\*\*\*\*\*最常用的部分

.(一个点）：可能通俗的理解为从属关系。比如：你.身体.胳膊.手.手指头。这里的.就表示了一系列的从属关系。

()括号：加在一个方法名后面表示让这个方法运行。

var:JS中定义变量的关键字，如果定义变量时不使var关键字，则此变量为全局变量。

window 是指浏览器对象，是JS编程中的顶级作用域，JS中的一切方法和属性，都是这个对象的后代

document 文档或文档对象，通俗的讲就是：凡是我们能看的见编码，就属于文档。是window的子对象

document.getElementById()通过id来获取一个HTML元素，以便我们用JS来控制操作。比如：document.getElementById('div1').style.color='red';获取id为div1的那个HTML元素使它的文字颜色为红色（.style.color);如果网页中（应该叫文档中）没有id是div1的元素，则得到一个null（空）

document.getElementsByTagName()在整个网页内通过标签名（tagName)获得一组HTML元素的集合对象。比如：document.getElementsByTagName(‘div’);就是在整个网页（文档）范围内获取所有的div元素，如果网页内没有div元素，返回一个空集合对象。

document. getElementsByTagName(‘div’)[3];从得到的一组div中取得第四个div元素。（3是索引值，从0开始，所以索引值为3就是第四个元素了）

document. getElementsByTagName(‘div’).length。获取到的这一组div的个数（length是长度、个数的意思）

body，指的是网页中body标记这个对象。要写成document.body才可使用

\*\*\*\*\*变量类型部分

null：空值

undefined：未定义的（在JS中其实是表示定义了而未赋值的变量）

true：逻辑值中的真

false：逻辑值中的假

Boolean:布尔型（逻辑型）

NaN:不是一个数（not a number）

number:数字

string:字符串

function:方法，是定义方法的关键字

Array:数组，最常用的JS数据对象，可以存放多个值，一般用 new Array()或一对[]来定义。比如：var arr=new Array() (等同于var arr=[])

typeof:用来计算变量类型的运算符，比如alert(typeof “abcd”)出输出string,表示这是个字符串类型

\*\*\*\*\*基本语法部分

object:对象，通俗的讲把一个东东当成一个整体来看待，可称做一个对象

switch：JS的基础语法之一，和case配合使用，表示判断的那个条件（请详见示例的解释）

case：原意为情况、实例的意思，在JS中和switch配合使用完成一个判断。

break:中断当前语句体的执行，执行下一个语句体（见示例来理解和return的区别）

return:中断程序的执行并返回。

= 一个等号是赋值运算，把右边的值赋给左边的变量，是从右向左计算。并且运算优先级最低。

== 两个等号是比较运算，会得到一个真（ture）或假（false）的逻辑值

?: 三元运算符，相当于if else。基本格式为：条件表达式?表达式一:表达式二 问号（？）之前的表示判断的条件，如果这个条件为真，则执行问号后边的语句。如果判断条件不成立，则执行冒号后边的语句

for(初始条件,判断条件,自增){//do something}

while(bool)：循环语法之一。

\*\*\*\*\*控制CSS的方法

currentStyle：计算后的样式

getComputedStyle：计算后的样式

obj.style.filter="alpha(opacity:50)"：IE中把某HTML对象的不透明度设为50%（半透明）

\*\*\*\*\*数组及字符串常用方法及属性

join(): 把一个数组的所有元素转换为字符串，并把这些字符串按参数指定的字符连接起来。如果没有参数，则默认以逗号连接。比如：var a=[1,2,3];alert(a.join());输出字符串1,2,3。alert(a.join(‘-‘));则输出1-2-3。

arr.Push()方法：压栈，把一个元素增加到数组的最后。

arr.Pop()方法：出栈，把数组的最后一个元素删除。

arr.reverse():用来颠倒数组的顺序（把原数组倒过来）。

arr.sort();对数组进行排序，不带参数是按增序重排数组。

arr.splice()：和alice方法一样也是删除数组中的元素，但和alice不同的是：这个里面的参数是被删除的位置，而不是保留的位置。

str.split():切割字符串

str.indexOf(str):取得一个字str这个字符串在这个字符串中的位置

str.lastIndexOf();

