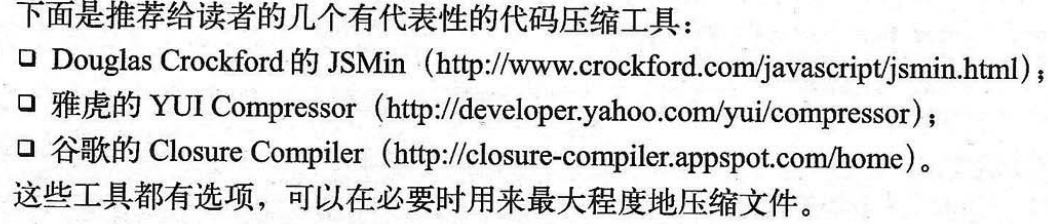
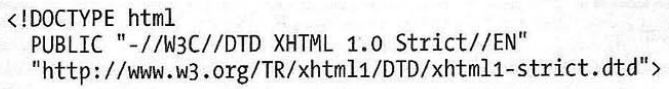
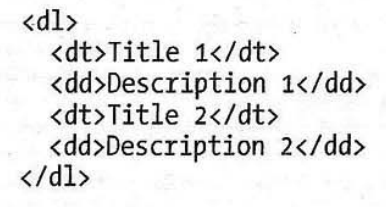
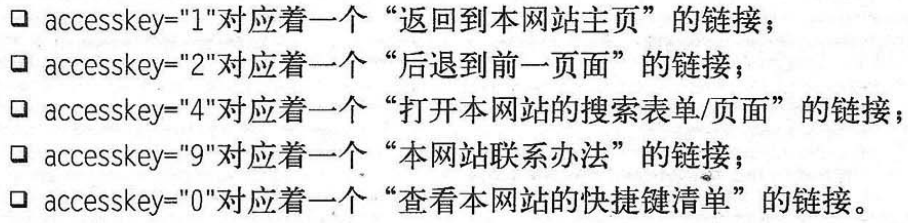
1. DOM是一种API（应用编程接口）。简单地说，API就是一组已经得到有关各方面共同认同的基本约定。在现实世界中，相当于API的例子包括（但不限于）摩尔斯码、国际时区、化学元素周期表。这些都是不同科学领域的标准，它们使得人们能过更方便地交流与合作。
2. 在软件编程领域中，虽然存在着多种不同的语言，但很多任务却是相同或相似的。这也是人们需要API的原因。一旦掌握了某个标准，就可以把它应用在许多不同的环境中。虽然语法会因为使用的程序设计语言而有所变化，但这些约定却总是保持不变的。
3. W3C对DOM的定义是：一个与系统平台和编程语言无关的接口，程序和脚本可以通过这个接口动态地访问和修改文档的内容、结果和样式。
4. 用JavaScript编写的脚本，与其他语言编写出来的脚本一样，都由一系列指令构成，这些指令叫做语句。JavaScript语句与英语中的句子相似，它们是构成任何一个脚本的基本单位。
5. 建议在每条语句的末尾都加上一个分号，这是一个良好的编程习惯，这样做可以让代码更容易阅读。让每条语句独占一行的做法能更容易跟踪JavaScript脚本的执行顺序。
6. JavaScript对大小写敏感。
7. JavaScript变量名允许包含字母、数字、美元符号和下划线（但第一个字符不允许是数字）。为了让比较长的变量名更容易阅读，可以在变量名中的适当位置插入下划线，也可以使用驼峰格式，删除中间的下划线，从第二个单词起每个单词首字母改为大写。通常驼峰格式是函数名、方法名和对象属性名命名的首选格式。
8. 有些其他语言要求在声明变量的同时还必须同时声明变量的数据类型，这种做法称为类型声明（typing）。必须明确类型声明的语言称为强类型（strongly typed）语言。JavaScript不需要进行类型声明，因此它是一种弱类型（weakly typed）语言。
9. JavaScript并不在意某个变量的值是一个字符串还是一个数，可以用新的值对之前赋的值进行覆盖。
10. 如果字符串中含有单引号或双引号，则要用转义字符表示，即在单引号或双引号前加个反斜杠。
11. 如果在某个函数中使用了var，那个变量就将被视为一个局部变量。它只存在于这个函数的上下文中；反之，如果没有使用var，那个变量就将被视为一个全局变量。
12. 函数在行为方面应该像一个自给自足的脚本，在定义一个函数时，我们一定要把它内部的变量全都明确地声明为局部变量。如果总是在函数里使用var关键字来定义变量，就能避免任何形式的二义性隐患。
13. JavaScript语言里的对象可以分为三种类型:
14. 用户定义对象（user-definedobject）：由程序员自行创建的对象。
15. 内建对象（native object）：由浏览器提供的对象。
16. 宿主对象（host object）：由浏览器提供的对象。
17. 为了把某一个或某几个与其他元素区别开来，需要使用class属性或id属性。
18. DOM中有许多类型的节点（node），我们先来看看其中的三种：元素节点、文本节点、属性节点。
19. 标签的名字就是元素的名字，关于元素节点。
20. 例如：<p>元素中包含的文本“记得买以下这些东西”，它是一个文本节点。
21. 在XHTML文档里，文本节点总是被包含在元素节点的内部。但并非所有的元素节点都包含有文本节点。
22. 属性节点用来对元素做出更具体的描述。因为属性总是被放在起始标签里，所以属性节点总是被包含在元素节点中，并非所有的元素都包含着属性，但所有的属性都被元素包含。
