

“The Missing Manual series is simply the most intelligent and usable series of guidebooks...”
—KEVIN KELLY, CO-FOUNDER OF WIRED

HTML5

the missing manual®

The book that should have been in the box®

Second
Edition



O'REILLY®

Matthew MacDonald



Answers found here!

HTML5 is more than a markup language—it's a collection of several independent web standards. Fortunately, this expanded guide covers everything you need in one convenient place. With step-by-step tutorials and real-world examples, *HTML5: The Missing Manual* shows you how to build web apps that include video tools, dynamic graphics, geolocation, offline features, and responsive layouts for mobile devices.

the missing manual®

The book that should have been in the box®

The important stuff you need to know

- **Add audio and video without plugins.** Build playback pages that work in every browser.
- **Create stunning visuals with Canvas.** Draw shapes, pictures, and text; play animations; and run interactive games.
- **Jazz up your pages with CSS3.** Add fancy fonts and eye-catching effects with transitions and animation.
- **Design better web forms.** Collect information from visitors more efficiently with HTML5 form elements.
- **Build it once, run it everywhere.** Use responsive design to make your site look good on desktops, tablets, and smartphones.
- **Include rich desktop features.** Build self-sufficient web apps that work offline and store the data users need.



Matthew MacDonald is a science and technology writer

with more than a dozen books to his name. He's known for books about building websites, including *Creating a Website: The Missing Manual* and *WordPress: The Missing Manual*, as well as quirky handbooks like *Your Brain: The Missing Manual* and *Your Body: The Missing Manual*.

US \$39.99

CAN \$41.99

ISBN: 978-1-449-36326-0



9 781449 363260



O'REILLY®

missingmanuals.com

twitter: @missingmanuals

facebook.com/MissingManuals

HTML5

Ấn bản lần 2

the **missing** manual®

Cuốn sách nên có trong hộp®

Matthew MacDonald

O'REILLY®

Bắc Kinh | Cambridge | Farnham | Köln | Sevastopol | Tokyo

HTML5: The Missing Manual, 2nd Edition của Matthew MacDonald

Bản quyền © 2014 Matthew MacDonald. Đã đăng ký Bản quyền.
In tại Hoa Kỳ.

Được xuất bản bởi O'Reilly Media, Inc.,
1005 Gravenstein Highway North, Sebastopol, CA 95472.

Sách của O'Reilly có thể được mua để sử dụng cho mục đích giáo dục, kinh doanh hoặc quảng cáo bán hàng.
Phiên bản trực tuyến cũng có sẵn cho hầu hết các đầu sách <http://my.safaribooksonline.com>).

(Để biết thêm thông tin, hãy liên hệ với bộ phận bán hàng của công ty/tổ chức của chúng
tôi: (800) 998-9938 hoặc ty@oreilly.com .

Tháng 8 năm 2011: Phiên bản đầu tiên.

Tháng 12 năm 2013: Phiên bản thứ hai

Lịch sử sửa đổi cho phiên bản thứ hai:

2013-12-09	Phát hành đầu tiên
Nhìn thấy http://oreil.ly/html5tmm_2e	để biết chi tiết phát hành.

Sách hướng dẫn bị thiêu là nhãn hiệu đã đăng ký của O'Reilly Media, Inc. Biểu tượng Sách hướng
dẫn bị thiêu và "Cuốn sách lê rạ phái ở trong hộp" là nhãn hiệu của O'Reilly Media, Inc. Nhiều ký
hiệu được sử dụng bởi các nhà sản xuất và người bán để phân biệt sản phẩm của họ được tuyên bố là
nhãn hiệu. Khi những ký hiệu đó xuất hiện trong cuốn sách này và O'Reilly Media biết về khiếu nại
nhãn hiệu, thì các ký hiệu đó được viết hoa.

Mặc dù mọi biện pháp phòng ngừa đã được thực hiện trong quá trình chuẩn bị cuốn sách này, nhà xuất
bản và tác giả không chịu trách nhiệm về những sai sót hoặc thiếu sót, hoặc những thiệt hại do việc sử
dụng thông tin trong đó.

ISBN-13: 978-1-4493-6326-0

[LSI]

nội dung

Các khoản tín dụng còn thiếu vii

Giới thiệu xi

Phần thứ nhất: Đánh dấu hiện đại

CHƯƠNG 1: Giới thiệu HTML5	3 Câu chuyện về
HTML5	
3 Ba nguyên tắc chính của HTML5.	
7 Cái nhìn đầu tiên của bạn về HTML5 Markup.	
10 Xem xét kỹ hơn về Cú pháp HTML5.	
16 Họ phần tử của HTML5.	
21 Sử dụng HTML5 ngay hôm nay.	
	26
CHƯƠNG 2: Cấu trúc các trang với các phần tử ngữ nghĩa	37 Giới thiệu các phần tử ngữ
nghĩa.	38 Trang bị thêm Trang
HTML truyền thống.	39 Khả năng
tương thích của trình duyệt đối với các yếu tố ngữ nghĩa.	
51 Thiết kế một trang web với các yếu tố ngữ nghĩa.	
53 Hệ thống phác thảo HTML5.	
	65
CHƯƠNG 3: Viết phần đánh dấu có ý nghĩa hơn	75
Các yếu tố ngữ nghĩa được xem xét lại.	76
Các tiêu chuẩn khác thúc đẩy ngữ nghĩa.	82
Một ví dụ thực tế: Trang bị thêm trang "Giới thiệu về tôi".	88 Cách Công cụ Tìm
kiếm Sử dụng Siêu dữ liệu.	93
CHƯƠNG 4: Xây dựng biểu mẫu web tốt hơn	103 Tím hiểu biểu
mẫu.	104 Cài
tiến Biểu mẫu HTML Truyền thông.	105
Xác thực: Dừng Lỗi.	
112 Trình duyệt Hỗ trợ Biểu mẫu Web và Xác thực.	119 Các
loại đầu vào mới.	
123 Nguyên Tố Mới.	
Trình soạn thảo HTML trong Trang Web.	136

Phần thứ hai: Video, Đồ họa và Glitz

CHƯƠNG 5: Âm thanh và Video	143
Sự phát triển của Web Video.	
144 Giới thiệu Âm thanh và Video HTML5.	145
Tìm hiểu Định dạng Phương tiện HTML5.	149
Cách Làm hài lòng Mọi Trình duyệt.	154
Trình phát của Bạn bằng JavaScript.	160
Video.	169
CHƯƠNG 6: Phông chữ ưa thích và hiệu ứng với CSS3.....	177
Sử dụng CSS3 ngay hôm nay.	178
Xây Dựng Những Chiếc Hộp Tốt Hơn.	
184 Tạo Hiệu ứng với Hiệu ứng Chuyển tiếp.	
185 Phông chữ Web.	206
CHƯƠNG 7: Thiết kế web đáp ứng với CSS3	221
Thiết kế đáp ứng: Khái niệm cơ bản.	
222 Điều chỉnh Bố cục của Bạn với Truy vấn Phương tiện.	231
CHƯƠNG 8: Vẽ cơ bản với Canvas.....	245
Bắt đầu với Canvas.	
246 Xây Dựng Chương Trình Paint	
Cơ Bản	263
Khả năng tương thích của trình duyệt cho Canvas.	271
CHƯƠNG 9: Canvas nâng cao: Tính tương tác và Hoạt ảnh	275
Những thứ khác bạn có thể vẽ trên Canvas.	
275 Bóng đồ và Tô màu ưa thích.	281
Làm cho hình dạng của bạn tương tác.	293
Canvas.	300
Một ví dụ thực tế: Trò chơi Mê cung.	307
Phần thứ ba: Xây dựng ứng dụng web	
CHƯƠNG 10: Lưu trữ dữ liệu của bạn	319
Thông tin cơ bản về lưu trữ web	
320 Tìm hiểu sâu hơn về Lưu trữ web.	
326 Đọc Tập Tin.	
332 IndexedDB: Công cụ cơ sở dữ liệu trong trình duyệt.	340
CHƯƠNG 11: Chạy ngoại tuyến	355
Tệp đệm với một Bản kê khai.	356
Kỹ Thuật Lưu Trữ Thực Tế.	366

CHƯƠNG 12: Giao tiếp với Web Server	375
Gửi tin nhắn đến máy chủ Web.	376
Sự kiện do máy chủ gửi.	386
Ở cảm web.	393

CHƯƠNG 13: Định vị địa lý, Công nhân web và Quản lý lịch sử...	401
Định vị địa lý.	402
Công nhân web.	414
Quản lý lịch sử.	425

Phần thứ tư: Phụ lục

PHỤ LỤC A: CSS cơ bản	435
Thêm kiểu vào một trang web.	435 Cấu tạo của Biểu định kiểu.
kiểu nâng cao hơn một chút.	436 Tờ định Tour.
	440 A Style Sheet
	445
PHỤ LỤC B: JavaScript: Bộ não của trang của bạn	451
Cách một trang web sử dụng JavaScript.	452
Một Vài Yếu Tố Cần Thiết Về Ngôn Ngữ.	
459 Tương tác với Trang.	470
Mục lục.....	477

Các khoản tín dụng còn thiếu

THÔNG TIN VỀ CÁC TÁC GIẢ



Matthew MacDonald là một nhà văn khoa học và công nghệ với hơn chục cuốn sách mang tên ông. Những người mới làm quen với web có thể rón rén truy cập Internet với [an toàn](#) trong [Office 2013: Các Còn thiếu hướng dẫn sử dụng](#), [Excel 2013: Các Còn thiếu Thủ công](#). Và con người của tất cả các mô tả có thể khám phá chỉ họ thực sự kỳ lạ làm sao trong những cuốn sổ tay kỹ quặc [Của bạn Não: Các Còn thiếu Thủ công và Của bạn Thân hình: Các Còn thiếu Thủ công](#).

GIỚI THIỆU VỀ ĐỘI NGŨ SÁNG TẠO

Nan Barber (biên tập viên) đã làm việc với sê-ri Missing Manual kể từ khi nó bắt đầu. Cô ấy sống ở Massachusetts với chồng và nhiều thiết bị Apple và Android khác nhau. E-mail: nanbarber@oreilly.com.

Kristen Brown (biên tập sản xuất) tốt nghiệp chương trình xuất bản tại Emerson College. Cô ấy sống ở khu vực Boston với chồng và bộ sưu tập lớn sách và trò chơi trên bàn cờ của họ. E-mail: kristen@oreilly.com.

Kara Ebrahim (chuyển đổi) sống, làm việc và vui chơi ở Cambridge, MA. Cô ấy yêu thích thiết kế đồ họa và tất cả những thứ ngoài trời. E-mail: kebrahim@oreilly.com.

Julie Van Keuren (nhân viên hiệu đính) bỏ công việc làm báo vào năm 2006 để chuyển đến Montana và sống với giấc mơ làm nghề tự do. Cô và chồng (người đang sống với giấc mơ viết tiểu thuyết) có hai con trai. E-mail: julie_media@yahoo.com

Julie Hawks (người lập chí mục) là một giáo viên và học sinh vinh cửu. Người ta có thể thấy cô ấy đang lang thang với chiếc máy ảnh trên tay. E-mail: juliehawks@gmail.com.

Shelley Powers (người đánh giá kỹ thuật) là cựu thành viên nhóm làm việc HTML5 và là tác giả của một số cuốn sách của O'Reilly. Trang mạng: <http://burningbird.net>.

Darrell Heath (nhà phê bình kỹ thuật) là một nhà thiết kế web/in ấn tự do và nhà phát triển web đến từ Newfoundland và Labrador, Canada, với nền tảng về Công nghệ thông tin và nghệ thuật thị giác. Anh ấy là tác giả của nội dung hướng dẫn hàng tuần cho NAPP, tạp chí và lớp [Planet Photoshop](#), và trong thời gian rảnh rỗi, anh ấy cung cấp các mẹo liên quan đến thiết kế và công nghệ thông qua blog của mình tại www.heatrowe.com/blog darrell@heatrowe.com.

SỰ NHИН NHẬN

Không tác giả nào có thể hoàn thành một cuốn sách mà không có một đội quân nhỏ gồm những cá nhân hữu ích. Tôi mang ơn sâu sắc toàn bộ nhóm Missing Manual, đặc biệt là biên tập viên Nan Barber của tôi, người dường như chưa bao giờ bối rối trước sự thay đổi liên tục của HTML5; và các chuyên gia đánh giá công nghệ Shelley Powers và Darrell Heath, những người đã giúp phát hiện ra các lỗi bất hợp pháp và luôn đưa ra lời khuyên hữu ích. Và, như mọi khi, tôi cũng mang ơn sâu sắc đối với nhiều người khác, những người đã làm việc cật lực ở hậu trường để lập chỉ mục các trang, vẽ hình và hiệu đính bản sao cuối cùng.

Cuối cùng, đối với những phần đời tôi tồn tại bên ngoài cuốn sách này, tôi muốn cảm ơn tất cả những người thân trong gia đình tôi. Họ bao gồm bố mẹ tôi, Nora và Paul; cha mẹ ruột của tôi, Razia và Hamid; vợ tôi, Faria; và các con gái của tôi, Maya, Brenna và Aisha.

Cảm ơn mọi người!

—Matthew MacDonald

SERIES HƯỚNG DẪN MẤT TÍCH

Sách hướng dẫn còn thiếu là những hướng dẫn dí dỏm, được viết tuyệt vời cho các sản phẩm máy tính không đi kèm với sách hướng dẫn in (chỉ là về tất cả chúng). Mỗi cuốn sách có một mục lục thủ công; tham chiếu chéo đến các trang cụ thể (không chỉ các chương); và RepKover, một loại gáy sách có gáy tách rời cho phép cuốn sách nằm phẳng hoàn toàn mà không cần sự hỗ trợ của quả nặng hoặc khói than.

