判断单，多张图片加载是否完成

（1）、单张图片（图片在文档中）

<img id='xiu' src="http://www.daqianduan.com/wp-content/uploads/2014/11/hs-xiu.jpg">

//js

$(document).ready(function(){

//jquery

$('#xiu').load(function(){

// 加载完成

});

//原生 onload

var xiu = document.getElementById('xiu')

xiu.onload = xiu.onreadystatechange = function(){

if(!this.readyState||this.readyState=='loaded'||this.readyState=='complete'){

// 加载完成

}

};

})

注：  
1、IE8及以下版本不支持onload事件，但支持onreadystatechange事件；  
2、readyState是onreadystatechange事件的一个状态，值为loaded或complete的时候，表示已经加载完毕。  
3、以下内容省略兼容

（2）、单张图片（图片动态生成）

//js

var xiu = new Image()

xiu.src = 'http://www.daqianduan.com/wp-content/uploads/2014/11/hs-xiu.jpg'

xiu.onload = function(){

// 加载完成

}

（3）、单张图片（结合ES6 Promise）

//js

new Promise((resolve, reject)=>{

let xiu = new Image()

xiu.src = 'http://www.daqianduan.com/wp-content/uploads/2014/11/hs-xiu.jpg'

xiu.onload = function(){

// 加载完成

resolve(xiu)

}

}).then((xiu)=>{

//code

})

（4）、多张图片

var img = [],

flag = 0,

mulitImg = [

'http://www.daqianduan.com/wp-content/uploads/2017/03/IMG\_0119.jpg',

'http://www.daqianduan.com/wp-content/uploads/2017/01/1.jpg',

'http://www.daqianduan.com/wp-content/uploads/2015/11/jquery.jpg',

'http://www.daqianduan.com/wp-content/uploads/2015/10/maid.jpg'

];

var imgTotal = mulitImg.length;

for(var i = 0 ; i < imgTotal ; i++){

img[i] = new Image()

img[i].src = mulitImg[i]

img[i].onload = function(){

//第i张图片加载完成

flag++

if( flag == imgTotal ){

//全部加载完成

}

}

}

（5）、多张图片（结合ES6 Promise.all()）

let mulitImg = [

'http://www.daqianduan.com/wp-content/uploads/2017/03/IMG\_0119.jpg',

'http://www.daqianduan.com/wp-content/uploads/2017/01/1.jpg',

'http://www.daqianduan.com/wp-content/uploads/2015/11/jquery.jpg',

'http://www.daqianduan.com/wp-content/uploads/2015/10/maid.jpg'

];

let promiseAll = [], img = [], imgTotal = mulitImg.length;

for(let i = 0 ; i < imgTotal ; i++){

promiseAll[i] = new Promise((resolve, reject)=>{

img[i] = new Image()

img[i].src = mulitImg[i]

img[i].onload = function(){

//第i张加载完成

resolve(img[i])

}

})

}

Promise.all(promiseAll).then((img)=>{

//全部加载完成

})

当网页加载完毕

js方法：

window.onload=function(){   
  
var userName="xiaoming";   
  
alert(userName);   
}

jQuery方法，需要引用jQuery文件

$(document).ready(function(){   
  
var userName="xiaoming";   
  
alert(userName);   
});

判断是否为IE浏览器

if(!!window.ActiveXObject || "ActiveXObject" in window){

});

Js动画知识点

**知识点一：速度动画。**

  1.首先第一步实现速度运动动画，封装一个函数，用到的知识是setInterval(function(){

oDiv.style.left=oDiv.offsetLeft+10+"px";  
　　},30).

　对于这里为什么要用到offsetLeft，我特意百度了一下，我得到的有用信息是：

　　a.offsetLeft和left的相同之处都是表示子节点相对于父节点的左位置。  
　　b.但是left是既可以读又可以写的，而offsetLeft是只读的；  
　　c.并且offsetLeft是没有单位的，获取子节点位置的时候后面不带px。

　2.让运动的节点停止下来，这里我们用if语句做一个验证，如果offsetLeft==0，clearInterval（timer），这里的timer应该事先初始化=null，然后将之前的那个运动动画赋值给它。

　　3.这里有一个问题，如果在运动结束之前再次触发的运动，那么运动的速度会累加，这里，只要在整个运动开始之前，clearInterval（timer）就可以了。

　　4.设置移入移除效果，给运动设置参数，一个是速度speed，一个是目标位置iTarget，我们发现速度也是可以通过ITarget的位置来判断的，所以只需要一个参数即可。

**知识点二：透明度渐变**

　　1.其实，跟刚刚是差不多的，只不过ITarget的值为透明度而已，流程还是清除定时器再开一个定时器判断等等。

　　2.定义一个参数alpha=透明度，注意定时器里面应该这样写：

alpha+=speed；  
　　oDiv.style.filter='alpha(opacity:'+alpha+')';         //这是非常重要的一种方法，注意是这样写的  
　　oDiv.style.opacity=alpha/100;            //注意不要忘记除以100

1. 以上都是行内样式。

**知识点三：缓冲运动**

　　1.缓冲运动就是距离越大，速度就愈大，距离越小，速度就越小，即速度与距离有关。

　　2.根据上面的说法，对速度进行重新的定义，一开始速度为0，而现在：

　var speed=iTarget-oDiv.offsetLeft;

　　重新定义定时器：  
　　oDiv.style.left=oDiv.offsetLeft+speed+'px';

