TURING 图灵程序设计丛书 Web开发系列

Apress®

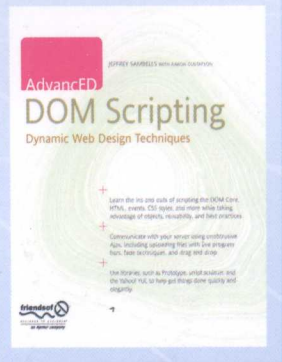
AdvancED DOM Scripting
Dynamic Web Design Techniques

JavaScript DOM 高级程序设计

[加] Jeffrey Sambells
[美] Aaron Gustafson 著

李松峰 李雅雯 等译

- 目前最深入的JavaScript力作之一
- 深入剖析Prototype、jQuery、YUI等JavaScript库的技术内幕
- 全景阐述JavaScript DOM程序开发的最佳实践



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 Web开发系列

AdvancED DOM Scripting
Dynamic Web Design Techniques
JavaScript DOM 高级程序设计

[加] Jeffrey Sambells
[美] Aaron Gustafson 著
李松峰 李雅雯 等译

人民邮电出版社
北 京

图书在版编目（CIP）数据

JavaScript DOM 高级程序设计 /（加）桑贝斯（Sambells，J.），（美）古斯塔夫森（Gustafson，A.）著；李松峰等译．—北京：人民邮电出版社，2008.7
（图灵程序设计丛书）
书名原文：AdvancED DOM Scripting：Dynamic Web Design Techniques
ISBN 978-7-115-18109-1

Ⅰ. J… Ⅱ. ①桑…②古…③李… Ⅲ. JAVA 语言－程序设计 Ⅳ. TP312

中国版本图书馆CIP数据核字（2008）第067862号

内 容 提 要

本书注重理论与实践的结合，全面讲述高级的 DOM 脚本编程。全书分为 3 个部分：第一部分“深入理解 DOM 脚本编程”，涉及 W3C DOM 规范的各方面，包括非标准的浏览器支持和不支持的内容；第二部分“浏览器外部通信”，以 Ajax 和客户端-服务器端通信为主题；第三部分“部分高级脚本编程资源”，集中介绍了一批第三方脚本编程资源，包括库和 API。同时，每部分的最后一章都为案例研究，将学到的内容应用于实践。通过学习全书内容，读者将能构建起属于自己的 DOM 实用方法库。

本书适合有 Web 开发和设计经验的读者阅读和参考。

图灵程序设计丛书

JavaScript DOM高级程序设计

◆ 著 ［加］Jeffrey Sambells ［美］Aaron Gustafson
译 李松峰 李雅雯等
责任编辑 杨 爽

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：800×1000 1/16
印张：29.5
字数：710千字 2008年7月第1版
印数：1－4 000册 2008年7月北京第1次印刷

著作权合同登记号 图字：01-2008-2031号
ISBN 978-7-115-18109-1/TP

定价：59.00元

读者服务热线：(010)88593802 印装质量热线：(010)67129223
反盗版热线：(010)67171154

译 者 序

读者的眼睛是雪亮的。为了让还没有成为这本书读者的朋友听听已经看过这本书的读者的意见，我一直在关注网上有关这本书的评论。

到中文版付梓时为止，Amazon上已经有4篇评论，总体上毁誉参半，两人给了5星，一个给出3星，另一个则给出1星。而业内几个知名的专业blog的评论中则普遍不乏溢美之辞。为什么会有这样大的差距呢？

Amazon上第一个发表评论的是该书的技术编辑Cameron Turner，评论题为Perfect in Every Way（一本十全十美的书），5星。他说："这是一本真正讲述构建下一代Web应用的书。到目前为止，这还是绝无仅有的。如果你需要为网站添加更多功能、灵活性和可访问性还有大势所趋的'耀眼的Web 2.0'特性，那么这本书是'必买'的。……需要提醒的是：如果你还是一个新手，可不要买这本书（因为它定位于'高级'这个层次上）。只有真正理解了CSS、JavaScript和HTML才能从本书中获益。相信本书将成为所有专业Web软件开发人员日常工作中时时查阅的必备图书。"

