**BOM**

BOM 是 Browser Object Model，浏览器对象模型。

DOM 是为了操作文档出现的接口，那 BOM 顾名思义其实就是**为了控制浏览器的行为**而出现的接口。

浏览器可以做什么呢？比如跳转到另一个页面、前进、后退等等，程序还可能需要获取屏幕的大小之类的参数。

所以 BOM 就是为了解决这些事情出现的接口。比如我们要让浏览器跳转到另一个页面，只需要

location.href = "http://www.xxxx.com";

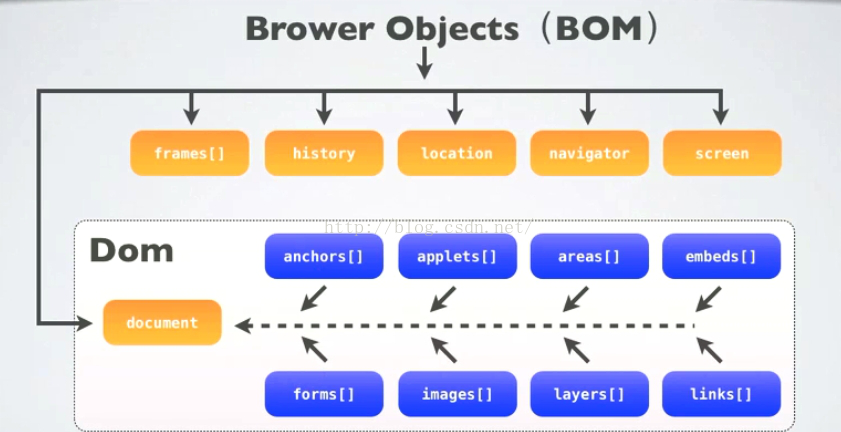
这个 location 就是 BOM 里的一个对象。

**window**

window 也是 BOM 的一个对象，除去编程意义上的“兜底对象”之外，通过这个对象可以获取窗口位置、确定窗口大小、弹出对话框等等。例如我要关闭当前窗口：

window.close();

