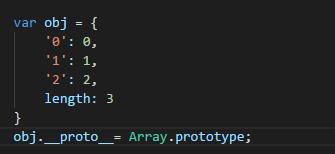
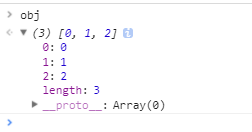
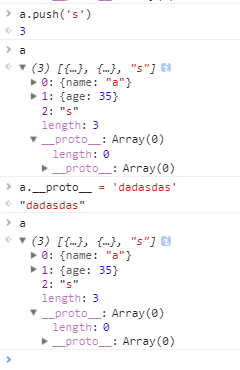
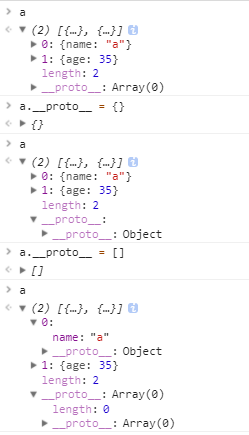
**1.** 字符串1.indexOf( 字符串2 ) indexOf是查看 字符串1里面是否含有括号里面字符串2，返回找到字符串2的索引，如果没有返回 -1；

**2.**标签里面可以自定义属性；

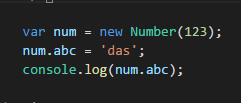
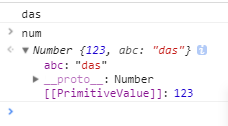
**3.** 数组slice的方法返回的是一个新数组，地址跟原数组不一样，但是如果数组里面有引用值的话引用值地址是一样的；

**4.**加\_\_proto\_\_可以把obj变成数组，还有改变原对象或数组的\_\_proto\_\_时，必须改成对象或数组的形式，改编成字符串是改变不了的；

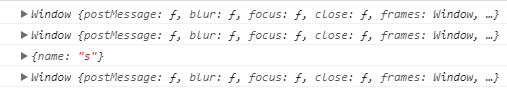
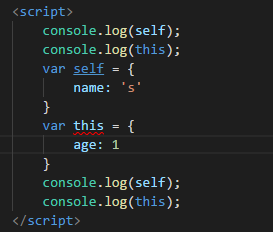
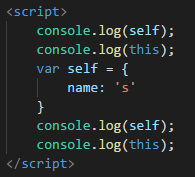


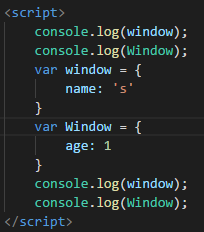
**5.**注意：包装类这样赋给别人后

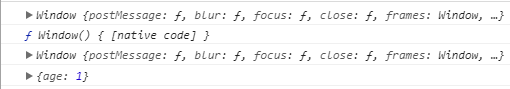
 

**6.**Relative:只根据自身进行定位，不管父级有没有定位

**7.**Javascript 中this和 self都指向window 但是 this改变不了 （会有报错），但是self可以改变，还有window也是改变不了的，但是Window可以改变

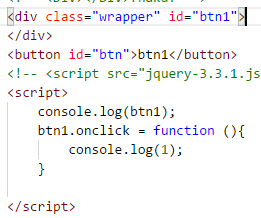
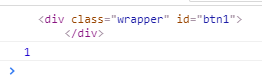






**8.**Img在载入页面后，再改变src是不好使的

**9.**有id值的可以直接这样绑定事件

**10.**Bind跟call和apply 一样改变this指向，但是它不会执行;

**11.**Var oImg = new Image()

oImg.onload = function ()｛

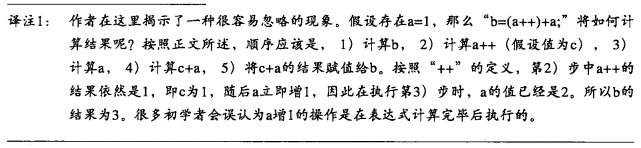
console.log(‘haha’);

Window.body.append(oImg);

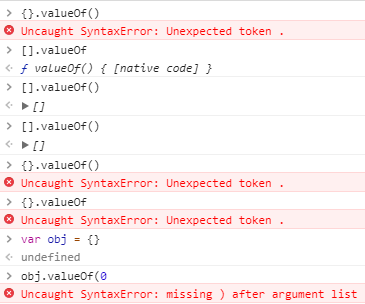
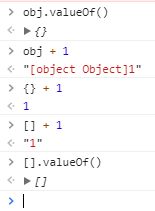
｝

oImg.src = ‘images/1.jpg’;

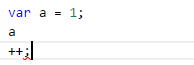
上述代码中onload只要src加载完了就可以打印出haha，而不是oImg添加到页面渲染完才触发onload;还有设置oImg.width/height接受的数字，如：oImg.Width = ‘50’或oImg.Width = 50;

