**十五 、javascript-运动**

**一、匀速运动**

电影 25帧 人眼能识别 18帧

**1.运动中的一些Bug**

(1)运动没有停止

(2)速度取某些值的时候，没有办法停止

(3)到达目的值以后，点击，还会去动

需要将运动停止分开 if...else

(4)重复点击速度会加快

保证运动期间只有一个定时器在运行。

【解决】每次启动定时器之前，将上一次的定时器给清除掉

**2.运动框架**

运动框架：前辈程序员们，经过长期的开发经验的累计，得出的一套写运动的规律，这套规律，叫做框架。

运动框架：写运动的套路。

**3.匀速运动**



**【存在的问题】**

(1)没有停的刚好

(2)停止的条件非常的不合理。

**【解决】**

泊车：一般情况，停的差不多就算停好了。

停的刚刚好的位置（物体离目标的距离的绝对值小于速度的绝对值）。

Math.abs((iCur - iTarget)) < Math.abs(speed);

【注】

if(oDiv.offsetLeft > iTarget){

speed = -7;

}else{

speed = 7;

}

//让物体可在目标两边运动；



**4.图片的淡入淡出**

【使用中间值替代透明度进行计算】



**二、缓冲运动**

**1.缓冲运动**

距离越远，速度越大

距离速度成正比

speed = (iTarget - iCur) //目标值-当前值

计算机 只能最小走1像素

speed = (iTarget - oDiv.offsetLeft) / 8;// 速度以差值除以8最合适；

【Math.ceil】向上取整，速度不足1时，取1，可保证最小速度为1；



**2.缓冲运动-反向**

【Math.floor】向下取整，速度不足-1时，取-1，可保证最小速度为反向的1；



**3.缓冲运动-进阶**

【将向上取整函数与向下取整函数通过三目运算写在一起】

var speed = (iTarget - oDiv.offsetLeft) / 8;

speed = speed > 0 ? Math.ceil(speed) : Math.floor(speed);