Các tiêu đề gần đây và sắp tới bao gồm:

Truy cập 2013: Các Còn thiếu Thủ công bởi Matthew MacDonald

adobe Bờ rìa sinh động: Các Còn thiếu Thủ công bởi Chris Grover

mua mót Nhà: Các Còn thiếu hướng dẫn sử dụng bởi Nancy Conner

Tạo một Trang mạng:Các Còn thiếu Thủ công, Ngày thứ ba Phiên bản bởi Matthew MacDonald

CSS3: Các Còn thiếu Thủ công, Ngày thứ ba Phiên bản của David Sawyer McFarland

David của Pogue Kỹ thuật số Nhiếp ảnh: Các Còn thiếu Thủ công bởi David Pogue

thợ dệt mộng CS6: Các Còn thiếu Thủ công của David Sawyer McFarland

thợ dệt mộng CC: Các Còn thiếu Thủ công của David Sawyer McFarland và Chris Grover

Excel 2013: Các Còn thiếu Thủ công bởi Matthew MacDonald

FileMaker Chuyên gia 12: Các Còn thiếu Thủ công của Susan Prosser và Stuart Gripman

Tốc biến CS6: Các Còn thiếu Thủ công bởi Chris Grover

ngân hà Chuyên hàng: Các Còn thiếu Thủ công của Preston Grala

Google+: Các Còn thiếu Thủ công của Kevin Purdy

iMovie '11 & iDVD: Các Còn thiếu Thủ công của David Pogue và Aaron Miller

máy tính bảng: Các Còn thiếu Thủ công, thứ sáu Phiên bản bởi JD Biersdorfer

điện thoại iPhone: Các Còn thiếu Thủ công, Thứ năm Phiên bản bởi David Pogue

điện thoại iPhone Ứng dụng Sự phát triển: Các Còn thiếu Thủ công bởi Craig Hockenberry

iPhoto '11: Các Còn thiếu Thủ công của David Pogue và Lesa Snider

iPod: Các Còn thiếu Thủ công, thứ mười một Phiên bản bởi JD Biersdorfer và David Pogue

Javascript & jQuery: Các Còn thiếu Thủ công, Thứ hai Phiên bản của David Sawyer McFarland

Thiếu Ngôn ngữ HD: Các Còn thiếu Thủ công bởi Peter Meyers

Cuộc sống Mùa xanh lá: Các Còn thiếu Thủ công bởi Nancy Conner

Microsoft Dự án 2013: Các Còn thiếu Thủ công của Bonnie Biafore

Motorola Xoom: Các Còn thiếu Thủ công của Preston Grala

Netbook: Các Còn thiếu Thủ công bởi JD Biersdorfer

NÓI HD: Các Còn thiếu Thủ công của Preston Grala

Văn phòng 2011 cho Macintosh: Các Còn thiếu Thủ công bởi Chris Grover

Văn phòng 2013: Các Còn thiếu Thủ công của Nancy Conner và Matthew MacDonald

X Núi con su từ: Các Còn thiếu Thủ công bởi David Pogue

X Mavericks: Các Còn thiếu Thủ công bởi David Pogue

Đầu tư cá nhân: Các Còn thiếu Thủ công của Bonnie Biafore

photoshop CS6: Các Còn thiếu Thủ công của Lesa Snider

photoshop CC: Các Còn thiếu Thủ công của Lesa Snider

Phần tử Photoshop 12: Các Còn thiếu Thủ công bởi Barbara Brundage

PHP & MySQL: Các Còn thiếu Thủ công, Thứ hai Phiên bản bởi Brett McLaughlin

Sách nhanh 2013: Các Còn thiếu Thủ công của Bonnie Biafore

chuyển sang Mac: Các Còn thiếu Thủ công, Núi Con su từ Phiên bản bởi David Pogue

chuyển sang Mac: Các Còn thiếu Thủ công, Mavericks Phiên bản bởi David Pogue

Windows 8.1: Sổ tay còn thiếu bởi David Pogue

WordPress: Các Còn thiếu Thủ công bởi Matthew MacDonald

Của bạn Thành hình: Các Còn thiếu Thủ công bởi Matthew MacDonald

Của bạn Não: Các Còn thiếu Thủ công bởi Matthew MacDonald

Của bạn Tiền bạc: Các Còn thiếu Thủ công bởi JD Roth

Để có danh sách đầy đủ tất cả các Sổ tay hướng dẫn còn thiếu được in, hãy truy cập www.missingmanuals.com/library.html.

Giới thiệu

Thoạt nhìn, bạn có thể cho rằng HTML5 là phiên bản thứ năm của HTML trang web-ngoài ngữ viết. Nhưng câu chuyện thực sự phức tạp hơn rất nhiều.

Tại

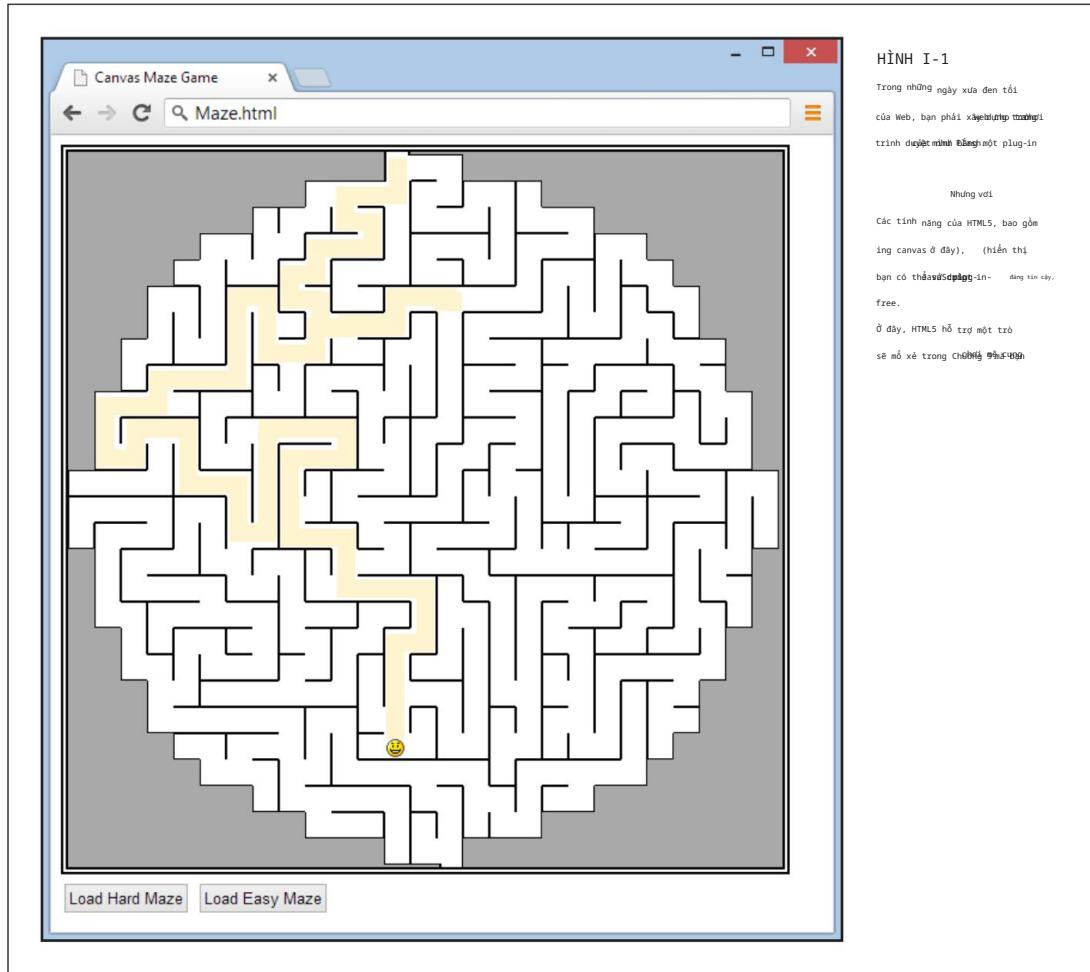
HTML5 là một kẻ nỗi loạn. Nó được mơ ước bởi một nhóm lồng léo gồm những người có suy nghĩ tự do, những người không chịu trách nhiệm về tiêu chuẩn HTML chính thức. Nó cho phép thực hành viết trang đã bị cấm một thập kỷ trước. Nó dành hàng nghìn từ một cách cẩn thận để nói với các nhà sản xuất trình duyệt cách xử lý các lỗi đánh dấu, thay vì từ chối chúng ngay lập tức. Cuối cùng, nó có thể phát lại video mà không cần trình cấm trình duyệt như Flash. Và nó giới thiệu một loạt các tính năng dựa trên JavaScript có thể cung cấp cho các trang web một số khả năng tương tác, phong phú của phần mềm máy tính để bàn truyền thống.

Hiểu HTML5 không phải là một kỳ công nhỏ. Một trở ngại là mọi người sử dụng từ này để chỉ một **HTML** hoặc phiên bản nhất định của **HTML5**. (Như bạn đã biết, một tiêu chuẩn là kết quả sau đó được chia thành nhiều phần dễ quản lý hơn.) Trên thực tế, HTML5 có nghĩa là “HTML5 và tất cả các tiêu chuẩn liên quan của nó” hoặc, thậm chí rộng hơn, “thế hệ tiếp theo của công nghệ viết trang web.” Đó là phiên bản của HTML5 mà bạn sẽ khám phá trong cuốn sách này: mọi thứ từ ngôn ngữ cốt lõi của HTML5 đến một vài tính năng mới đều được tích hợp trong HTML5 mặc dù chúng

không bao giờ một phần của tiêu chuẩn.

Thách thức thứ hai của HTML5 là hỗ trợ trình duyệt. Các trình duyệt khác nhau hỗ trợ HTML5 ở các mức độ khác nhau. Sự tụt hậu đáng chú ý nhất là Internet Explorer 8, hỗ trợ rất ít HTML5 và vẫn được tìm thấy trên một trong số 20 máy tính lướt web. (Ít nhất đó là vào thời điểm viết bài này. Trang 30 giải thích cách bạn có thể nhận thông kê sử dụng trình duyệt mới nhất.) May mắn thay, có những cách giải quyết có thể thu hẹp khoảng cách hỗ trợ trình duyệt—một số dễ dàng và một số xấu. Trong cuốn sách này, bạn sẽ tìm hiểu một chút về cả hai trong quá trình tìm cách sử dụng HTML5 trong các trang web của [mình hay](#).

Bắt chấp những thách thức mà HTML5 đưa ra, có một sự thật mà không ai phải bàn cãi—HTML5 là tương lai. Các công ty phần mềm khổng lồ như Apple, Google và Microsoft đã ủng hộ nó, và W3C (World Wide Web Consortium) đã từ bỏ công việc của mình trên XHTML để chính thức hóa và xác nhận nó. Với cuốn sách này, bạn cũng có thể tham gia bữa tiệc và sử dụng HTML5 để tạo các trang thú vị như trang được minh họa trong Hình I-1.



Những gì bạn cần để bắt đầu

Cuốn sách này đề cập đến HTML5, phiên bản mới nhất và lớn nhất của tiêu chuẩn HTML. Và mặc dù bạn không cần phải là chuyên gia đánh dấu để đọc nó, nhưng bạn [cần có một số kinh nghiệm](#) đó. Đây là tóm tắt chính thức:

- Viết trang web. Cuốn sách này giả định rằng bạn đã từng viết ít nhất một vài trang web trước đó (hoặc ít nhất, bạn hiểu cách sử dụng các phần tử HTML để cấu trúc nội dung thành các tiêu đề, đoạn văn và danh sách). Nếu bạn là người mới trong lĩnh vực thiết kế web, tốt hơn hết bạn nên có phần giới thiệu nhẹ nhàng hơn, giống như phần giới thiệu của tôi [một trang web](#): [Các Còn thiếu Thủ công](#), [Ngày thứ ba Phiên bản](#). (Nhưng đừng lo lắng; bạn sẽ không bị mắc kẹt quá khứ, như tất cả các ví dụ trong ân bản thứ ba của [Tạo một trang web](#) trong tài liệu HTML5.)
- Style sheet trải nghiệm. Không thể có trang web hiện đại nào nếu không có CSS-tiêu chuẩn Cascading Style Sheet-cung cấp bối cảnh và định dạng cho các trang web. Để theo dõi cuốn sách này, bạn nên biết những kiến thức cơ bản về biểu định kiểu: cách tạo chúng, nội dung bên trong và cách định kèm biểu định kiểu vào một trang. Nếu bạn hơi mơ hồ về chủ đề này, bạn có thể tìm hiểu thêm trong Phụ lục A, "CSS cơ bản". Nhưng nếu bạn cần thêm trợ giúp hoặc nếu bạn chỉ muốn nâng cao kỹ năng CSS của mình để tạo bối cảnh và phong cách thực sự thú vị, hãy xem một cuốn sách bổ sung [HTML5: Các Còn thiếu Thủ công](#) của David Sawyer McFarland.
- Kinh nghiệm JavaScript. Không, bạn không cần JavaScript để tạo trang HTML5. Tuy nhiên, bạn cần có JavaScript nếu muốn sử dụng nhiều tính năng mạnh mẽ nhất của HTML5, chẳng hạn như vẽ trên canvas hoặc giao tiếp với máy chủ web. Nếu bạn có ít kinh nghiệm lập trình nhưng không biết nhiều về JavaScript, thì Phụ lục B, "JavaScript: Bộ não của trang của bạn" có thể giúp bạn bắt kịp tốc độ. Nhưng ý tưởng viết mã nghe có vẻ thoái mái như leo lên giường với một con trăn đã trốn thoát, thì cuối cùng bạn sẽ bỏ qua rất nhiều tài liệu trong cuốn sách này, hoặc bạn sẽ cần lắp đầy khoảng trống bằng một cuốn sách như [JavaScript & jQuery: Các Còn thiếu Thủ công](#) của David Sawyer McFarland.

Viết HTML5 Bạn có thể

viết các trang HTML5 bằng chính phần mềm mà bạn sử dụng để viết các trang HTML. Điều đó có thể đơn giản như một trình soạn thảo văn bản cấp thấp, như Notepad (trên Windows) hoặcTextEdit (trên Mac). Nhiều công cụ thiết kế hiện tại, như Adobe Dreamweaver và Microsoft Visual Studio, có các mẫu cho phép bạn nhanh chóng tạo các tài liệu HTML5 mới. Tuy nhiên, cấu trúc cơ bản của trang HTML5 đơn giản đến mức bạn có thể sử dụng bất kỳ trình chỉnh sửa web nào để tạo một trang, ngay cả khi nó không được thiết kế riêng cho HTML5.

GHI CHÚ

Và, tất nhiên, việc bạn lướt web và tạo trang web trên Windows không thành vấn đề. PC hoặc MacBook Pro mới nhất-HTML5 không quan tâm đến hệ điều hành bạn sử dụng.

