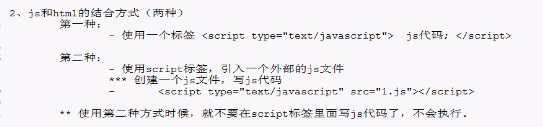
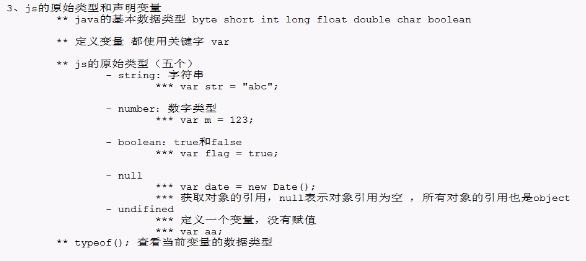
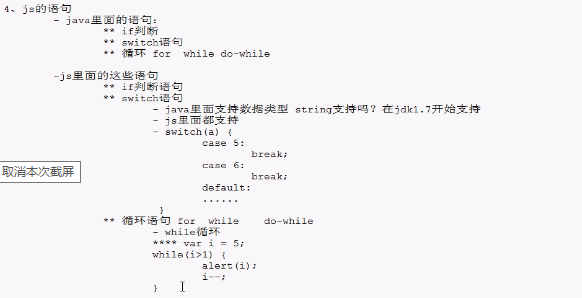
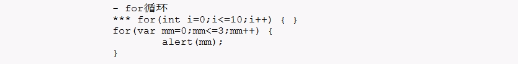


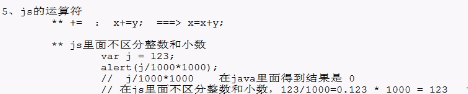
Javascript代码要写在<script></script>标签中，JavaScript的代码被浏览器解析就执行了。浏览器解析HTML代码是从上到下解析的。

