

HTML5: Up and Running
Dive into the Future of Web Development



HTML5

揭秘

Mark Pilgrim 著

常可 胡金埔 赵静 译 赵泽欣 审校

O'REILLY® | Google™ PRESS



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

O'REILLY®

HTML5 揭秘

HTML5: Up and Running

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内容简介

本书全面而深入地対 HTML5 相关的技术进行详细介绍和剖析。“从开始到现在”道出 HTML5 的坎坷发展史;“HTML5 特性检测”介绍了多种针对不同特性的检测方法;“本地存储”揭开了如何把“数据库”搬到客户端的神秘面纱;“离线应用”展示了脱机状态下依然能让 Web 应用完好无损的神奇技艺;“疯狂的表单”大秀了一把炫酷无比的下一代 Web 表单……它几乎涵盖了 HTML5 标准中描述的所有新特性。本书在以诙谐的文字结合生动的实例介绍 HTML5 特性的同时,还深入剖析其内部原理。让读者不仅知其然,而且知其所以然。无论是刚接触 Web 前端技术的新人,还是经验丰富的老手,只要是 HTML5 技术爱好者,都会从本书中受益。

978-0-596-80602-6 HTML5: Up and Running © 2010 by O'Reilly Media, Inc. Simplified Chinese edition, jointly published by O'Reilly Media, Inc. and Publishing House of Electronics Industry, 2010. Authorized translation of the English edition, 2010 O'Reilly Media, Inc., the owner of all rights to publish and sell the same. All rights reserved including the rights of reproduction in whole or in part in any form.

本书中文简体版专有出版权由 O'Reilly Media, Inc. 授予电子工业出版社, 未经许可, 不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字: 01-2010-6579

图书在版编目(CIP)数据

HTML5 揭秘 / 皮尔格林 (Pilgrim, M.) 著; 常可, 胡金埔, 赵静译.

—北京: 电子工业出版社, 2010.12

书名原文: HTML5: Up and Running

ISBN 978-7-121-12408-2

I. ①H… II. ①皮… ②常… ③胡… ④赵… III. ①超文本标记语言, HTML—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 227311 号

策划编辑: 卢鹤翔

责任编辑: 杨绣国

项目管理: 杨绣国

印刷: 北京市天竺颖华印刷厂

装订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱

邮编: 100036

开本: 787×980 1/16 印张: 14 字数: 280 千字

印次: 2010 年 12 月第 1 次印刷

定价: 45.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn, 服务热线: (010) 88258888。

O'Reilly Media, Inc.介绍

为了满足读者对网络 and 软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权电子工业出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在 Unix、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是在线出版的先锋。

从最畅销的“The Whole Internet User's Guide & Catalog”（被纽约公共图书馆评为20世纪最重要的50本书之一）到 GNN（最早的Internet 门户和商业网站），再到 WebSite（第一个桌面 PC 的Web服务器软件），O'Reilly Media, Inc. 一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc. 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得 O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc. 依靠他们及时地推出图书。因为 O'Reilly Media, Inc. 紧密地与计算机业界联系着，所以 O'Reilly Media, Inc. 知道市场上真正需要什么图书。

译者序

HTML5 可能是眼下 IT 前端圈子里最为热门的词语，在乔布斯看来，未来的客户端也将是 HTML5 的天下。作为 Web 开发工程师更有理由来更新自我的知识体系，学好这门新技术。在如今这个信息爆炸的时代，一本好书往往就是一位良师益友，能帮你解开学习道路中的一个又一个谜题。《HTML5 揭秘》就是一本这样的书。

本书全面而又深入地对 HTML5 技术作了介绍和剖析。从文档类型到全新 HTML 标签，从视频到画布，从本地存储到离线应用，涵盖方方面面；从 MIME 类型的发展史到其详细含义，从本地存储方式到其内部实现原理，从 Form2.0 的全新标签到其使用注意事项，讲解深入浅出。

本书不是一本关于 HTML5 的手册，而是一本可以教会你使用 HTML5 的书。如果以学习修车为例，本书很少教你一步步如何操作，取而代之的是有关引擎的内部原理及其发展历史。比如第 4 章“Canvas 绘图”，并非只教给你怎么画，更多的是教你绘图的整个过程和原理；第 5 章“网络视频”，讲到了很多平时被我们忽视的视频容器、编解码器，以及一些相关历史。从而，我们越发地觉得，在一个领域用心研究的人，他的记忆是有时间点的，也就是熟悉那个领域相关技术的发展史。