Xem HTML5 Bạn sẽ nhận

được hỗ trợ cho hầu hết các tính năng HTML5 trong phiên bản mới nhất của bất kỳ trình duyệt hiện đại nào, kể cả các trình duyệt dành cho thiết bị di động chạy trên các thiết bị Apple và Android. Miễn là trình duyệt của bạn được cập nhật, HTML5 sẽ hoạt động tốt-và bạn sẽ có thể thử các ví dụ trong cuốn sách này.

KHI NÀO HTML5 SẴN SÀNG?

Hiện tại, không có trình duyệt nào hỗ trợ chi tiết cuối cùng của HTML5, một phần vì HTML5 thực sự là một tập hợp các tiêu chuẩn có liên quan với nhau. Google Chrome thường dẫn đầu cuộc đua trình duyệt về hỗ trợ HTML5, với Firefox và Opera đang theo sát. Safari chậm hơn một chút và Internet Explorer vẫn còn ở phía sau. Vấn đề thực sự nằm ở chỗ các bản Internet Explorer cũ không thể cập nhật được vì chúng đang chạy trên các hệ điều hành ợp ẹp như Windows Vista hoặc Windows XP (hiện vẫn đang chạy trên 1/5 số máy tính để bàn trên thế giới). Trang 26 có một cái nhìn sâu hơn về vấn đề này và một số lời khuyên về cách giải quyết nó.

Khi nào HTML5 sẽ sẵn sàng?

Câu trả lời ngắn gọn là “bây giờ”. Ngay cả Internet Explorer 6 bị coi thường, đã 10 năm tuổi và chưa đầy những tính năng phá vỡ trang web, cũng có thể hiển thị các tài liệu HTML5 cơ bản. Đó là bởi vì tiêu chuẩn HTML5 đã được tạo ra một cách có chủ ý theo cách bao hàm và mở rộng HTML truyền thống.

Câu trả lời chi tiết hơn là “nó phụ thuộc.” Như bạn đã biết, HTML5 là tập hợp các tiêu chuẩn khác nhau với mức độ hỗ trợ trình duyệt khác nhau. Vì vậy, mặc dù ngày nay mọi nhà phát triển web đều có thể chuyển sang tài liệu HTML5 (và nhiều trang web lớn, như Google, YouTube và Wikipedia, đã có), nhưng có thể mất một thời gian trước khi sử dụng tất cả các tính năng mới ưa thích của HTML5 một cách an toàn nhất là mà không cần thêm một số loại cơ chế dự phòng cho các trình duyệt kém thông minh hơn.

GHI CHÚ

Trước khi khuyễn khích bạn sử dụng một tính năng HTML5 mới, cuốn sách này chỉ rõ mức độ hỗ trợ trình duyệt hiện tại của tính năng đó. Tuy nhiên, các phiên bản trình duyệt thay đổi tương đối nhanh, vì vậy bạn sẽ muốn thực hiện nghiên cứu cập nhật của riêng mình trước khi sử dụng bất kỳ tính năng nào có thể gây ra sự cố. Trang web <http://caniuse.com> cung cấp các tính năng cụ thể và cho bạn biết chính xác phiên bản trình duyệt nào hỗ trợ chúng. (Bạn sẽ tìm hiểu thêm về công cụ hữu ích này ở trang 27.)

Là một nhà phát triển quan tâm đến tiêu chuẩn, bạn cũng có thể quan tâm đến việc biết các tiêu chuẩn khác nhau đã đi được bao xa trong hành trình hướng tới trạng thái chính thức. Điều này phức tạp bởi thực tế là những người mơ ước về HTML5 có một triết lý phi mỉa mỉa hơi lật đổ và họ thường chỉ ra rằng những gì trình duyệt hỗ trợ quan trọng hơn những gì tiêu chuẩn chính thức nói. Nói cách khác, bạn có thể tiếp tục và sử dụng mọi thứ bạn muốn ngay bây giờ, nếu bạn có thể làm cho nó hoạt động. Tuy nhiên, các nhà phát triển web, các công ty lớn, chính phủ và các tổ chức khác thường đưa ra tín hiệu về việc liệu một ngôn ngữ đã sẵn sàng để sử dụng hay chưa bằng cách xem trạng thái tiêu chuẩn của ngôn ngữ đó.

Tại thời điểm viết bài này, ngôn ngữ HTML5 đang ở giai đoạn [khởi đầu phát triển](#), nghĩa là tiêu chuẩn này phần lớn đã được giải quyết nhưng các nhà sản xuất trình duyệt vẫn đang hoàn thiện các triển khai HTML5 của họ. Giai đoạn tiếp theo và cũng là giai đoạn cuối cùng để tiêu chuẩn này trở thành một tiêu chuẩn [đã định hình](#) và HTML5, dù hiện chưa đạt, sẽ có mặt trên phần mềm năm 2010. Trong giai đoạn chuẩn, gọi là HTML 5.1. (Để được trợ giúp thêm về ý nghĩa của tất cả các phiên bản khác nhau nhôp ở trang tiếp theo.)

CÂU HỎI THƯỜNG GẶP

Sự khác biệt giữa HTML5 và HTML 5.1

có một phiên bản khác của And what's with the Is HTML? Mới khoảng cách không nhất quán?

Như bạn sẽ tìm hiểu trong Chương 1, HTML5 đã trải qua hai giai đoạn. Quá trình này đã để lại một số điều kỳ quặc phía sau, bao gồm cả một hệ thống phiên bản hơi phân liệt.

Những người ban đầu đã tạo ra HTML5-các thành viên của WHATWG mà bạn sẽ gặp ở trang 5-không quan tâm nhiều đến số phiên bản. Họ coi HTML5 là một ngôn ngữ sống. Họ khuyên khích các nhà phát triển web chú ý đến hỗ trợ trình duyệt, thay vì lo lắng về số phiên bản chính xác.

Tuy nhiên, WHATWG đã chuyển HTML5 cho những người giữ tiêu chuẩn web chính thức-W3C-de họ có thể hoàn thiện nó. W3C là một tổ chức cần thận hơn, có phương pháp hơn. Những người ở đó

muốn có một cách để tách ẩn phẩm đầu tiên của họ về tiêu chuẩn HTML5 khỏi những ẩn phẩm kế thừa đã được tinh chỉnh và làm sạch một chút mà chắc chắn sẽ tuân theo. Do đó, W3C đã quyết định đặt tên cho bản phát hành đầu tiên của tiêu chuẩn HTML5 là HTML 5.0 (lưu ý khoảng trắng). Bản phát hành thứ hai sẽ là HTML 5.1, tiếp theo là bản phát hành thứ ba có tên là HTML 5.2. Thật khó hiểu, tất cả các phiên bản này vẫn được coi là HTML5.

Ngẫu nhiên, các lần lặp lại sau của tiêu chuẩn HTML5 không có khả năng bổ sung các thay đổi lớn. Thay vào đó, các tính năng mới sẽ xuất hiện trong các thông số kỹ thuật bổ sung, riêng biệt. Bằng cách này, các nhóm nhỏ người có thể nhanh chóng phát triển các tính năng HTML5 mới, hữu ích mà không cần phải đợi bản sửa đổi ngôn ngữ hoàn toàn mới.

Về đề cương

Cuốn sách này bao gồm một hướng dẫn toàn diện về HTML5 thành 13 chương. Đây là những gì bạn sẽ tìm thấy:

Phần thứ nhất: Làm quen với ngôn ngữ mới

- Chương 1 giải thích cách HTML biến thành HTML5. Bạn sẽ gặp tài liệu HTML5 đầu tiên của mình, xem ngôn ngữ đã thay đổi như thế nào và xem hỗ trợ trình duyệt.
- Chương 2 giải quyết vấn đề của HTML5 ~~không có nghĩa~~ phần tử có thể đưa ý nghĩa vào phần đánh dấu của bạn. Nếu được sử dụng đúng cách, thông tin bổ sung này có thể giúp trình duyệt, trình đọc màn hình, công cụ thiết kế web và công cụ tìm kiếm hoạt động minh hơn.
- Chương 3 đi sâu hơn vào thế giới ngữ nghĩa với các tiêu chuẩn bổ trợ như . Và mặc dù nó có vẻ hơi ~~vì dữ liệu~~ lý thuyết, nhưng có một phần thường béo bở dành cho các nhà phát triển web hiểu nó nhất: danh sách tốt hơn, chi tiết hơn trong các công cụ tìm kiếm như Google.
- Chương 4 khám phá những thay đổi của HTML5 đối với các thành phần biểu mẫu web-hộp văn bản, danh sách, hộp kiểm và các tiện ích con khác mà bạn sử dụng để thu thập thông tin từ khách truy cập của mình. HTML5 thêm một vài kiểu cách và một số công cụ cơ bản để bắt lỗi nhập dữ liệu.

Phần thứ hai: Video, Đồ họa và Glitz

- Chương 5 đề cập đến một trong những tính năng thú vị nhất của HTML5: hỗ trợ phát lại âm thanh và video. Bạn sẽ học cách tồn tại trong Cuộc chiến giải mã video trên web để tạo các trang phát lại hoạt động trong mọi trình duyệt và thậm chí bạn sẽ thấy cách tạo trình phát tùy chỉnh của riêng mình.
- Chương 6 giới thiệu phiên bản mới nhất của tiêu chuẩn CSS3, bổ sung hoàn hảo cho HTML5. Bạn sẽ học cách làm nổi bật văn bản của mình bằng các phông chữ ưa thích và thêm các hiệu ứng bắt mắt với các hiệu ứng chuyển tiếp và hoạt ảnh.
- Chương 7 khám phá các truy vấn phương tiện CSS3. Bạn sẽ học cách sử dụng chúng để tạo các thiết kế đáp ứng-bổ cục trang web tự điều chỉnh liền mạch với các thiết bị di động khác nhau.
- Chương 8 giới thiệu bề mặt vẽ hai chiều được gọi là Tranh sơn dầu.
Bạn sẽ học cách vẽ nó bằng các hình dạng, ảnh và văn bản, thậm chí là xây dựng một chương trình vẽ cơ bản (với một lượng mã JavaScript vừa phải).
- Chương 9 nâng cao kỹ năng canvas của bạn. Bạn sẽ tìm hiểu về bóng đèn và các mẫu ưa thích, cùng với các kỹ thuật canvas đầy tham vọng hơn như hình dạng có thể nhấp, tương tác và hoạt ảnh.

Phần Ba: Xây dựng Ứng dụng Web • Chương

10 bao gồm tính năng lưu trữ web cho phép bạn lưu trữ các mẩu thông tin nhỏ trên máy tính của khách truy cập. Bạn cũng sẽ tìm hiểu về các cách xử lý tệp do người dùng chọn trong mã JavaScript trang web của bạn, thay vì trên máy chủ web.

- Chương 11 khám phá tính năng bộ nhớ đệm HTML5 có thể cho phép trình duyệt lưu chạy một trang web, ngay cả khi nó mất kết nối web.
- Chương 12 đi sâu vào thế giới đầy thách thức của giao tiếp máy chủ web. Bạn sẽ bắt đầu với đối tượng XMLHttpRequest lâu đời, đối tượng này cho phép mã JavaScript của bạn liên hệ với máy chủ web và yêu cầu thông tin. Sau đó, bạn sẽ chuyển sang hai tính năng mới hơn: sự kiện phía máy chủ và ô cảm biến web tham vọng hơn.
- Chương 13 đề cập đến ba tính năng khác giải quyết các thách thức trong các ứng dụng web hiện đại. Trước tiên, bạn sẽ thấy cách định vị địa lý có thể xác định vị trí của khách truy cập. Tiếp theo, bạn sẽ sử dụng nhân viên web để chạy các tác vụ tốn nhiều thời gian ở chế độ nền. Cuối cùng, bạn sẽ tìm hiểu về tính năng lịch sử trình duyệt, cho phép bạn đồng bộ hóa URL của trang web với trạng thái hiện tại của trang.

Ngoài ra còn có hai phụ lục có thể giúp bạn bắt kịp các nguyên tắc cơ bản cần thiết để thành thạo HTML5. Phụ lục A, "CSS cơ bản," đưa ra bản tóm tắt rút gọn về CSS; Phụ lục B, "JavaScript: Bộ não của trang của bạn" cung cấp một cái nhìn tổng quan ngắn gọn về JavaScript.

Giới thiệu về Tài nguyên Trực tuyến

Là chủ sở hữu của Sổ tay bị mất tích, bạn không chỉ có một cuốn sách để đọc. Trực tuyến, bạn sẽ tìm thấy các tệp ví dụ cũng như mèo, bài báo và thậm chí có thể là một hoặc hai video.

Bạn cũng có thể liên lạc với nhóm Missing Manual và cho chúng tôi biết điều bạn yêu thích (hoặc ghét) về cuốn sách. Đi qua hoặc đi trực tiếp www.missingmanuals.com,

Đĩa CD bị thiếu Cuốn

sách này không có đĩa CD dán bên trong bìa sau, nhưng bạn không bỏ sót bất cứ thứ gì. Truy cập để tải xuống các ví dụ trang web được tạo ra ngay chính bây giờ trong cuốn sách này. <http://www.missingmanuals.com/cds/html5.html> Khi gõ các địa chỉ web dài, trang Missing CD cung cấp danh sách các liên kết có thể nhấp được tới các trang web được đề cập trong mỗi chương.

TIỀN BOA

Nếu bạn đang tìm kiếm một ví dụ cụ thể, đây là một cách nhanh chóng để tìm thấy nó: Hãy xem hình tương ứng trong cuốn sách này. Tên tệp thường hiển thị ở cuối văn bản trong hộp đĩa chỉ của trình duyệt web. Ví dụ: nếu bạn thấy đường dẫn [Tệp HTML SuperSimple.html](#) hiển thị tệp ví dụ tương ứng là [SuperSimpleHTML5.html](#).

Trang web dùng thử

Có một cách khác để sử dụng các ví dụ: trên trang web ví dụ tại www.prosetech.com/html5. Ở đó, bạn sẽ tìm thấy các phiên bản trực tiếp của mọi ví dụ từ cuốn sách này, bạn có thể chạy các phiên bản này trong trình duyệt của mình. Sự tiện lợi này có thể giúp bạn bắt đầu đầu, bởi vì HTML5 bao gồm một số tính năng yêu cầu sự tham gia của một máy chủ web thực sự. (Nếu bạn đang chạy các trang web từ ổ cứng trên máy tính cá nhân của mình, thì các tính năng này có thể phát triển những điểm kỳ quặc bí ẩn hoặc ngừng hoạt động hoàn toàn.) Bằng cách sử dụng trang web trực tiếp, bạn có thể xem một ví dụ được cho là hoạt động như thế nào trước khi bạn tải trang xuống và bắt đầu thử nghiệm của riêng bạn.