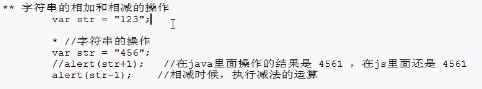


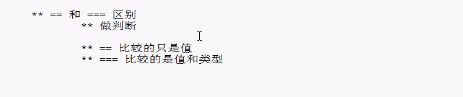


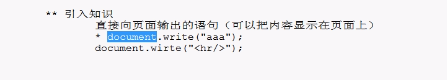








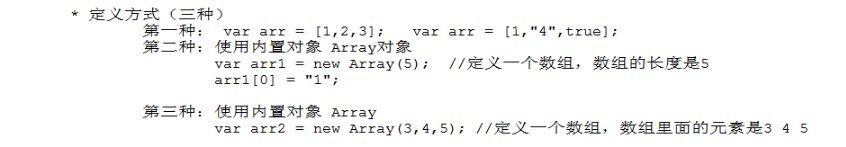




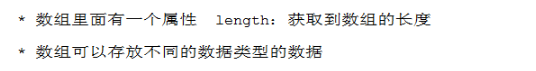


document.write方法输出html代码时，双引号要改为单引号

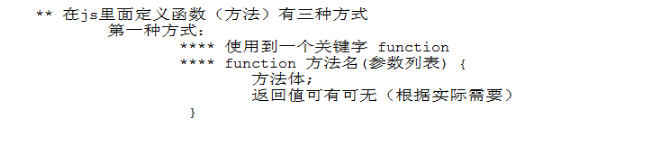


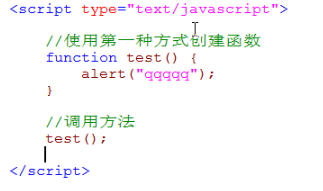




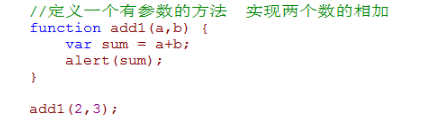


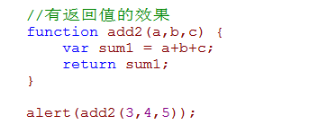
二维数组的定义： 

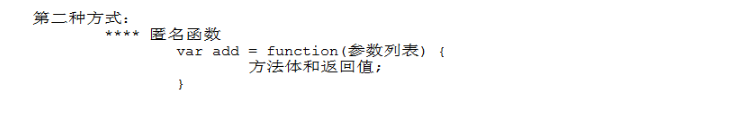


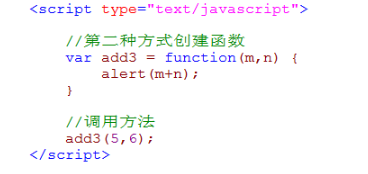


