**这些的JavaScript编程黑科技，装逼指南，高逼格代码，让你惊叹不已**

[风靡](https://www.zhihu.com/people/feng-mi-91-45)[风靡](https://www.zhihu.com/people/feng-mi-91-45)

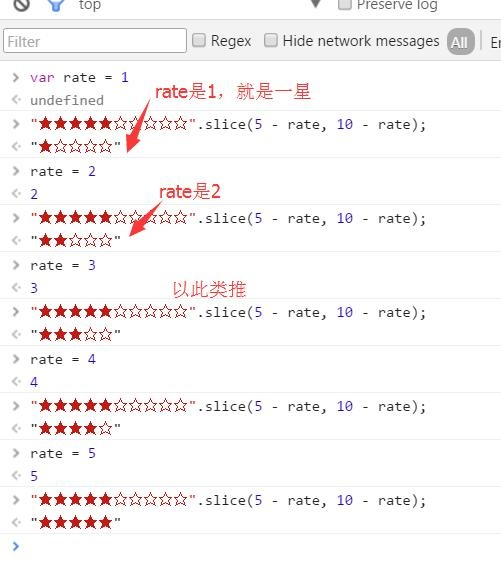
2 小时前

Javascript是一门很吊的语言，我可能学了假的JavaScript的，哈哈，大家还有什么推荐的，补充送那啥邀请码。  
本文秉承着：你看不懂是你SB，我写的代码就要牛逼。

**1，单行写一个评级组件**

"★★★★★☆☆☆☆☆".slice(5 **-** rate, 10 **-** rate);

定义一个变量率是1到5的值，然后执行上面代码，看图



才发现插件什么的都弱爆了



来源：[来自知乎用户蜗牛老湿的回答](https://www.zhihu.com/question/46943112/answer/113583615)

**2，如何装逼用代码骂别人SB**

(**!**(**~+**[])**+**{})[**--**[**~+**""][**+**[]]**\***[**~+**[]] **+** **~~!+**[]]**+**({}**+**[])[[**~!+**[]]**\*~+**[]]

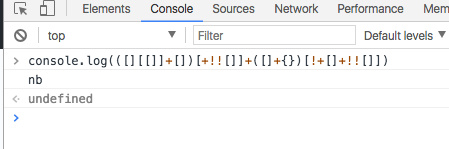


了解为什么请移步：[一行能装逼的的JavaScript代码](http://link.zhihu.com/?target=http%3A//www.jfh.com/jfperiodical/article/3224" \t "_blank)

**3，如何用代码优雅的证明自己NB**

这个牛逼了

console.log(([][[]]**+**[])[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]])



**4，JavaScript错误处理的方式的正确姿势**

😂😂😂，舅服你

**try** {

something

} **catch** (e) {

window.location.href **=**

"http://stackoverflow.com/search?q=[js]+" **+**

e.message;

}



**5、从一行代码里面学点JavaScript**

[].forEach.call($$("\*"),**function**(a){

a.style.outline**=**"1px solid #"**+**(**~~**(Math.random()**\***(1**<<**24))).toString(16)

})

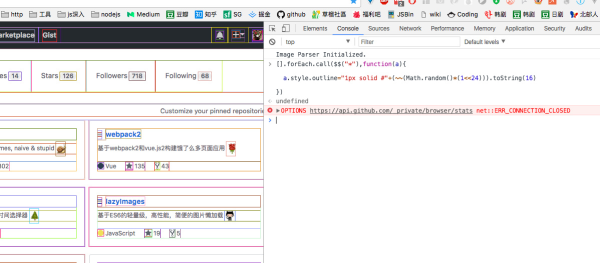
翻译成正常语言就是这样的

Array.prototype.forEach.call(document.querySelectorAll('\*'),

dom **=>** dom.style.outline **=** `1px solid #${parseInt(Math.random() **\***

Math.pow(2,24)).toString(16)}`)

接下来在浏览器控制看看：something magic happens

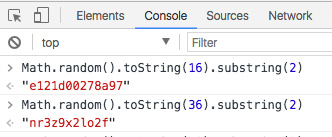


具体分析请参见这篇文章：[从一行代码里面学点JavaScript](http://link.zhihu.com/?target=https%3A//sdk.cn/news/3025" \t "_blank)

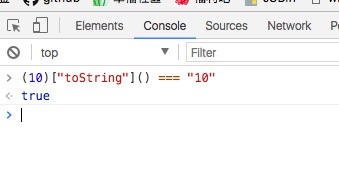
**6、论如何优雅的取随机字符串**

Math.random().toString(16).substring(2)

