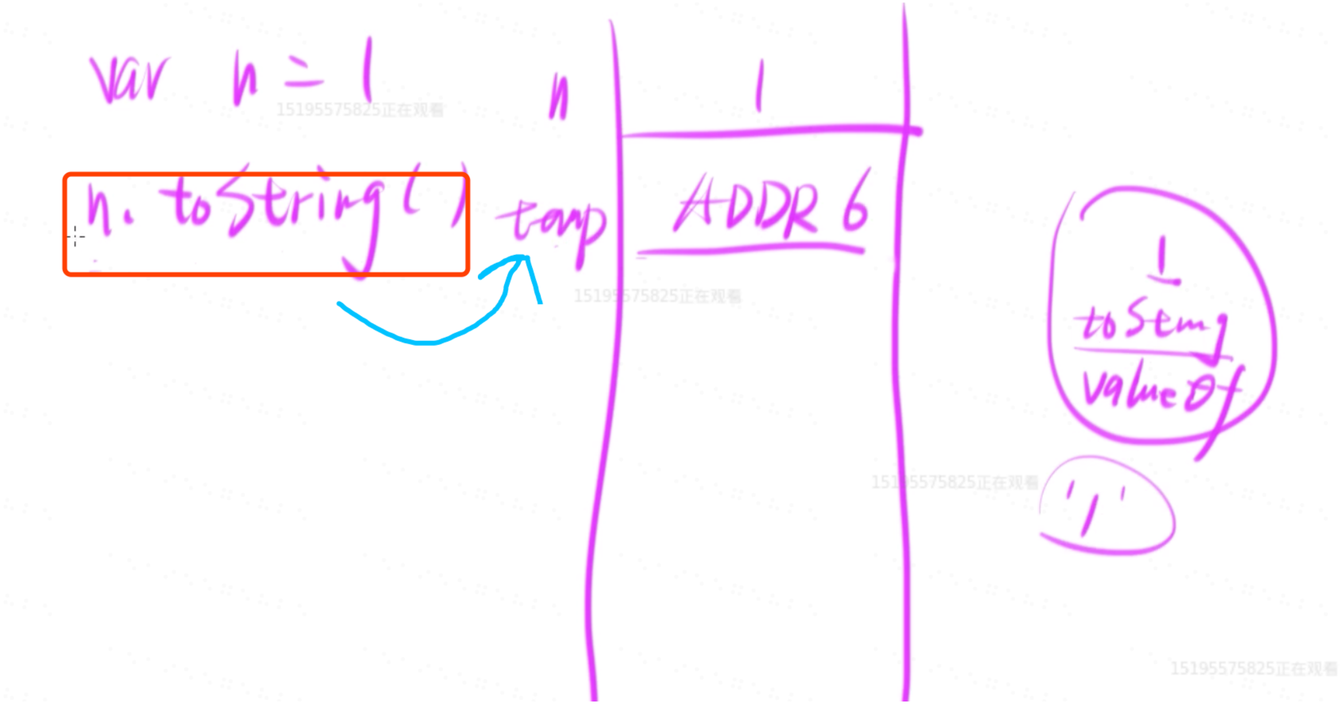


```
> var n2 = new Number(1)
< undefined
> console.log(n2)
    ▼Number {1} i
      ▼__proto__: Number
        ▶constructor: ƒ Number()
        ▶toExponential: ƒ toExponential()
        ▶toFixed: ƒ toFixed()
        ▶toLocaleString: ƒ toLocaleString()
        ▶toPrecision: ƒ toPrecision()
        ▶toString: ƒ toString()
        ▶valueOf: ƒ valueOf()
        ▶__proto__: Object
         [[PrimitiveValue]]: 0
       [[PrimitiveValue]]: 1
< undefined
> n2.toExponential
< ƒ toExponential() { [native code] }
> n2.toFixed
< ƒ toFixed() { [native code] }
> n2.toFixed()
< "1"
> n2.toString()
< "1"
> n2.toExponential()
< "1e+0"
>
```