GHI CHÚ

Đừng lo-khi bạn bắt gặp một tính năng HTML5 cần lưu trữ máy chủ web, cuốn sách này sẽ cảnh báo bạn.

Đăng ký Nếu bạn

đăng ký cuốn sách này tại www.oreilly.com (ưu www.oreilly.com), bạn sẽ đủ điều kiện nhận .
đãi-chẳng hạn như giảm giá cho các lần xuất bản HTML5 còn [Hỗ trợ đăng ký: Đăng](#) Đăng ký vào trong tương lai chỉ mất vài cú nhấp [ichuột](#). [Nhậpregisterbook](#) trình duyệt của bạn để hop trực tiếp vào trang Đăng ký.

Nhận xét

Có câu hỏi nào không? Cần thêm thông tin? Tưởng tượng mình là một nhà phê bình sách? Trên trang Phản hồi của chúng tôi, bạn có thể nhận được câu trả lời của chuyên gia cho các câu hỏi này ra trong khi đọc, chia sẻ suy nghĩ của bạn về Sở tay hướng dẫn còn thiếu này và tìm các nhóm người có chung sở thích với bạn trong việc tạo trang web của riêng họ.

Để có tiếng nói của bạn, đi đến www.missingmanuals.com/feedback.

sai sót

Để giữ cho cuốn sách này được cập nhật và chính xác nhất có thể, mỗi lần chúng tôi in thêm bản sao, chúng tôi sẽ thực hiện bất kỳ chỉnh sửa nào được xác nhận mà bạn đề xuất. Chúng tôi cũng lưu ý những thay đổi như vậy trên trang web của cuốn sách, vì vậy bạn có thể đánh dấu những phần sửa chữa quan trọng vào bản sao cuốn sách của mình, nếu muốn. Truy cập để báo cáo lỗi và xem các chính sách tại <http://tinyurl.com/html52e-mm>.

Safari® Sách trực tuyến

Safari® Books Online là một thư viện kỹ thuật số theo yêu cầu cho phép bạn tìm kiếm hơn 7.500 cuốn sách và video về công nghệ.

Với đăng ký, bạn có thể đọc bất kỳ trang nào và xem bất kỳ video nào từ thư viện của chúng tôi.

Truy cập các tiêu đề mới trước khi chúng có sẵn trong bản in. Sao chép và dán các mẫu mã, sắp xếp các mục yêu thích của bạn, tải xuống các chương, đánh dấu các phần quan trọng, tạo ghi chú, in các trang và hưởng lợi từ rất nhiều tính năng tiết kiệm thời gian khác.

O'Reilly Media đã tải cuốn sách này lên dịch vụ Safari Books Online. Để có toàn quyền truy cập kỹ thuật số vào cuốn sách này và những cuốn khác về các chủ đề tương tự từ O'Reilly và các nhà xuất bản khác, hãy đăng ký miễn phí tại <http://my.safaribooksonline.com>.

PHẦN

đánh dấu hiện đại

1

CHƯƠNG 1:

Giới thiệu HTML5

CHƯƠNG 2:

Cấu trúc các trang với các yếu tố ngữ nghĩa

CHƯƠNG 3:

Viết đánh dấu có ý nghĩa hơn

CHƯƠNG 4:

Xây dựng biểu mẫu web tốt hơn

Giới thiệu HTML5

Nếu HTML là một bộ phim, thì HTML5 sẽ là bước ngoặt bất ngờ của nó. HTML không có nghĩa là tồn tại trong thế kỷ 21. Tổ chức tiêu chuẩn web chính thức, W3C (viết tắt của World Wide Web Consortium), đã rời bỏ HTML vào năm 1998. W3C đã gìn giữ các kế hoạch tương lai của mình vào một đặc điểm kỹ thuật có tên là XHTML, mà nó dự định là người kế thừa được hiện đại hóa và làm sạch của HTML. Nhưng XHTML vấp ngã, và một nhóm phiến quân bị tước quyền đã hồi sinh HTML, đặt nền tảng cho các tính năng mà bạn sẽ khám phá trong cuốn sách này.

Trong chương này, bạn sẽ nắm được thông tin sót dép về lý do tại sao HTML chết và cách nó hoạt động trở lại. Bạn sẽ tìm hiểu về triết lý và các tính năng của HTML5 và bạn sẽ xem xét vấn đề nhức nhối về hỗ trợ trình duyệt. Bạn cũng sẽ có cái nhìn đầu tiên về tài liệu HTML5 đích thực.

Câu chuyện về HTML5

Ý tưởng cơ bản đằng sau HTML—mà bạn sử dụng để cấu ~~kiết~~^{tố} nội dung của mình—đã không thay đổi kể từ những ngày đầu tiên của Web. Trên thực tế, ngay cả những trang web cũ nhất vẫn hoạt động hoàn hảo trong các trình duyệt web hiện đại nhất.

Già và thành công cũng mang theo một số rủi ro — cụ thể là mọi người đều muốn thay thế bạn. Vào năm 1998, W3C ngừng hoạt động trên HTML và cố gắng cài tiến nó với phiên bản kế nhiệm dựa trên XML có tên là XHTML 1.0.

XHTML 1.0: Bất nghiêm ngặt XHTML có hầu hết các quy ước cú pháp giống như HTML, nhưng nó thực thi các quy tắc chặt chẽ hơn. Phần lớn đánh dấu câu thả mà HTML truyền thống cho phép lại không thể chấp nhận được trong XHTML.

Ví dụ: giả sử bạn muốn in nghiêng từ cuối cùng trong tiêu đề, như sau:

```
<h1>Cuộc đời của <i>Vịt</i></h1>
```

Và bạn vô tình tráo đổi hai thẻ cuối cùng:

```
<h1>Cuộc đời của <i>Vịt</h1></i>
```

Khi một trình duyệt gặp phần đánh dấu hơi lộn xộn này, nó có thể tìm ra những gì bạn thực sự muốn. Nó in nghiêng từ cuối cùng mà không hề có một lời phàn nàn lịch sự nào. Tuy nhiên, các thẻ không khớp sẽ vi phạm các quy tắc chính thức của XHTML. Nếu bạn cắm trang của mình vào trình xác thực XHTML (hoặc sử dụng công cụ thiết kế web như Dreamweaver), bạn sẽ nhận được cảnh báo chỉ ra lỗi của mình. Từ quan điểm thiết kế web, tính chặt chẽ của XHTML hữu ích ở chỗ nó cho phép bạn phát hiện những lỗi nhỏ có thể gây ra kết quả không nhất quán trên các trình duyệt khác nhau (hoặc có thể gây ra sự cố lớn hơn khi bạn chỉnh sửa và nâng cao trang).

Lúc đầu, XHTML là một câu chuyện thành công. Các nhà phát triển web chuyên nghiệp, thắt vọng với những điều kỳ quặc của trình duyệt và tình trạng thiết kế web bất di bất dịch, đã dỗ xô đến XHTML. Trên đường đi, họ buộc phải áp dụng những thói quen tốt hơn và từ bỏ một số tính năng định dạng nửa vời của HTML. Tuy nhiên, nhiều lợi ích tưởng tượng của XHTML—như khả năng tương tác với các công cụ XML, xử lý trang dễ dàng hơn cho các chương trình tự động, khả năng di chuyển sang các nền tảng di động và khả năng mở rộng của chính ngôn ngữ XHTML—đã không bao giờ thành hiện thực.

Tuy nhiên, XHTML đã trở thành tiêu chuẩn cho hầu hết các nhà thiết kế web nghiêm túc. Và trong khi mọi người đều có vẻ khá hài lòng, thì có một bí mật bẩn thỉu: Mặc dù các trình duyệt hiểu đánh dấu XHTML, nhưng chúng không thực thi kiểm tra lỗi nghiêm ngặt mà tiêu chuẩn yêu cầu. Điều đó có nghĩa là một trang có thể vi phạm các quy tắc của XHTML và trình duyệt sẽ không nhấp nháy hai lần. Trên thực tế, không có gì ngăn cản một nhà phát triển web tập hợp một mớ hỗn độn các đánh dấu câu thả và nội dung HTML lỗi thời rồi gọi nó là một trang XHTML. Không có một trình duyệt nào trên hành tinh này sẽ phản nản.

Và khiến những người chịu trách nhiệm về tiêu chuẩn XHTML vô cùng khó chịu.

XHTML 2: Thất bại ngoài dự kiến XHTML 2 được cho là cung cấp giải pháp cho sự cầu thả này. Nó được thiết lập để thắt chặt các quy tắc xử lý lỗi, buộc các trình duyệt từ chối các trang XHTML 2 không hợp lệ.

XHTML 2 cũng loại bỏ nhiều điều kỳ quặc và quy ước có nguồn gốc từ HTML. Ví dụ: hệ thống tiêu đề được đánh số (`<h1>`, `<h2>`, `<h3>`, v.v.) được thay thế bằng một phần tử `<h>` mới, có ý nghĩa phụ thuộc vào vị trí của nó trong trang web. Tương tự, phần tử `<a>` bị lu mờ bởi một tính năng cho phép các nhà phát triển web chuyển đổi bất kỳ phần tử nào thành liên kết và phần tử `` mất thuộc tính `alt` để thay thế bằng một cách mới để cung cấp nội dung thay thế.

Những thay đổi này là điển hình của XHTML 2. Về lý thuyết, chúng được tạo ra để đánh dấu logic hơn, sạch hơn. Trên thực tế, những thay đổi buộc các nhà thiết kế web phải thay đổi cách họ viết trang web (không nói gì đến việc cập nhật các trang web họ đã có) và không thêm tính năng mới nào để làm cho tất cả những thứ đó hoạt động có giá trị. XHTML 2 thậm chí còn loại bỏ một số yếu tố lỗi thời mà một số nhà thiết kế web vẫn yêu thích, như **** cho văn bản in đậm, **<i>** cho chữ nghiêng và **<iframe>** để nhúng một trang web vào trong một trang web khác.

Nhưng có lẽ vấn đề tồi tệ nhất là tốc độ thay đổi chóng mặt. Quá trình phát triển trên XHTML 2 kéo dài trong 5 năm và sự nhiệt tình của nhà phát triển dần mất đi.

HTML5: Trở về từ cõi chết

Gần như cùng lúc đó—bắt đầu từ năm 2004—một nhóm người bắt đầu nhìn vào tương lai của Web từ một góc độ khác. Thay vì cố gắng tìm ra điều gì sai (hoặc chỉ là “không trong sạch về mặt triết học”) trong HTML, họ tập trung vào những gì còn thiếu, xét về những điều mà các nhà phát triển web muốn hoàn thành.

Rốt cuộc, HTML đã bắt đầu cuộc sống của nó như một công cụ để hiển thị tài liệu. Với việc bổ sung JavaScript, nó đã biến thành một hệ thống để phát triển các ứng dụng web, như công cụ tìm kiếm, cửa hàng thương mại điện tử, công cụ lập bản đồ, ứng dụng email, v.v. Và trong khi một ứng dụng web xảo quyệt có thể làm được nhiều điều ẩn tượng, thì không dễ để tạo ra một ứng dụng như vậy. Hầu hết các ứng dụng web đều dựa trên hỗn hợp JavaScript viết tay, một hoặc nhiều bộ công cụ JavaScript phổ biến và một mớ-jun mã chạy trên máy chủ web.

Đó là một thách thức để làm cho tất cả các phần này tương tác nhất quán trên các trình duyệt khác nhau. Ngay cả khi bạn làm cho nó hoạt động, bạn vẫn cần lưu ý đến băng keo và kim ghim để giữ mọi thứ lại với nhau.

Những người tạo trình duyệt đặc biệt quan tâm đến tình trạng này. Vì vậy, một nhóm các cá nhân có tự duy tiên bô từ Opera Software (người tạo ra trình duyệt Opera) và Mozilla Foundation (người tạo ra Firefox) đã vận động hành lang để có được XHTML nhằm giới thiệu nhiều tính năng hướng đến nhà phát triển hơn. Khi họ thất bại, Opera, Mozilla và Apple đã thành lập WHATWG (Nhóm làm việc về công nghệ ứng dụng siêu văn bản trên web) để nghĩ ra các giải pháp mới.

WHATWG không phải để thay thế HTML mà là để thay thế nó theo một cách liền mạch, tương thích ngược. Phiên bản đầu tiên của công việc này có hai thông số kỹ thuật bổ sung được gọi là Ứng dụng web 1.0 và Biểu mẫu web 2.0. Cuối cùng, các tiêu chuẩn này đã phát triển thành HTML5.

GHI CHÚ

Con số 5 trong tên đặc tả HTML5 được cho là chỉ ra rằng tiêu chuẩn tiếp tục ở nơi HTML còn lại (đó là phiên bản HTML 4.01, có trước XHTML). Tuy nhiên, điều này không thực sự chính xác, bởi vì HTML5 hỗ trợ mọi thứ đã xảy ra với các trang web trong thập kỷ kể từ khi HTML 4.01 được phát hành, bao gồm cả phap kiều XHTML nghiêm ngặt (nếu bạn chọn sử dụng nó) và một loạt cải tiến JavaScript. Tuy nhiên, cái tên vẫn nói lên một điểm rõ ràng: HTML5 có thể hỗ trợ **truy cập** của XHTML, nhưng nó thực thi các **quy tắc** của HTML.

Đến năm 2007, trại WHATWG đã thu hút sự chú ý của các nhà phát triển web ở khắp mọi nơi. Sau một số suy nghĩ đau đớn, W3C đã quyết định giải tán nhóm đang làm việc trên XHTML 2 và thay vào đó làm việc để chính thức hóa tiêu chuẩn HTML5. Tại

tại thời điểm này, HTML5 ban đầu đã được chia thành nhiều phần dễ quản lý hơn và nhiều tính năng ban đầu được gọi là HTML5 đã trở thành các tiêu chuẩn riêng biệt (để biết thêm thông tin, hãy xem hộp trên trang này).

TIỀN BOA

Bạn có thể đọc phiên bản W3C chính thức của tiêu chuẩn HTML5 tại

www.w3.org/TR/html5.

LÊN ĐẾN TỐC ĐỘ

HTML5 bao gồm những gì?