Math.random().toString(36).substring(2)



**7、(10)["toString"]() === "10"**



解析请移步：[js奇淫技巧1](http://link.zhihu.com/?target=http%3A//qylanikin.lofter.com/post/1cbb3f55_cc787c9" \t "_blank)

**8、匿名函数自执行**

这么多写法你选择哪一种？我选择死亡。

( **function**() {}() );

( **function**() {} )();

[ **function**() {}() ];

**~** **function**() {}();

**!** **function**() {}();

**+** **function**() {}();

**-** **function**() {}();

**delete** **function**() {}();

**typeof** **function**() {}();

**void** **function**() {}();

**new** **function**() {}();

**new** **function**() {};

**var** f **=** **function**() {}();

1, **function**() {}();

1 **^** **function**() {}();

1 **>** **function**() {}();

*// ...*



**9、另外一种undefined**

从来不需要声明一个变量的值是undefined，因为JavaScript会自动把一个未赋值的变量置为undefined。所有如果你在代码里这么写，会被鄙视的

**var** data **=** **undefined**;

但是如果你就是强迫症发作，一定要再声明一个暂时没有值的变量的时候赋上一个undefined。那你可以考虑这么做：

**var** data **=** **void** 0; *// undefined*

void在JavaScript中是一个操作符，对传入的操作不执行并且返回undefined。void后面可以跟()来用，例如void(0)，看起来是不是很熟悉？没错，在HTML里阻止带href的默认点击操作时，都喜欢把href写成javascript:void(0)，实际上也是依靠void操作不执行的意思。

当然，除了出于装逼的原因外，实际用途上不太赞成使用void，因为void的出现是为了兼容早起ECMAScript标准中没有undefined属性。void 0的写法让代码晦涩难懂。

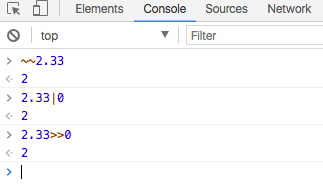


**10、论如何优雅的取整**

**var** a **=** **~~**2.33

**var** b**=** 2.33 **|** 0

**var** c**=** 2.33 **>>** 0



**11、如何优雅的实现金钱格式化：1234567890 --> 1,234,567,890**

用正则魔法实现：

var test1 **=** '1234567890'

var format **=** test1**.**replace**(/**\B**(?=(**\d**{**3**})+(?!**\d**))/**g**,** ','**)**

console**.**log**(**format**)** *// 1,234,567,890*

非正则的优雅实现：

**function** formatCash(str) {

**return** str.split('').reverse().reduce((prev, next, index) **=>** {

**return** ((index **%** 3) **?** next **:** (next **+** ',')) **+** prev

})

}

console.log(formatCash('1234567890')) *// 1,234,567,890*



**12、这个我服，还有这个你很机智**

我服

**while** (1) {

alert('牛逼你把我关了啊')

}

你很机智，好一个障眼法

清除缓存： **<**a href**=**"javascript:alert('清除成功');"**>**清除缓存**<**/a>



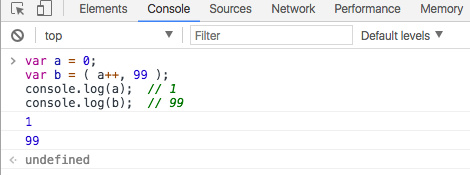
**13、逗号运算符**

**var** a **=** 0;

**var** b **=** ( a**++**, 99 );

console.log(a); *// 1*

console.log(b); *// 99*



**14、论如何最佳的让两个整数交换数值**

常规办法：

**var** a**=**1,b**=**2;

a **+=** b;

b **=** a **-** b;

a **-=** b;

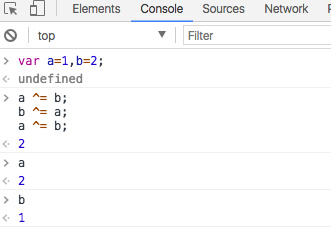
缺点也很明显，整型数据溢出，对于32位字符最大表示数字是2147483647，如果是2147483645和2147483646交换就失败了。

黑科技办法：

a **^=** b;

b **^=** a;

a **^=** b;



