# Js短期知识总结

## 基础知识

首先，我们接触的是变量，js里的变量都是弱变量，就是说声明变量的方式没那么严格，不像c语言那样声明变量有严格的定义，有各自不同的声明方式，这个是比较方便的。

然后就是一些书写规范（1.注意变量的大小写2.开头英文3.不能用关键字保留字例如if，else while等之类的4,.不能中文开头）。

六种数据类型：未定义undefined；布尔值boolean；字符串 string；数字number；空值null；对象object（object中包含array，function，object）；

接着是各种执行函数及用法

for循环

for循环是针对数组的多个元素连续执行或者连续判断，一个for循环可以省略若干行代码，非常实用，也非常常见。

var a=[0,1,2,3,4,5,6]

for (var i=0; i<a.length; i++) {

alert(a[i]);

}

其中i是变量名，i<是判断条件，i++是循环方式，{}里写所要执行的函数;上面的代码意思是将array中的每一个元素弹出。

if else 判断语句

var a=1；

if（a==1）{

alert（true）；

}else{

alert（false）；

}

If后的（）里写判断条件，整个函数的意思是判断a是否等于1，如果等于1，弹出true，如果不等于，弹出false；if else语句中可以嵌套多个if，加多个判断；if语句是非常实用的判断语句，在各种编程语言中都有很大的作用，js也一样；

Function

可以形象的理解为函数包，里面可以放若干个函数，将他们放在一起，在需要的时候执行，function有两种写法；

Function（）{} 匿名函数，大括号里放进需要执行的函数，这种写法是立即执行的；

Function a（）{}是一个名字为a的function，它在写完后还需要再写a（）来执行它。

while语句

while语句与for循环相似

写法1

while（判断条件）{

执行函数

}

写法2

do{执行函数}

while（判断条件）；

作用于for基本相同

parseint和parsefloat

parseInt是取整型的数字

parsefloat是取浮点型的数字，也就是说取带有小数点的数字以及整数

特点：

1.parseint可以识别16进制《前加0x或0X》和8进制的数字《前加0》，

2.parsefloat 能识别带有小数点的数字以及带有e用10的幂来计算的数字，

3.以0开头的数字parseint识别为8进制数，parsefloat识别的是十进制的数

4.他们都不能识别0字节的字符串和null，undefined

5.他们都是从第一个数字开始识别，到最后一个不是数字的字符为止，parsefloat遇到e除外

indexof（）与lastindexof（）

1.他们都是用来从字符串中找所需要的字符

2.indexof是从左往右查找字符，从0开始计数

3，lastindexof是从右往左查找字符，但是计算字符的位置也是从左往右从0开始；

4，他们都可以设置跳跃某一段字符，从哪个字符开始查找，但是结果都是从第一个开始计算{lastIndexOf（a，5）a代表声明的字符串，5代表从第五位开始找。

实例

var str="Hello world!"

document.write(str.indexOf("Hello") + "<br />")          0

document.write(str.indexOf("World") + "<br />")        -1

document.write(str.indexOf("world"))                            6

push语句应用实例

 var c=“what's up”

var a= new Array（）；

a.push（indexOf（p））；

a=[8];

常用js事件

Onclick；onmouseover；onmouseout；onmousedown；onmouseup；onscroll，onresize；

innerHTML与innerText与value的区别

innerHTML是针对一个元素内部添加内容的标签，可以用innerHTML添加标签及标签里的内容，value只是针对有value属性的标签，innerText只能给标签中加文本内容，适用大部分标签；

continue与break

continue用于for循环时会让for循环跳过一次执行

break则直接打断停止

代码示例

for （var i=0; i<5; i++）{

    if (i==2){

        continue;

       alert(i);

    }

}

当i为2 的时候，不会弹出2，直接继续执行for

for （var i=0; i<5; i++）{

     if (i==2){

         break；

         alert(i);

    }

}

当i为2时，直接停止for循环

关于传参的问题

传参的作用是将函数需要的参数通过从外到内的方式传递进去，这样可以将相似的function写成通用类，节省代码，理清思路

例：

function a（参数1，参数2）{

执行函数（参数1，参数2）

}

a（参数1值1，参数2值1）；

a（参数1值2，参数2值2）；

下面a的两次执行函数，充分利用了一个function包，只改变参数值，将同一个函数包以不同的内容执行了两次

从console中找bug

当使用console.log(……)时，会把括号内的字符串输出到控制台，当然，在IE中这个是没有的，要报错。相同的还有console.info(……),console.debug(……)，区别是有些能通过控制台的输出点进去对应的语句，适用于chrome和firefox

## 学习过程中所遇问题

1. ie9不识别javascript里的backgroundPositionX，识别background-position-x，但是ie6不识别backgroundPositionX；
2. setinterval 时间不精确，cpu忙就会溜掉执行，settimeout比setinterval强一点点；
3. 定位仍然是比较严重的问题，因为在js阶段与定位配合实现的效果比较多，定位的多层嵌套有时候很难理解，需要记住的是position：relative；是相对于自己定位的，position：absolute；是相对于有定位的父标签定位的；
4. 参数对应，写for循环时判断的参数一定要与其里面的参数对应，不然就没有效果，同时要想在for循环里加function再次用到外层for循环的参数，可以使用闭包的方法省略function里再次写for循环，例如：

var pag=-1;

for (var i=0; i<navs.length; i++) {

navs[i].onclick=(function (i) {

return function () {

pag=i;

shownav();

}

})(i);

}

此段代码首先是用闭包的方法让function里的i同时与for循环里的i对应，然后将pag里的i传出；

在写大图滚动时，第一张图不能加上所要的滚动相应效果，此时给互相联系的全局变量声明从-1（var index=-1;）开始就可以了；但是这样写还有一个问题：当滚动刚开始的时候不能给第一张图加上效果，此时可以在全局中将相应的效果写上就可以了，例如：

navs[0].className="mouse";

navsimg[0].className="mouseimg"

setTimeout(function () {

bands[0].style.display="block";

},1000);

1. Js中不能识别css中的%，所以在css中设置了相应的百分比值时，需要在js中重新计算赋值，或者是改进库中的内容；
2. 用onresize可以很容易的解决元素位置自适应的问题，同时window.innerHeight与window.innerWidth可以获取到当前窗口的内宽高，但是当window.innerHeight这种方法失效时，可以用另一种获取方法document.documentElement.clientHeight与document.documentElement.clientHeight来实现；