23. 继承（inheritance）是CSS技术中的一项强大功能。类似于DOM，CSS也把文档的内容视为一棵节点树，节点树上的各个元素将继承其父元素的样式属性。例如，如果我们为body元素定义了一些颜色或字体，包含在body元素里的所有元素都将自动获得那些样式。
24. 在某些场合，当把样式应用于一份文档时，我们其实只想让那些样式作用于某个特定的元素。为了获得如此精细的控制，需要在文档里插入一些能够把这段文本与其他段落区别开来的特殊标志。为了把某一个或某几个元素与其他元素区别开来，需要使用class属性或id属性。
25. 可以在所有的元素上任意应用class属性，id属性的用途是给网页里的某个元素加上独一无二的标识符（即id属性只能使用一次）。
26. 有3种DOM方法可获得元素节点，分别是通过元素ID、通过标签名和通过类名字来获取。
27. DOM提供了一个名为getElementById的方法，这个方法将返回一个与那个有着给定id属性值的元素节点对应的对象。这个方法是document对象特有的函数。在脚本代码里，函数名后面必须跟一对圆括号，这对圆括号包含着函数的参数。getElementById方法只有一个参数，即就是你想获得的那个元素的id属性的值，这个id值必须放在单引号或双引号里。
28. getElementsByTagName方法返回一个对象数组，每个对象分别对应着文档里有着给定标签的一个元素。这个方法也只有一个参数，它的参数是标签的名字。
29. getElementsByTagName允许把一个通配符作为它的参数，而这意味着文档里的每个元素都将在这个函数所返回的数组里占有一席之地。通配符（星号字符“\*”）必须放在引号里，这是为了让通配符与乘法操作符有所区别。如果想知道某份文档里总共有多少个元素节点，像下面这样使用通配符即可：alert(document.getElementsByTagName("\*").length);
30. getElementsByClassName值接受一个参数，就是类名。这个方法的返回值与getElementsByTagName类似，也是一个具有相同类名的元素的数组。例如，下面这行代码返回的是一个数组，其中包含类名为“sale”的所有元素：document.getElementsByClassName(“sale”);使用这个方法还可以查找那些带有多个类名的元素，要指定多个类名，只要在字符串参数中用空格分隔类名即可。
31. 文档中的每个元素节点都是一个对象。
32. getElementById将返回一个对象，该对象对应着文档里的一个特定的元素节点。getElementsByTagName和getElementsByClassName将返回一个对象数组，它们分别对应着文档里的一组特定的元素节点。
33. 每个节点都是一个对象。
34. getAttribute是一个函数，它只有一个参数-你打算查询的属性的名字。getAttribute只能通过元素节点对象调用，例如，可以与getElementByTagName方法合用，获取每个<p>元素的title属性值。
35. setAttribute方法有两个参数：一个是属性名，另一个是属性值。可以用来改变一个元素的属性值。
36. 最适合于用浏览器来查看的尺寸为400像素\*300像素。
37. DOM是一种适用于多种环境和多种程序设计语言的通用性API。
38. 事件处理函数的作用是在特定事件发生时调用特定的JavaScript代码。
39. 在一棵节点树上，childNodes属性可以用来获取任何一个元素的所有子元素，它是一个包含这个元素全部子元素的数组。
40. nodeType属性总共有12种可取值，但其中仅有3种具有实用价值：
    1. 元素节点的nodeType属性值是1；
    2. 属性节点的nodeType属性值是2；
    3. 文本节点的nodeType属性值是3.