作为译者，我们一方面深感荣幸，另一方面也倍感压力。我们知道，中文和英文思维有着很大的区别，倘若带着英文思维去直译，往往读者就会觉得晦涩，甚至误解了原文的本意，这对于译者而言就是莫大的罪过了。正如 Gilbert Highet 所说：“一本书写得不好，那只是一种过错，而将一本好书翻译得糟糕那就是犯罪了”。带着这种心态，我们希望能保持原文意思的基础上，尽可能地采用中文思维来翻译，呈现给读者一个中文的语境。

在这里，特别感谢博文视点的编辑们，是他们的见识使这本好书有可能尽早地与国内读者见面。由于本书覆盖面广，翻译难度也较大，我们虽然在翻译中下了不少功夫，但译文仍难免会出现一些疏漏，恳请认真阅读的同行和朋友们不吝赐教。

常可 胡金埔 赵静
2010 年 11 月

前言

Preface

引言

Diving In

HTML5 是什么？HTML5 是下一代的 HTML，将会取代 HTML 4.01，XHTML 1.0 及 XHTML 1.1。HTML5 将会提供现代 Web 应用程序所必需的新功能。它同时也会将许多用于 Web 平台上的技术进行标准化，开发者们已经使用了这些技术多年，但从未有标准委员会对其进行审核和归档。（“Window 对象从未被正式写进标准文档里”——你听到这个会不会惊讶？除了新功能/特性，还有许多那些已经出现多年，“既成事实”的浏览器标准功能，HTML5 将会首次尝试将这些东西都正式地文档化。）

和它的前辈一样，HTML5 也被设计为是跨平台的。你并不须要一定运行 Windows 或者 Mac OS X 或者 Linux 或者 Multics 或者其他任何特定的操作系统，即可享受 HTML5 的优势。唯一需要的就是一个现代的浏览器。所有主流操作系统上都有免费的现代浏览器。你或许已经在使用一个支持部分 HTML5 特性的浏览器了。比如最新版的 Apple Safari、Google Chrome、Mozilla Firefox 及 Opera，它们都支持许多的 HTML5 特性。（在本书中附有详细的浏览器兼容性列表。）在 iPhone、iPad 及 Android 系统上预装的移动设备浏览器也都对 HTML5 提供极好地支持。甚至微软也已经宣布即将问世的 Internet Explorer 9 会支持部分 HTML5 的功能。

本书的内容集中在以下八个主题：

- 新的语义元素，如<header>、<footer>和<section>（详见第 3 章）。
- Canvas，一个可以通过 JavaScript 编程控制的二维绘图界面（详见第 4 章）。
- 无须安装第三方插件就可以在网页中嵌入视频（详见第 5 章）。
- 地理定位，使得访问者可以通过 Web 应用程序分享他们的所在位置（详见第 6 章）。
- 无须使用第三方插件就能实现的永久性本地数据存储（详见第 7 章）。

- 离线 Web 应用程序，即使在没有连接网络的情况下也可以运作（详见第 8 章）。
- 经过改进的 HTML Web 表单（详见第 9 章）。
- 通过微数据创建你自己的 HTML5 扩展词汇表，赋予网页新的语义性（详见第 10 章）。

HTML5 被设计为尽可能地后向兼容现有的 Web 浏览器。新的特性都建立在已有特性的基础之上，使得你可以为旧版本的浏览器提供容错内容。如果需要更全面的控制，你还可以检测浏览器对各项 HTML5 特性的支持情况（参见第 2 章），这只需几行 JavaScript 就能做到。不要依赖脆弱的对浏览器的侦测来断定 HTML5 的支持情况！相反，使用 HTML5 本身来测试你想要用到的功能。

本书中所用的约定

Conventions Used in This Book

本书使用以下字体排版约定：

斜体 (*Italic*)

表示专业词汇、链接 (URL)、文件名和文件扩展名。

等宽字体 (Constant width)

表示广义上的计算机编码，它们包括变量或函数名、数据库、数据类型、环境变量、语句和关键字。



该图标表示一个提示、建议或一般的说明。



该图标表示警告或提醒。

示例代码的使用

Using Code Examples

本书的目的是为了帮助你搞定工作任务。在一般情况下，你都可以在程序和文档中使用本书中的代码。除非需要复制这些示例代码的相当大部分，否则无须联系我们以获得许可。比如说，当你编写的程序用到了本书中的若干示例代码，这并不需要特别许可。但是，销售或分发含有 O'Reilly 书籍附带的示例程序的光盘则需要获得许可。当你在回答他人问题时援引本书内容，或者引用书中的范例代码，也不用申请许可。而如果要把本书中的代码大量地引用到你的产品文档中，则需要许可。