当然，Turner可以算是本书的参与者，有些偏爱在所难免。不过他指出了本书针对中高级读者，而不适合新手，这一点非常关键。实际上，Amazon上两位不满意的读者中，Richard（3星）就是因为看不太懂而发了牢骚。而另一位读者T.Dalmasso（1星）则对书中出现的错误非常恼火。幸运的是，他提到的错误在图灵公司给我的电子版文件里大部分已经修正。在翻译过程中，确实还发现了一些错误（主要是拼写和排版错误），但在我翻译过的书中已经算是比较少的了，远远没有多到令人生厌的程度，而且我基本上都已经解决了。

Amazon上最后发表评论的是资深.NET工程师David Betz "quantzai"（5星），他一上来就语出惊人："这是我见过的最好的一本现代JavaScript、DOM脚本编程和Ajax的书。这本书包括了JavaScript中从经常令人误解的变量作用域到与DOM深入交互等方方面面的内容。……本书是'研究生层次'的书，深入了Ajax的内幕，将使你成为专家，当然，也要求你一开始就具备思考力。"评论最后，David Betz "quantzai"对前面读者提出的拼写错误等问题给出了"反击"。他说："这又不是一本讲英语的书（出一两处错误在所难免），那些小错误根本无伤大雅；而且，即使没有这些错误你该理解不了，还是理解不了。"他甚至用"井蛙不可以语于海，夏虫不可以语于冰"，来表示对给出1星和3星的两位读者的不屑一顾。

在业内的知名blog圈里，瑞典哥德堡的资深Web工程师Roger Johansson这样说："我读了许多

blog和书，想搞清楚作用域、闭包、面向对象等JavaScript概念，但是一直苦苦挣扎，读了本书后，我想问题终于解决了。”DOMAssistant库的作者Robert Nyman则评价：“如果你是一位中级JavaScript开发人员，还想更上一层楼，那么这将是使你梦想成真的绝妙好书。”Godbit项目的Nathan Smith也给予了很高的赞誉：“我要说，这是我读过的最好的JavaScript图书之一。”

看到这里，你有什么想法？没错。还记得小马过河的故事吗？无论是老黄牛，还是小松鼠，都有自己的角度和立场。因此，听别人的评论虽然能够大致了解一本书的内容，还是代替不了自己的判断。以我的经验，要购买一本自己感兴趣的、专注于某一技术领域的书，一是要听听网友的评论，二是仔细看一看书的目录，三是挑挑书的装帧，四是拿着书走向收银台（或者单击“放入购物车”按钮）。但是别忘了，对于一本外版书而言，译者也是中文版的第一读者。因此，听一听这位读者的看法也很重要（终于轮到我了，呵呵）。