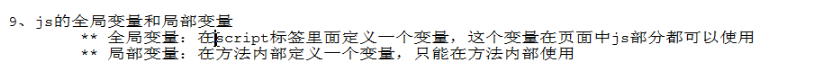
如果要定义有参数的方法，参数的前面不要用var修饰，直接写参数名。

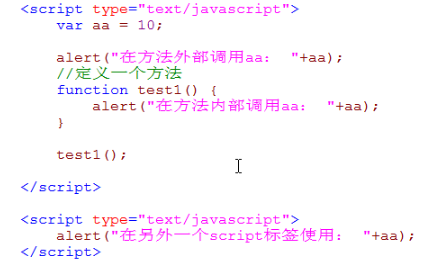




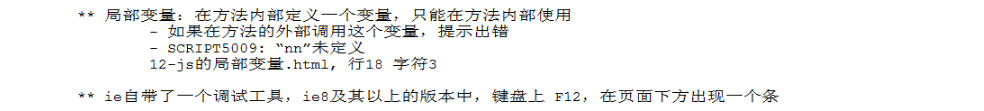


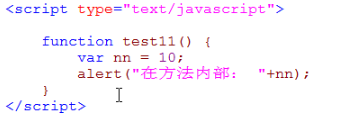




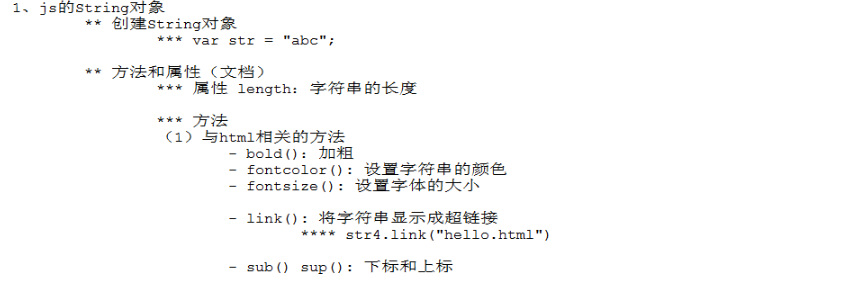


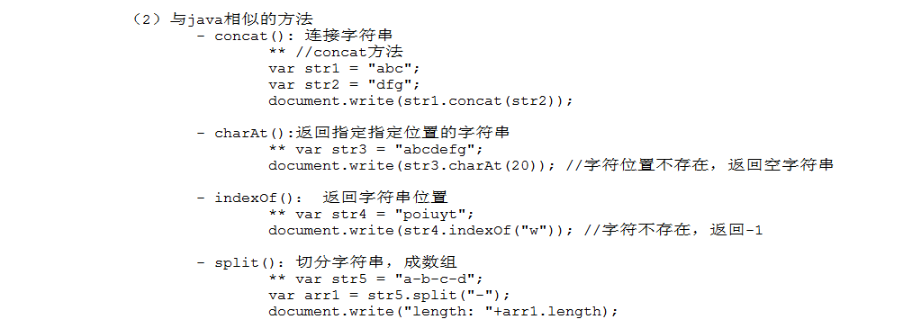


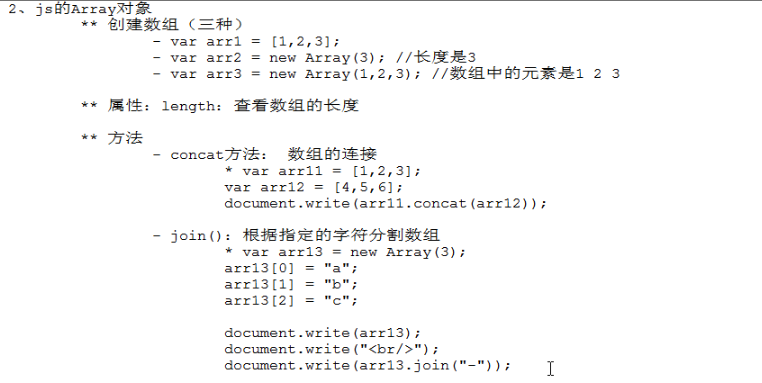


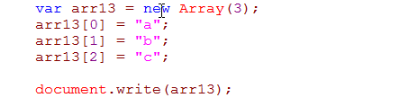






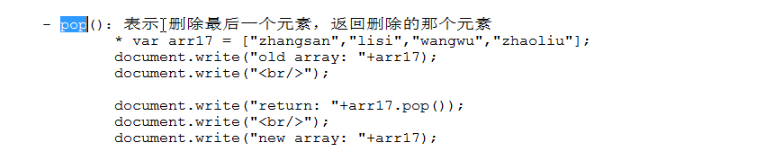




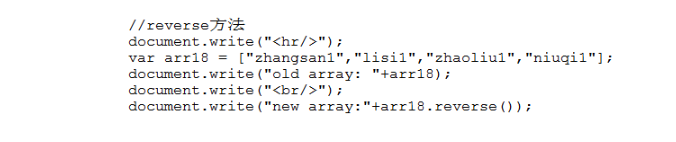


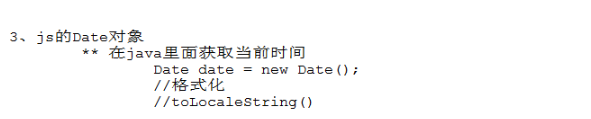
输出的是数组的内容。