哈哈😄，看不懂的童鞋建议去补习一下C语言的位操作，我就不去复习了，以前学嵌入式时候学的位操作都忘了

**15、实现标准JSON的深拷贝**

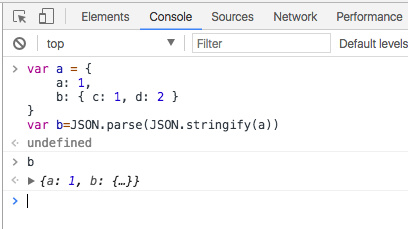
**var** a **=** {

a**:** 1,

b**:** { c**:** 1, d**:** 2 }

}

**var** b**=**JSON.parse(JSON.stringify(a))



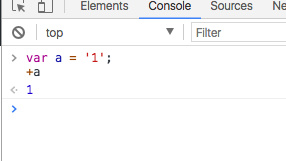
不考虑IE的情况下，标准JSON格式的对象蛮实用，不过对于undefined和function的会忽略掉。

**16、不用Number、parseInt和parseFloat和方法把"1"字符串转换成数字**

哈哈，不准用强制类型转换，那么就想到了强大了隐式转换

**var** a **=**1

**+**a

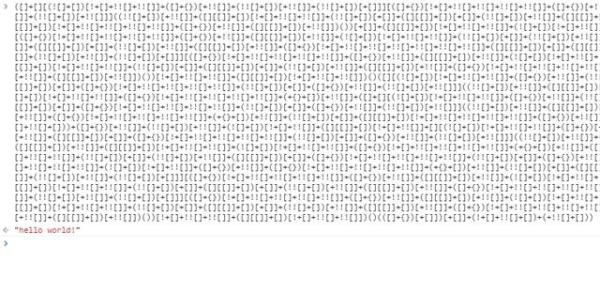


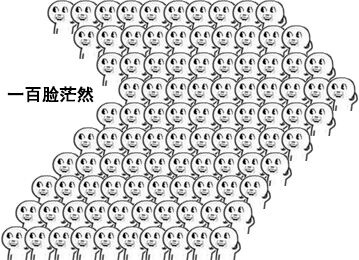
**17、如何装逼的写出"hello world!"**

滚动条很长哦😯

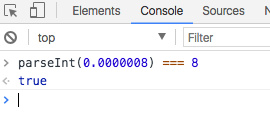
([]**+**[][(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]][([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]]((**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([][[]]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**+**{}**+**[])[**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]])())[**+**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**[][(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]][([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]]((**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([][[]]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([][[]]**+**[])[**+**[]]**+**([][[]]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**+**{}**+**[])[**+!!**[]]**+**([]**+**[][(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]][([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]]((**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([][[]]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**+**{}**+**[])[**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]])())[**!+**[]**+!!**[]**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]])()([][(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]][([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]]((**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([][[]]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**+**{}**+**[])[**+!!**[]]**+**([]**+**[][(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]][([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]]((**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([][[]]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**+**{}**+**[])[**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]])())[**!+**[]**+!!**[]**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]])()(([]**+**{})[**+**[]])[**+**[]]**+**(**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+**[])**+**(**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+**[]))**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]]**+**[][(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]][([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]]((**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([][[]]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([][[]]**+**[])[**+**[]]**+**([][[]]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**+**{}**+**[])[**+!!**[]]**+**([]**+**[][(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]][([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]]((**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([][[]]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**+**{}**+**[])[**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]])())[**!+**[]**+!!**[]**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]])()([][(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]][([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]]((**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([][[]]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**+**{}**+**[])[**+!!**[]]**+**([]**+**[][(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]][([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([]**+**{})[**+!!**[]]**+**(**!!**[]**+**[])[**+!!**[]]]((**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([][[]]**+**[])[**+**[]]**+**(**!!**[]**+**[])[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**!**[]**+**[])[**!+**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([]**+**{})[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**(**+**{}**+**[])[**+!!**[]]**+**(**!!**[]**+**[])[**+**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]**+!!**[]**+!!**[]]**+**([]**+**{})[**+!!**[]]**+**([][[]]**+**[])[**+!!**[]])())[**!+**[]**+!!**[]**+!!**[]]**+**([][[]]**+**[])[**!+**[]**+!!**[]**+!!**[]])()(([]**+**{})[**+**[]])[**+**[]]**+**(**!+**[]**+!!**[]**+**[])**+**(**+!!**[]**+**[]))

居然能运行，牛逼的隐式转换



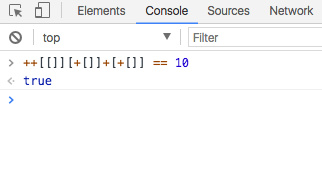


**18、parseInt(0.0000008) === 8**





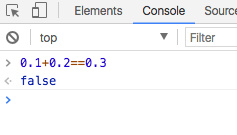
**19、++[[]][+[]]+[+[]] == 10**



强大的隐式转换，23333



**20、0.1 + 0.2 == 0.3**



0.1 +0.2 == 0.3 竟然是不成立的。。。。所以这就是为什么数据库存储对于货币的最小单位都是分。

简单说，0.1和0.2的二进制浮点表示都不是精确的，所以相加后不是0.3，接近（不等于）  
0.30000000000000004。

所以，比较数字时，应该有个宽容值。ES6中这个宽容值被预定义了：Number.EPSILON。

**21、最短的代码实现数组去重**

[...**new** Set([1, "1", 2, 1, 1, 3])]