对于引用时注明出处，我们表示感谢，但并不硬性要求。署名信息通常包括书名、作者、出版商及 ISBN。例如：“HTML5: Up and Running，由 Mark Pilgrim 撰写。O'Reilly 媒体公司 2010 年版权所有，978-0-596-80602-6。”

如果你觉得你使用示例代码的情况超出了以上描述的不需要许可的范围，请随时联系我们 permissions@oreilly.com。

关于本书的版本说明

A Note on the Editions of This Book

本书源自最初的 HTML5 网页版本，位于 <http://diveintohtml5.org/>，由作者本人维护。电子书和 Safari Books Online 版本包括所有的原始超链接，而印刷版只包括了其中一部分。如果你阅读的是印刷版，请参照其他版本，或原始在线版本，以获得更多信息。作者所维护的 <http://diveintohtml5.org/> 网站上包括了本书中的范例的现实版本，本书中，许多相关内容为了便于出版经过了一些修改。请访问 <http://diveintohtml5.org/> 来查看这些例子，但是要注意，不同的浏览器可能有不同的呈现结果。

如何联系我们

How to Contact Us

请将对本书的有关意见和问题告知出版商：

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街2号成铭大厦C座807室（100035）
奥莱利技术咨询（北京）有限公司

与本书有关的在线信息如下所示：

<http://oreilly.com/catalog/9780596806026/>（原书）
<http://www.oreilly.com.cn/book.php?bn=978-7-121-12408-3>（中文版）

北京博文视点资讯有限公司（武汉分部）

湖北省 武汉市 洪山区 吴家湾 邮科院路特1号 湖北信息产业科技大厦1402室

邮政编码: 430074

电话: (027)87690813 传真: (027)87690013

读者服务网页: <http://bv.csdn.net>

E-mail:

reader@broadview.com.cn (读者信箱)

bvtougao@gmail.com (投稿信箱)

目录

Table of Contents

前言	1
第 1 章 从开始到现在	1
1.1 引言	1
1.2 MIME 类型	1
1.3 很长的题外话：一份标准是如何诞生的？	2
1.4 未曾间断的路线	7
1.5 HTML 发展史：从 1997 到 2004 年	9
1.6 你所知道的关于 XHTML 的一切都是错误的	10
1.7 一个竞争愿景	11
1.8 WHAT 工作小组？	12
1.9 回到 W3C	13
1.10 后记	14
1.11 扩展阅读	14
第 2 章 HTML5 特性检测	15
2.1 引言	15
2.2 检测技术	15
2.3 Modernizr：一个 HTML5 特性检测库	16
2.4 画布	16
2.5 画布文本	17
2.6 视频	18
2.7 视频格式	19
2.8 本地存储	21
2.9 Web Workers	23
2.10 离线 Web 应用	23
2.11 地理位置	24
2.12 输入框类型	25
2.13 占位文本	27
2.14 表单自动聚焦	27
2.15 微数据	28

2.16 扩展阅读	29
第3章 从这一切的含义	31
3.1 引言	31
3.2 文档类型	31
3.3 根元素	33
3.4 <head>元素	34
3.4.1 字符编码	35
3.4.2 朋友和（链接）关系	36
3.5 HTML5 中新增的语义元素	41
3.6 题外话：浏览器如何处理未知元素	42
3.7 页头	45
3.8 文章	47
3.9 日期和时间	49
3.10 导航	51
3.11 页脚	52
3.12 扩展阅读	55
第4章 Canvas 绘图	57
4.1 引言	57
4.2 简单的图形	58
4.3 Canvas 坐标系	60
4.4 路径	61
4.5 文本	63
4.6 颜色渐变	67
4.7 图片	70
4.8 IE 怎么办？	73
4.9 一个完整的例子	75
4.10 扩展阅读	79
第5章 网络上的视频	81
5.1 前言	81
5.2 视频容器	81
5.3 视频编解码器	83
5.3.1 H.264	84
5.3.2 Theora	84
5.3.3 VP8	85
5.4 音频编解码器	85
5.4.1 MPEG-1 音频层 3	86
5.4.2 高级音频编码	87
5.4.3 Vorbis	87
5.5 在网页中怎么工作	88
5.6 H.264 视频的授权问题	90