41. 在所有的JavaScript特效当中，最臭名昭著的莫过于那些在人们打开网页时弹出的广告窗口。弹出的广告窗口和内容覆盖是一个典型的滥用JavaScript的例子。
42. 从技术上讲，弹出窗口本身是一项很实用的功能，它解决了网页设计工作中的一个难题：如何向用户发送信息。
43. 如果要使用JavaScript，就要确认：这么做会对用户的浏览体验产生怎样的影响？还有个更重要的问题：如果用户的浏览器不支持JavaScript时该怎么办？
44. 如果正确地使用了JavaScript脚本，就可以让访问者在他们的浏览器不支持JavaScript的情况下仍能顺利地浏览你的网站，这就是所谓的平稳退化（graceful degradation）。就是说，虽然某些功能无法使用，但最基本的操作仍能顺利完成。渐进增强的实现必然支持平稳退化。如果你按照渐进增强的原则去充实内容，你为内容添加的样式和行为就自然支持平稳退化，那些缺乏必要的CSS和DOM支持的访问者仍可以访问到你的核心内容。
45. JavaScript使用window对象的open（）方法来创建新的浏览器窗口。这个方法有三个参数，并且这三个参数都是可选的：window.open(url,name,features)
    1. 第一个参数是想在新窗口里打开的网页的URL地址。如果省略这个参数，屏幕上将弹出一个空白的浏览器窗口。
    2. 第二个参数是新窗口的名字。可以在代码里通过这个名字与新窗口进行通信。
    3. 最后一个参数是一个以逗号分隔的字符串，其内容是新窗口的各种属性。这些属性包括新窗口的尺寸（宽度和高度）以及新窗口被启用或禁用的各种浏览功能（工具条、菜单条、初始显示位置等）。对于这个参数应该掌握以下原则：新窗口的浏览功能要少而精。
46. “真”协议用来在因特网上的计算机之间传送数据包，如HTTP协议、FTP协议等。伪协议是一种非标准化的协议。“JavaScript：”伪协议让我们通过一个链接调用JavaScript函数。在HTML文档里里通过“JavaScript：”伪协议调用JavaScript代码的做法非常不好。
47. 如果一个指令中使用了return false语句，这个链接不会真的被打开。
48. CSS技术的最大优点是，它能够帮助你将Web文档的内容结构（标记）和版面设计（样式）分离开来。真正能从CSS技术获益的方法，是把样式全部转移到外部文件中去。作为CSS技术的突出优点，文档结构与文档样式的分离可以确保网页都能平稳退化，具备CSS支持的浏览器固然可以把网页呈现得美轮美奂，不支持或禁用了CSS功能的浏览器同样可以把网页的内容按照正确的结构显示出来。
49. 标记良好的内容就是一切。
50. 所谓“渐进增强”（progressive enhancement）就是用一些额外的信息层去包裹原始数据。按照“渐进增强”原则创建出来的网页几乎都符合“平稳退化”原则。渐进增强原则基于这样一种思想：你应该总是从最核心的部分，也就是内容开始，应该根据内容使用标记实现良好的结构；然后再逐步加强这些内容，这些增强工作既可以是通过CSS改进呈现效果，也可以是通过DOM添加各种行为。
51. 最佳的使用CSS的方法是：先把样式信息存入一个外部文件，再在文档的head部分用<link>标签来调用这个文件，即<link rel=“stylesheet” href=“layout.css” media=”screen”>
52. 性能考虑：