HTML5 thực sự là một trang web gồm các tiêu chuẩn có liên quan với nhau. Cách tiếp cận này là cá tát và xấu. Thật tốt vì các trình duyệt có thể nhanh chóng triển khai các tính năng trưởng thành trong khi các trình duyệt khác tiếp tục phát triển. Thật tệ vì nó buộc người viết trang web phải lo lắng về việc kiểm tra xem trình duyệt có hỗ trợ từng tính năng mà họ muốn sử dụng hay không. Bạn sẽ học được một số kỹ thuật đơn giản và không quá đau đớn để làm như vậy trong cuốn sách này.

Dưới đây là các danh mục tính năng chính thuộc phạm vi của HTML5:

- Lỗi HTML5. Phần này của HTML5 tạo nên phiên bản W3C chính thức của thông số kỹ thuật. Nó bao gồm các yếu tố ngữ nghĩa mới (Chương 2 và Chương 3), tiện ích biểu mẫu web mới và nâng cao (Chương 4), hỗ trợ âm thanh và video (Chương 5) và canvas để vẽ bằng JavaScript (Chương 8 và Chương 9).
- Các tính năng đã từng là HTML5. Các tính năng này xuất phát từ đặc tả HTML5 ban đầu được chuẩn bị bởi

CÁI GÌ. Hầu hết trong số này là thông số kỹ thuật cho các tính năng yêu cầu JavaScript và hỗ trợ các ứng dụng web phong phú. Điều quan trọng nhất bao gồm lưu trữ dữ liệu cục bộ (Chương 10), ứng dụng ngoại tuyến (Chương 11) và nhẫn tin (Chương 12), nhưng bạn sẽ tìm hiểu thêm về một số điều khác trong cuốn sách này.

- Các tính năng đôi khi được gọi là HTML5. Đó là các tính năng thế hệ tiếp theo thường được gộp chung với HTML5, mặc dù chúng chưa từng là một phần của tiêu chuẩn HTML5. Danh mục này bao gồm CSS3 (Chương 6 và Chương 7) và định vị địa lý (Chương 13).

Ngay cả W3C cũng đang làm mờ ranh giới giữa "thực" HTML5 (những gì thực sự có trong tiêu chuẩn) và phiên bản "tiếp thi" (bao gồm mọi thứ là một phần của HTML5 và nhiều thông số kỹ thuật bổ sung). Ví dụ: trang web logo chính thức của W3C () khuyến khích bạn tạo các logo HTML5 [quảng bá CSS3](#) / [Vai trò của ngôn ngữ](#) / [giản hóa](#) / [tạo trước](#) khi HTML5 xuất hiện.

HTML: Ngôn ngữ sống Việc chuyển đổi từ W3C

sang WHATWG và quay trở lại W3C một lần nữa đã dẫn đến một sự sắp xếp khá bất thường. Về mặt kỹ thuật, W3C chịu trách nhiệm xác định đâu là và đâu không phải là HTML5 chính thức. Nhưng đồng thời, WHATWG tiếp tục công việc mở ước về các tính năng HTML trong tương lai. Chỉ bây giờ, họ không còn coi công việc của họ là HTML5 nữa. Họ chỉ đơn giản gọi nó là HTML, giải thích rằng HTML sẽ tiếp tục như một

cuộc sống

ngôn ngữ .

Vì HTML là một ngôn ngữ sống nên một trang HTML sẽ không bao giờ trở nên lỗi thời và ngừng hoạt động. Các trang HTML sẽ không bao giờ sử dụng số phiên bản (ngay cả trong loại tài liệu) và các nhà phát triển web sẽ không bao giờ cần "nâng cấp" đánh dấu của họ từ phiên bản này sang phiên bản khác để làm cho nó hoạt động trên các trình duyệt mới. Đồng thời, các tính năng mới có thể được thêm vào HTML bất kỳ lúc nào.

Khi các nhà phát triển web nghe về kế hoạch này, phản ứng đầu tiên của họ thường là kinh hoàng. Rốt cuộc, ai muốn đối phó với một thế giới hỗ trợ tiêu chuẩn thay đổi dữ dội, nơi các nhà phát triển cần chọn và chọn các tính năng họ sử dụng dựa trên khả năng các tính năng này sẽ được hỗ trợ? Tuy nhiên, khi suy ngẫm lại, hầu hết các nhà phát triển web đều nhận ra một cách miễn cưỡng: Dù tốt hay xấu, đây chính xác là cách các trình duyệt đã hoạt động kể từ buổi bình minh của Web.

Như đã giải thích trước đó, các trình duyệt ngày nay hài lòng với mọi tính năng được hỗ trợ. Bạn có thể lấy một trang XHTML hiện đại và thêm một thứ gì đó gây tai tiếng về phía sau như phần tử `<marquee>` (một tính năng lỗi thời để tạo văn bản cuộn) và sẽ không có trình duyệt nào phản ứng. Tương tự như vậy, các trình duyệt có những lỗi hỏng nội tiêng trong việc hỗ trợ ngay cả những tiêu chuẩn lâu đời nhất. Ví dụ: các nhà sản xuất trình duyệt bắt đầu triển khai CSS3 trước khi hỗ trợ CSS2 kết thúc và nhiều tính năng CSS2 sau đó đã bị loại bỏ. Sự khác biệt duy nhất là giờ đây HTML5 đã biến trạng thái “ngôn ngữ sống” trở thành chính thức. Tuy nhiên, không có gì là trớ trêu khi HTML đang bắt tay vào một chương mới, trong chương mới, cuối cùng nó đã quay trở lại vòng tròn đầy đủ về cội nguồn của nó.

TIỀN BOA

Để xem bản nháp HTML hiện tại, đang phát triển bao gồm nội dung được gọi là HTML5 và một tập hợp các tính năng mới, không được hỗ trợ đang phát triển, hãy truy cập <http://whatwg.org/html>.

Ba nguyên tắc chính của HTML5

Đến thời điểm này, có lẽ bạn đang mong muốn bắt đầu với một trang HTML5 thực sự. Nhưng trước tiên, nó đáng để đi sâu vào tâm trí của những người đã xây dựng HTML5. Một khi bạn hiểu được triết lý đằng sau ngôn ngữ đó, thì những điều kỳ quặc, phức tạp và thỉnh thoảng khiến bạn đau đầu mà bạn gặp phải trong cuốn sách này sẽ trở nên có ý nghĩa hơn rất nhiều.

1. Đừng Phá Web

“Đừng phá vỡ Web” có nghĩa là một tiêu chuẩn không được đưa ra những thay đổi khiến trang web của người khác ngừng hoạt động. May mắn thay, loại đồng đồ nát này hiếm khi xảy ra.

“Đừng phá vỡ trang Web” có nghĩa là một tiêu chuẩn không nên tùy tiện thay đổi các quy tắc và trong quá trình đó, các trang web hiện tại hoàn toàn tốt sẽ trở nên lỗi thời (ngay cả khi chúng vẫn hoạt động). Ví dụ, XHTML 2 đã phá vỡ trang web bởi vì nó đòi hỏi một sự thay đổi mạnh mẽ, ngay lập tức trong cách viết các trang web. Có, các trang cũ sẽ vẫn hoạt động—nhờ khả năng tương thích ngược được tích hợp trong trình duyệt. Nhưng nếu bạn muốn chuẩn bị cho tương lai và giữ cho trang web của mình được cập nhật, bạn buộc phải lãng phí vô số thời gian để sửa những “lỗi” mà XHTML 2 đã cấm.

HTML5 có một quan điểm khác. Mọi thứ hợp lệ trước HTML5 vẫn hợp lệ trong HTML5. Trên thực tế, mọi thứ hợp lệ trong HTML 4.01 cũng vẫn hợp lệ trong HTML5.

GHI CHÚ

Không giống như các tiêu chuẩn trước đó, HTML5 không chỉ cho các nhà sản xuất trình duyệt biết cần hỗ trợ những gì-nó còn ghi lại và chính thức [Hỗ trợ cách này](#). Vì thế, nếu trình duyệt không hỗ trợ một tính năng nào, nó sẽ không hiển thị nó.

LÉN ĐẾN TỐC ĐỘ

Cách HTML5 xử lý các phần tử lỗi thời

Vì HTML5 hỗ trợ tất cả HTML nên nó hỗ trợ nhiều tính năng được coi là lỗi thời. Chúng bao gồm các phần tử định dạng như ``, các phần tử hiệu ứng đặc biệt bị coi thường như `<blink>` và `<marquee>` và hệ thống khung HTML khó hiểu.

Sự cởi mở này là một điểm gây nhầm lẫn cho nhiều người học HTML5. Một mặt, HTML5 hoàn toàn nên cấm các phần tử lỗi thời này, vốn không xuất hiện trong thông số kỹ thuật chính thức trong nhiều năm (nếu có). Mặt khác, các trình duyệt hiện đại vẫn lặng lẽ hỗ trợ các yếu tố này và HTML5 được cho là phản ánh cách các trình duyệt web thực sự hoạt động. Vì vậy, một tiêu chuẩn để làm gì?

Để giải quyết vấn đề này, đặc tả HTML5 có hai phần tốc độ riêng biệt. Phần đầu tiên-là những gì bạn sẽ xem xét trong cuốn sách này-nhằm đến các nhà phát triển web. Các nhà phát triển cần tránh những thói quen xấu và những yếu tố bị loại bỏ trong quá khứ. Bạn có thể đảm bảo rằng mình đang tuân thủ phần này của tiêu chuẩn HTML5 bằng cách sử dụng trình xác thực HTML5.

Phần thứ hai, dài hơn nhiều của đặc tả HTML5 nhằm đến các nhà sản xuất trình duyệt. Trình duyệt cần hỗ trợ mọi thứ

đã từng tồn tại trong HTML, để tương thích ngược. Lý tưởng nhất là tiêu chuẩn HTML5 phải có đủ thông tin để ai đó có thể xây dựng trình duyệt từ đầu và làm cho trình duyệt đó hoàn toàn tương thích với các trình duyệt hiện đại ngày nay, cho dù trình duyệt đó đang xử lý đánh dấu mới hay cũ. Phần tiêu chuẩn này cho trình duyệt biết cách xử lý các phần tử lỗi thời chính thức không được khuyến khích nhưng vẫn được hỗ trợ.

Ngẫu nhiên, đặc tả HTML5 cũng chính thức hóa cách trình duyệt xử lý nhiều loại lỗi (ví dụ: thẻ bị thiếu hoặc không khớp). Điểm này rất quan trọng, vì nó đảm bảo rằng một trang bị lỗi sẽ hoạt động giống nhau trên các trình duyệt khác nhau, ngay cả khi liên quan đến các vấn đề tế nhị như cách một trang được mô hình hóa trong DOM (đó là Mô hình đối tượng tài liệu, cây của bộ nhớ trong), các đối tượng đại diện cho trang và được cung cấp cho mã JavaScript). Để tạo ra phần tiêu chuẩn dài và tẻ nhạt này, những người tạo ra HTML5 đã thực hiện các thử nghiệm toàn diện trên các trình duyệt hiện đại để tìm ra hành vi xử lý lỗi không có giấy tờ của chúng. Sau đó, họ đã viết nó ra.

2. Mở đường cho bò Đường bò

là con đường gồ ghề, có nhiều người giảm đạp đưa mọi người từ điểm này sang điểm khác. Một đường bò tồn tại bởi vì nó đang được sử dụng. Nó có thể không phải là cách tốt nhất có thể để di chuyển, nhưng tại một số điểm, đó là giải pháp hiệu quả nhất.

HTML5 tiêu chuẩn hóa các kỹ thuật không chính thức (nhưng được sử dụng rộng rãi) này. Nó có thể không gọn gàng bằng việc xây dựng một đường cao tốc được trải nhựa đẹp đẽ với cách tiếp cận hoàn toàn mới, nhưng nó có cơ hội thành công cao hơn. Đó là bởi vì việc chuyển sang các kỹ thuật mới có thể nằm ngoài khả năng hoặc sở thích của nhà thiết kế trang web trung bình. Và tệ hơn nữa, các kỹ thuật mới có thể không hoạt động đối với những khách truy cập đang sử dụng các trình duyệt cũ hơn. XHTML 2 đã cố gắng đẩy mọi người ra khỏi lối mòn và nó đã thất bại thảm hại.

GHI CHÚ

Việc lát đường bờ có một lợi ích rõ ràng: Nó sử dụng các kỹ thuật đã được thiết lập sẵn đã có một số mức độ hỗ trợ của trình duyệt. Nếu bạn cho các nhà phát triển web lựa chọn giữa một tính năng mới được thiết kế đẹp mắt, hoạt động trên 70% trình duyệt web hiện có và một bản hack lộn xộn hoạt động ở mọi nơi, thì họ sẽ chọn bản hack lộn xộn và luôn có lượng khán giả lớn hơn.

Cách tiếp cận “mở đường cho bờ” cũng đòi hỏi một số thỏa hiệp. Đôi khi nó có nghĩa là nắm lấy một tính năng được hỗ trợ rộng rãi nhưng được thiết kế kém. Một ví dụ là khả năng kéo và thả của HTML5 (trang 337), hoàn toàn dựa trên hành vi mà Microsoft đã tạo cho IE 5. Mặc dù tính năng kéo và thả này hiện được hỗ trợ trong tất cả các trình duyệt, nhưng nó bị mọi người ghét vì vụng về và quá phức tạp. Sự cao cả này đã khiến một số nhà thiết kế web phàn nàn rằng “HTML5 không chỉ khuyến khích hành vi xấu mà còn xác định hành vi đó”.

3. Hãy thực tế

Nguyên tắc này rất đơn giản: Các thay đổi nên có mục đích thiết thực. Và sự thay đổi càng đòi hỏi khắt khe thì phần thường càng lớn. Các nhà phát triển web có thể thích các tiêu chuẩn được thiết kế độc đáo, nhất quán, không có gì sai sót, nhưng đó không phải là lý do đủ chính đáng để thay đổi một ngôn ngữ đã được sử dụng để tạo ra vài tỷ trang.

Tất nhiên, việc quyết định mối quan tâm của ai là quan trọng nhất vẫn tùy thuộc vào ai đó. Một manh mối tốt là xem những trang web nào đang làm—hoặc đang cố gắng làm.

Ví dụ: trang web phổ biến thứ ba trên thế giới (tại thời điểm viết bài này) là YouTube. Nhưng vì HTML không có tính năng video thực sự trước HTML5, YouTube đã phải dựa vào plug-in trình duyệt Flash. Giải pháp này hoạt động tốt một cách ngạc nhiên vì trình cắm Flash có mặt trên hầu hết tất cả các máy tính được kết nối web.

