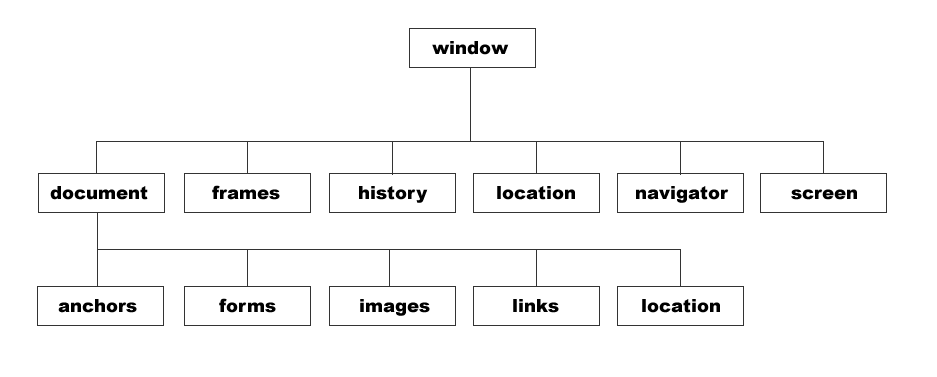
## 1：BOM概念

BOM：Browser Object Model。

window对象是BOM中所有对象的核心。



## 2：window属性

self：self代表自己，相当于window。

parent：返回父窗口。

top：返回顶层窗口，和parent作用一样。

opener：窗口开启者。

status：设置窗口状态栏的文本。（IE）

## 3：window方法

1.window.open(url, name, feature, replace);

url：一个可选的字符串，声明了要在新窗口中显示的文档的 URL。如果省略了这个参数，或者它的值是空字符串，那么新窗口就不会显示任何文档。

name：一个可选的字符串，该字符串是一个由逗号分隔的特征列表，其中包括数字、字母和下划线，该字符声明了新窗口的名称。

这个名称可以用作标记 <a> 和 <form> 的属性 target 的值（\_self,\_blank,\_parent）。

feature：**自行扩展。**replace：**自行扩展。**

**谷歌默认会把系统自动打开的新网页阻止掉，但不阻止通过事件打开新的网页。**

2.close：关闭浏览器**。（火狐不支持！）（支持通过open打开的窗口的关闭）**

**但都支持通过别的网页打开的新的网页关闭。**

3.alert(显示的文本)：弹出窗。

4.confirm(对话框提示的文字)：该方法有返回值，点击确定返回true，点击取消返回false。

5.prompt(提示信息)：输入框。点击确定返回字符串，点击取消返回null。（跟alert的弹出方式一样，就是多了取消按钮，并有返回值）

## 4：history对象

该对象包含浏览器访问过的url。

1.属性

length 返回浏览器历史记录的数量

2.方法

back() 后退，加载前一个url。

forward() 前进。

go(number) 如果参数是正数，那么就是前进相应的数目，如果是负数那么反之，如果是0那么就是刷新。

## 5：location对象

包含有当前url的相关信息，而history对象不能具体反应url的相关信息。

属性：

href：设置或返回完整的url。可以为相对路径，也可以为绝对路径。

search：返回url?后面的查询部分。

hash ：是一个可读可写的字符串，该字符串是 URL 的锚部分（从 # 号开始的部分）。

方法：

assign(url)：加载新的文档。

reload(boolean)：重新加载文档，当参数是true，任何时候都会重新加载，false的时候，只有在文档改变的时候才会加载，否则直接读取内存当中的。

replace(url)：用新的文档代替当前的文档。但不会在 History 对象中生成一个新的记录。当使用该方法时，新的 URL 将覆盖 History 对象中的当前记录。

## 6：navigator对象

appName：浏览器名称。

appVersion：浏览器版本。

platform：操作系统。

**userAgent：**用户代理信息，通过该属性可获取浏览器及操作系统信息。

## 应用：

1：广告弹出窗（自动关闭）。



## 7：window事件

onload：加载事件网页加载完毕后执行。

onscroll：滚动事件。

onresize：窗口缩放事件。

// 获取浏览器视窗宽度/高度

document.documentElement.clientWidth;

document.documentElement.clientHeight;

