function addloadEvent(func){

var oldonload=window.onload;

if(typeof window.onload !="function"){

window.onload=func;

}

else{

window.onload=function(){

if(oldonload){

oldonload();

}

func();

}

}

}

//window.onload可以是一个函数。如果是，这个函数将在页面加载完成之后被执行。

这里typeof是检测类型，函数的类型就是"function"这个字符串。那么，如果(typeof window.onload != 'function')也就是说window.onload还没有被赋值，就直接把它赋值为func。

否则，说明有程序先对其赋值了，那么在页面加载完成之后，应该先执行现有函数再执行自己的新的func。

这里现有函数是oldonload()有问题，因为没有赋值。应该是：else {var oldonload = window.onload;window.onload=function(){oldonload();func();}}这里先用oldonload变量把现有的window.onload存下来。

然后【应该先执行现有函数再执行自己的新的func】是两件事。为了按顺序完成两件事，所以这里又多写了一个函数，函数内容是{oldonload();func();}然后这个函数被赋值给了window.onload。

//简单来说：

// addLoadEvent函数主要是完成如下的操作：

//1、把现有的window.onload事件处理函数的值存入到oldonload中。

//2、如果在这个处理函数上还没有绑定任何函数，就将该函数添加给它。

//3、如果在这个处理函数上已经绑定了一些函数，就把该函数追加到现有指定的末尾。

getTime（）

getTime() 方法可返回距某日期（比如： 1970 年 1 月 1 日）之间的毫秒数

Math.floor（） 向下取整

getFullYear()

getFullYear() 方法可返回一个表示年份的 4 位数字。

getMonth（）

getMonth() 方法可返回表示月份的数字。

getDate()

getDate() 方法可返回月份的某一天。

getDay（）

getDay() 方法可返回一周（0~6）的某一天的数字

getHours（）

getHours() 方法可返回时间的小时字段。

getMinutes（）

getMinutes() 方法可返回时间的分钟字段。

getSeconds()

getSeconds() 方法可返回时间的秒。返回值是 0 ~ 59 之间的一个整数。

charAt（）

charAt() 从某个字符串取得具体的字符。