    1. 尽量少访问DOM和尽量减少文档中的标记数量。无论何时，只要是查询DOM中的某些元素，浏览器都会搜索整个DOM树，从中查找可能匹配的元素。过多不必要的元素只会增加DOM树的规模，进而增加遍历DOM树以查找特定元素的时间。
    2. 合并和放置脚本。将多个JS文件合并到一个脚本文件中，可以减少加载页面时发送的请求数量。将所有的<script>标签都放在文档末尾，</body>标记之前，可以让页面变得更快。
    3. 压缩脚本。所谓压缩脚本，指的是把脚本文件中不必要的字节，如空格和注释，统统删除，从而达到“压缩”文件的目的。有许多工具可以做这件事。压缩（精简）之后的代码虽然不容易看懂，却能大幅减少文件大小。多数情况下，你应该有两个版本，应该是工作副本，可以修改代码并添加注释；另一个是精简副本，用于放在站点上。通常，为了与非精简版本区分开，最好在精简版本的文件名中加上min字样。
53. 与DOM脚本编程工作有关的问题不外乎平稳退化、向后兼容和分离JavaScript几大类，这些问题的解决方式和解决程度影响着网页的可用性和可访问性。
54. 网页的结构由标记负责创建，JavaScript函数只用来改变某些细节而不改变其底层结构。这是绝大多数JavaScript函数的工作原理。
55. document.write的最大缺点是它违背了“行为应该与表现分离”的原则。即使把document.write语句挪到外部函数里，也还是需要在标记的<body>部分使用<script>标签才能调用那个函数。
56. 把结构、行为和样式分开永远都是一个好主意。只要有可能，就应该用外部CSS文件代替<font>标签去设定和管理网页的样式信息，最好用外部JavaScript文件去控制网页的行为。应该避免在<body>部分乱用<script>标签，避免使用document.write方法。
57. 在需要把一大段HTML内容插入一份文档时，innerHTML属性可以让你又快又简单地完成这项任务。不过，innerHTML属性不会返回任何对刚插入的内容的引用。如果想对刚插入的内容进行处理，则需要使用DOM提供的精确的方法和属性。
58. createElement方法用来创建一个新元素。不论何时何地，只要使用了createElement方法，就应该把新创建出来的元素赋予一个变量。
59. 把新创建的节点插入某个文档的节点树的最简单的方法是，让它成为这个文档某个现有节点的一个子节点。
60. 用CreateTextNode方法来创建一个文本节点。
61. Ajax用于概括异步加载页面内容的技术。使用Ajax可以做到只更新页面中的一小部分，其他内容---标识、导航、头部、脚部，都不用重新加载。它的主要优势是对页面的请求以异步方式发送到服务器，而服务器不会用整个页面来响应请求，它会在后台处理，与此同时用户还能继续浏览页面并与页面交互。你的脚本则可以按需加载和创建页面内容，而不会打断用户的浏览体验。
62. Ajax技术的核心是XMLHttpRequest对象。这个对象充当着浏览器中的脚本（客户端）与服务器之间的中间人的角色。以往的请求都由浏览器发出，而JavaScript通过这个对象可以自己发送请求，同时也自己处理响应。
63. Ajax应用的一个特色是减少重复加载页面的次数。
64. 构建Ajax网站的最好方法，是先构建一个常规的网站，然后Hijax。Hijax指的是“渐进增强地使用Ajax”。
65. Ajax应用主要依赖于服务器端处理，而非客户端处理。
66. 若要使用XHTML DOCTYPE，应该将下列内容写在文档开头：



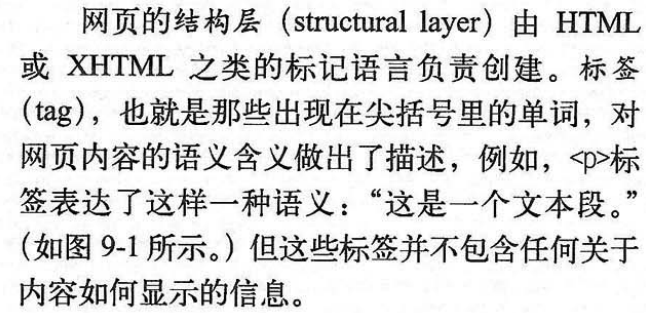