\*\*\*\*\*DOM方法

createElement:创建元素的方法，比如：var o=document.createElement(‘div’);创建一个DIV元素。

appendChild:追加子元素的方法，比如document.body.appentChild(o);(承上句)

firstChild:第一个子节点。

lastChild:最后一个子节点

childNodes:所有的子节点的集合。

previousSibling:上一个兄弟节点。（sibling是兄弟的意思）

nextSibling:下一个兄弟节点。

parentNode:父节点

getAttribute(name):获取元素属性的值（attribute是属性的意思）

insertBefore(newNode,targetNode)：把newNode这个节点插入到targetNode的前面。

clone()：复制节点。

removeChild():删除子节点。

\*\*\*\*\*时间和数学常用方法

Math.random():成生一个随机数，介于0到1之间的。

Math.round():四舍五入

Math.abs()：取绝对值

\*\*\*\*\*\*其它常用全局方法

parseInt():截取字符串前边的数字，如果一个字符串前部不是数字，则返回NaN。比如parseInt(‘12px’),则返回12.

setTimerOut(fn,interval):定时器，在指定的时间（由第二个参数interval指定）之后运行某个方法（就是第一个参数fn）一次，然后停下来

clearTimeOut(fn,interval):

setInterval(fn,interval):

\*\*\*\*\*与盒子模型相关的属性与方法（附图）

clientWidth: 是对象可见的宽度，包括width和padding的宽，但不包括滚动条和边线。 offsetWidth：是对象的可见宽度，包滚动条等边线，会随窗口的显示大小改变。（不包括margin）?

scrollWidth ：是对象的实际内容（比如文字占的实际宽）的宽，不包边线宽度，会随对象中内容的多少改变（内容多了可能会改变对象的实际宽度），但内容溢出之后，下内边距不会被计算。

document.documentSrcElement.clientWidth||document.body.clientWidth:获取浏览器可见区域的宽。

offsetTop :当前对象到其上级层顶部边的距离。（常用）

offsetLeft :当前对象到其上级层左边的距。（常用）

scrollWidth:获取对象的滚动宽度 。

scrollHeight: 获取对象的滚动高度。

scrollLeft:设置或获取位于对象左边界和对象中目前可见内容的最左端之间的距离(width+padding为一体),其实就是用来移动左右滚动条。

scrollTop:设置或获取位于对象最顶端和对象中可见内容的最顶端之间的距离；(height+padding为一体)，其实就是用来移动上下滚

clientLeft: 获取对象的border宽度

clientTop：获取对象的border高度

offsetParent :当前对象的上级层对

\*\*\*\*\*和事件相关的属性与方法

event:事件，在IE浏览器中是个全局属性，表示浏览器的事件对象。

target:目标，指的是在标准浏览器中触发事件的那个源头，就是事件传翻的开始的那个元素。

srcElement:意义同上，不过是IE浏览器中用的。

setCapture();IE专用方法，让一个HTML元素一直捕捉鼠标事件，使这个元素不会丢掉鼠标事件。当一个元素上调用这个方法时，所有后续的鼠标事件都被引导到这个元素上。

releaseCapture():和上一个方法配合使用，当一个元素上调用这个方法时，它释放对鼠标事件的捕捉。IE专用。

ele.addEventListener(eventType,fnHandler,boolean)：DOM2级事件处理方法：给ele（ele是事先定义的变量名，下同）元素的eventTypt这个事件绑定

ele.removeEventListenter(evntType,fnHandler,boolean)：DOM2级事件处理方法：给ele元素的eventTypt这个事件移除fnHandler这个处理方法。（IE9以下浏览器不支持）

oEle.attachEvent(onEventType,fnHandler)：同addEventListener 方法，IE专用

oEle.detachEvent(onEventType,fnHandler)：同removeEventListenter方法，IE专用

e.cancelBubble=true：IE中取消事件冒泡的方法。

preventDefault:标准浏览器中阻止HTML元素默认行为的的方法。