1、Ajax请求

创建过程

a、创建Ajax对象

b、连接服务器

c、发送请求

d、接受返回

\*全局变量全局函数都属于window.

\*用一个不存在的变量：出错；用一个不存在的属性：undefined

请求状态监控（onreadystatechange事件）

readyState属性：请求状态

>0（未初始化），还没有调用open()方法；

>2（载入）已调用send()方法，正在发送请求

>3（载入完成）send()方法完成，已收到全部相应内容

>4（完成）相应内容解析完成，可以在客户端调用了

status属性：请求结果

responseText服务器返回的内容；

2、运动知识

\*当代码相同时，但是运行的条件不同，可以把判定条件的值作为参数传入。

\*多物体运动时，所有的东西都不能共用。

\*offsetWidth 等于物体宽度+物体的边宽。

\*尽量避免小数，可以使用parseInt()转换为整数。

\*Jon只能用for(…in…)进行循环。

设置对象属性：（通用）

function getStyle(obj,attr,value)

{

obj.style[attr] = value;

}

缓冲运动

a、获取属性的值（例如获取对象的opacity、width、height值）

function getStyle(obj, attr)

{

if(obj.currentStyle)

{

return obj.currentStyle[attr];

}

else

{

return getComputedStyle(obj, false)[attr];

}

}

下例子兼容了各种浏览器

b、防止出现小数（正数向上取整，负数向下取整）

var iSpeed=(iTarget-iCur)/8;

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

链式运动

a、当一个动作执行完成后，继续执行下一个运动。多次调用startMove函数。

startMove(oDiv, 'width', 300, function (){

startMove(oDiv, 'height', 300, function (){

startMove(oDiv, 'opacity', 100);

});

});

b、函数的参数格式

function startMove(obj, attr, iTarget, fn)

c、防止无参数函数时，即“startMove(oDiv, 'opacity', 100);”运行出错，则在函数内写入以下函数：

if(fn){ fn();}

完美运动

a、两个动作或两个以上的动作同时执行

b、利用Json的数据类型格式来实现完美运动

startMove(oDiv, {width: 102, height: 200, opacity: 100});

c、参数格式如下

function startMove(obj, json, fn)

d、利用循环执行函数

for(var attr in json){…}

弹性运动

弹性公式

速度+=(目标值-oDiv.offsetLeft)/5

速度\*=0.7

3、面对对象编程

1、工厂方式

function createPerson(name,sex)

{

var obj = new Object();

obj.name = name;

obj.sex = sex;

obj.showName = function()

{

alert("My name is " +this.name);

};

obj.showSex = function()

{

alert("I am a "+this.sex);

};

return obj;

}

var p1 = createPerson("huang","female");

var p2 = createPerson("zhao","male");

p1.showName();

p1.showSex();

问题：

1、每个对象都有自己的一套函数，浪费资源。

2、没有显式创建对象（没有new符号）。

4、封装框架

获取对象样式

//获取对象的样式，非行间样式

function getStyle(obj,attr)

{

if(obj.currentStyle)

{

return obj.currentStyle[attr];

}

else

{

return getComputedStyle(obj,false)[attr];

}

}

currentStyle，不同于全局方法getComputedStyle，它是作为DOM元素属性存在的，兼容IE。

绑定事件

//绑定事件函数

function myAddEvent(obj,sEv,fn)

{

if(obj.attachEvent)

{

//IE

obj.attachEvent("on"+sEv,fn);

}

else

{

//非IE

obj.addEventListener(sEv,fn,false);

}

}

attachEvent方法只支持IE浏览器．与其功能相同的指令是addEventListener,该指令支持FF等浏览器，并且是W3C标准

语法：Element.attachEvent(Etype,EventName)

返回值：[tag:return\_value /]

参数Element:要为该元素动态添加一个事件．

Etype:指定事件类型．比如：onclick,onkeyup,onmousemove等．．

EventName:指定事件名称．也就是你写好的函数．

语法：Element.addEventListener(Etype,EventName,boole)返回值：[tag:return\_value /]  
参数Element:要为该元素绑定一个事件．可以是任意的html元素．  
Etype:事件类型.比如：click,keyup,mousemove.注意使用addEventListener绑定事件时，设置参数事件类型时不必写on．否则会出错．  
EventName:要绑定事件的名称．也就是你写好的函数．  
boole:该参数是一个布尔值：false或true必须填写．false代表支持浏览器事件捕获功能，true代表支持浏览事件冒泡功能．