其实运行至n.toString()的时候,内存里面会生成一个临时对象temp【注意,这句话执行完之后temp就会立即消失！！】;(看下面两个图！)



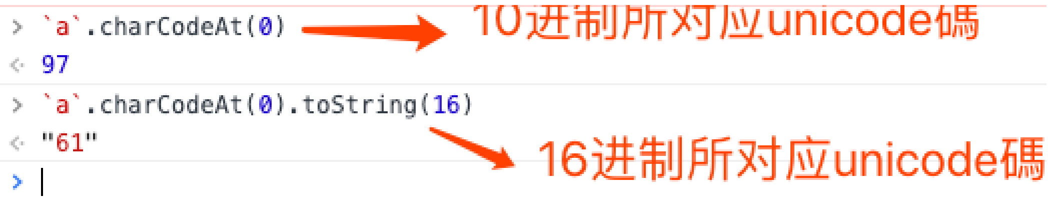
```
> var n = 1
< undefined
> n.xxx=2
< 2
> n.xxx
< undefined
>
```



故很少有js程序员用new Number(1)这种形式！

String

var s = new String('hello') 创建一个 String 对象
'hello' 与 new String('hello') 的区别是什么？

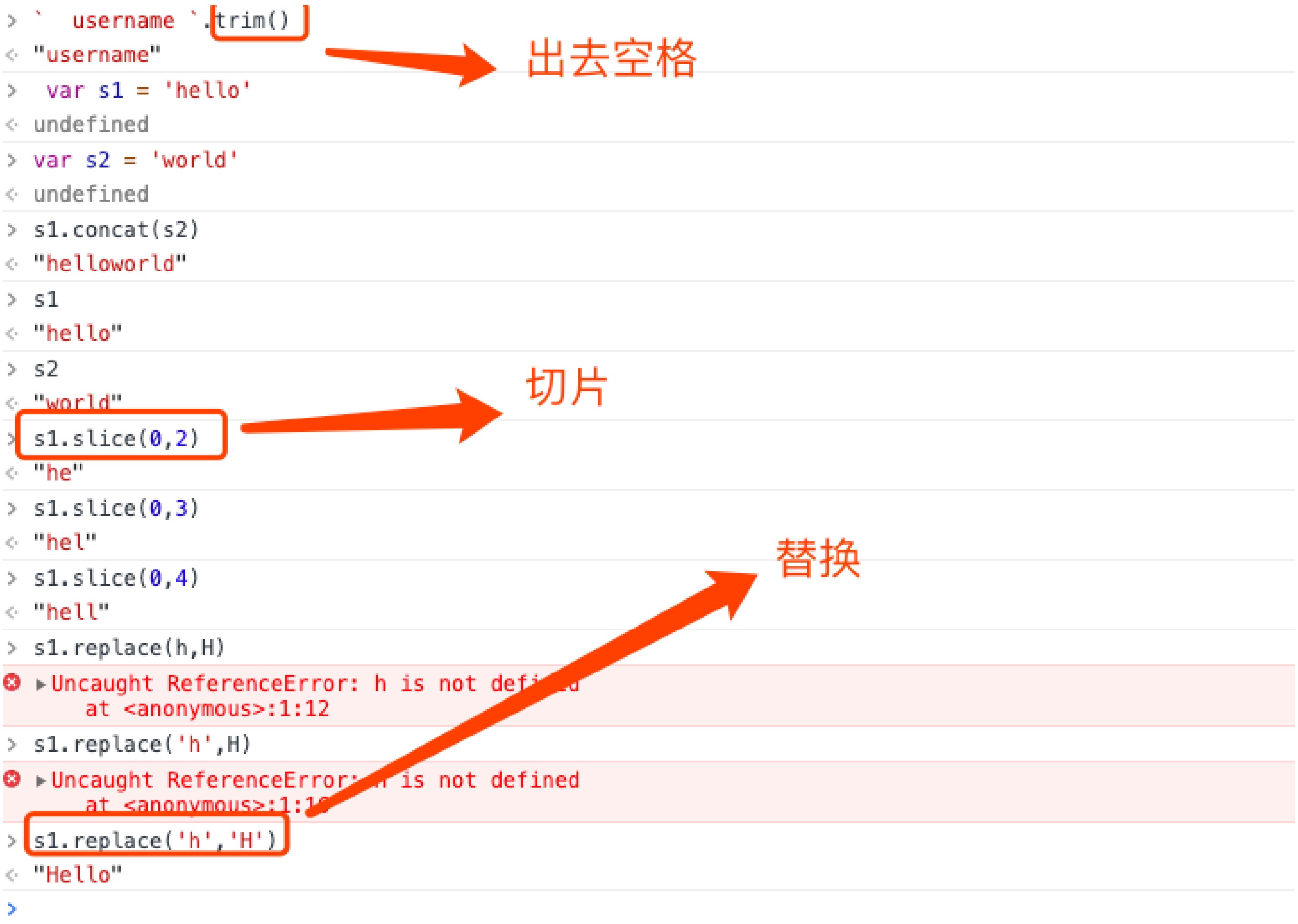


```
> `a`.charCodeAt(0)
< 97
> `a`.charCodeAt(0).toString(16)
< "61"
> |
```