**4.实例**

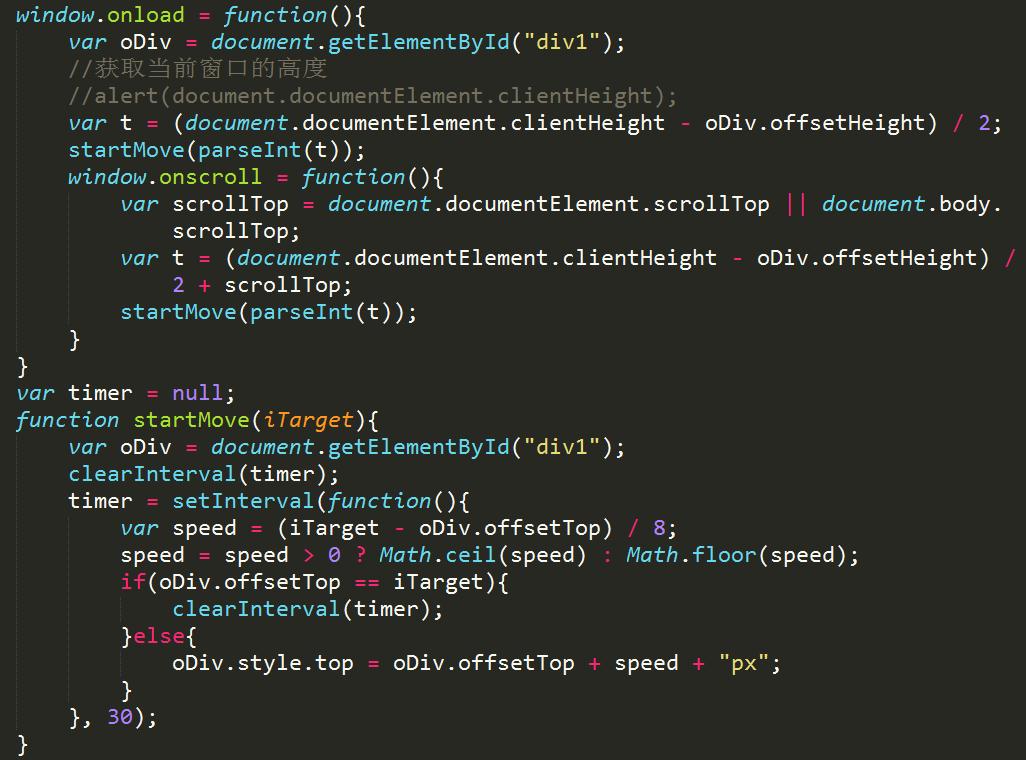
【缓冲菜单】：实现当窗口刷新和滚动时，物体可一直处于中间位置。

(1)获取当前窗口的高度/宽度：

document.documentElement.clientHeight；

(2)获取当前窗口滚动高度：

scrollTop = document.documentElement.scrollTop || document.body.scrollTop;