1. 如果使用HTML5，它的文档类型声明是<!DOCTYPE html>，并且这个文档声明同样也支持HTML和XHTML标记。
2. 节点集合就是一个由节点构成的数组。
3. 就像我们可以用自己的CSS样式去取代浏览器所使用的默认样式，我们同样可以用DOM去改变浏览器的默认行为。
4. 当你需要把一系列数据保存起来时，数组是理想的存储媒介。JavaScript中定义一个数组是这样的，如需定义一个名为defs的新数组：var defs = new Array()；
5. 定义列表是表现缩略语及其解释的理想结构。定义列表（<dl>）由一系列“定义标题”（<dt>）和相应的“定义描述”（<dd>）构成：



1. 在编写DOM脚本时，你会想当然地认为某个节点肯定是一个元素节点，这是一种相当常见的错误。如果没有百分之百的把握，就一定要去检查nodeType属性值。有很多DOM方法只能用于元素节点，如果用在了文本节点上，就会出错。
2. 一般来说，在适用于Windows系统的浏览器里，快捷键的用法是在键盘上同时按下Alt键和特定按键；在使用于Mac系统的浏览器里，快捷键的用法是同时按下Ctrl键和特定按键。一些基本的快捷键都有约定俗成的设置办法：



1. 在需要对文档里的现有信息进行检索时，以下DOM方法最有用：
   1. getElementById
   2. getElementsByTagName
   3. getAttribute
2. 在需要把信息添加到文档里去时，以下DOM方法最有用：
   1. createElement
   2. createTextNode
   3. appendChild
   4. insertBefore
   5. setAttribute
3. JavaScript脚本只应该用来充实文档的内容，要避免使用DOM技术来创建核心内容。
4. DOM技术不仅可以用来改变网页结构，还可以用来更新HTML页面元素的CSS样式。
5. 我们在浏览器里看到的网页其实是由以下三层信息构成的一个共同体（三位一体）：
   1. 结构层



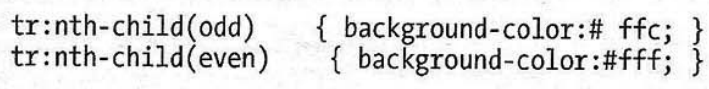
* 1. 表示层



* 1. 行为层

这是JavaScript语言和DOM主宰的领域。

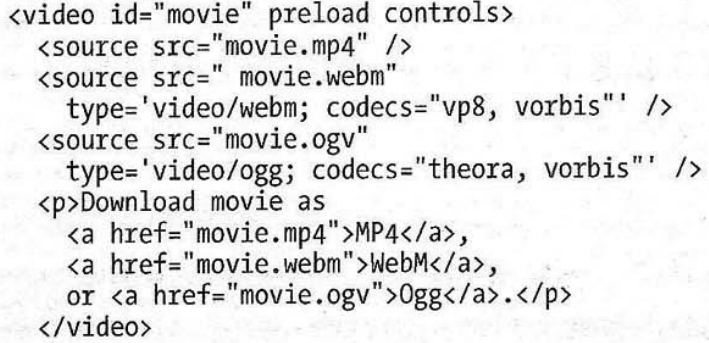
1. 网页的表示层和行为层总是存在的，即使未明确给出任何具体的指令也是如此。此时，Web浏览器将应用它的默认样式和默认事件处理函数。
2. 在所有的产品设计活动中，选择最适用的工具去解决问题是最基本的原则。具体到网页设计工作，这意味着：
   1. 使用（X）HTML去搭建文档的结构；
   2. 使用CSS去设置文档的呈现效果；
   3. 使用DOM脚本去实现文档的行为。
3. CSS利用伪类走进DOM的领地，同样，利用DOM样式给元素设定样式。
4. 文档里的每个元素都是一个对象，每个元素都有一个style属性，style 属性也是一个对象。
5. 减号和加号之类的操作符是保留字符，不允许用在函数或变量的名字里。这同时也意味着他们也不能用在方法和属性的名字里（方法和属性其实是关联在某个对象上的函数和变量）。
6. 当需要引入一个中间带减号的CSS属性时，DOM要求你用驼峰命名法。不管CSS样式属性的名字里有多少个连字符，DOM一律采用驼峰命名法来表示它们。DOM属性fontFamily的值与CSS属性font-family的值是一样的。