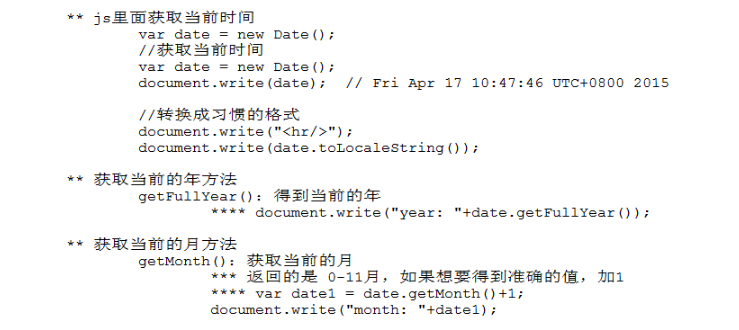


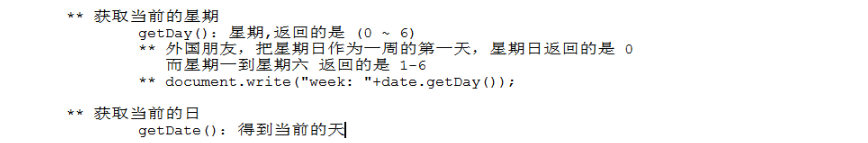


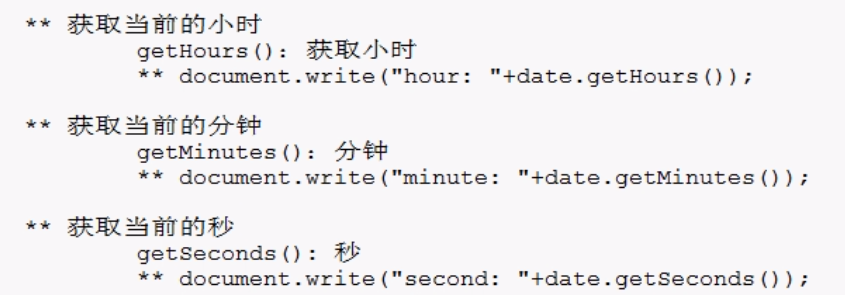


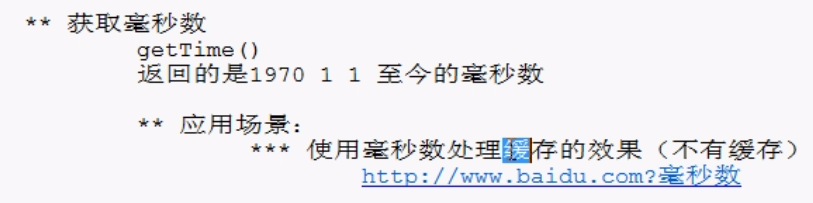


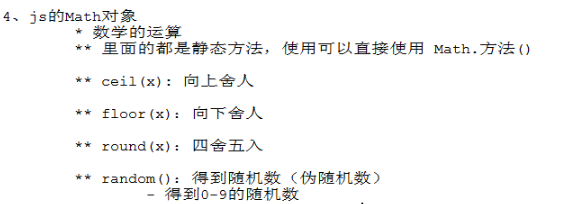


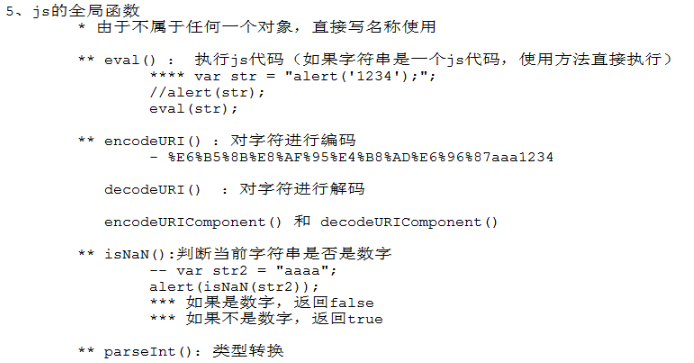


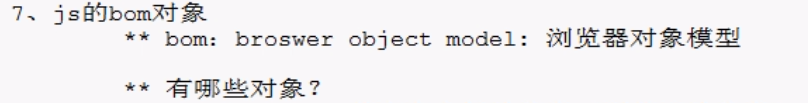


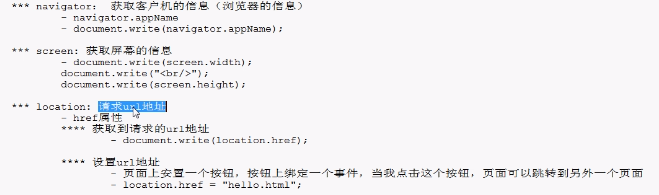




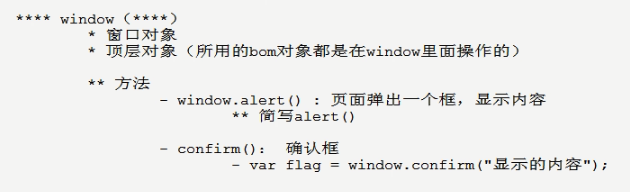




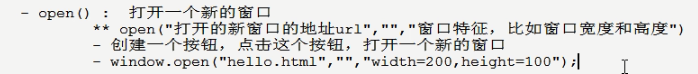


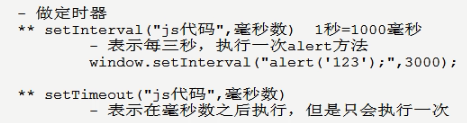


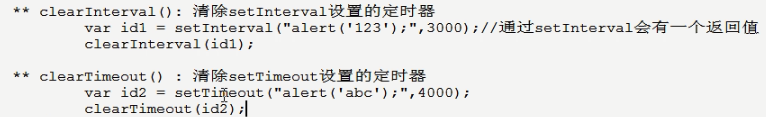