**DOM 是为了操作文档出现的 API，document 是其的一个对象；  
BOM 是为了操作浏览器出现的 API，window 是其的一个对象。**



**DOM和BOM的关系**

**3.1Window对象**

**Window 对象是 JavaScript层级中的顶层对象。**

**Window 对象代表一个浏览器窗口或一个框架。**

**Window 对象会在 <body>或<frameset>每次出现时被自动创建。**

**1）对象属性**

**window //窗户自身, window=window.self可使用全局属性window访问　Window对象**

**document 对 Document 对象的只读引用。请参阅Document对象。**

**history 对 History 对象的只读引用。请参数History对象。**

**location 用于窗口或框架的 Location 对象。请参阅Location对象。**

**screen 对 Screen 对象的只读引用。请参数Screen对象。**

**navigator 对 Navigator 对象的只读引用。请参数Navigator对象。**

**defaultStatus 设置或返回窗口状态栏中的默认文本。**

**innerheight 返回窗口的文档显示区的高度。**

**innerwidth 返回窗口的文档显示区的宽度。**

**outerheight 返回窗口的外部高度。**

**outerwidth 返回窗口的外部宽度。**

**pageXOffset 设置或返回当前页面相对于窗口显示区左上角的 X 位置。**

**pageYOffset 设置或返回当前页面相对于窗口显示区左上角的 Y 位置。**

**name 设置或返回窗口的名称。**

**parent 返回父窗口。**

**top 返回最顶层的先辈窗口。**

**status 设置窗口状态栏的文本。**

**window.location //URL地址，配备布置这个属性可以打开新的页面**

**2）对象方法**

**window.close(); //关闭窗口**

**window.alert("message"); //弹出一个具有OK按钮的系统消息框，显示指定的文本**

**window.confirm("Are you sure?"); //弹出一个具有OK和Cancel按钮的询问对话框，返回一个布尔值**

**window.prompt("What's your name?", "Default"); //提示用户输入信息，接受两个参数，即要显示给用户的文本和文本框中的默认值，将文本框中的值作为函数值返回**

**window.status //可以使状态栏的文本暂时改变**

**window.defaultStatus //默认的状态栏信息，可在用户离开当前页面前一直改变文本**

**window.setTimeout("alert('xxx')", 1000); //设置在指定的毫秒数后执行指定的代码，接受2个参数，要执行的代码和等待的毫秒数**

**window.clearTimeout("ID"); //取消还未执行的暂停，将暂停ID传递给它**

**window.setInterval(function, 1000); //无限次地每隔指定的时间段重复一次指定的代码，参数同setTimeout()一样**

**window.clearInterval("ID"); //取消时间间隔，将间隔ID传递给它**

**window.history.go(-1); //访问浏览器窗口的历史，负数为后退，正数为前进**

**window.history.back(); //同上**

**window.history.forward(); //同上**

**window.history.length //可以查看历史中的页面数**

**clearInterval() 取消由 setInterval() 设置的timeout。**

**clearTimeout() 取消由 setTimeout() 方法设置的timeout。**

**createPopup() 创建一个 pop-up 窗口。**

**moveBy() 可相对窗口的当前坐标把它移动指定的像素。**

**moveTo() 把窗口的左上角移动到一个指定的坐标。**

**open() 打开一个新的浏览器窗口或查找一个已命名的窗口。**

**print() 打印当前窗口的内容。**

**resizeBy() 按照指定的像素调整窗口的大小。**

**resizeTo() 把窗口的大小调整到指定的宽度和高度。**

**scrollBy() 按照指定的像素值来滚动内容。**

**scrollTo() 把内容滚动到指定的坐标。**

**setInterval() 按照指定的周期（以毫秒计）来调用函数或计算表达式。**

**setTimeout(方法,秒数) 在指定的毫秒数后调用函数或计算表达式。**

**timeOutEvent = setTimeout("longPress('" + obj + "')",1500);定时器传参数**

**3）成员对象**

**window.event**

**window.document //见document对象详解**

**window.history**

**window.screen**

**window.navigator**

**Window.external**

**扩展**

**① 如果文档包含框架（frame 或iframe标签），浏览器会为HTML文档创建一个window对象，并为每个框架创建一个额外的window对象。**

**② window.frames 返回窗口中所有命名的框架**

**③parent是父窗口（如果窗口是顶级窗口，那么parent==self==top）**

**top是最顶级父窗口（有的窗口中套了好几层frameset或者iframe）**

**self是当前窗口（等价window）**

**opener是用open方法打开当前窗口的那个窗口**

**④与消息框有关的方法：alert(String)、confirm(String)、prompt(String)**

**⑤两种定时器：setTimeout(code,latency) 和setInterval(code,period)**

**注：setTimeout只执行一次code，如果要多次调用，可以让code自身再次调用setTimeout()；setInteval()会不停地调用code，直到clearInterval()被调用。**

**3.2 history对象**

**window.history.length //浏览过的页面数**

**history.back() //在浏览历史里后退一步**

**history.forward() //在浏览历史里前进一步**

**history.go(i) //到汗青详细登记单的第i位**

**//i>0进步,i<0撤退退却**

**-------------------------------------------------- -------------------**

**3.3 screen对象**

**screen对象：用于获取某些关于用户屏幕的信息，也可用window.screen引用它**

**window.screen.width //屏幕宽度**

**window.screen.height //屏幕高度**

**window.screen.colorDepth //屏幕颜色深度**

**window.screen.availWidth //可用宽度(除去任务栏的高度)**

**window.screen.availHeight //可用高度(除去任务栏的高度)**

**-------------------------------------------------- -------------------**

**3.4 external对象**

**window.external.AddFavorite("地址","标题" ) //把网站新增到保藏夹**

**-------------------------------------------------- -------------------**

**3.5 navigator对象**

**navigator`对象：包含大量有关Web浏览器的信息，在检测浏览器及操作系统上非常有用**

**window.navigator.appCodeName //浏览器代码名**

**window.navigator.appName //浏览器步伐名**

**window.navigator.appMinorVersion //浏览器补钉版本**

**window.navigator.cpuClass //cpu类型x86**

**window.navigator.platform //操作体系类型win32**

**window.navigator.plugins**

**window.navigator.opsProfile**

**window.navigator.userProfile**

**window.navigator.systemLanguage //客户体系语言zh-cn简体中文**

**window.navigator.userLanguage //用户语言,同上**

**window.navigator.appVersion //浏览器版本(包括 体系版本)**

**window.navigator.userAgent//用户代理头的字符串表示**

**window.navigator.onLine //用户否在线**

**window.navigator.cookieEnabled //浏览器是否撑持cookie**

**window.navigator.mimeTypes**

**3.6 Document对象**

**1）对象属性**

**document.body//提供对 <body>元素的直接访问。