前不久面试阿里就问了这道题，哈哈，所以也写上一下

**22、用最短的代码实现一个长度为m(6)且值都n(8)的数组**

Array(6).fill(8)

这个够短了吧，好像是当初哪里看到的一个面试题，就自己想到了ES6的一些API

**23、短路表达式**

条件判断

**var** a **=** b **&&** 1

*// 相当于*

**if** (b) {

a **=** 1

} **else** {

a **=** b

}

**var** a **=** b **||** 1

*// 相当于*

**if** (b) {

a **=** b

} **else** {

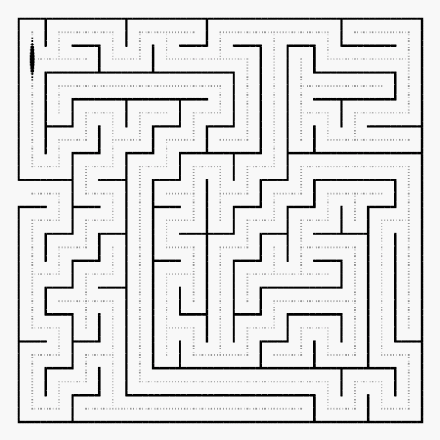
a **=** 1

}

**24、JavaScript版迷宫**



逃出迷宫，2333

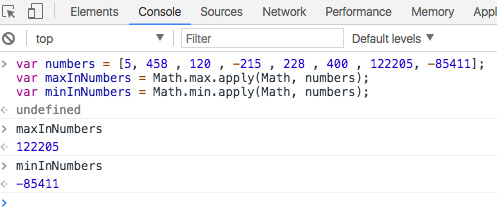


**25、取出一个数组中的最大值和最小值**

**var** numbers **=** [5, 458 , 120 , **-**215 , 228 , 400 , 122205, **-**85411];

**var** maxInNumbers **=** Math.max.apply(Math, numbers);

**var** minInNumbers **=** Math.min.apply(Math, numbers);



**26、将argruments对象转换成数组**

**var** argArray **=** Array.prototype.slice.call(arguments);

或者ES6：

**var** argArray **=** Array.from(arguments)

**27、javascript高逼格之Function构造函数**

很多JavaScript教程都告诉我们，不要直接用内置对象的构造函数来创建基本变量，例如var arr = new Array(2); 的写法就应该用var arr = [1, 2];的写法来取代。

但是，Function构造函数（注意是大写的Function）有点特别。Function构造函数接受的参数中，第一个是要传入的参数名，第二个是函数内的代码（用字符串来表示）。

**var** f **=** **new** Function('a', 'alert(a)');

f('jawil'); *// 将会弹出窗口显示jawil*

这种方式可以根据传入字符串内容来创建一个函数 是不是高大上？！



**28、从一个数组中找到一个数，O(n)的算法，找不到就返回 null。**

正常的版本:

**function** find (x, y) {

**for** ( **let** i **=** 0; i **<** x.length; i**++** ) {

**if** ( x[i] **==** y ) **return** i;

}

**return** **null**;

}

**let** arr **=** [0,1,2,3,4,5]

console.log(find(arr, 2))

console.log(find(arr, 8))

结果到了函数式成了下面这个样子（好像上面的那些代码在下面若影若现，不过又有点不太一样，为了消掉if语言，让其看上去更像一个表达式，动用了 ? 号表达式）：

*//函数式的版本*

**const** find **=** ( f **=>** f(f) ) ( f **=>**

(next **=>** (x, y, i **=** 0) **=>**

( i **>=** x.length) **?** **null** **:**

( x[i] **==** y ) **?** i **:**

next(x, y, i**+**1))((...args) **=>**

(f(f))(...args)))

**let** arr **=** [0,1,2,3,4,5]

console.log(find(arr, 2))

console.log(find(arr, 8))

[如何读懂并写出装逼的函数式代码](http://link.zhihu.com/?target=http%3A//coolshell.cn/articles/17524.html)

*原文：*[*这些JavaScript编程黑科技，装逼指南，高逼格代码，让你惊叹不已*](http://link.zhihu.com/?target=https%3A//github.com/jawil/blog/issues/24)