7. 通过DOM style属性获取样式有很大的局限性，style属性只能返回内嵌样式，即就是只有把CSS style属性插入到标记里，才可以用DOM style属性去查询那些信息。这不是使用样式的好办法---表现信心与结构混杂在一起了。更好的办法是用一个外部样式表去设置样式，来自外部文件的样式不能用DOM style属性检索出来。如果把样式添加到HTML文档的<head>部分的的style标签里，同样不能用DOM style属性检索出来。
8. 用DOM处理CSS样式的意义在于：用DOM设置的样式，就可以用DOM再把它们检索出来。
9. 在绝大数场合，还是应该使用CSS 去声明样式。就像不应该利用DOM去创建重要的内容那样，你也不应该利用DOM为文档设置重要的样式。不过，在使用CSS不方便的场合，可以利用DOM对文档的样式做一些小的增强。
10. 在用CSS安排你的内容时，千万不要人云亦云地认为表格都是不好的。虽然利用表格来做页面布局不是好主意，但利用表格来显示表格数据却是理所应当的。
11. 让表格里的行更可读的常用技巧是交替改变它们的背景色，从而形成斑马线效果，使相邻的两行泾渭分明。通过分别设置奇数行和偶数行样式的办法可实现这种效果。如果浏览器支持CSS3，那就很简单，只需要如下两行样式：



1. 如果nth-child()不可用，要获取同样的效果可在JavaScript中用while或for循环可以实现，可以编写一个函数来为表格添加斑马线效果，只要隔行设置样式就行了。
2. 只要有可能，最好选用CSS为文档设置样式。但是当CSS不能处理或者难以部署时，DOM可以帮上大忙。
3. CSS提供的：hover等伪class属性允许我们根据HTML元的状态来改变样式。
4. 绝大多数的现代浏览器，虽然对CSS伪类的支持很不完整，但对DOM却都有着良好的支持。在浏览器们对CSS的支持进一步完善之前，在事件发生时用DOM改变HTML元素的样式更切合实际。
5. 当一个场合既可以用CSS又可以用DOM来设置样式时，需要考虑以下因素：
   1. 这个问题最简单的解决方案是什么；
   2. 哪种解决方案会得到更多浏览器的支持。

要做出明智的选择，就必须对CSS和DOM技术都有足够深入的了解。如果你只喜欢使用CSS，你十有八九会选择一个CSS解决方案，而不考虑JavaScript解决方案的效果会不会更好。反之，如果你只懂得写DOM脚本，你往往会立刻动手编写JavaScript函数，而不考虑用CSS来解决

1. 如果想要改变某个元素的呈现效果，使用CSS；如果想改变某个元素的行为，使用DOM；如果想根据某个元素的行为去改变它的呈现效果，则根据上一条两个因素考虑具体用哪种方法。
2. className属性是一个可读可写的属性，凡事元素节点都有这个属性，可以用className属性得到一个元素的class属性。
3. 把一个非常具体的东西改进为一个较为通用的东西的过程叫做抽象。
4. 如果我们想随着时间的变化而不断改变某个元素的样式，则只能使用JavaScript。JavaScript能够按照预定的时间间隔重复调用一个函数，而这意味着我们可以随着时间的推移而不断改变某个元素的样式。
5. JavaScript函数setTimeout能够让某个函数在经过一段预定的时间之后才开始执行。这个函数带有两个参数：第一个参数通常是一个字符串，其内容是将要执行的那个函数的名字；第二个参数是一个数值，它以毫秒为单位设定了需要经过多长时间后才开始执行第一个参数所给出的函数。
6. 未使用关键字var声明的变量为全局变量，使用var声明的变量为局部变量。