Tuy nhiên, đôi khi có những trường hợp ngoại lệ, chẳng hạn như máy tính công ty bị khóa không cho phép Flash hoặc thiết bị di động không hỗ trợ Flash (như iPhone, iPad và Kindle). Và cho dù có bao nhiêu máy tính có Flash, vẫn có một trường hợp tốt để mở rộng tiêu chuẩn HTML để nó hỗ trợ trực tiếp một trong những cách cơ bản nhất mà mọi người sử dụng các trang web ngày nay—để xem video.

Có một động lực tương tự đằng sau nỗ lực của HTML5 để thêm nhiều tính năng tương tác hơn—hỗ trợ kéo và thả, nội dung HTML có thể chỉnh sửa, bắn vẽ hai chiều trên canvas, v.v. Bạn không cần phải tìm kiếm đâu xa để tìm các trang web sử dụng tất cả các tính năng này ngay bây giờ, một số trang web có phần bổ trợ như Adobe Flash và Microsoft Silverlight và những trang web khác có thư viện JavaScript hoặc (tốn công sức hơn) có các trang JavaScript được viết tùy chỉnh mã số. Vậy tại sao không thêm hỗ trợ chính thức vào tiêu chuẩn HTML và đảm bảo các tính năng này hoạt động ổn định trên tất cả các trình duyệt? Đó là những gì HTML5 bắt đầu thực hiện.

GHI CHÚ

Các plug-in trình duyệt như Flash sẽ không biến mất trong một sớm một chiều. Mặc dù có nhiều cải tiến, nhưng vẫn cần nhiều công việc hơn để xây dựng các ứng dụng đồ họa phức tạp trong HTML5. Nhưng tóm nhìn cuối cùng của HTML5 rất rõ ràng: cho phép các trang web cung cấp video, tính tương tác phong phú và hàng đồng thứ rưởm rà mà không yêu cầu phần bổ trợ.

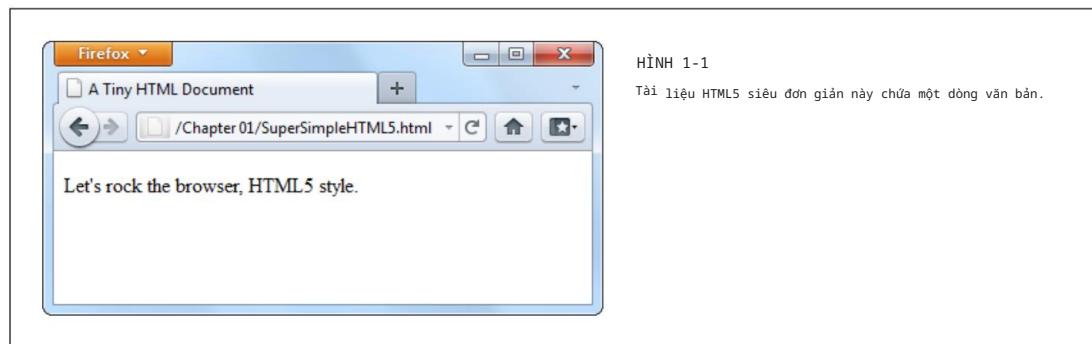
Cái nhìn đầu tiên của bạn về đánh dấu HTML5

Đây là một trong những tài liệu HTML5 đơn giản nhất mà bạn có thể tạo:

```
<!DOCTYPE html>
<title>Một tài liệu HTML nhỏ</title>
<p>Hãy khuấy động trình duyệt theo phong cách HTML5.</p>
```

Nó bắt đầu với loại tài liệu HTML5 (một mã đặc biệt được giải thích ở trang 11), tiếp theo là tiêu đề và sau đó là một số nội dung. Trong trường hợp này, nội dung là một đoạn văn bản.

Bạn đã biết giao diện này trông như thế nào trong trình duyệt, nhưng nếu bạn cần yên tâm, hãy xem Hình 1-1.



Bạn có thể giảm bớt tài liệu này hơn một chút. Ví dụ: tiêu chuẩn HTML5 không thực sự yêu cầu thẻ </p> cuối cùng, vì các trình duyệt biết đóng tất cả các phần tử đang mở ở cuối tài liệu (và tiêu chuẩn HTML5 làm cho hành vi này trở thành chính thức). Tuy nhiên, các phím tắt như thế này tạo ra các đánh dấu khó hiểu và có thể dẫn đến những lỗi không mong muốn.

Tiêu chuẩn HTML5 cũng cho phép bạn bỏ qua phần tử <title> nếu thông tin tiêu đề được cung cấp theo cách khác. Ví dụ: nếu bạn đang gửi tài liệu HTML trong thư email, bạn có thể đặt tiêu đề vào tiêu đề của thư email và đặt phần đánh dấu còn lại—loại tài liệu và nội dung—vào nội dung thư.

Nhưng đây rõ ràng là một kịch bản chuyên biệt.

Thông thường hơn, bạn sẽ muốn hoàn thiện tài liệu HTML5 cơ bản này. Hầu hết các nhà phát triển web đồng ý rằng việc sử dụng các phần <head> và <body> truyền thống có thể tránh nhầm lẫn, bằng cách tách biệt rõ ràng thông tin về trang của bạn (phần đầu) và nội dung thực của nó (phần thân). Cấu trúc này đặc biệt hữu ích khi bạn bắt đầu thêm tập lệnh, biểu định kiểu và phần tử meta.

```
<!DOCTYPE html>
<đầu>
    <title>Một tài liệu HTML nhỏ</title> </
head> <body> Hãy khuấy động trình duyệt
theo phong cách HTML5.</p> </body>
```

Như mọi khi, việc thực lè (ở đầu dòng ba và dòng sáu) hoàn toàn là tùy chọn.

Ví dụ này sử dụng nó để làm cho cấu trúc của trang dễ nhìn hơn ngay từ cái nhìn đầu tiên.

Cuối cùng, bạn có thể chọn bọc toàn bộ tài liệu (không bao gồm loại tài liệu) trong phần tử `<html>` truyền thống. Đây là những gì trông giống như:

```
<!DOCTYPE html>
<html>
<đầu>
    <title>Một tài liệu HTML nhỏ</title> </
head> <body> Hãy khuấy động trình duyệt
theo phong cách HTML5.</p> </body> </html>
```

Cho đến HTML5, mọi phiên bản của đặc tả HTML chính thức đều yêu cầu bạn sử dụng phần tử `<html>`, mặc dù thực tế là nó không có tác dụng đối với trình duyệt.

Tuy nhiên, HTML5 làm cho chi tiết này hoàn toàn không bắt buộc.

GHI CHÚ

Việc sử dụng các phần tử `<html>`, `<head>` và `<body>` chỉ đơn giản là vẫn dễ về phong cách. Bạn có thể bỏ chúng đi và trang của bạn sẽ hoạt động hoàn hảo, ngay cả trên các trình duyệt cũ không biết gì về HTML5. Trên thực tế, trình duyệt sẽ tự động giả định những chi tiết này. Vì vậy, nếu bạn sử dụng JavaScript để xem nhanh DOM (tập hợp các đối tượng lập trình đại diện cho trang của bạn), bạn sẽ tìm thấy các đối tượng cho các phần tử `<html>`, `<head>` và `<body>`, ngay cả khi bạn không thêm bản thân họ.

Hiện tại, ví dụ này nằm ở đâu đó giữa tài liệu HTML5 đơn giản nhất có thể và điểm bắt đầu chính xác của một trang web HTML5 thực tế. Trong các phần tiếp theo, bạn sẽ điền phần còn lại của những gì bạn cần và tìm hiểu sâu hơn một chút về phần đánh dấu.

Loại tài liệu HTML5 Dòng đầu

tiên của mọi tài liệu HTML5 là một mã đặc biệt được gọi là phần mở rộng . Loại tài liệu chỉ rõ tiêu chuẩn được sử dụng để viết phần đánh dấu tài liệu sau đó. Đây là cách một trang thông báo rằng nó tuân thủ tiêu chuẩn HTML5:

```
<!DOCTYPE html>
```

Điều đầu tiên bạn sẽ nhận thấy về loại tài liệu HTML5 là tính đơn giản nổi bật của nó.

Ví dụ, so sánh nó với loại tài liệu vô duyên mà các nhà phát triển web cần khi sử dụng XHTML 1.0 nghiêm ngặt:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Ngay cả các nhà phát triển web chuyên nghiệp cũng buộc phải sao chép và dán loại tài liệu XHTML từ tài liệu này sang tài liệu khác. Tuy nhiên, loại tài liệu HTML5 ngắn và linh hoạt, vì vậy bạn sẽ không gặp nhiều khó khăn khi nhập bằng tay.

Loại tài liệu HTML5 cũng đáng chú ý vì thực tế là nó không bao gồm phiên bản đặc tả chính thức (đó là phiên bản trong HTML5). Thay vào đó, [Sơ tóm tài liệu và các tính năng mới về HTML5 trang 6](#) (trang 6). Khi các tính năng mới được thêm vào ngôn ngữ HTML, chúng sẽ tự động có sẵn trên trang của bạn mà không yêu cầu bạn chỉnh sửa loại tài liệu.

Tất cả những điều này đặt ra một câu hỏi hay-nếu HTML5 là ngôn ngữ sống, tại sao trang web của bạn lại yêu cầu bắt kỳ loại tài liệu nào?

Câu trả lời là loại tài liệu vẫn còn vì những lý do lịch sử. Nếu không có loại tài liệu, hầu hết các trình duyệt (bao gồm cả Internet Explorer và Firefox) sẽ mất hiệu lực. Ở chế độ này, [chúng sẽ cố gắng hiển thị các trang theo các quy tắc hơi lỗi mà chúng đã sử dụng trong các phiên bản cũ hơn](#). Vấn đề là chế độ quirks của một trình duyệt khác với chế độ tiếp theo, vì vậy các trang được thiết kế cho một trình duyệt có khả năng nhận được các phông chữ có kích thước không nhất quán, bối rối lộn xộn và các trực trặc khác trên một trình duyệt khác.

Khi bạn thêm một loại tài liệu, trình duyệt sẽ nhận ra rằng bạn muốn sử dụng loại tài liệu chặt chẽ [chế độ tiêu chuẩn](#), hơn để đảm bảo rằng trang web được hiển thị với định dạng và bố cục nhất quán trên mọi trình duyệt hiện đại. Trình duyệt thậm chí không quan tâm đến loại tài liệu bạn sử dụng (chỉ [cái mà với một vài ngoại lệ](#)). Thay vào đó, nó chỉ kiểm tra xem bạn có doctype hay không. Loại tài liệu HTML5 [đơn giản](#) là loại tài liệu hợp lệ ngắn nhất, vì vậy nó luôn kích hoạt chế độ tiêu chuẩn.

TIẾN BOA Loại tài liệu HTML5 kích hoạt chế độ tiêu chuẩn trên tất cả các trình duyệt có chế độ tiêu chuẩn, kể cả những trình duyệt không biết gì về HTML5. Vì lý do đó, bạn có thể sử dụng loại tài liệu HTML5 ngay bây giờ, trong tất cả các trang của mình, ngay cả khi bạn cần tắt một số tính năng ít được hỗ trợ của HTML5.

Mặc dù loại tài liệu chủ yếu nhằm báo cho trình duyệt web biết phải làm gì, nhưng các tác nhân khác cũng có thể kiểm tra nó. Điều này bao gồm trình xác thực HTML5, công cụ tìm kiếm, công cụ thiết kế và những người khác khi họ đang cố gắng tìm ra hướng vị đánh dấu mà bạn đã chọn cho trang của mình.

Mã hóa ký tự

Đây là [tiêu chuẩn](#) cho máy tính biết cách chuyển đổi văn bản của bạn thành một chuỗi byte khi nó được lưu trữ trong tệp—và cách chuyển đổi ngược lại khi tệp được mở. Vì những lý do lịch sử, có nhiều cách mã hóa ký tự khác nhau trên thế giới. Ngày nay, hầu như tất cả các trang web tiếng Anh đều sử dụng mã hóa

được gọi là UTF-8, nhỏ gọn, nhanh chóng và hỗ trợ tất cả các ký tự không phải tiếng Anh mà bạn cần.

Thông thường, máy chủ web lưu trữ các trang của bạn được định cấu hình để cho các trình duyệt biết rằng nó đang phân phát các trang có một loại mã hóa nhất định. Tuy nhiên, vì bạn không thể chắc chắn rằng máy chủ web của mình sẽ thực hiện bước này (trừ khi bạn sở hữu máy chủ) và vì trình duyệt có thể gặp sự cố bảo mật khó hiểu khi chúng cố gắng đoán mã hóa của trang, nên bạn luôn phải thêm mã hóa thông tin cho đánh dấu của bạn.

HTML5 làm cho điều đó trở nên dễ dàng. Tất cả những gì bạn cần làm là thêm phần tử `<meta>` được hiển thị bên dưới vào đầu phần `<head>` của bạn (hoặc ngay sau loại tài liệu, nếu bạn không xác định phần tử `<head>`):

```
<đầu>
<meta charset="utf-8">
<title>Một tài liệu HTML nhỏ</title> </
head>
```

Các công cụ thiết kế như Dreamweaver tự động thêm chi tiết này khi bạn tạo một trang mới. Họ cũng đảm bảo rằng các tệp của bạn đang được lưu bằng mã hóa UTF. Tuy nhiên, nếu đang sử dụng trình soạn thảo văn bản thông thường, bạn có thể cần thực hiện thêm một bước để đảm bảo các tệp của mình được lưu chính xác. Ví dụ: khi chỉnh sửa tệp HTML trong Notepad (trên Windows), trong hộp thoại Lưu dưới dạng, bạn phải chọn UTF-8 từ danh sách Mã hóa (ở dưới cùng). TrongTextEdit (trên máy Mac), trong hộp thoại Lưu dưới dạng, trước tiên bạn cần chọn Định dạng Tạo văn bản thuần túy để đảm bảo chương trình lưu trang của bạn dưới dạng tệp văn bản thông thường, sau đó chọn “Unicode (UTF-8)” từ menu bật lên Mã hóa văn bản thuần túy.

Ngôn ngữ Nó được coi

là phong cách tốt để chỉ ra rằng trang web của bạn đòi hỏi ngôn ngữ tự nhiên . Thông tin này hữu ích cho người khác-ví dụ: các công cụ tìm kiếm có thể sử dụng nó để lọc kết quả tìm kiếm để chúng chỉ bao gồm các trang phù hợp với ngôn ngữ của người tìm kiếm.