**三、多物体运动框架**

**1.多个运动的物体共用一个定时器。**

【解决】保证每一个运动物体，都有一个自己的定时器。

【结论】定时器不能共用。

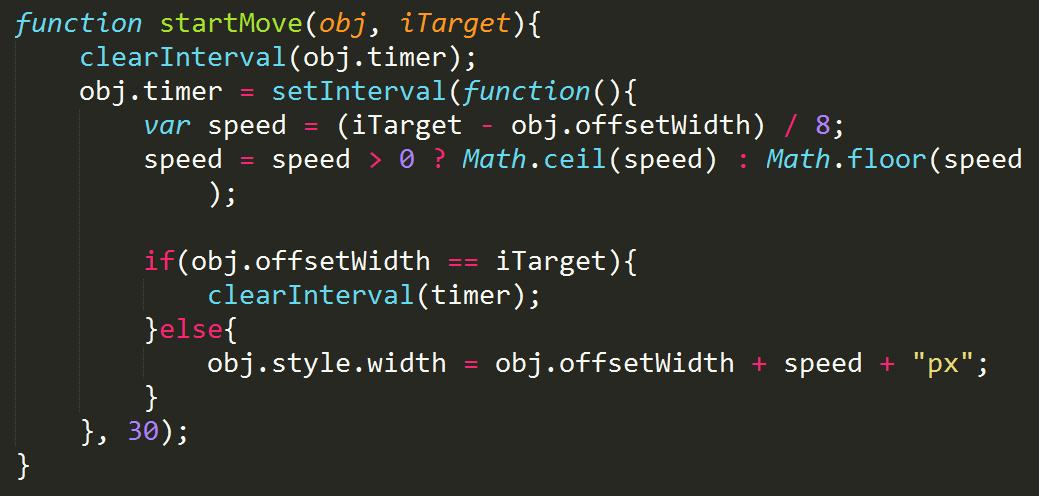
【作用】

(1)实现一个多物体，多样式运动框架

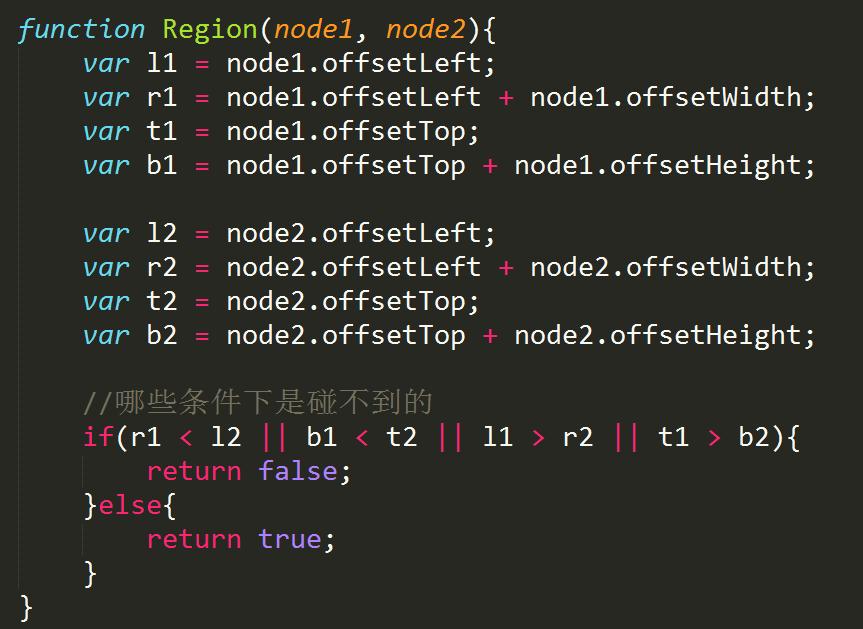
(2)实现碰撞小球

(3)打砖块

【通过在定时器timer前面加参数obj，来给每一个obj都添加一个定时器】



**2.【碰撞检测】**



**3.多物体的淡入淡出**

【结论】多物体的透明度发生变化的时候，中间变量不能共用。

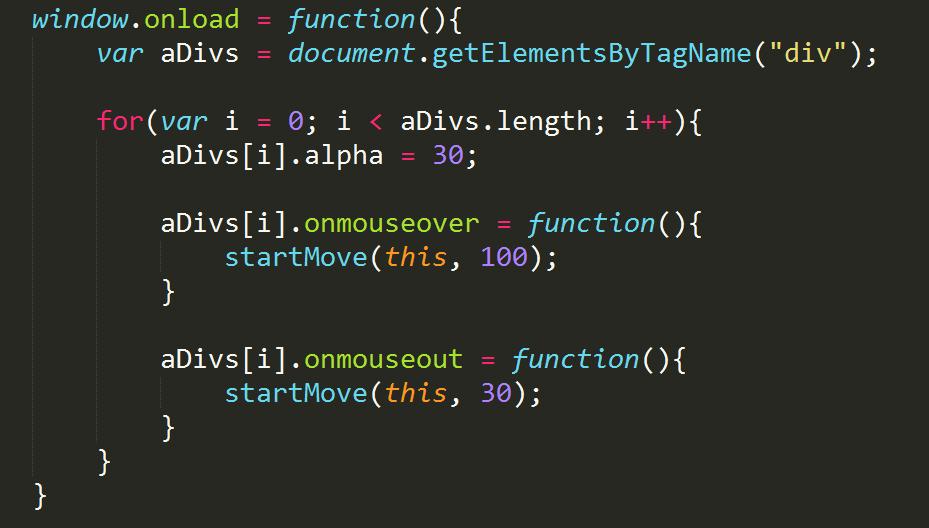
【注】不但定时器不能共有，所有的中间变量都不能共用。

【例】此时的透明度（

