本章内容：

1. 平稳退化：确保网页在没有JavaScript 的情况下也能正常工作；
2. 分离JavaScript：把网页的结构和内容与 JavaScript 脚本的动作行为分开；
3. 向后兼容性：确保老版本的浏览器不会因为你的JavaScript 脚本而死掉；
4. 性能考虑：确定脚本执行的性能最优。

平稳退化：

在访问者的浏览器不支持JavaScript 的情况下仍能顺利地浏览网站，就是说，虽然某些功能无法使用，但是最基本的操作仍能顺利完成。

JavaScript 使用 window 对象的 open() 方法来创建新的浏览窗口：

*window.open(url, name, features)*

三个参数都是可选的。

1. url：想在新窗口里打开的网页的URL地址。如果省略这个参数，屏幕上将打开一个空白的浏览器窗口；
2. name：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. features：这是一个以逗号分割的字符串，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者禁用的各种浏览功能（工具条、菜单条、初始显示位置， 等等）。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。

示例：

***function popUp(winURL){***

***window.open(winURL, “popup”, “width=320,height=480”);***

***}***

“JavaScript:” 伪协议：

语法：

***<a href=”javascript:popUp(‘http://www.example.com/’);”>Example</a>***

内嵌的事件处理函数：

***<a href=”#” onclick=”popUp(‘http://www.example.com/’);return false;”>Example</a>***

“#” 符号仅供文档内部使用的标记符号。在某些浏览器里，# 链接指向当前文档的开头。把 href 属性的值设置为 # 只是为了创建一个空链接。实际工作全部由 onclick 属性负责完成。

这个技巧与 ”javascript:” 伪协议都不能平稳退化。如果用户已经金庸了浏览器的JavaScript功能，这样的链接将毫无用处。

具体到popUp() 函数，为其中的JavaScript 代码预留出退路很简单：在链接里把 href 属性设置为真实存在的URL 地址，让它成为一个有效的链接：

***<a href=”http://www.example.com/” onclick=”popUp(‘http://www.example.com/’);return false;”>Example</a>***

***<a href=”http://www.example.com/” onclick=”popUp(this.getAttribute(‘href’));return false;”>Example</a>***

***<a href=”http://www.example.com/” onclick=”popUp(this.href);return false;”>Example</a>***

href 属性现在有合法的值，即使JavaScript 被禁用，这个链接也是可用的。虽然这个链接在功能上打了折扣，但是没有彻底失效。

向CSS 学习

结构与样式的分离：