7. 真正的动画是一个渐变的过程，元素应该从出发点逐步地移动到目的地，而不是从出发点一下子跳跃到目的地。
8. JavaScript函数parseInt可以字符串里的数值信息提取出来。如果把一个以数字开头的字符串传递给这个函数，它将返回一个数值：parseInt（string）。例如：parseInt（“39 steps”），这个函数调用将返回数值39。ParseInt函数的返回值通常是整数，如果需要提取的是带小数点的数值（浮点数），应该使用相应的parseFloat函数。
9. 除非浏览器运行用户“冻结”移动着的内容，否则就应该避免让内容在页面中移动。如果页面上有移动着的内容，就应该用脚本或插件的机制允许用户冻结这种移动或动态更新行为。
10. HTML5简介：HTML5是HTML语言当前及未来的新标准。HTML规范从HTML4到XHTML，再到Web APPs 1.0，最后又回到了HTML5，整个成长过程充满了艰辛和争议。
11. 谈到Web设计，最准确的理解是把网页看成三个层：
    1. 结构层
    2. 样式层
    3. 行为层
12. 这三个层分别对应不同的技术，分别是：
    1. 超文本标记语言（HTML）
    2. 层叠样式表（CSS）
    3. JavaScript和文档对象模型（DOM）
13. JavaScript允许我们为元素创建属性：element.property = value;这很像创建一个变量，但区别是这个变量专属于某个特定的元素。
14. 像movie.mp4这样的视频，其实是一个包含很多东西的容器。扩展名mp4表示视频是使用基于苹果QuickTime技术的MPEG4打包而成的。这个容器规定了不同的音频和视频轨道在文件中的位置，以及其他与回放相关的特性。其他容器还有m4v（另一个MPEG4扩展名）、avi（Audio Video Interleave，音频视频交错）、flv（Flash Video）等。
15. 在每个影片容器中，音频和视频轨道都使用不同的编解码器来编码。编解码器决定了浏览器在播放时应该如何解码音频和视频。编解码器的核心是一个算法，用于压缩和存储视频，以减少原始文件的大小，同时可能会也可能不会损失品质。视频解码器有很多种，其中有代表性的有三个：H.264、Theora和VP8。同样，音频文件也有相应的编解码器，常见的有mp3（MPEG-1 Audio Layer3）、aac（Advanced Audio Coding）和ogg（Ogg Vorbis）。
16. 为了保证每个人都能看到视频，必须制作多种格式的视频并在<video>元素中包含多个来源：



1. Modernizr库可以进行兼容性检查。
2. HTML5这个规范至今仍然没有尘埃落定，很多地方都有可能发生变化。在浏览器支持不是特别完善的情况下，全面转入HTML5还为时过早。
3. 为了准备站点的制作，创建三个文件夹，一个叫images，保存要用的图片；一个叫styles，保存CSS文件；一个叫scripts，保存JavaScript文件。
4. 不管为哪个元素应用什么颜色，都要同时给它一个背景色。否则，就有可能导致意外，看不到某些文本。
5. 所谓库，就是可重用的代码包，具有如下一些优点：
   1. 库代码经过了大量用户的测试和验证；
   2. 库能够很容易地与已有的开发框架集成；
   3. 库为大多数日常琐碎的DOM编程工作提供了方便、简洁的方案，每个函数都能节省很多行代码；
   4. 库很好地解决了跨浏览器的问题，让你更省心。