opacity: 0.3;

filter: alpha(opacity=30);）

实现所依靠的中间变量alpha需要通过循环给每个物体添加。





**四、关于offset系列**

【注】运动的时候，使用offset系列会有隐患。

比如offsetWidth的值会包括边框的宽度，而style.width的宽度是不包括边框的。

【注】在运动中，尽量避免使用offset系列，可使用getStyle()，即

**//获取当前有效样式**

**function getStyle(element, attr){**

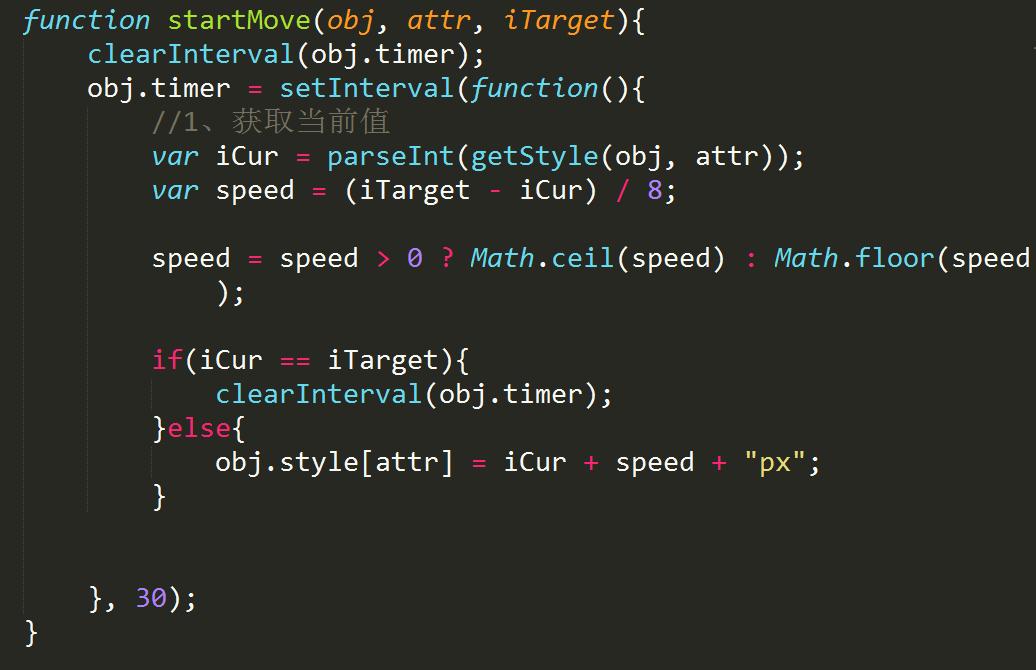
**return element.currentStyle ? element.currentStyle[attr] : getComputedStyle(element)[attr];**