Window对象是javascript中的顶层对象。

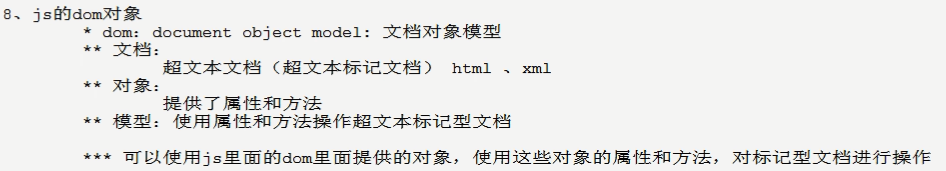


javascript中在两个双引号中间的两个双引号要改为两个单引号，否则会出现bug。







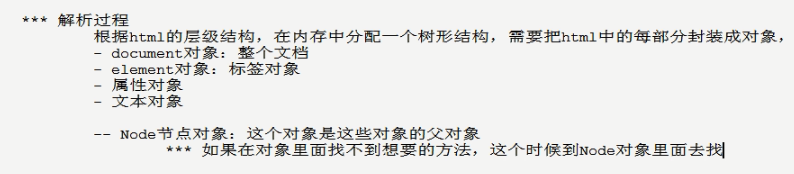


**javascript 是要用js HTML dom中提供的对象才能对html文档中的标签进行操作的。也就是说写到<script></script>中的一定是对象。**

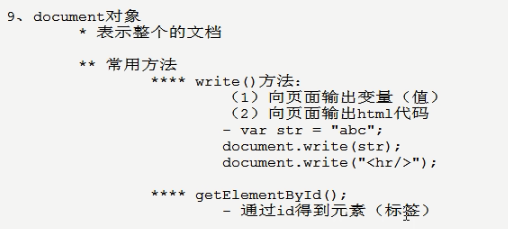
Javascript的html dom为每一种HTML标签都提供了对应的一种标签对象。

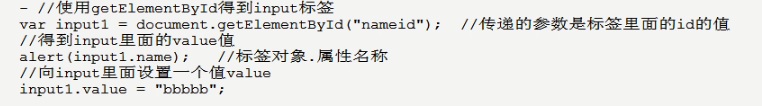
系统会自动将那个要解析的文档的各个组成部分封装成各个对象。

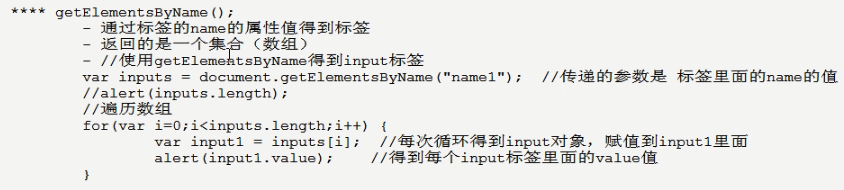
一个document对象（代表整个html文档） ， 各个element对象（html文档中的各个标签对象，一个标签对象就是一个element对象）