作为译者，我来谈一谈自己对这本书的看法，供读者参考。

这本书在面向标准的Web编程领域是名副其实的扛鼎之作，也难怪它有些曲高和寡。全书的内容，都是作为一名专业的Web开发人员（或者真正的高手）所必须了解和掌握的高级知识，没一点多余的内容，洋洋550页中绝无浮华不实之辞。而且，书中对核心JavaScript原理的总结和概括（如常见陷阱、作用域链解析、闭包、面向对象等）、对最佳实践的倡导和践行（包括对面向未来的现代Web开发趋势的归纳和宣传，即脚本必须不唐突和增强而不是提供行为等）、对DOM规范讲解的提纲挈领（好像还没有哪本书这么详细地讲解过DOM）、对浏览器外部通信（Ajax）的反思与解决之道、对Web 2.0内容整合（Mashup）的分类与讲说等，无一不折射出这本书是作者博观约取、厚积薄发的心血力作。最后（最后说的往往最重要），如果你也醉心于Prototype、Base、jQuery、YUI、Ext、Mochikit、DOMAssitant、Script.aculo.us、Moo.fx等这些优秀的JavaScript库，不知道多少次被它们的魅力所倾倒，也想见微知著地真正理解这些库背后的工作原理，甚至于希望创建自己的库，那么这本书恰好适合你——一名JavaScript高手的需要，因为学习完这本书，你就会拥有自己跨平台的ADS库了（你必须Get your hands dirty——动手编写这个库的每一行代码），这还不够酷吗？应该说，本书是一本全景式的、沟通历史和未来的Web开发经典好书，是对现有JavaScript DOM程序开发最佳实践的一次大检阅和大放送，是推动Web标准化和向下一代Web开发挺进的里程碑式著作。而且，根据译者（就是我）的个人体会，这些话绝非溢美之辞，句句都言之有据，译者也愿意和读者就本书内容进行交流，互相学习。

然而，为什么在Amazon上本书不像*DOM Scripting*（中译本《JavaScript DOM编程艺术》，人民邮电出版社）那么广受关注，甚至大受欢迎呢——*DOM Scripting*有近60人评论，给1星的只有两人，3星三人，4星15人，其余均为5星？

比较了一下两书的内容，可以发现，*DOM Scripting*一书针对的是初学者，尤其是编程经验并不是很丰富的Web前端开发和设计人员，所以行文浅显，门槛比较低，而且学习曲线也非常平滑，对于这些读者，本书可以说是比较完美的。讲JavaScript非常好懂，而且字里行间渗透着现代的Web开发思想。这也是国内很多读者都嫌内容太浅的原因，他们往往都是已经有不少经验的程序员了。对于这些读者，本书才是他们真正想找的那本。我们有理由相信它会受到大家的欢迎。

这是我与图灵公司合作的第一本书，也是我最喜欢的一本书。在此，我要感谢刘江老师的热

情邀请，感谢傅志红老师不厌其烦地回答我的问题和修改我的译稿，也感谢武卫东老师的悉心指导。同时，还要感谢本书的责任编辑杨爽，正是因为她创造性与我沟通，才使得本书在付梓前又消除了一些问题。不过，囿于个人水平和能力，翻译中的错误和不当之处在所难免。如果读者发现了书中的问题，请在我的个人网站http://www.cn-cuckoo.com中给予指出，或者将电子邮件发送到lsf.email@yahoo.com.cn。

本书由李松峰负责编译，参加翻译工作的还有李雅雯、程宝杰、宋会敏、闫建旺、左玉春、曹建辉、崔淑云、张井文、熊俊佳、左艳波、熊俊芹、刘英、赵淑云、贾爱华等。

译 者

2008年2月于北京

前　言

DOM（Document Object Model，文档对象模型）脚本编程经常会被误解为Web上的某种脚本编程，实际上，纯粹的DOM脚本编程只包括W3C DOM规范中所涵盖的特性和方法。也就是说，不包括任何专有的浏览器特性。在理想的世界里，我们可以遵循标准，忽略专有特性，最终完成可以在任何设备中运行的脚本。但这个世界不是理想的世界——目前还不是。众所周知，并非所有设备或浏览器都合乎W3C标准，那像我们这样的程序设计人员要满足每个人的要求该怎么办呢，怎样才能继续严格遵守W3C DOM规范呢？

当试图回答这些问题并在保持真正的DOM符合性基础上处理多浏览器时，我们萌生了写这本书的想法。本书不仅对以上问题给出了答案，而且还涉及下列主题。

- 深入W3C DOM规范，并筛选出经常容易被误解的细节，同时仍然为非标准浏览器提供等价选项。
- 进一步探讨新方法，例如Ajax客户端-服务器端通信，冲破Ajax的局限性以提供更具交互性的体验。
- 体验一些主要的第三方资源，通过它们省掉一些平淡的日常工作。
- 理解并创建自己日常所用的DOM方法库。