5.7	使用 Firefogg 编码 Ogg 视频.....	91
5.8	使用 ffmpegtheora 批量编码 Ogg 视频.....	98
5.9	使用 HandBrake 编码 H.264 视频.....	100
5.10	使用 HandBrake 批量编码 H.264 视频.....	107
5.11	使用 ffmpeg 编码 WebM 视频.....	108
5.12	最后, 标记.....	110
5.12.1	MIME 类型很重要.....	113
5.13	IE 怎么办?	114
5.14	完整的例子.....	114
5.14	扩展阅读.....	115
第 6 章	地理位置.....	117
6.1	引言.....	117
6.2	地理位置 API.....	117
6.3	代码展示.....	118
6.4	容错处理.....	120
6.5	方案! 我要方案!	121
6.6	那 IE 怎么办?	123
6.7	geo.js 来拯救.....	123
6.8	一个完整的例子.....	125
6.9	扩展阅读.....	126
第 7 章	Web 应用本地存储的过去、现在和未来.....	127
7.1	引言.....	127
7.2	HTML5 之前的伪本地存储简史.....	128
7.3	HTML5 存储介绍.....	129
7.4	使用 HTML5 存储.....	130
7.4.1	跟踪 HTML5 存储区的改动.....	131
7.4.2	现有浏览器的局限性.....	132
7.5	HTML5 存储实践.....	132
7.6	超越键值对的存储形式.....	134
7.7	扩展阅读.....	135
第 8 章	离线 Web 应用.....	137
8.1	引言.....	137
8.2	缓存清单.....	138
8.2.1	“网络”段.....	139
8.2.2	“默认”部分.....	140
8.3	事件流.....	141
8.4	调试的艺术——杀了我! 现在就杀了我!	142
8.5	让我们来构建一个离线 Web 应用!	145
8.6	扩展阅读.....	146

第 9 章 疯狂的表单	147
9.1 引言	147
9.2 占位文本	147
9.3 自动聚焦	148
9.4 Email 地址	150
9.5 Web 地址	151
9.6 数字类型输入框：数字选择器	153
9.7 数字类型输入框：滑块	155
9.8 日期选择器	156
9.9 搜索框	158
9.10 颜色选择器	160
9.11 还有一点……	160
9.12 扩展阅读	161
第 10 章 “分布式”、“可扩展性”及其他华丽词藻	163
10.1 引言	163
10.2 什么是微数据？	164
10.3 微数据的数据模型	165
10.4 标注“人”	168
10.4.1 Google Rich Snippets 介绍	174
10.5 标注“组织”	176
10.6 标注“事件”	180
10.6.1 Google Rich Snippets 的回归	184
10.7 标注“点评”	186
10.8 扩展阅读	190
附录 A 全方位特性检测指南	191
元素列表	191
扩展阅读	199
索引	201

从开始到现在

How Did We Get Here?

1.1 引言

Diving In

前不久，我偶然读到某 Mozilla 开发人员的一个观点，内容是关于在创建一种标准的过程中存在的冲突 (<http://lists.w3.org/Archives/Public/public-html/2010Jan/0107.html>)：

一份技术规范和它的具体实现必须要做到步调一致。实现先于规范完成不是什么好事情，因为人们会开始依赖这些已实现的细节，这样会对规范形成制约。然而，你也不希望在规范已经完成时还没有任何相关的具体实现和实践经验，因为这样规范就得不到任何反馈。这里面如果存在着无法避免的冲突，而我们也只能硬着头皮去克服了。

请你先牢记这番话，接下来我开始讲述 HTML5 是如何横空出世的。

1.2 MIME 类型

MIME Types

本书的内容是关于 HTML5，不是旧版本的 HTML，也不是任何版本的 XHTML。但要理解 HTML5 的历史及其背后的发展动力，你需要先了解一些技术细节。具体来说，就是 MIME 类型。

每当 Web 浏览器请求一个页面时，Web 服务器会在发送实际页面内容之前，先发送一些头信息 (header)。这些头信息通常是不可见的，当然如果你有兴趣，也可以借助许多 Web 开发工具来查看到这些信息。看不到不代表不重要，事实上，浏览器需要这些信息来决定如何解析随后的页面内容。最重要的头信息是 Content-Type，比如：

```
Content-Type: text/html
```