对于定义了框架集的文档，该属性引用最外层的<frameset>。**

**document.cookie 返回与当前文档有关的所有 cookie。**

**document.title //返回文档标题等价于HTML的title标签**

**document.domain 返回当前文档的域名。**

**document.bgColor //返回页面背景色**

**document.fgColor //返回前景色(文本颜色)**

**document.linkColor //未点击过的链接颜色**

**document.alinkColor //激活链接(焦点在此链接上)的颜色**

**document.vlinkColor //已点击过的链接颜色**

**document.URL //设置URL属性从而在同一窗口打开另一网页**

**document.fileCreatedDate //文件建立日期，只读属性**

**document.fileModifiedDate //文件修改日期，只读属性**

**document.lastModified 返回文档被最后修改的日期和时间。**

**document.fileSize //文件大小，只读属性**

**document.cookie //设置和读出cookie**

**document.charset //返回字符集 简体中文:gb2312**

**document.URL 返回当前文档的 URL。**

**document.referrer 返回载入当前文档的文档的 URL。**

**document.styleSheets 返回样式表的集合,返回值CSSStyleSheet[]**

**document.styleSheets[0].cssRules.style.paddingTop=”10px”设置样式,样式名去掉连字符,**

**2）常用对象方法**

**document.write() //动态向页面写入内容**

**document.writeln() 等同于 write() 方法，不同的是在每个表达式之后写一个换行符。**

**document.createElement(<Tag>) //用指定标签类型创建一个新的element对象)**

**document.getElementById(ID) //获得指定ID值的对象**

**document.getElementsByName(Name) //获得指定Name值的对象**

**getElementsByTagName() 返回带有指定标签名的对象集合。**

**document.body.appendChild(oTag)**

**———————————————————————**

**3）body-主体子对象**

**document.body //指定文档主体的开始和结束等价于body>/body>**

**document.body.bgColor //设置或获取对象后面的背景颜色**

**document.body.link //未点击过的链接颜色**

**document.body.alink //激活链接(焦点在此链接上)的颜色**

**document.body.vlink //已点击过的链接颜色**

**document.body.text //文本色**

**document.body.innerText //设置body>…/body>之间的文本**

**document.body.innerHTML //设置body>…/body>之间的HTML代码**

**document.body.topMargin //页面上边距**

**document.body.leftMargin //页面左边距**

**document.body.rightMargin //页面右边距**

**document.body.bottomMargin //页面下边距**

**document.body.background //背景图片**

**document.body.appendChild(oTag) //动态生成一个HTML对象**

**4）常用对象事件**

**document.body.onclick=”func()”//鼠标指针单击对象是触发**

**document.body.onmouseover=”func()”//鼠标指针移到对象时触发**

**document.body.onmouseout=”func()”//鼠标指针移出对象时触发**

**———————————————————————**

**5）location-位置子对象**

**location对象：表示载入窗口的URL，也可用window.location引用它**

**location.href //当前载入页面的完整URL，如http://www.somewhere.com/pictures/index.htm**

**location.portocol //URL中使用的协议，即双斜杠之前的部分，如http**

**location.host //服务器的名字，如www.wrox.com**

**location.hostname //通常等于host，有时会省略前面的www**

**location.port //URL声明的请求的端口，默认情况下，大多数URL没有端口信息，如8080**

**location.pathname //URL中主机名后的部分，如/pictures/index.htm**

**location.search //执行GET请求的URL中的问号后的部分，又称查询字符串，如?param=xxxx**

**location.hash //如果URL包含#，返回该符号之后的内容，如#anchor1**

**location.assign("http:www.baidu.com"); //同location.href，新地址都会被加到浏览器的历史栈中**

**location.replace("http:www.baidu.com"); //同assign()，但新地址不会被加到浏览器的历史栈中，不能通过back和forward访问**

**location.reload(true | false); //重新载入当前页面，为false时从浏览器缓存中重载，为true时从服务器端重载，默认为false**

**document.location.reload(URL) //打开新的网页**

**6）selection-选区子对象**

**document.selection**

**7）forms集合(页面中的表单)**

**a)通过集合引用**

**document.forms //对应页面上的form标签**

**document.forms.length //对应页面上/formform标签的个数**

**document.forms[0] //第1个/formform标签**

**document.forms[i] //第i-1个/formform标签**

**document.forms[i].length //第i-1个/formform中的控件数**

**document.forms[i].elements[j] //第i-1个/formform中第j-1个控件**

**b)通过标签name属性直接引用**

**/formform name=”Myform”>input name=”myctrl”/>/form**

**document.Myform.myctrl //document.表单名.控件名**

**c)访问表单的属性**

**document.forms[i].name //对应form name>属性**

**document.forms[i].action //对应/formform action>属性**

**document.forms[i].encoding //对应/formform enctype>属性**

**document.forms[i].target //对应/formform target>属性**

**document.forms[i].appendChild(oTag) //动态插入一个控件**

**document.all.oDiv //引用图层oDiv**

**document.all.oDiv.style.display=” //图层设置为可视**

**document.all.oDiv.style.display=”none”//图层设置为隐藏**

**document.getElementId(”oDiv”) //通过getElementId引用对象**

**document.getElementId(”oDiv”).style=”**

**document.getElementId(”oDiv”).display=”none”**

**/\*document.all表示document中所有对象的集合**

**只有ie支持此属性，因此也用来判断浏览器的种类\*/**