10进制所对应unicode碼

16进制所对应unicode碼

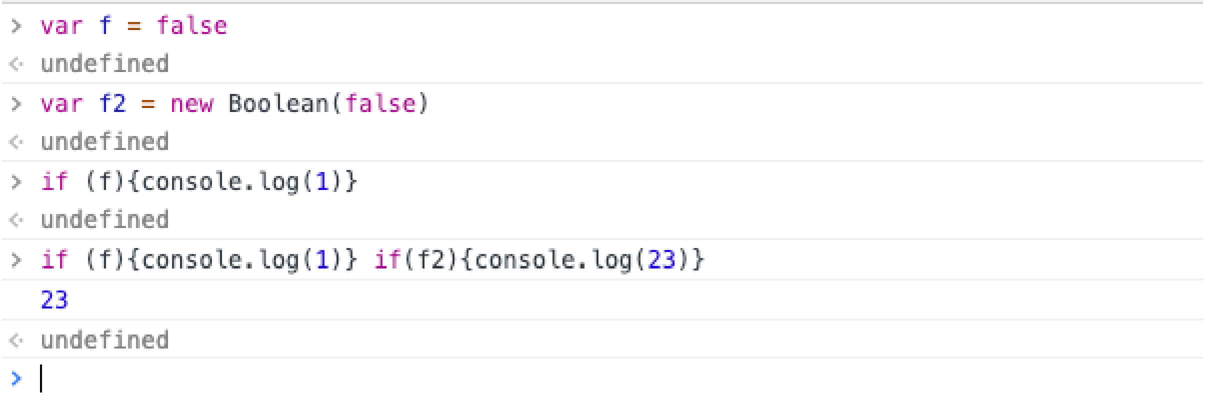
常见的几个api:



```
> `  username `.trim()
< "username"
>  var s1 = 'hello'
< undefined
> var s2 = 'world'
< undefined
> s1.concat(s2)
< "helloworld"
> s1
< "hello"
> s2
< "world"
> s1.slice(0,2)
< "he"
> s1.slice(0,3)
< "hel"
> s1.slice(0,4)
< "hell"
> s1.replace(h,H)
⊗ ▸Uncaught ReferenceError: h is not defined
      at <anonymous>:1:12
> s1.replace('h',H)
⊗ ▸Uncaught ReferenceError: H is not defined
      at <anonymous>:1:16
> s1.replace('h','H')
< "Hello"
>
```

Boolean

var b = new Boolean(true) 创建一个 Boolean 对象
true 与 new Boolean(true) 的区别是什么？看内存图



```
> var f = false
< undefined
> var f2 = new Boolean(false)
< undefined
> if (f){console.log(1)}
< undefined
> if (f){console.log(1)} if(f2){console.log(23)}
  23
< undefined
> |
```

因为所有对象都是true!

Object

var o1 = {}

var o2 = new Object()

o1 和 o2 没区别

小结

① 7种类型 ② 5个 falsy ③ 内存图