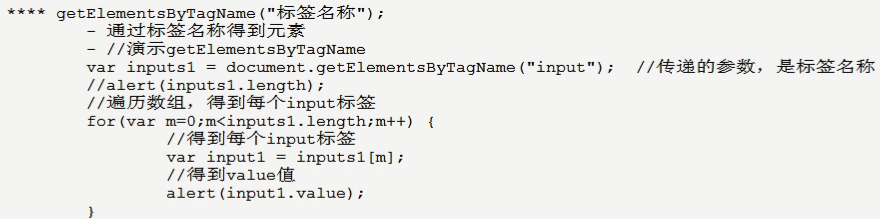


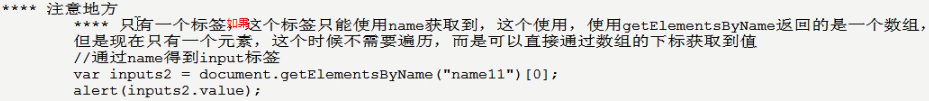
可以通过document对象得到html文档中的各个组成部分。



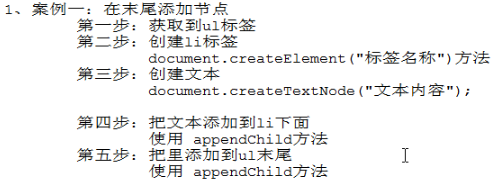


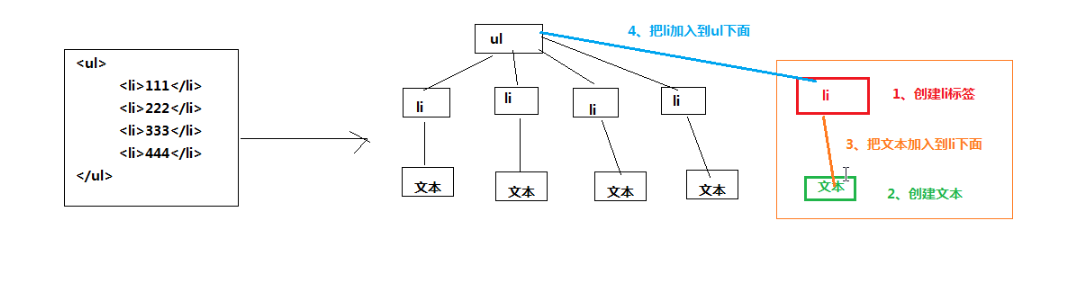


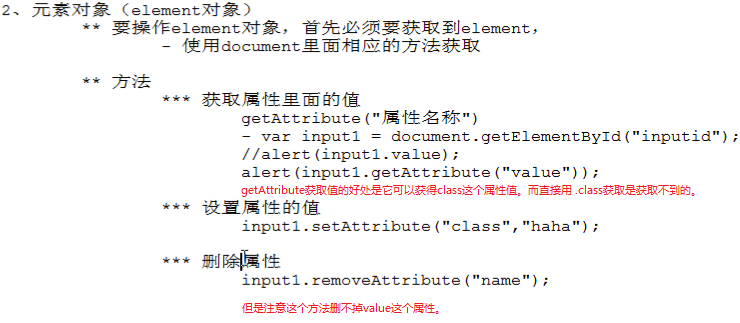




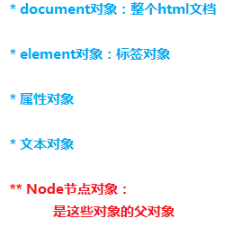
由于javaScript应该要具有很高的安全性（即不应该访问本地文件），但此时实际上是访问了本地的文件。而谷歌浏览器由于安全性做的很好，它不会允许访问本地文件，所以就不会出现效果。问题的解决方式是：将html文件部署到tomcat上。





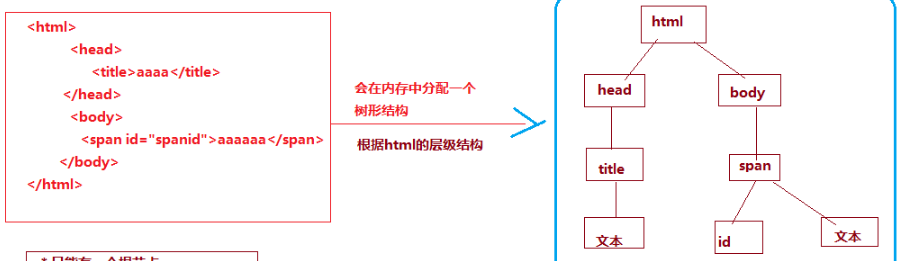


获取一个标签的子标签的唯一有效办法就是 标签名.getElementsByTagName(); （注意不是被document对象调用）而不要用childNodes属性( childNodes属性会产生bug )。

也就是说document对象，标签对象，属性对象，文本对象都是节点对象。注意属性对象和属性值是不同的，一个是对象，一个是值，属性值不是节点对象。

标签节点对象的NodeType都是1，属性节点对象的NodeType是2，文本节点对象的NodeType是3.