这些新的能力也带来了许多诱惑。我们在进行DOM脚本编程时，往往因为热衷于一些华而不实的新特性，而背离了良好清晰的Web设计原则。因而，纵贯全书作者都会强调最佳实践的价值，提供很多强调可用性和可访问性的解决方案，这样对最终用户和你——开发者或设计者而言，都是有益的。

你可以把这本书放在计算机旁作为参考，也可以从头到尾读完它。无论采取哪种方式，只要你坚持学习完本书中的理论、代码、例子和案例研究，就会深刻地发现自己已经很好地理解了书中那些高级概念的含义，不仅知其然，而且更知其所以然。

本书读者对象

本书适合对DOM感兴趣并希望进一步提升自己的所有Web开发者和设计者。通过本书通俗易懂的讲解，读者能够轻松地理解高级的DOM编程概念。如果读者对DOM脚本编程和Web标准有一些基本的经验，那么通过学习本书收获会更大。

本书组织方式

本书分三部分，通过学习全书内容，读者将能构建起属于自己的DOM实用方法库。书中的每一章都以前一章学习的概念为依托，因而本书的每一部分都是一个完整、独立的主题，而每一章则并非完全独立。

第一部分，“深入理解DOM脚本编程”，涉及W3C DOM规范的方方面面，包括非标准的浏览器支持和不支持的内容。从一开始就以最佳实践为榜样，然后你将了解到DOM2 HTML和DOM2核心规范，同时还有DOM2事件和DOM2样式规范。本部分中的每一章都会给出一些不针对特定浏览器的例子。而且，你也将着手构建自己的脚本程序库，并往其中添加各种方法去访问和操纵DOM、样式以及事件。这些方法将不针对特定的浏览器，因此你可以很容易地在公共方法（你将自己创建）的基础上建立自己的应用程序。第一部分最后的第6章将会完成一个案例研究，在这一章中，你将学会建立一个交互式裁剪和调整图像大小的工具。

在介绍了操纵和访问文档的各个方面知识之后，第二部分，“浏览器外部通信”，将以Ajax和客户端-服务器端通信为主题。在这一部分中，作者没有停留在介绍简单的做法上，而是深入解释了相应的内部工作机制，同时，也没有忘记介绍整合Ajax界面时可能遇到的麻烦。第二部分最后把这些技能用于实战检验，综合运用传统和当前的通信方法，创建了一个带有实时进度条的文件上传程序。

最后，在第三部分，“部分高级脚本编程资源”中，作者集中介绍了一批第三方脚本编程资源，包括库和API。你将在这一部分学习到如何利用主要的DOM脚本库来提高自己的开发效率，也包括使用一些视觉效果，为自己的Web应用程序添彩。同时，你还将学习如何通过可自由使用的API来整合交互式地图和项目管理工具。这些资源将为你提供高级编程能力，同时最大限度地减少你的重复性工作——但只有在对第一部分和第二部分内容深入理解的基础上，才能较好地体会到这些资源的价值。本书以Aaron Gustafson撰写的一个案例研究作为结尾，这个案例把`select`元素提高到了一个全新的水平。

作者没有提供附录，而是向读者公布了一个网站http://advanceddomscripting.com。在这个网站中，读者可以下载到本书的源代码及额外一些例子和参考文献。作者将在这个网站中发布与DOM脚本编程相关的最新的重要消息，读者可以经常访问这个网站，以便与时俱进。

本书约定

为保证本书内容清晰，容易理解，全书将遵守如下约定：

(1) 代码以等宽字体表示。

(2) 新加的或修改过的代码通常会以加粗的等宽字体表示。

(3) 伪代码和变量输入以斜体等宽字体表示。

(4) 对于需要提醒读者注意的内容，会像下面这样突出排版：

> 嗯，别说我没提醒过你！

(5) 有时候某一行代码可能会超出书本的宽度，这时将使用像下面这样的箭头图标➥：

```
这一部分的代码非常、非常、非常地长，应该全都写在不带有换行符的➥
同一行中。
```