**}**

**五、多物体多样式运动**

**1.【将元素和属性都作为函数的参数】**





**2.加入透明度属性【需要单独判断】**





**六、链式运动**

1、移入 先将宽变成500，接下来高变成500，再将透明度变成30

2、移出 按原路径恢复原状

【注】当一个动画结束以后，再去完成另一个动画

【关键点】什么时候，动画执行完成。

【注】把这种将函数当做参数传入的操作，叫做回调函数。





**七、多物体多样式同时运动**

**【注】如果同时想设置多个样式，通过json对象来进行传参。**

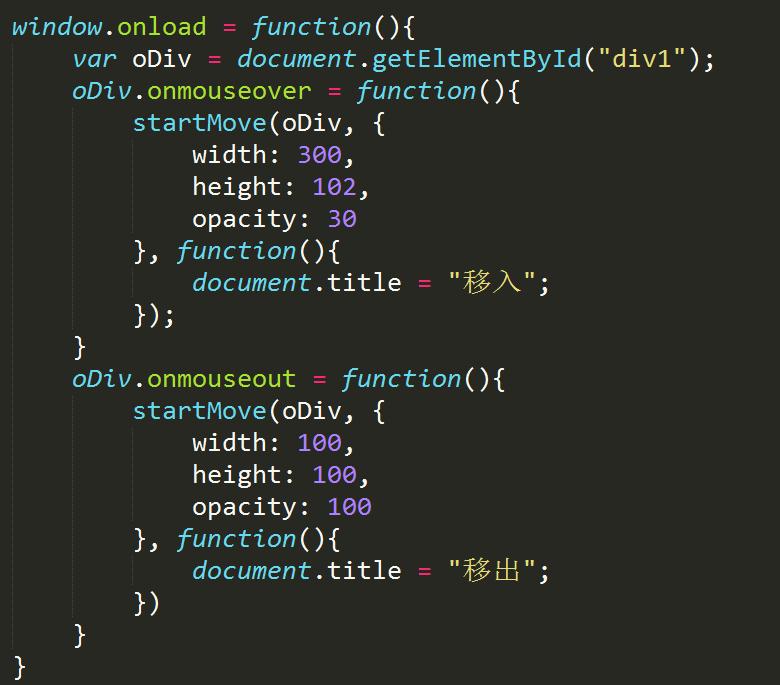


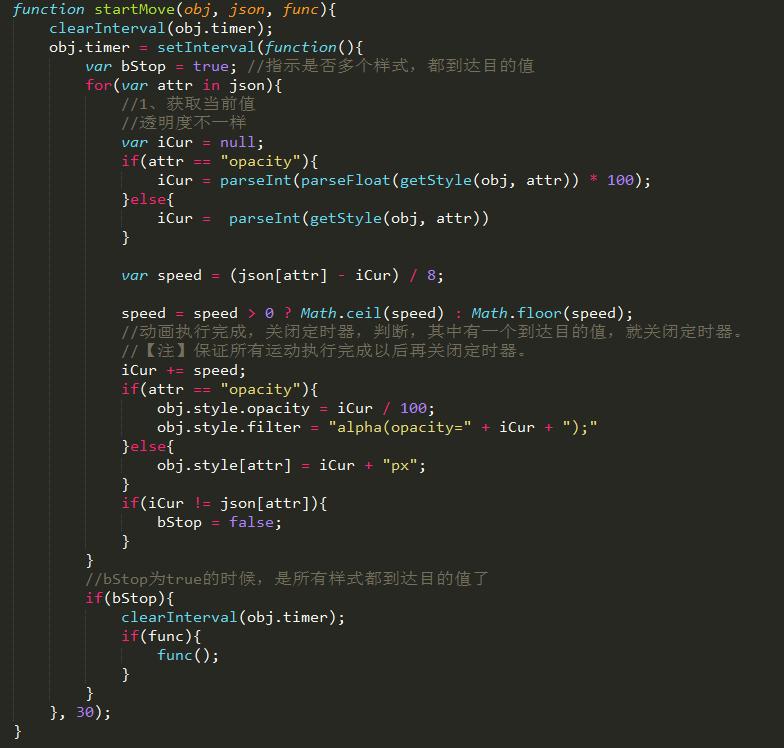
**多物体多样式同时运动**

(1)设定一个值，如bStop = true; //用来指示是否多个样式，都到达目的值

(2)在循环末尾加入判断，如果有未完成的运动，则bStop = false;

(3)在循环外判断bStop的值，如为true，则此时所有物体的运动都已完成，关闭定时器。





**八、运动总结**

**1、匀速运动 介绍运动框架基本信息**

<1>运动和停止分开 if...else

<2>每次启动定时器之前，必须将上一次定时器关闭掉

<3>运动停止条件

Math.abs(iCur - iTarget) < Math.abs(speed);

将物体挪动到目的值。

**2、缓冲运动 速度 = (iTarget - iCur) / 8**

startMove(iTarget)

淡入淡出的缓冲运动 透明度 中间值去进行运动

**3、多物体运动**

多个物体不能共用一个定时器 每一个运动的物体配了一个定时器 obj.timer

startMove(obj, iTarget)

**4、多物体多样式运动**

startMove(obj, attr, iTarget);

offset系列有bug隐患 getStyle获取当前样式

**5、多物体多样式运动透明度**

通过判断 对透明度和别的而样式进行了区分

**6、链式运动 运动结束的时间点 func回调函数**

startMove(obj, attr, iTarget, func)

**7、想同时将元素宽高发生变化**

完美运动

startMove(obj, json, func)