Để chỉ định ngôn ngữ của một số nội dung, bạn sử dụng thuộc tính lang trên bất kỳ phần tử nào, cùng với mã ngôn ngữ thích hợp. Đó là dành cho tiếng Anh `đối gặn mān họ̄ng bạ̣o k̄y l̄b̄eht b̄im t̄p̄áy`

<http://tinyurl.com/l-codes> .

Cách dễ nhất để thêm thông tin ngôn ngữ vào trang web của bạn là sử dụng phần tử `<html>` với thuộc tính lang:

```
<html lang="vi">
```

Chi tiết này cũng có thể giúp trình đọc màn hình nếu một trang có văn bản từ nhiều ngôn ngữ. Trong trường hợp này, bạn sử dụng thuộc tính lang để chỉ ra ngôn ngữ của các phần khác nhau trong tài liệu của bạn; ví dụ: bằng cách áp dụng nó cho các phần tử `<div>` khác nhau bao bọc nội dung khác nhau. Trình đọc màn hình sau đó có thể xác định phần nào sẽ đọc to.

Thêm biểu định kiểu Hầu như mọi

trang web trong một trang web chuyên nghiệp, được thiết kế phù hợp đều sử dụng biểu định kiểu CSS. Bạn chỉ định các biểu định kiểu bạn muốn sử dụng bằng cách thêm các phần tử `<link>` vào phần `<head>` của tài liệu HTML5, như sau:

```
<đầu>
    <meta charset="utf-8">
    <title>Một tài liệu HTML nhỏ</title>
    <link href="styles.css" rel="stylesheet"> <
head>
```

Phương pháp này ít nhiều giống với cách bạn định kèm biểu định kiểu vào tài liệu HTML truyền thống, nhưng đơn giản hơn một chút.

GHI CHÚ

Ghi chú Vì CSS là ngôn ngữ biểu định kiểu duy nhất nên không cần thêm thuộc tính type="text/css" mà các trang web thường yêu cầu.

Thêm JavaScript

bắt đầu hoạt động như một cách để thêm sự lấp lánh và quyền rũ phủ phiếm vào các trang web. Ngày nay, JavaScript ít nói về giao diện người dùng rườm rà và nhiều hơn về các ứng dụng web mới, bao gồm các ứng dụng email siêu tiên tiến, trình xử lý văn bản và công cụ lập bản đồ chạy ngay trong trình duyệt.

Bạn thêm JavaScript vào trang HTML5 giống như cách bạn thêm JavaScript vào trang HTML truyền thống. Đây là một ví dụ tham chiếu đến một tệp bên ngoài có mã JavaScript:

```
<đầu>
  <meta charset="utf-8">
  <title>Một tài liệu HTML nhỏ</title>
  <script src="scripts.js"></script> </
head>
```

Không cần bao gồm thuộc tính language="JavaScript". Trình duyệt có nghĩa là bạn muốn có JavaScript trừ khi bạn chỉ định khác và bởi vì JavaScript là ngôn ngữ kịch bản HTML duy nhất có hỗ trợ rộng rãi, bạn sẽ không bao giờ muốn như vậy. Tuy nhiên, vẫn cần nhớ rằng nó ~~đang~~ là ngôn ngữ để lập trình, ngôn ngữ của nó là HTML để áp dụng cho trang web.

Nếu bạn dành nhiều thời gian kiểm tra các trang hỗ trợ JavaScript của mình trong Internet Explorer, bạn cũng có thể muốn thêm một nhận xét đặc biệt có tên `đánh dấu` (`đánh dấu`) vào phần `head` của trang web, sau mã hóa ký tự. Nó trông như thế này:

```
<đầu>
<meta charset="utf-8">
<!-- được lưu từ url=(0014)about:internet -->
<title>Một tài liệu HTML nhỏ</title> <script
src="scripts.js"></script> </head>
```

Nhận xét này yêu cầu Internet Explorer coi trang như thể nó đã được tải xuống từ một trang web từ xa. Nếu không, IE sẽ chuyển sang chế độ khóa đặc biệt, bật lên cảnh báo bảo mật trong thanh thông báo và sẽ không chạy bất kỳ mã JavaScript nào cho đến khi bạn nhấp rõ ràng vào "Cho phép nội dung bị chặn".

Tất cả các trình duyệt khác đều bỏ qua nhận xét "mark of the Web" và sử dụng cùng một cài đặt bảo mật cho các trang web từ xa và các tệp cục bộ.

Sản phẩm cuối cùng

Nếu bạn đã làm theo các bước này, bạn sẽ có một tài liệu HTML5 trông giống như sau:

```
<!DOCTYPE html>
<html lang="vi">
<head>
<meta charset="utf-8">
<title>Một tài liệu HTML nhỏ</title>
<link href="styles.css" rel="stylesheet">
<script src="scripts.js"></script> </head>

<body>
<p>Hãy khuấy động trình duyệt, phong cách HTML5.</p>
</body> </html>
```

Mặc dù đây không còn là tài liệu HTML5 ngắn nhất có thể, nhưng đó là điểm khởi đầu hợp lý cho bất kỳ trang web nào bạn muốn xây dựng. Và mặc dù ví dụ này có vẻ cực kỳ buồn tẻ, nhưng đừng lo lắng-trong chương tiếp theo, bạn sẽ bước tới một trang thực tế chứa đầy nội dung được sắp xếp cẩn thận và tất cả được gói gọn trong CSS.

GHI CHÚ

Tất cả cú pháp HTML5 mà bạn đã học trong phần này—loại tài liệu mới, phần tử meta để mã hóa ký tự, thông tin ngôn ngữ, biểu định kiểu và tham chiếu JavaScript, đều hoạt động trong trình duyệt cả mới và cũ. Đó là bởi vì chúng dựa vào các giá trị mặc định và các phương pháp sửa lỗi tích hợp sẵn mà tất cả các trình duyệt sử dụng.

Xem xét kỹ hơn về cú pháp HTML5

Như bạn đã biết, HTML5 nới lỏng một số quy tắc. Đó là bởi vì những người tạo ra HTML5 muốn ngôn ngữ này phản ánh chính xác hơn thực tế của trình duyệt web-nói cách khác, họ muốn thu hẹp khoảng cách giữa “các trang web hoạt động” và “các trang web được coi là hợp lệ theo tiêu chuẩn”. Trong phần tiếp theo, bạn sẽ xem xét kỹ hơn các quy tắc đã thay đổi như thế nào.

GHI CHÚ

Vẫn còn rất nhiều thực tiễn lỗi thời mà các trình duyệt hỗ trợ nhưng tiêu chuẩn HTML5 đó thực sự nghiêm ngặt.

Không khuyến khích. Để được trợ giúp nắm bắt những điều này trong các trang web của riêng bạn, bạn sẽ cần một trình xác thực HTML5 (trang 17).

Các quy tắc nới lỏng

Trong lần đầu tiên xem tài liệu HTML5, bạn đã phát hiện ra rằng HTML5 đặt các phần tử `<html>`, `<head>` và `<body>` thành tùy chọn (mặc dù chúng vẫn có thể khá hữu ích). Nhưng thái độ thoải mái của HTML5 không dừng lại ở đó.

HTML5 bỏ qua cách viết hoa, cho phép bạn viết đánh dấu như sau:

```
<P>Chữ hoa và chữ thường <EM>không quan trọng</EM> trong tên thẻ.</p>
```

HTML5 cũng cho phép bạn bỏ qua dấu gạch chéo đóng từ một phần tử **trống** – đó là một yếu tố không có nội dung lồng nhau, chẳng hạn như `` (hình ảnh), `
` (ngắt dòng) hoặc `<hr>` (dường ngang). Dưới đây là ba cách tương đương để thêm ngắt dòng:

```
tôi không thẻ<br />
lùi lại<br>
hoặc chuyển tiếp.<br/>
> Tôi bị bắt
```

HTML5 cũng thay đổi quy tắc cho các thuộc tính. Giá trị thuộc tính không cần dấu ngoặc kép nữa, miễn là giá trị không bao gồm ký tự bị hạn chế (thường gọi là `>`, = hoặc dấu cách). Đây là một ví dụ về phần tử `` tận dụng khả năng này:

```
<img alt="Tinh vân Đầu ngựa" src=Horsehead01.jpg>
```

Các thuộc tính không có giá trị cũng được cho phép. Vì vậy, trong khi XHTML yêu cầu cú pháp hơi dư thừa để đặt một hộp kiểm ở trạng thái đã chọn.

```
<loại đầu vào="hộp kiểm" đã chọn ="đã kiểm tra" />
```

giờ đây bạn có thể phục hồi truyền thống HTML 4.01 ngắn hơn bao gồm tên thuộc tính của riêng nó.

```
<loại đầu vào="hộp kiểm" đã chọn >
```

Điều khiển một số người đặc biệt lo lắng không phải là việc HTML5 cho phép những thứ này. Thực tế là các nhà phát triển không nhất quán có thể tình cờ chuyển đổi qua lại giữa phong cách nghiêm ngặt hơn và phong cách lỏng lẻo hơn, thậm chí sử dụng cả hai trong cùng một tài liệu.

Tuy nhiên, trên thực tế, XHTML cho phép có sự không nhất quán tương tự. Trong cả hai trường hợp,

phong cách tốt là trách nhiệm của nhà thiết kế web và trình duyệt chấp nhận bất cứ điều gì bạn có thể ném vào nó.

Dưới đây là tóm tắt nhanh về những gì tạo nên phong cách HTML5 tốt – và những quy ước mà các ví dụ trong cuốn sách này tuân theo, ngay cả khi chúng không bắt buộc phải tuân theo:

- Bao gồm các phần tử <html>, <body> và <head> tùy chọn. Phần tử <html> là một vị trí thuận tiện để xác định ngôn ngữ tự nhiên của trang (trang 13); và các phần tử <body> và <head> giúp tách nội dung trang khỏi các chi tiết trang khác.
- Sử dụng thẻ chữ thường (như <p> thay vì <P>). Chúng không cần thiết, nhưng chúng phổ biến hơn nhiều, dễ gõ hơn (vì bạn không cần phím Shift) và gần như không ôn ào.
- Sử dụng dấu ngoặc kép xung quanh các giá trị thuộc tính. Dấu ngoặc kép tồn tại là có lý do – để bảo vệ bạn khỏi những sai lầm quá dễ mắc phải.
Không có dấu ngoặc kép, một ký tự không hợp lệ có thể phá vỡ toàn bộ trang của bạn.

Mặt khác, có một số quy ước cũ mà cuốn sách này bỏ qua (và bạn cũng có thể bỏ qua). Các ví dụ trong cuốn sách này không đóng các phần tử trống, bởi vì hầu hết các nhà phát triển không bận tâm thêm dấu gạch chéo (/) khi họ chuyển sang HTML5.

Tương tự, không có lý do gì để ủng hộ dạng thuộc tính dài khi tên thuộc tính và giá trị thuộc tính giống nhau.

Xác thực HTML5

Phong cách mới, thoải mái của HTML5 có thể phù hợp với bạn. Hoặc, chính suy nghĩ rằng có thể có mã đánh dấu không nhất quán, đầy lỗi ẩn dang sau một trình duyệt hoàn toàn ổn định có thể đủ khiến bạn mất ngủ. Nếu bạn rơi vào trường hợp thứ hai, bạn sẽ rất vui khi biết rằng một công cụ xác thực có thể tìm kiếm đánh dấu không tuân thủ các tiêu chuẩn được khuyến nghị của HTML5, ngay cả khi nó không gây khó khăn cho trình duyệt.

Dưới đây là một số vấn đề tiềm ẩn mà trình xác thực có thể phát hiện:

- Thiếu các phần tử bắt buộc (ví dụ: phần tử <title>)
- Thẻ bắt đầu không có thẻ kết thúc phù hợp
- Các thẻ được lồng không chính xác
- Thẻ thiếu thuộc tính (ví dụ: phần tử không có phần tử src thuộc tính)
- Các yếu tố hoặc nội dung ở sai vị trí (ví dụ: văn bản được đặt trực tiếp trong phần <head>)

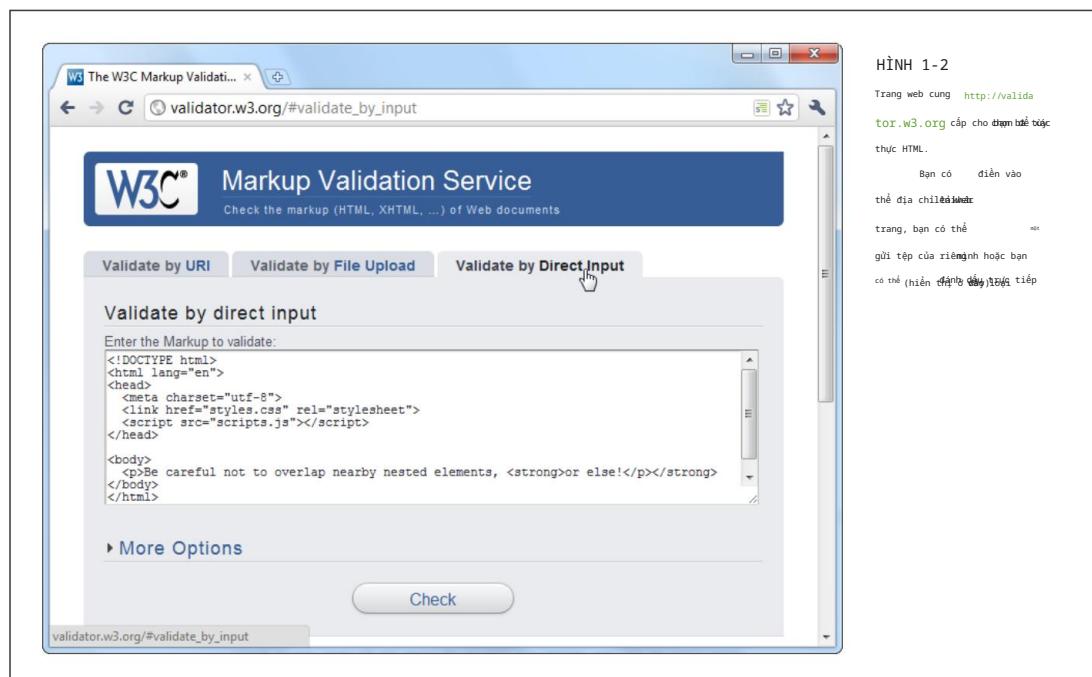
Các công cụ thiết kế web như Dreamweaver thường có trình xác thực riêng. Nhưng nếu bạn không muốn chi phí hoặc sự phức tạp của trình chỉnh sửa web chuyên nghiệp, bạn có thể nhận được thông tin tương tự từ công cụ xác thực trực tuyến. Dưới đây là cách sử dụng trình xác thực phổ biến do tổ chức tiêu chuẩn W3C cung cấp:

1. Trong trình duyệt web của bạn, hãy truy cập <http://validator.w3.org> (Hình 1-2).

Trình xác thực W3C cung cấp cho bạn ba lựa chọn, được biểu thị bằng ba tab riêng biệt: "Xác thực bằng URI" (đối với trang đã trực tuyến), "Xác thực bằng cách tải tệp lên" (đối với trang được lưu trữ trong tệp trên máy tính của bạn) và "Xác thực bằng cách nhập trực tiếp" (đối với một loạt đánh dấu bạn tự nhập).