(6) 为了节省版面，许多例子代码中都会包含“……省略的代码……”这样的文本行。这行文本表示相应位置处省略了一部分代码。这样做可以使当前例子中的主要代码更突出。在有些情形下，也会存在代码前后都有“……省略的代码……”字样的情况，这也是为了更好地在上下文中显示代码。例如，下面的代码：

```
(function(){
window['ADS'] = {};

……省略的代码……

function helloWorld(message) {
    alert(message);
}
……省略的代码……

})();
```

表示函数helloWorld()处于以(function(){开始和以})();结尾的代码块中。如果代码在所在文件中的位置也很重要，同样会加以标识。

(7) 在要求读者按照作者的指示构建例子的情况下，会以一个带注释的结构开头，比如：

```
// 定义一个变量
// 通过helloWorld()函数给出警告
```

这样，读者就可以在每条注释语句后面添加一些代码，以便构成例子：

```
// 定义一个变量
var example = 'DOM Scripting is great!';

// 通过helloWorld()函数给出警告
helloWorld(example);
```

这种方式可以使读者轻松地编写完成每个例子，并有助于理解每一步的意图。

(8) 在下载的本书源代码中，读者可能也会注意到书中没有出现的一些注释。书中删除了这部分注释，保证印刷出来的代码不致于太冗长。

阅读本书的条件

创造力、兴趣和学习的渴望就是阅读本书的全部条件，熟悉一些DOM、JavaScript、Web标准和CSS的基础知识当然也是有益无害的。只要有可能，作者会尝试着讲解一些高级主题，而且尽力以一名Web初学者都能够理解的方式来进行。

本书中的例子代码已经在下列浏览器中测试通过了。在书中，作者还将针对每种浏览器给出相应的提示：

- Firefox 1.5+

- Microsoft IE 6+
- Safari 2+
- Opera 9+

在其他兼容标准的浏览器中，同样可以运行本书的例子。但比较早的浏览器只能降级使用不含DOM脚本的版本。

如何下载代码

如果担心输入本书的全部代码会把手指头累得抽筋，作者建议你访问friends of ED的网站（http://friendsofed.com）并从中下载本书的源代码。同时在这个网站上，你还可以找到与Web标准、DOM、CSS以及Flash相关的其他好书。

如果你觉得不满足，那么还可以访问本书的站点（http://advanceddomscripting.com），其中除了源代码之外，还会提供与DOM脚本编程相关的更多信息。

如何联系作者

读者如果有任何问题、意见和想法，可以随时与作者联系，电子邮件地址是jeff@advanceddomscripting.com。同样，读者也可以经常浏览一下本书的网站 http://advanceddomscripting.com，作者将在此发表一些更新的内容和勘误信息，以及其他对读者有帮助的第一手资料。

Aaron Gustafson 的电子邮件地址是ads-book@easy-designs.net。

致谢

这几年来，在我的人生旅途中，有许多人都在影响着我——家人、朋友、相识的和不相识的，正是所有这些人的帮助才使得本书得以出版。要提及每个人恐怕不易，所以这里就一并致谢了。

感谢Chris Mills给我这次机会，让我能够深入到自己热爱的主题中。如果没有他的指导，我根本无法理清头脑中那些混乱的想法。而且，也要感谢friends of ED和Apress出版社的团队成员，包括Kylie Johnston、Heather Lang、Laura Cheu以及隐身于幕后的所有人，正是来自你们的反馈才使得这本书更加完善，从而让我有了一番不同的感受。

感谢Aaron Gustafson撰写了一个极好的案例。学习完这个案例一定会让读者因为超越了基本知识而感到惬意。

感谢Cameron Turner和Victor Sumner，他们测试了我编写的代码。相信他们测试出的或没测试出的问题一定不少，但读者的评论和鼓励一定会让我更加努力。

