# 浏览器BOM对象

之前学习过dom对象，d是document，o,obj对象，m是model模型

那么，顾名思义，什么是BOM

B，代表浏览器，borwser；o代表对象，object，m代表模型，model

那么，bom就是浏览器对象模型

其中，有一个很重要的东西，就是window，window是一个浏览器的实例，在浏览器中，window有双重角色，它既是通过javascript访问浏览器窗口的一个对象，有事ECMAScript定制的Global（世界）；

Window对象的方法：

语法：window.alert(“hello ”);//带有一段小心和确认按钮的警告框

一般使用中，window对象中的window可以省略；

Window.confirm(“啦啦啦”)

显示一个带有指定消息和ok及取消按钮的对话框

如果用户点击确定按钮，则confirm()返回true,点击取消，则返回false

Window.prompt(‘text,defaultText’);

Text：对话框中显示的纯文本；

DefaultText：默认的输入文本

返回值：如果用户单机提示框的取消按钮，返回null；

如果单机确认按钮，返回输入字段当前显示的文本；

Window.open(pageURL,name，parameters);

功能：打开一个新的浏览器窗口或者查找一个已命名的窗口

参数：pageURL:子窗口路径；name：子窗口句柄；parameters：窗口参数

参数：newwindow(打开新窗口)；toolbar设置工具条，no为不要manubar菜单条scrollbars滚动条

Window.close（）：关闭浏览器窗口

//任何我们定义的全局变量，函数，对象，都会变成window对象的属性

## location对象

location对象提供了与当前窗口中加载的文档有关的信息，还提供了一些导航的功能，它既是window对象的属性，也是document对象的属性

location对象常用属性：

location.href：返回当前加载页面的完整URL；

location.hash:返回URL中的hash，如果不包含，则返回空字符串（#后面跟的东西）；

location.host:返回服务器名称和端口号（如果有的话）

location.hostname:返回不带端口号的服务器名称

location.pathname:返回URL中的目录和文件名

location.port:返回URL中制定的端口号，如果没有返回空字符串

location.protocol:返回页面所使用的协议

location.search:返回UR的查询字符串，这一坨以问号开头

location.replace（url）:重定向URL：：他没有浏览记录

location.reload():重新加载当前显示的页面（就是刷新）；

location.reload()括号里面可以跟值，如果不写值，那么有缓存就从缓存重新加载，没有缓存就从服务器加载

如果location.reload(true),则不管你本地有没有缓存，都从服务器加载

# History

历史对象

History.back():回到历史记录的上一步

History.forward():回到历史记录的下一步

History.go(n)：回到历史记录的后N步

History.go(1)== History.forward()

History.go(-n):回到历史记录的前N步

History.go(-1)== History.back();

Screen对象属性

语法：screen.availWidth

功能：返回可用的屏幕宽度

语法：screen.availHeight

功能：返回可用的屏幕高度  
//返回的不是浏览器的宽度和高度，是屏幕的宽度和高度

Navigator:用来识别浏览器名称、版本、引擎、以及操作系统等信息

API就是实现定义好的一个函数，调用他的接口方法（数据接口）；

访问API就是访问别人提供好的数据接口，以此来获取数据

navigator.geolocation.getCurrentPosition(function(position){//获取坐标的方法函数

console.log(position.coords.latitude+'和'+position.coords.longitude);

});

获取经度纬度

## Offset家族

Offset就是 ‘自己的’的意思；

目的，这玩意的作用：JS中一套方便的获取元素尺寸的办法，就是Offset家族

offsetWidth offsetHeight

得到对象的宽度和高度（自己的啊，和其他人，其他元素没任何关系）；

div.style.width和Offsetwidth的区别是什么呢？

offsetWidth的值==width+border+padding

div{

width:220px;

border:1px solid red;

padding:10px;

}

div.style.width只能得到行内的数值，限制很大，不好使，并且，返回的值带有px的字符串

offsetWidth?=242

所以，在我们的js中，需要用到元素自身属性计算的时候，都用offset

offsetLeft offsetTop

返回元素自身距离上级（最近的带有定位）盒子的坐标（上面）的位置；

如果父级都没有定位，则以body为准

这里的上级指的是上一级，不仅仅指爸爸，可以使爷爷，曾爷爷，十八代祖宗、

OffsetLeft从上一级的padding开始算，上级的border不算

总结一下就是：盒子到定位的父盒子的边框到边框的距离；

如果需要计算图片距离父级的top，用个div把图片包起来，然后获取div距离上级的top值，防止出现误差；

offsetParent

返回该对象最近的带有定位的父级（不一定是亲爸爸）；

Offset和style的区别，也是最重要的一点

Offset只是单纯的可读，style可读可写

Offset仅仅只是可以获取值，查看值，而不能修改值

如果要修改，请用style

## event事件对象

之前学过很多事件：onclick、onmouseover、……

你知道这些事件发生的过程中，浏览器产生了哪些数据吗？

div.onclick=function(event){执行语句}

event翻译过来，就是事件的意思

那么，event就是事件对象，它指向的事件，就是onclick

我们在触发DOM上的某一个事件的时候，都会产生event对象，但是支持的方式不同

比如，鼠标操作的时候，会添加鼠标位置的相关信息到事件对象中

还有要注意的一点，event有兼容问题

常规浏览器可以直接识别，比如你function(event)，常规浏览器可以直接读取event,IE不行

IE必须读取window.event

常用的event属性

data :返回拖拽对象的URL字符串

width:该窗口或框架的宽度

height:该窗口或框架的高度

pageX:鼠标光标相对于该网页的水平位置（IE不支持）；

pageY: 鼠标光标相对于该网页的垂直位置（IE不支持）；

screenX:光标相对于该屏幕的水平位置；

screenY: 光标相对于该屏幕的垂直位置；

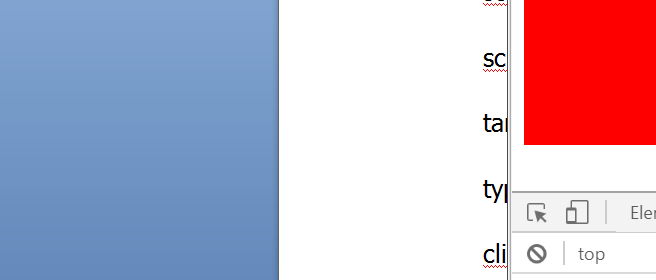
target:该事件被传送到的对象

type:事件类型

clientX:光标相对于该网页的水平位置(当前可见区域)

clientY:光标相对于该网页的垂直位置(当前可见区域)

主要看看pageX、screenX、clientX的区别



ScreenX是点击的位置距离屏幕最边缘的位置

pageX、Y距离网页最顶端，包括了被卷起来的部分

clientX、Y只是我们可以看得到的哪一块区域的位置；

onmousedown鼠标按下事件，只要鼠标不松开，就一直触发

onmousemove鼠标移动事件