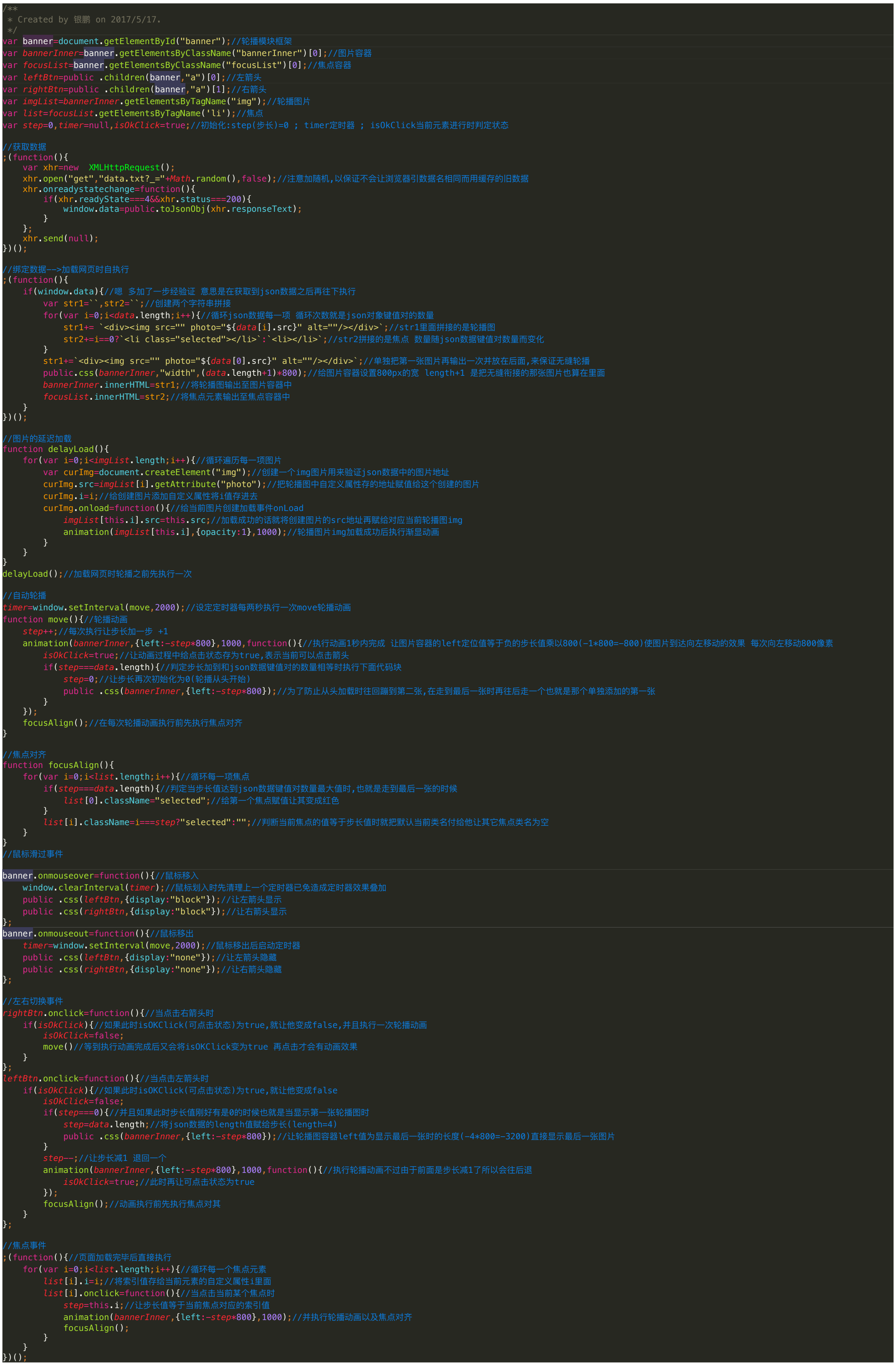

```
/**
 * Created by 银鹏 on 2017/5/17.