**12.**

**13.**有趣的问题(valueof方法和toString()方法)

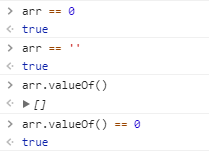
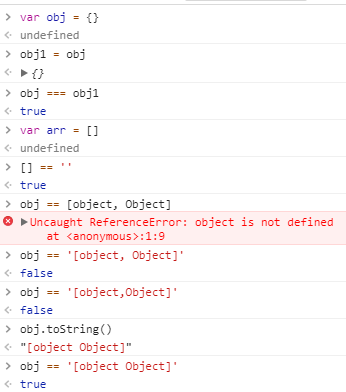
 

**14.**有趣的问题(a++,++a)

**15.**有趣的问题(对象，数组，字符串的比较)，通常字符串与数组和对象比较是对象都是看情况调用toString和 valueOf方法

**16.** padding 在设置百分比时都是相对于父级的宽度的 ；line-height不能设置百分比；margin-top/margin-bottom设置百分比是按照父级的高度，另外两个是按照父级宽度

**17.**

atan 和 atan2 都是求反正切函数，如：有两个点 point(x1,y1), 和 point(x2,y2);

那么这两个点形成的斜率的角度计算方法分别是：

float angle = atan( (y2-y1)/(x2-x1) );

或

float angle = atan2( y2-y1, x2-x1 );

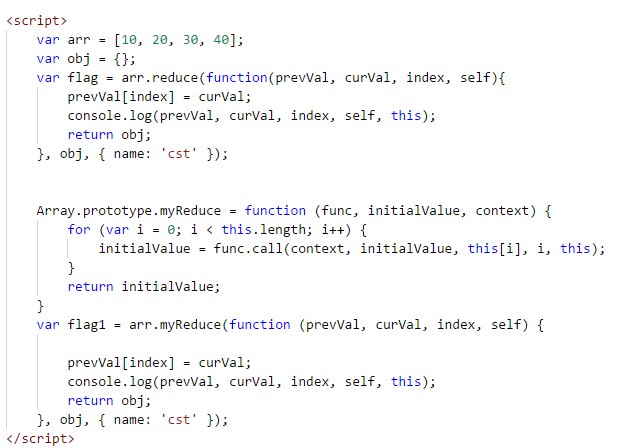
atan 和 atan2 区别：

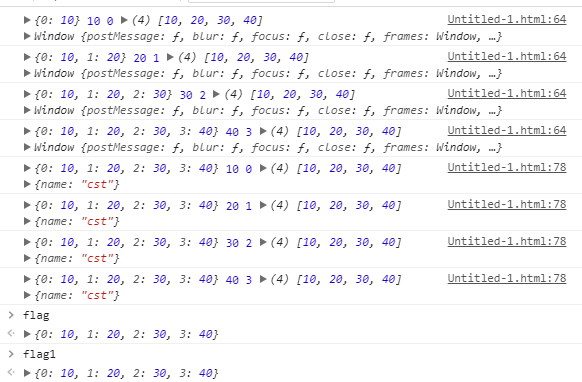
1：参数的填写方式不同；

2：atan2 的优点在于 如果 x2-x1等于0 依然可以计算，但是atan函数就会导致程序出错；

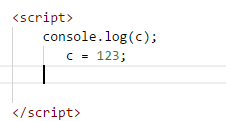
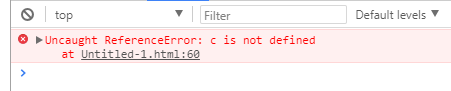
结论： atan 和 atan2函数，建议用 atan2函数；（来自网络摘抄）

**18**.一个自己封装的myReduce和数组本身的reduce（思彤哥课上封装的和原reduce两个不一样）

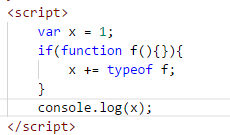




**19.**以下出现错误的原因是：c是暗示全局变量,预编译没他事,赋值的时候作用域没找到c就把c在全局里创建并赋值;

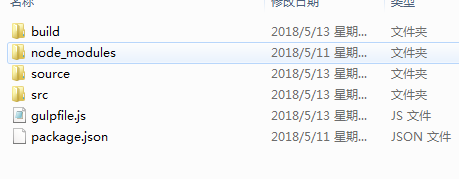
 

**20.**出错的题目：因为函数f被括号括起来就变成函数表达式了，就不是函数定义了，f就从此之后消失了

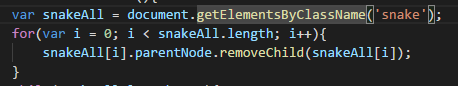
**21.**offsetHeight/offsetWidth获取的宽度高度是包含padding和border的