感谢那些慷慨地共享他们知识和思想的每一个人。如果没有他们深刻的评论、博客、图书、谈话和讨论，不可能有DOM脚本编程、Web标准的进步和创新。谢谢大家。

特别感谢我的妻子Stephanie和她给我的爱。感谢她在照顾我们刚出生的孩子和我时所表现出的耐心及给予我的极大的理解。没有她我将一事无成。

最后，感谢亲爱的读者花时间看这本书。我衷心地祝愿你理解我所讲的一切内容。

Jeffrey Sambells

有这么多人我得感谢，不过我只是撰稿人之一，所以还是简略点儿好。感谢Brothercake，他的表达天分着实给我的书稿增色不少。感谢Jeremy Keith，他让我说得更头头是道。感谢Shaun Inman，This is Cereal的设计真让人深受启发啊。感谢Jeffrey Zeldman，他坚定了我当一名好作者的信心。感谢Molly Holzschlag，我取得的成功皆应归功于她，在此，不管怎么表达谢意都是不够的（但我还是要表达我的谢意）。最后（也是最重要的），感谢Kelly，有多少个深夜和周末，我都在忙于编写程序，再写书分析程序，她都默默忍受了。我要对她说：我爱你。

Aaron Gustafson

目　录

Part 1 第一部分

深入理解 DOM 脚本编程

本部分内容

第1章

遵循最佳实践

1

你很兴奋，你的客户也很兴奋。你刚刚为客户安装启用了新版的网站，一切都很顺利。网站很花哨，它历经了数小时汗水和泪水的浇灌，每一处设计细节——扩展式菜单、交互的Ajax都精心调试过，所有新式花样无所不包。它看上去很不错，运行得也很完美，所有人都沉浸在喜悦中。然而，一周之后，噩梦开始了。客户慌里慌张地打来电话，好像是据他们的部分顾客来电话反应说主页上的链接打不开了，而另外一些顾客在填写反馈表单时也碰到了问题。但在你和你的客户那里却不存在这些的问题。还有一些人也打来电话，抱怨下载网站的每一个页面都要等很长时间，即使看上去网页内容并不很多，而你从来没有发现下载时间有问题。更糟糕的是，随后，你的客户发现网站在搜索引擎中的排名一落千丈。看来事情没有想象的那么乐观，但问题出在哪里呢？下面我们就一起来查找原因。

最佳实践是人们做事时应该遵循的、被公认和经过验证的模式。虽然不一定是唯一的，甚至不是最佳的方式，但这些方式是大多数人认同的做事方式。一般的书在最后部分会提到一些最佳实践，这更多地是一种提示，即在你已学会了每件事并按自己的方式行事时，告诉你还会有一种最适当的方式。我之所以把最佳实践放在前面来讲，就是为了让你在开始学习新知识之前，先明确正确的方向。如果有阳关大道，那又何必去走独木桥呢？

1.1 不唐突和渐进增强

XHTML（Extensible HyperText Markup Language，可扩展超文本标记语言）、CSS（Cascading Style Sheet，层叠样式表）和使用JavaScript的DOM（Document Object Model，文档对象模型）脚本是Web设计的三个主要部分。其中，XHTML用于提供文档结构的语义标记，CSS为文档布局提供定位和样式，而DOM脚本编程用于增强文档的行为和交互性。发现了吗？我刚才说DOM脚本“增强”，而不是为文档“提供”行为和交互性。“增强”和“提供”之间的差异暗示了一个重要区别。我们都学过XHTML语义，知道验证文档是否符合W3C规范，而且也都在用CSS来为严格型XHTML文档标记应用适当的样式（对吧？）。但是，作为第三个主要部分的DOM脚本，虽然它可以把事情做得格外漂亮，为我们的Web应用程序增光添彩，却有可能是一个唐突的家伙。