“text/html”即是这个页面的“内容类型 (content type)”，或者称作“MIME 类型”。这条头信息将唯一确定某个资源本质上是什么，也因而决定了它应该如何被呈现。图片文件有自己的 MIME 类型 (JPEG 图像的类型是 image/jpeg, PNG 图像则是 image/png, 等等)。JavaScript 文件有它自己的 MIME 类型。CSS 样式表也有自己的 MIME 类型。任何

资源都有自己的 MIME 类型。整个 Web 都依靠 MIME 类型运作。

当然，现实情况比这还要复杂得多。非常早期的 Web 服务器（我指的是 1993 年）并不会发送 Content-Type 头信息，因为它当时压根还不存在。（1994 年以前还没有“内容类型”这一概念。）为了兼容到这样的情况，某些流行的浏览器还会在一些特定情况下忽略 Content-Type 头信息。（这一行为被称为“内容嗅探”。）但作为一个一般性的经验法则，凡是在一个网页上能看到的东西，比如 HTML，图片，脚本，视频，PDF 文件，所有这些可以用一个 URL 地址定位的东西，都具有特定的 MIME 类型——在 content-type 头信息中指明。

把上面这些内容先塞到脑子里吧。在后面还会复习到。

1.3 很长的题外话：一份标准是如何诞生的？

A Long Digression into How Standards Are Made

为什么会存在这个元素？这个问题想必没有人会经常考虑。显然，必定是有人创造了它。这种东西才不会自己从石头里蹦出来。每个元素，每个属性，每个你所用过的 HTML 特性都是这样——某些人创造了它们，决定它们的用法，并且把全部相关事项都记录了下来。这些人不是造物主，不会完美无暇。他们是人——当然，是很聪明的人——但毕竟也只是人。

“公开”地制定标准有一大好处，你可以重返过去，寻找上述问题的答案。与之相关的讨论都会出现在邮件列表里，这些邮件通常留有存档，并且可以被公开搜索。所以，我决定做一下对电子邮件的考古，来尝试回答这个关于元素的问题。为此我得追溯到万维网联盟(W3C)诞生之前。回到 Web 时代最开始的日子，那时候 Web 服务器的总数还屈指可数——好吧，也许再加上脚趾。

在 1993 年 2 月 25 日，马克·安德雷森（Marc Andreessen）写道：¹

我想提议一个新的可选 HTML 标签：

IMG

其必需参数是 SRC="url"。

它指明一个位图或像素图文件，让浏览器尝试通过网络获取并解析为一幅图像，显示于页面上标签所在的位置。

例如：

```
<IMG SRC="file:///foobar.com/foo/bar/blargh.xbm">
```

（没有结束标记，它是一个独立标签。）

¹译注：<http://1997.webhistory.org/www.lists/www-talk.1993q1/0182.html>。点击“下一条消息”和“上一条消息”的链接可以跟踪阅读这份长达数页的讨论。

这个标签可以像其他东西一样嵌在锚点元素里，此时，它就变成一个图标链接，和文字链接一样可以被激活。

对于支持何种图像格式，浏览器应该有足够的灵活性。例如，Xbm 和 Xpm 就应该被支持。如果浏览器无法解析给定的格式，它可以自行决定做何种代替处理（X Mosaic 会显示一个默认图像作为占位符）。

对这个标签提供支持是 X Mosaic 的必要功能；我们已经将其实现，并且至少会在内部使用它。我也乐于听到任何有关如何在 HTML 中处理图像的建议；如果你有更好的主意，烦请告知。我知道，关于图像格式这一点，这里的提议还有些模糊，但我也想不出其他办法，只能说“让浏览器自行处理”，然后等待完美解决方案的出现（MIME？也许吧）。

以上内容需要一些解释。Xbm 和 Xpm 是当时 Unix 系统上流行的图像格式。

“Mosaic”是最早的浏览器之一。（“X Mosaic”是其运行于 Unix 系统上的版本。）当马克在 1993 年初写下这篇帖子时，他尚未成立那家将让他一举成名的公司：Mosaic 通信公司，也还没有开始设计公司的主打产品“Mosaic Netscape。”（你可能更熟悉它们后来的名字，“网景公司”（Netscape Corp.）和“网景浏览器。”（Netscape Navigator））