每个标签根据层级结构对应的节点关系图：

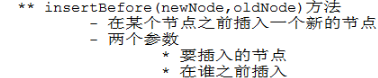


注：id属性对象是文本对象的兄弟节点，id是span的子节点，span是id的父节点。

一个标签对象的firstChild属性值，是这个标签对象的第一个子节点对象的引用。nextSibling返回相邻的后一个兄弟节点对象的引用。



appendChild() 方法是从一个节点对象拿出子节点对象移动到调用这个方法的节点对象中(注意是移动而不是复制，移走了就没了,**编程时如果涉及到数组，数组的长度会由于appendChild而改变，要特别注意**)。







标签对象.cloneNode( boolean ) : 创建一个与自己内容相同的对象（即复制），返回值是新对象的地址。

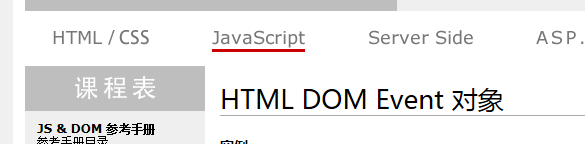
Element的属性innerHTML（注意大小写要完全一致）: 就是设置和返回一个标签对象对应的标签内部的**html代码**。所以也可以用来获取或者设置这个标签对象中的文本内容，并且用这个方式获取时不会有浏览器兼容性问题产生，所以实际中都用这个属性获取文本内容。





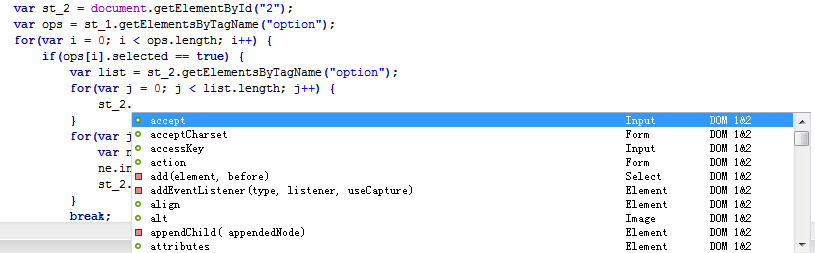
在<select> 标签中添加onchange事件后，当select标签中选择的option发生改变，就意味着onchange事件发生了。

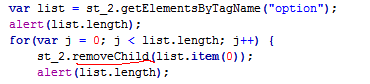
查看一个标签对象支持哪些事件：







注意：由左图可知，JavaScript是一种弱类型语言，JavaScript中的方法返回的变量的类型不是固定的，这里**如果st\_2是可以当成一个Select标签对象的，**此时如果你把st\_2当做是一个Select标签对象就可以调用add()方法**（注意如果st\_2不可以当成一个Select标签对象，而你去调用了add()方法，是会报错的）**。如果将st\_2当做是Input对象就可以调用accept()方法,如果将st\_2当做是Element对象就可以调用等方法。

要注意当前节点调用.返回的集合的长度是会自动更新的，例如中的removeChild(list.item(0))的操作会导致**list的长度发生变化，所以这个程序中是有bug的，我原本以为程序会循环list.length次**。正确的程序应该是。

调用的时候很容易选成, **要特别注意**。

如果一个标签对象有很多子节点对象，并且此时想改变这个标签对象的子节点对象，也可以通过设置这个标签对象的innerHTML属性的值的方式（即改变与这个标签对象对应的标签的内部的html代码），而不是一定要通过调用removeChild(),appendChild()等方法的这种方式去改变子节点对象。