// 获取浏览器滚动条隐藏的宽度/高度

chrome(body)、火狐、IE(documentElement)

document.documentElement.scrollLeft || document.body.scrollLeft;

document.documentElement.scrollTop || document.body.scrollTop;（上面隐藏高度）（谷歌）

## 8：DOM的概念和作用

DOM：Document Object Model。

document对象是DOM核心对象。

作用：对html中的内容，属性，样式进行操作。

**节点树中节点之间的关系：父子，兄弟。**

## 9：DOM常用属性

title：返回或设置当前文档的标题。

all：返回所有元素的集合。有兼容性问题。不适用于火狐。

forms：返回对文档中所有form对象的引用。

通过集合来访问相应的对象：

1. 通过下标的形式。



2.通过name形式。



## 10：DOM查询方法

1.getElementById(id)：返回拥有指定id的（第一个）对象的引用。

2.getElementsByTagName(tagName)：返回有指定标签名的对象集合(数组)。

**注：上面两个方法没有兼容性问题。**

3.getElementsByName(name)：返回带有指定name指定名称的对象的集合。**有兼容性问题。**

**在IE9及其以下，**如果该对象的标准属性包含有name,（比如说input，在input中name是input的默认属性，所以可以正常使用，但是在div中name并不是div的默认属性，所以不能正常显示），那么可以正确的使用。否则不可以使用。**在火狐中，该方法可以适用于任何情况。**

**结论： getElementsByName(name)主要是适用于表单。**

4.write：输出内容到页面。

5.getElementsByClassName()：返回带有指定className指定名称的对象的集合。有兼容性问题，适用于火狐浏览器，在IE浏览器中不可用（IE8及以下不可用）。

兼容IE6,7,8的获取指定className的方法：



**练习：封装获取className值的DOM节点的兼容性的函数。**



## 11：操作内容

1.innerHTML：用来设置或获取对象起始和结束标签内（例如div框架，它只会获取div里面的内容，而不会获取div）的内容(识别html标签) 。

2.innerText：用来设置或获取对象起始和结束标签内的内容（不识别HTML标签） (只是获取文字内容)(适用于IE，最新版火狐已支持)。

textContent用来设置或获取对象起始和结束标签内的内容 (只是获取文字内容)(适用于火狐，IE8及其以下不支持)。

3.outerHTML 用来设置或获取包括本对象在内的起始和结束标签内（例如div框架，不仅会获取div里面的内容，也会获取div自身的内容）的内容(识别html标签) 。

4.outerText 用来设置或获取包括本对象在内的起始和结束标签内的内容(只是获取文字内容)（火狐不支持）。

## 12：操作属性

1.直接操作

对象.属性。 // 获取属性。某些属性有兼容性问题，例如name（如果不是标签特有的属性，在chrome/firfox访问不到，在IE8及其以下浏览器能获取的到）。

对象.属性=值 // 设置、添加属性(属性值)。

2.通过方法

getAttribute(“属性名称”)。 // 获取给定的属性的值。

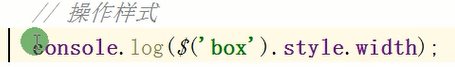
setAttribute(“属性”,“值”)。 // 设置或是添加属性。

removeAttribute(“属性”)。 // 删除属性。

## 13：操作样式

1.行内样式

对象.style.属性 // 获取样式属性

**返回字符串**；

对象.style.属性=值 // 设置、添加样式属性。



**注： 遇到属性是“-”链接的，要将“-”去掉，后面的单词的首字母大写。**

2.行内样式和css层叠样式通用的方式

对象.currentStyle.属性 IE 用来获得实际的样式属性。

getComputedStyle (对象,null).属性 火狐 用来获得实际的样式属性。



注意理解，形参在函数中作为对象属性的写法，不可以用点+形参，而应该写成中括号的形式，因为点的话会变成固定的字符而不是变量。

**注：只能获取不能设置。**

## 14：DOM增删改

一：创建节点（注释和文档节点一般不需要创建）

1：创建元素节点

document.createElement("元素标签名");

2：创建属性节点

var oAttr = document.createAttribute(“属性名”);（不常用）

oAttr.value = "attr-name";

oDiv.setAttributeNode(oAttr);

对象.属性=属性值;（常用）

3：创建文本节点

对象.innerHTML = "";

var oText = document.createTextNode(“文本”); // 文本中的HTML标签实体化

oDiv.appendChild(oText);

二：追加到页面当中

父对象.appendChild(newNode) // 插入到父对象内最后。

父对象.insertBefore(newNode, existsNode) // 插入到什么对象之前。

三：修改(替换)节点

父对象.replaceChild(newNode,existsNode); // 前面的替换后面的

四：删除节点

父对象.removeChild(oldNode);

如果确定要删除节点，最好也清空内存 对象=null;

五：表格操作

table.tBodies[0].rows[0].cells[0].innerHTML;

插入行，插入列：

var oNewRow = table.insertRow(“行位置”);

oNewRow.insertCell(“列位置”);索引值

## 应用：

1：顶部悬浮。

2：回到顶部。

3：动态创建表格。

## 综合应用：

1：表格删除。

2：获取元素索引值：