“MIME？也许吧”这句话影射的是内容协商(content negotiation)，这是一个 HTTP 特性，其中：客户端（如 Web 浏览器）借此告诉服务器（如 Web 服务器）它支持什么类型的资源（如 image/jpeg），然后服务器就可以按客户端需要的格式返回内容。最初的超文本传输协议(HTTP)定义于 1991 年（1993 年 2 月时有了唯一的实现版本），而其中还没有办法能让客户端告诉服务器它所能支持的图像格式，这就是马克在设计上所面临的困境。

几个小时后，托尼·约翰逊（Tony Johnson）回复道：

我在 Midas 2.0（用于 SLAC，很快就要对外发布了）中也有很类似的东西，除了名称不同，以及还有一个额外参数 NAME="name"。在功能上，和你提出的 IMG 标签则几乎是一模一样的。例如：

```
<ICON name="NoEntry" href="http://note/foo/bar/NoEntry.xbm">
```

给出这个 name 属性是为了允许浏览器使用一系列的“内置”图像。如果 name 匹配了“内置”图像，浏览器就直接使用它，而不用再去网络上抓取。这个属性也可以作为针对“行模式”浏览器的提示信息，告诉它在图像位置显示什么符号。

我不太关心具体参数或者标签名称，不过如果我们都用相同的命名，倒是更为明智。我也不很在意缩写，比如为什么不用写全的 IMAGE=以及 SOURCE=。不过我倒更倾向于使用 ICON 作为标签名称，因为它意味着应该用较小的图片。不过也许 ICON 又是个意思太多的词？

这里提到的 Midas 是另一个早期的 Web 浏览器，和 X Mosaic 同时代。它是跨平台的，在 Unix 和 VMS 上都能运行。“SLAC”是斯坦福直线加速器中心——现在的斯坦福国家加

速器实验室——的名字缩写。美国第一台 Web 服务器（也是欧洲以外的第一台 Web 服务器）就位于这里。在托尼写下这篇帖子时，SLAC 已经是万维网的老前辈了，它的 Web 服务器上已存放有五个网页达 441 天之久。

托尼继续写道：

既然我们正在讨论关于新的标签这一话题，我还有一个类似的标签，想在 Midas 2.0 中提供支持。它大致是：

```
<INCLUDE HREF = "...">
```

这个标签的作用是把另一个文档包含进来，置于当前文档中该标签所出现的位置。原则上，所包含的文件可以是任何类型，但其主要目的还是为了在文档中嵌入图片（此例中图片可以是任意大小）。此外，当 HTTP2 成形时，包含文档的具体格式就可以另外协商了。

“HTTP2”指的是 1992 年诞生的基本超文本传输协议(Basic HTTP)。1993 年初的时候，这一协议很大程度上仍然没有任何实现。被称为“HTTP2”的规范草案持续进化，并最终标准化，成为“HTTP 1.0”。HTTP 1.0 确实包括有用于内容协商的请求头信息（request headers），也就是那个“只是也许”的“MIME”。

托尼继续说：

我正在考虑另一种选择是：

```
<A HREF = "... " INCLUDE>See photo</A>
```

我不是很喜欢给<A>标签添加更多的功能，这里的写法是为了那些对不支持 INCLUDE 参数的浏览器也保持兼容性。我的想法是，那些支持 INCLUDE 参数的浏览器，会把链接里的文字（本例中的“See photo”）替换为将被包含的文件（图片），那些老旧的、愚蠢的浏览器就干脆完全忽略掉 INCLUDE 标签。

这一提议从未付诸实施，尽管这个“在图像缺失的情况下显示一段替代文本”是一个重要的，提高网页可用性的技术。而这一点在马克最初的提案里是没有的。多年以后，这个特性被规定为属性，Netscape 则错误地将它实现成一个 tooltip 提示信息。

在托尼发帖几个小时之后，蒂姆·伯纳斯·李回帖说：

我原本觉得图像应该这样标记

```
<a name=fig1 href="fghjkdfghj" REL="EMBED, PRESENT">Figure</a>
```

其中 rel 属性的值的意思是：

EMBED 呈现时嵌入链接所指内容

PRESENT 呈现源文件时，始终呈现这个标签内容

请注意，这些值可以任意组合，如果浏览器一个都不支持，它也不应该崩溃。

[我]发现要使用这种方法创建可选择的图标，就意味着嵌套的链接标记。嗯……但我不要一个特殊的标签。