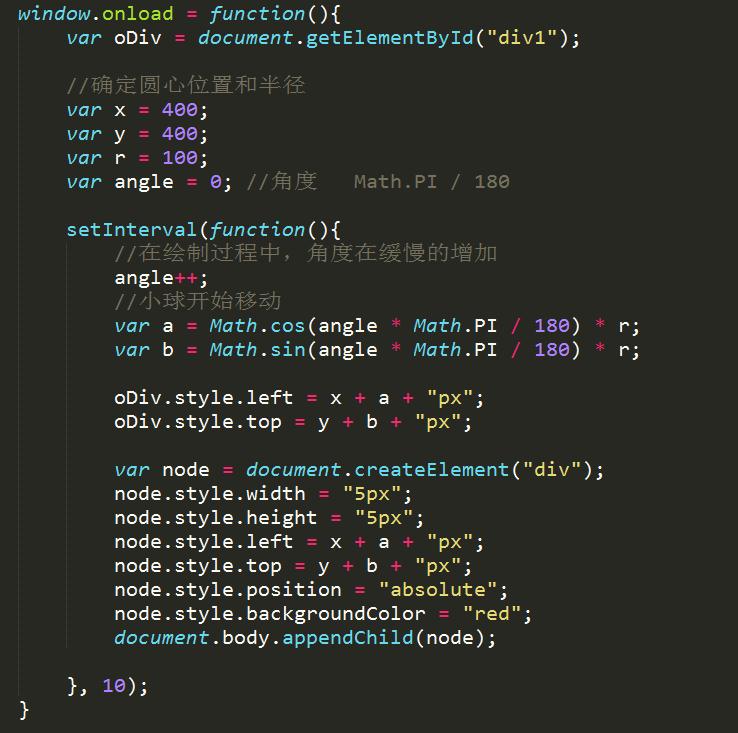
**九、圆周运动**

**实现：**

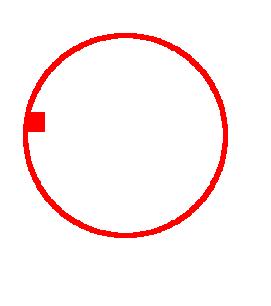
（1）确定圆心位置和半径

（2）设置角度// Math.PI / 180 弧度

（3）绘制，角度增加，小球移动

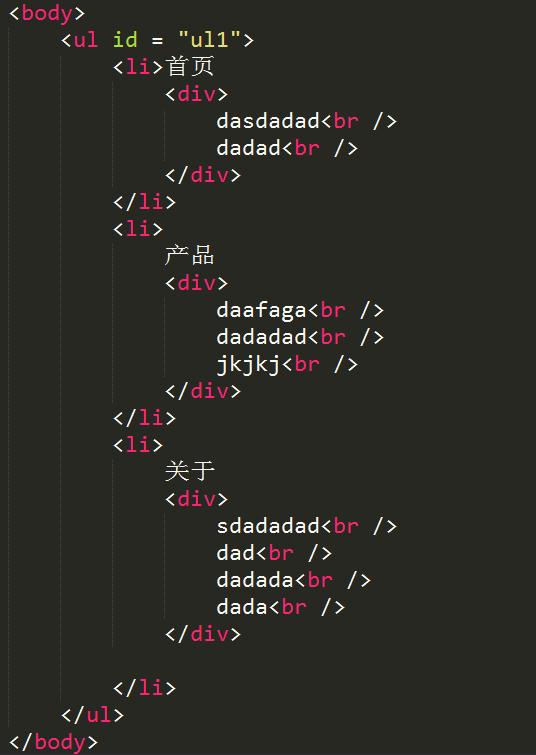


**效果:**

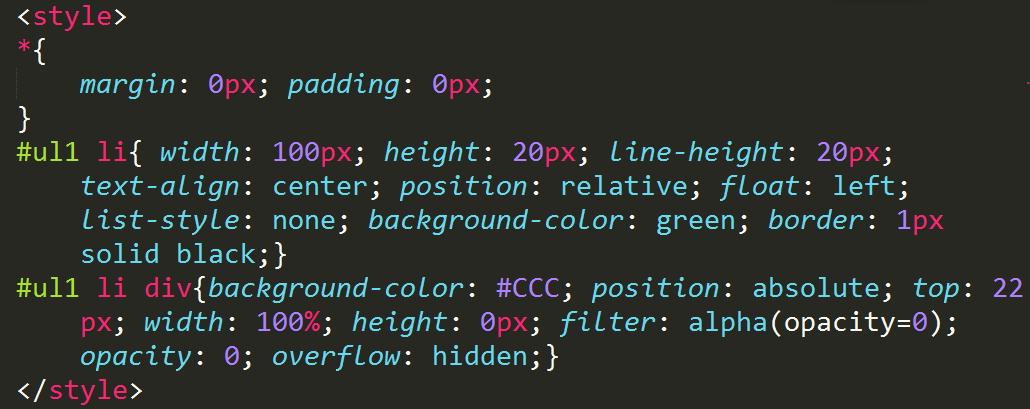


**十、伸缩菜单-淡入淡出**

**1.布局：**



**2.设置样式**



**3.javascript**

(1)找出li对应的div

(2)找到当前菜单的高，将高赋值给oDiv.iHeight；

(3)将高设置回去，开始动画。



**十一、新浪微博效果**

**要求：点击发布按钮，文字发布到显示框，后发布的在顶部，高度由0到正常，透明度opacity从0到1；**

（1）获取显示区域中div1的div，判断div1中是否存在div，如果有，则用insertBefore方法插入到原有节点之前，如果没有，则用appendChild插入

（2）开始动画，链式动画，先将高撑开，然后透明度变换。

1、记录一下当前插入的信息的高

2、将高设置成0，开始动画



**十二、图片无缝滚动（轮播图）**