 */
var banner=document.getElementById("banner");//轮播模块框架
var bannerInner=banner.getElementsByClassName("bannerInner")[0];//图片容器
var focusList=banner.getElementsByClassName("focusList")[0];//焦点容器
var leftBtn=public .children(banner,"a")[0];//左箭头
var rightBtn=public .children(banner,"a")[1];//右箭头
var imgList=bannerInner.getElementsByTagName("img");//轮播图片
var list=focusList.getElementsByTagName('li');//焦点
var step=0,timer=null,isOkClick=true;//初始化:step(步长)=0 ; timer定时器 ; isOkClick当前元素进行时判定状态

//获取数据
;(function(){
    var xhr=new  XMLHttpRequest();
    xhr.open("get","data.txt?_="+Math.random(),false);//注意加随机,以保证不会让浏览器引数据名相同而用缓存的旧数据
    xhr.onreadystatechange=function(){
        if(xhr.readyState===4&&xhr.status===200){
            window.data=public.toJsonObj(xhr.responseText);
        }
    };
    xhr.send(null);
})();

//绑定数据-->加载网页时自执行
;(function(){
    if(window.data){//嗯 多加了一步经验证 意思是在获取到json数据之后再往下执行
        var str1=``,str2=``;//创建两个字符串拼接
        for(var i=0;i<data.length;i++){//循环json数据每一项 循环次数就是json对象键值对的数量
            str1+= `<div><img src="" photo="${data[i].src}" alt=""/></div>`;//str1里面拼接的是轮播图
            str2+=i==0?`<li class="selected"></li>`:`<li></li>`;//str2拼接的是焦点 数量随json数据键值对数量而变化
        }
        str1+=`<div><img src="" photo="${data[0].src}" alt=""/></div>`;//单独把第一张图片再输出一次并放在后面,来保证无缝轮播
        public.css(bannerInner,"width",(data.length+1)*800);//给图片容器设置800px的宽 length+1 是把无缝衔接的那张图片也算在里面
        bannerInner.innerHTML=str1;//将轮播图输出至图片容器中
        focusList.innerHTML=str2;//将焦点元素输出至焦点容器中
    }
})();

//图片的延迟加载
function delayLoad(){
    for(var i=0;i<imgList.length;i++){//循环遍历每一项图片
        var curImg=document.createElement("img");//创建一个img图片用来验证json数据中的图片地址
        curImg.src=imgList[i].getAttribute("photo");//把轮播图中自定义属性存的地址赋值给这个创建的图片
        curImg.i=i;//给创建图片添加自定义属性将i值存进去
        curImg.onload=function(){//给当前图片创建加载事件onLoad
            imgList[this.i].src=this.src;//加载成功的话就将创建图片的src地址再赋给对应当前轮播图img
            animation(imgList[this.i],{opacity:1},1000);//轮播图片img加载成功后执行渐显动画
        }
    }
}
delayLoad();//加载网页时轮播之前先执行一次

//自动轮播
timer=window.setInterval(move,2000);//设定定时器每两秒执行一次move轮播动画
function move(){//轮播动画
    step++;//每次执行让步长加一步 +1
    animation(bannerInner,{left:-step*800},1000,function(){//执行动画1秒内完成 让图片容器的left定位值等于负的步长值乘以800(-1*800=-800)使图片到达向左移动的效果 每次向左移动800像素
        isOkClick=true;//让动画过程中给点击状态存为true,表示当前可以点击箭头
        if(step===data.length){//判定步长加到和json数据键值对的数量相等时执行下面代码块
            step=0;//让步长再次初始化为0(轮播从头开始)
            public .css(bannerInner,{left:-step*800});//为了防止从头加载时往回蹦到第二张,在走到最后一张时再往后走一个也就是那个单独添加的第一张
        }
    });
    focusAlign();//在每次轮播动画执行前先执行焦点对齐
}

//焦点对齐
function focusAlign(){
    for(var i=0;i<list.length;i++){//循环每一项焦点
        if(step===data.length){//判定当步长值达到json数据键值对数量最大值时,也就是走到最后一张的时候
            list[0].className="selected";//给第一个焦点赋值让其变成红色
        }
        list[i].className=i===step?"selected":"";//判断当前焦点的值等于步长值时就把默认当前类名付给他让其它焦点类名为空
    }
}
//鼠标滑过事件

banner.onmouseover=function(){//鼠标移入
    window.clearInterval(timer);//鼠标划入时先清理上一个定时器已免造成定时器效果叠加
    public .css(leftBtn,{display:"block"});//让左箭头显示
    public .css(rightBtn,{display:"block"});//让右箭头显示
};
banner.onmouseout=function(){//鼠标移出
    timer=window.setInterval(move,2000);//鼠标移出后启动定时器
    public .css(leftBtn,{display:"none"});//让左箭头隐藏
    public .css(rightBtn,{display:"none"});//让右箭头隐藏
};

//左右切换事件
rightBtn.onclick=function(){//当点击右箭头时
    if(isOkClick){//如果此时isOKClick(可点击状态)为true,就让他变成false,并且执行一次轮播动画
        isOkClick=false;
        move()//等到执行动画完成后又会将isOKClick变为true 再点击才会有动画效果
    }
};
leftBtn.onclick=function(){//当点击左箭头时
    if(isOkClick){//如果此时isOKClick(可点击状态)为true,就让他变成false
        isOkClick=false;
        if(step===0){//并且如果此时步长值刚好有是0的时候也就是当显示第一张轮播图时
            step=data.length;//将json数据的length值赋给步长(length=4)
            public .css(bannerInner,{left:-step*800});//让轮播图容器left值为显示最后一张时的长度(-4*800=-3200)直接显示最后一张图片
        }
        step--;//让步长减1 退回一个
        animation(bannerInner,{left:-step*800},1000,function(){//执行轮播动画不过由于前面是步长减1了所以会往后退
            isOkClick=true;//此时再让可点击状态为true
        });
        focusAlign();//动画执行前先执行焦点对其
    }
};

//焦点事件
;(function(){//页面加载完毕后直接执行
    for(var i=0;i<list.length;i++){//循环每一个焦点元素
        list[i].i=i;//将索引值存给当前元素的自定义属性i里面
        list[i].onclick=function(){//当点击当前某个焦点时
            step=this.i;//让步长值等于当前焦点对应的索引值
            animation(bannerInner,{left:-step*800},1000);//并执行轮播动画以及焦点对齐
            focusAlign();
        }
    }
})();
```