DOM脚本编程依赖于JavaScript。如果你在Web上搜索“Unobtrusive JavaScript”这两个关键词，将会发现网上泛滥着不同的描述，这只不过是因为你无处不需要考虑不唐突性（unobtr-

usiveness)。不过，这并不意味着不唐突性是“创可贴（Band-Aid）”，随便贴到你的代码上就万事大吉了。JavaScript中没有不唐突的对象（嗯……或许有？），而且也不可能下载到“不唐突”的库来解决代码中的问题。不唐突性只能来自于对Web应用程序正确的规划和准备。当需要用户和Web开发者彼此合作时就必须考虑不唐突性。你的脚本对用户来说必须是不唐突的，即消除不必要的行为和令人讨厌的功能（过去曾令JavaScript背负坏名声的一些东西）。脚本也必须是不唐突的，即在没有JavaScript的情况下，能使页面和标记持续有效，虽然不再那么优雅了。最后，不唐突的脚本必须在标记中容易实现，也就是通过ID和类属性来关联行为，进而保证脚本与标记的分离。要想更好地理解不唐突性，以及DOM脚本编程、XHTML和CSS整合的本质，需要考虑我们在前面情形中所看到的结果，并理解基本原理及其在代码中的应用。

我们经常会听到与不唐突性有关的两个术语——“渐进增强（progressive enhancement）”和“平稳退化（graceful degradation）”。某些技术能够实现，当浏览器支持相应功能时文档会得到增强（渐进增强），而当浏览器不支持相应功能时，文档被退化（平稳退化）。通过使用这些技术，不支持相应功能的浏览器也会获得同一文档的相同信息量但却不同的视图。这两个术语经常交替使用，但二者都承认现实：并非所有浏览器都遵循相同的标准创建，而且不能对所有浏览器一视同仁。同理，谁也不能为了迎合少数人而强迫所有人都接受一种低质量的服务。

Yahoo的Nate Koechley关于渐进增强必要性的论述，我特别赞同。下面是他在介绍浏览器支持的时候总结出来的相关内容（http://developer.yahoo.com/yui/articles/gbs/gbs.html）：

“支持并不意味着每个人都会得到完全一样的结果。期望两个使用不同浏览器的用户拥有相同的体验就是不认同Web的异构本性（heterogeneous essence）。事实上，要求所有用户都具有相同的体验会对参与性造成障碍。应该把内容的有效性和可访问性作为首要目标。”

如果你在使用JavaScript时妨碍了“内容的有效性和可访问性”，那么你的做法就是错误的。

同时，由于渐进增强并不是必需的，因而也就意味着要提供一种全有或全无JavaScript的方案。问题在于，JavaScript与那些在专用服务器或计算机中运行的编程及脚本语言不同，它是一种运行在Web浏览器中的解释型语言。因此，对于脚本需要处理的各种各样的软硬件和操作系统的组合我们无法控制。为了应对几乎无限种这样的组合，必须小心谨慎地基于浏览器的能力和可用技术来创建行为性的增强。这意味着要么提供一个仍然较少依赖于JavaScript脚本的平稳退化方案，要么提供一个借助于传统、没有JavaScript方法的方案。

虽然在本书中会强调不唐突性、平稳退化和渐进增强这些观点，但我也要声明自己并不是一个狂热的信徒。我知道你的网站确实需要支持具有与标准不兼容的特性的浏览器。在很多情况下，一些专有的方法仍然是必需的。这其中的关键是，要走与标准兼容的道路，只在必要时使用专有方法。

当使用DOM脚本编程并将它们整合到网站中时，你编写的脚本必须：

- **与标准兼容**。面向未来开发应用程序，确保Web应用程序能够在更新更好的浏览器中继续运行。
- **容易维护**。综合运用可重用和容易理解的方法，以便你和其他人能够集中关注业务逻辑，而不是反复重写代码。