**22.**这样的路径从最外边开始找 ’/source/song\_1.mp3’

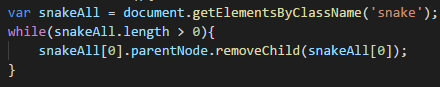


-----

**22.**要注意document取到的dom节点都是实时的，如下面去掉className为snake的所有元素节点的做法是错误的，因为循环一次snakeAll的长度就减少;

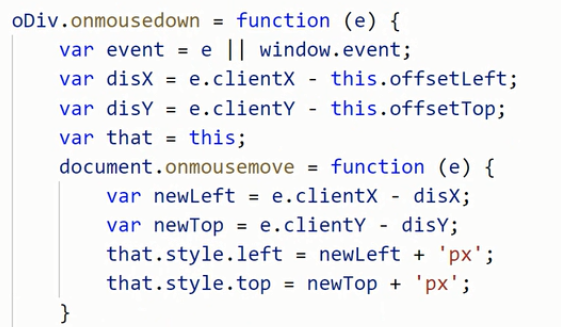
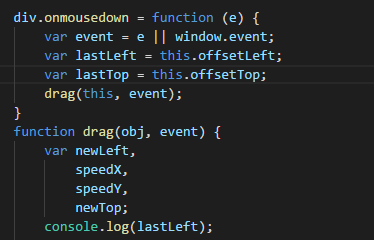
.

正确做法：



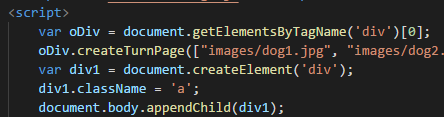
**23．**Div[0].style.offset…可以访问到css中的属性，访问的是最终属性，返回的数据类型是Number;

3.在传this可以这样传 ; 4.下面这样访问不到lastLeft;

**24.**在做点击事件时要考虑到连续点击可能出现的BUG;

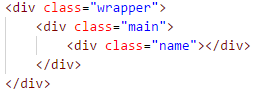
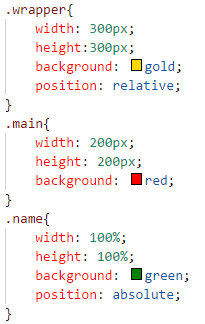
**25.**在创建元素是var的变量就代表那个元素的节点，跟获取dom节点是一样的道理;

 C:\Users\ADMINI~1\AppData\Local\Temp\1521017304(1).png

**26.**inline-block 设置margin: 0 auto;时auto没用，span设置margin-top/bottom无效，设置margin-left/right有效;

**27.**pading设置的百分比是按照其父级的宽度，父级width为20px，子级padding为20%，那么padding的值为20\*20% = 4;

**28.**在一个div中设置width为100%，那么它是根据父级来决定其大小的，如果他有设置定位absolute那么他的width是根据其相对定位的父级来决定宽度的，还有left/right是根据其相对定位的父级的宽度width来决定的，top/bottom是根据其相对定位的父级的高度height来决定的;

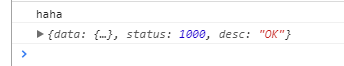
 



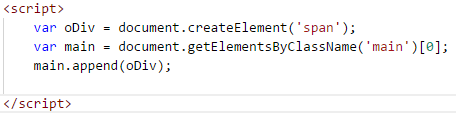
**29.一个神奇的问题：**

这个是jsonp的脚本，这里面的doJson能执行，下面src是异步执行的，大部分异步是同时触发，**定时器是个特殊的，定时器实际上可以理解为回调函数**，异步加载，可以理解为 我只是给src上面加了一个地址，当他渲染的时候会自动触发这个函数。下面先执行‘haha’然后才执行对象，这个就是时间问题了（**异步执行会出现这样的问题，说不定哪个现执行到**），谁执行时间短，谁先输出;

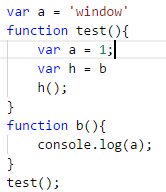




**30.**发现原生js也可以用append来添加节点，IE不行

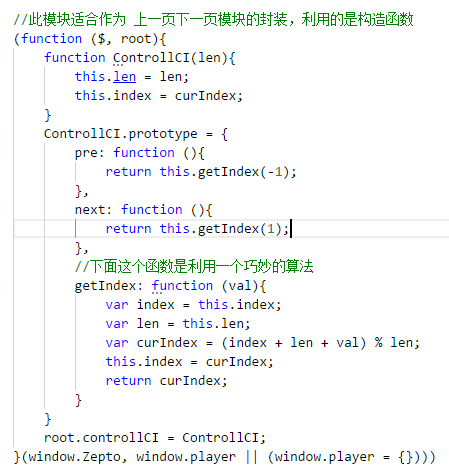


**31.**做这个的时候认为打印的应该是1，但是结果是window，因为变量h的只是在全局下函数b，等于它拿的是地址



**数学算法**

数学优化的小算法（可以运用到获取数组的index值）



**32.**div在position：absolute是设置with：100%是继承有定位父级的width的100%（这个div不会默认继承父级的宽度，要手动设置，普通的没有设置定位的父子级，子级没有设置宽度，默认是继承父级的宽度的，但是高度没设置是没有默认继承的，为0）,即使这个父级设置了box-sizing: border-box;它继承的还是父级width的值，而不是内容宽度的值，但是这个父级下没有定位的元素设置width：100%继承的是父级内容宽度的100%；

**33.**translate改变不会叠加，它是从原始位置开始移动的，下面只移动了100px