　　此时我们发现速度太大了，可以这样：  
　　var speed=(iTarget-oDiv.offsetLeft)/10;

　　3.此时会有一个严重的问题，因为屏幕的最小单位为px，所以会出现最终的left值为小数，而不为目标的iTarget,可以通过判断来解决，这里要引入Math.floor（），这是向下取整，同样还有Math.ceil()，这是向上取整。在定义speed后我们这样写：  
　　speed=speed>0?Math.ceil(speed):Math.floor(speed);

这样就可以完全保证速度都是整数，并且在临界值上都为0。

**知识点四：多物体运动**

　　1.先获取所有的物体，形成一个数组，然后再用for循环来做（这种方法多么经典呀！），在for循环中添加节点事件，在函数中可以用this代替当前的节点，eg：startMove（this，iTarget）,定义函数时startMove（obj，iTarget）.

　　2.取当前宽度offsetWidth的时候就要用obj的值了。

　　3.当鼠标移动特别快的时候，节点的宽度未能恢复原状，这是因为定时器是大家公用的定时器，上一个节点还未恢复原状下一个节点就已经清除定时器了，解决的办法就是给每一个节点加一个index，就是在上面的for循环中加上aDiv[i].timer=null;然后再定义函数中用obj.timer替换timer。由此我们要注意共用定时器会出事的。

　　4.透明度的运动中，alpha代替了speed，但是即使定时器不共用，多物体的运动也会出现问题，这是因为alpha公用的原因，导致各物体相互撕扯，解决办法就是可以像timer那样在for循环中给每个节点都分配alpha。

      总结：解决冲突问题，要么初始化，要么个性化。

**知识点五.获取样式**

　　1.在改变节点宽度（移入大，移除小）的定时器中，如果给节点添加一个border边框，那么在移入的时候比目标节点小，移出的时候比目标节点大。注意width+padding+scrollbar（滚动条）+border，所以原因就是每次offset都会增加border\*2-（定时器中每次减少的数值）。

　　2.解决以上问题的办法就是在行内写入width，并且使用parseInt(oDiv.style.width)代替offsetLeft，但是并不能总是写在行内，所以我们定义一个函数，获取链入样式：

　function getStyle(obj,attr){  
　　　　if(obj.currentStyle){  
　　　　　　return obj.currentStyle[attr];　　　                    //ie浏览器　  
　　　　}  
　　　　else{  
　　　　　　return getComputerStyle(obj,false)[attr];          //其他浏览器  
　　　　}  
　　}

　3.对于font-size这种，在js里只有fontSize这一种写法。

**知识点六：任意属性值**

　　1.所有的offset-的都会有小bug，要用getStyle函数，这个函数经常和parseInt（）一起用，并通常用变量保存。

　　2.在写style.width的时候，也可以写成style['width']。

　　3.对于多物体的属性值调整，可以将样式作为参数封装一下，这样多物体属性的函数就包括了（obj,attr,iTarget）这三个属性值。

　　4.以上的这个运动框架是不适合于透明度变化的，因为透明度都是小数的，因为两个原因，第一是parseInt，第二是attr=...+px，这里我们可以用一个if判读来对透明度进行单独处理，将parseInt换成parseFloat，去掉px。

　　5.计算机本身是有一个bug的，0.07\*100并不等于7，所以我们引入一个函数是Math.round()，这是一个四舍五入的值。

**知识点七：链式运动**

　　1.引入move.js框架。

　　2.传入一个回调函数fn（），用if判断，如果有fn（），那么执行fn（）。

**知识点八：同时运动**

　　1.如果写两个运动函数来控制同时运动，会发生函数覆盖。

　　2.使用json这个知识点，json的循环是使用for（i in json)，运动函数的参数为obj，json，fn。

　　3.没有iTarget这个值了，取而代之的是json[attr].

　　写到这里，就已经完全结束了，希望大家能够喜欢。也希望对大家学习js运动动画能够有所帮助。

**字体**

font-family: STXihei, "Microsoft YaHei";

font-family: STXihei, "华文细黑", "Microsoft YaHei", "微软雅黑";

**Html5ie9兼容代码：**

<!--[if lt IE9]>

<script src="http://cdn.static.runoob.com/libs/html5shiv/3.7/html5shiv.min.js"></script>

<![endif]-->

载入后，初始化新标签的CSS

article,aside,dialog,footer,header,section,footer,nav,figure,menu{display:block}

**Css路径**

"." 代表当前所在目录，相对路径。如:<a href="./abc">文本</a>或<img src="./abc" />；

".." 代表上一层目录，相对路径。如:<a href="../abc">文本</a>或<img src="../abc"/>；

"../../" 代表的是上一层目录的上一层目录，相对路径。 如:<img src="../../abc" />；

"/" 代表根目录,绝对路径。 如：<a href="/abc">文本</a>或<img src="/abc"/>；

"D:/abc/" 代表根目录,绝对路径。"." 代表当前所在目录，相对路径。如:<a href="./abc">文本</a>或<img src="./abc" />；

".." 代表上一层目录，相对路径。如:<a href="../abc">文本</a>或<img src="../abc"/>；

"../../" 代表的是上一层目录的上一层目录，相对路径。 如:<img src="../../abc" />；

"/" 代表根目录,绝对路径。 如：<a href="/abc">文本</a>或<img src="/abc"/>；

"D:/abc/" 代表根目录,绝对路径。