**对象：DOM中的“O”**

1. window对象对应着浏览器窗口本身，这个对象的属性和方法通常称为BOM（浏览器对象模型），但是称为Window Object Model（窗口对小数模型更为贴切。

**模型：DOM中的“M”**

1. 文档中的元素的关系用节点树来表示。

**节点**

1. 有三种节点：元素节点、文本节点、属性节点。

**获取元素**

1. 有三种DOM方法可以获取元素节点，分别是通过元素ID、通过元素名、通过类名字。这些方法是document对象特有的函数。
2. 事实上文档中的每一个元素都是一个对象，都有对应的方法和属性。
3. 要查找带有多个类名的元素，需指定多个类名，只要在字符串参数中用空格分隔类名即可。ps：document.getElementByClassName(“sale important”);类名的实际顺序并不重要，就算元素还带有更多的类名也没有关系。

**获取和设置属性**

1. getAttribute是一个函数。它只有一个参数——你打算查询的属性的名字。此方法不属于document对象，所以不能通过document对象调用。它只能通过元素节点对象调用。
2. 当检查某项数据是否是null值时，我们其实是在检查它是否存在。这种检查可以简化为直接把被检查的数据用作if语句的条件。
3. setAttribute实际上完成了两项操作：先创建这个属性，然后设置它的值，如果setAttribute用在一个本身就有这个属性的元素节点上，这个属性的值就会被覆盖掉。
4. 有一个值得注意的细节：通过setAttribute对文档做出修改后，在通过浏览器的查看源代码选项查看源代码时看到的仍是改变前的属性值，也就是说，setAttribute做出的修改不会反映在文档本身的源代码里。这种“表里不一”的现象源自DOM的工作模式：先加载文档的静态内容，再动态刷新，动态刷新不影响文档的静态内容。这正是DOM的真正威力：对页面内容进行刷新却不需要在浏览器里刷新页面。