2. Nhấp vào tab bạn muốn và cung cấp nội dung HTML của bạn.

- Xác thực bằng URI cho phép bạn xác thực một trang web hiện có. Bạn chỉ cần nhập URL của trang vào hộp Địa chỉ (ví dụ: <http://www.MySloopySite.com/FlawedPage.html>).
- Xác thực bằng Tải lên cho phép bạn tải lên bất kỳ tệp nào từ máy tính của mình. Đầu tiên, nhấp vào nút Duyệt qua (trong Chrome, nhấp vào Chọn tệp). Trong hộp thoại Mở, chọn tệp HTML của bạn rồi bấm Mở.
- Xác thực bằng cách nhập trực tiếp cho phép bạn xác thực bất kỳ đánh dấu nào—you chỉ cần nhập đánh dấu đó vào một hộp lớn. Cách dễ nhất để sử dụng tùy chọn này là sao chép phần đánh dấu từ trình soạn thảo văn bản của bạn và dán nó vào hộp trên trang xác thực W3C.



Trước khi tiếp tục, bạn có thể nhấp vào Tùy chọn khác để thay đổi một số cài đặt, nhưng có thể bạn sẽ không làm như vậy. Tốt nhất là để trình xác thực tự động phát hiện loại tài liệu —theo cách đó, trình xác nhận sẽ sử dụng loại tài liệu được chỉ định trong trang web của bạn. Tương tự, sử dụng tính năng phát hiện tự động cho bộ ký tự trừ khi bạn có

một trang HTML được viết bằng ngôn ngữ khác và trình xác thực gặp sự cố khi xác định bộ ký tự chính xác.

3. Nhập vào nút Kiểm tra.

Nhấp chuột này sẽ gửi trang HTML của bạn đến trình xác thực W3C. Sau một khoảng thời gian ngắn, báo cáo xuất hiện. Bạn sẽ thấy liệu tài liệu của mình có vượt qua quá trình kiểm tra xác thực hay không và nếu thất bại, trình xác thực đã phát hiện ra lỗi gì (xem Hình 1-3).

GHI CHÚ

Ngay cả trong tài liệu HTML hoàn toàn hợp lệ, bạn có thể nhận được một số cảnh báo vô hại, bao gồm mã hóa ký tự được xác định tự động và dịch vụ xác thực HTML5 được coi là một tính năng thử nghiệm, chưa hoàn thiện.

HÌNH 1-3

Trình xác thực đã phát hiện **tài liệu** có **4 lỗi**.

cả hai đều thiếu **title**,
lịch sử của trang.
< tiêu đề >

Thứ hai, nó đóng phần **<P>**
từ trước **** bên
**** trong.
(Bề giải
quyết **thông** có thể
</p>
</p>

Ngẫu nhiên, tài liệu này
vẫn **đóng** **strong** **đã** **đến**.

Sự trở lại của XHTML

Như bạn đã biết, HTML5 báo hiệu sự kết thúc cho vị vua trước đây của Web- XHTML. Tuy nhiên, thực tế không hoàn toàn đơn giản như vậy và những người hâm mộ XHTML không cần phải từ bỏ tất cả những điều họ yêu thích về thế hệ ngôn ngữ đánh dấu trước đây.

Đầu tiên, hãy nhớ rằng cú pháp XHTML vẫn tồn tại. Các quy tắc mà XHTML thực thi vẫn là nguyên tắc (ví dụ: lòng các phần tử chính xác) hoặc vẫn được hỗ trợ dưới dạng các quy ước tùy chọn (ví dụ: bao gồm dấu gạch chéo ở cuối trên các phần tử trống).

Nhưng nếu bạn muốn các quy tắc của pháp XHTML thì sao? Có thể bạn đang lo lắng rằng bạn (hoặc những người làm việc cùng bạn) sẽ vô tình rơi vào những quy ước lỏng lẻo hơn của HTML thông thường. Để ngăn điều đó xảy ra, bạn cần sử dụng XHTML5—một tiêu chuẩn ít phô biến hơn mà về cơ bản là HTML5 với các hạn chế dựa trên XML được đặt lên hàng đầu.

Để biến tài liệu HTML5 thành tài liệu XHTML5, bạn cần thêm rõ ràng không gian tên XHTML vào phần tử `<html>`, đóng mọi phần tử, đảm bảo bạn sử dụng thẻ chữ thường, v.v. Đây là ví dụ về một trang web thực hiện tất cả các bước sau:

```
<!DOCTYPE html>
<html lang="vi" xmlns="http://www.w3.org/1999/xhtml"> <head>

    <meta charset="utf-8"/>
    <title>Một tài liệu HTML nhỏ</title> <link
        href="styles.css" rel="stylesheet"/> <script
        src="scripts.js"></script>
    </head>

    <body>
        <p>Hãy khuấy động trình duyệt, phong cách XHTML5.</p>
    </body> </html>
```

Giờ đây, bạn có thể sử dụng trình xác thực XHTML5 để kiểm tra lỗi nghiêm ngặt hơn nhằm thực thi các quy tắc XHTML kiêng cũ. Trình xác thực W3C tiêu chuẩn sẽ không làm điều đó, nhưng trình xác thực sẽ tùy ý miễn là bạn nhập vào nút **đã** chọn và **chọn** **XHTML5** trong danh sách **Lỗi** tại **Cung cấp** khi bạn đang sử dụng phương pháp nhập trực tiếp và dán đánh dấu của mình vào hộp văn bản.

Bằng cách làm theo các bước này, bạn có thể tạo và xác thực tài liệu XHTML. Tuy nhiên, sẽ vẫn xử lý trang của bạn dưới dạng tài liệu HTML5—một tài liệu chỉ xảy ra để có một phức hợp thấp kém XML. Họ sẽ không cố gắng áp dụng bất kỳ quy tắc bổ sung nào.

Nếu bạn muốn chuyển sang XHTML5 hoàn toàn, bạn cần định cấu hình máy chủ web của mình để phục vụ trang của bạn với loại MIME ứng dụng/xhtml+xml hoặc ứng dụng/xml, thay vì văn bản/html tiêu chuẩn. (Xem trang 152 để biết thông tin chi tiết về các loại MIME.)

Nhưng trước khi bạn gọi cho công ty lưu trữ web của mình, hãy cảnh báo rằng thay đổi này sẽ ngăn không cho trang của bạn hiển thị trên bất kỳ phiên bản Internet Explorer nào trước IE 9. Vì lý do đó, XHTML5 thực sự là một công cụ phá vỡ thỏa thuận ngay lập tức trong trình duyệt.

Ngẫu nhiên, các trình duyệt hỗ trợ XHTML5 xử lý nó khác với HTML5 thông thường. Họ cố gắng xử lý trang dưới dạng tài liệu XML và nếu quá trình đó không thành công (vì bạn đã để lại lỗi), trình duyệt sẽ từ bỏ phần còn lại của tài liệu.

Điểm mấu chốt? Đối với đại đa số các nhà phát triển web, từ những người bình thường đến những chuyên gia nghiêm túc, XHTML5 không đáng để bạn tâm. Các ngoại lệ duy nhất là các nhà phát triển có

ghi nhớ mục tiêu cụ thể liên quan đến XML; ví dụ: các nhà phát triển muốn thao tác nội dung trong trang của họ bằng các tiêu chuẩn liên quan đến XML như XQuery và XPath.

TIỀN BOA Nếu tò mò, bạn có thể lừa trình duyệt của mình chuyển sang chế độ XHTML. Chỉ cần đổi tên tệp của bạn để nó kết thúc bằng .xhtml hoặc .xht. Sau đó mở nó từ ổ cứng của bạn. Hầu hết các trình duyệt (bao gồm Firefox, Chrome và IE 9 trở lên) sẽ hoạt động như thế bạn đã tải xuống trang từ máy chủ web có loại MIME XML. Nếu có một lỗi nhỏ trong trang, cửa sổ trình duyệt sẽ hiển thị trang được xử lý một phần (IE), thông báo lỗi XML (Firefox) hoặc kết hợp cả hai (Chrome).

Họ phần tử của HTML5

Cho đến giờ, chương này đã tập trung vào những thay đổi đối với cú pháp của HTML5. Nhưng điều quan trọng hơn là các phép cộng, phép trừ và các thay đổi đối với HTML mà nó hỗ trợ. Trong các phần sau, cái nhìn tổng quan về cách chúng đã thay đổi.

Các yếu tố đã thêm

Trong các chương tiếp theo, bạn sẽ dành phần lớn thời gian để tìm hiểu về các thành phần mới—những thành phần chưa từng tồn tại trên các trang web cho đến bây giờ. Bảng 1-1 có phần xem trước những gì đang có trong cửa hàng (và nơi bạn có thể đọc thêm về nó).

BẢNG 1-1 Mới phần tử HTML5

THỂ LOẠI	YẾU TỐ	ĐƯỢC THẢO LUẬN TRONG
Các yếu tố ngữ nghĩa để cấu trúc một trang	<article>, <aside>, <figcaption>, <figure>, <footer>, <header>, <nav>, <section>, <details>, <summary>	chương 2
Yếu tố ngữ nghĩa cho văn bản	<mark>, <time>, <wbr> (trước đây được hỗ trợ nhưng hiện là một phần chính thức của ngôn ngữ này)	Chương 3
Biểu mẫu web và tính tương tác	<input> (không mới, nhưng có nhiều kiểu con mới) <datalist>, <keygen>, <meter>, <progress>, <command>, <menu>, <output>	Chương 4
Âm thanh, video và plug-in	<audio>, <video>, <source>, <embed> (trước đây được hỗ trợ nhưng hiện là một phần chính thức của ngôn ngữ này)	Chương 5
Thanh sơn dầu	<canvas>	Chương 8
Hỗ trợ ngôn ngữ không phải tiếng Anh	<bdo>, <rp>, <rt>, <ruby>	Đặc tả HTML5 tại danh dấu http://dev.w3.org/html5/

Các phần tử đã xóa

Mặc dù HTML5 bổ sung thêm các phần tử mới, nhưng nó cũng khởi động một số phần tử ra khỏi họ chính thức. Các phần tử này sẽ tiếp tục hoạt động trong trình duyệt, nhưng bất kỳ trình xác thực HTML5 đang hoàng nào cũng sẽ hút chúng ra khỏi nơi ẩn náu và lớn tiếng phàn nàn.

Rõ ràng nhất, HTML5 giữ nguyên triết lý (lần đầu tiên được tạo ra bằng XHTML) không được hoan yếu tố trình bày nghênh trong ngôn ngữ này. Các phần tử trình bày là các phần tử chỉ đơn giản là có để thêm định dạng cho các trang web và ngay cả nhà thiết kế web thân thiện nhất cũng biết đó là công việc của biểu định kiểu. Từ chối bao gồm các yếu tố mà các nhà phát triển chuyên nghiệp đã không sử dụng trong nhiều năm (như `<big>`, `<center>`, ``, `<tt>` và `<strike>`). Các thuộc tính trình bày của HTML đã chết cùng một cái chết, vì vậy không có lý do gì để thử lại tất cả chúng ở đây.

Ngoài ra, HTML5 còn bổ sung thêm cát vào nấm mò nơi các nhà phát triển web đã chôn vùi tính năng khung HTML. Khi nó được tạo lần đầu tiên, các khung HTML dường như là một cách tuyệt vời để hiển thị nhiều trang web trong một cửa sổ trình duyệt. Nhưng giờ đây, các khung được biết đến nhiều hơn như một cơn ác mộng về khả năng truy cập vì chúng gây ra sự cố với công cụ tìm kiếm, phần mềm hỗ trợ và thiết bị di động. Thật thú vị, phần tử `<iframe>` cho phép các nhà phát triển đặt một trang vào bên trong một trang khác—phát ra tiếng kêu.

Đó là bởi vì các ứng dụng web sử dụng `<iframe>` cho một loạt các tác vụ tích hợp, chẳng hạn như kết hợp các cửa sổ YouTube, quảng cáo và hộp tìm kiếm của Google trong một trang web.

Một vài phần tử khác đã bị loại bỏ vì chúng dư thừa hoặc là nguyên nhân của các lỗi phổ biến, bao gồm `<acronym>` (dùng `<abbr>` thay thế) và `<applet>` (vì `<object>` được ưu tiên hơn). Nhưng phần lớn họ phần tử tồn tại trong HTML5.

GHI CHÚ

Đối với những người theo dõi, HTML5 bao gồm một nhóm chỉ hơn 100 phần tử. Trong số này, gần 30 là mới và khoảng 10 là thay đổi đáng kể. Bạn có thể duyệt danh sách các phần tử (và xem lại phần tử nào mới hoặc đã thay đổi) tại <http://dev.w3.org/html5/markup>.

Các phần tử được điều chỉnh

HTML5 có một thủ thuật kỳ quặc khác: Đôi khi nó điều chỉnh một tính năng cũ cho một mục đích mới. Ví dụ: hãy xem xét phần tử `<small>`, không được ưa chuộng vì là một cách vụng về để thu nhỏ kích thước phông chữ của một khối văn bản—một nhiệm vụ được thực hiện đúng cách hơn với biểu định kiểu. Nhưng không giống như phần tử `<big>` bị loại bỏ, HTML5 giữ lại phần tử `<small>` với một sự thay đổi. Bây giờ, phần tử `<small>` đại diện cho “chữ in nhỏ”—ví dụ: ngôn ngữ pháp lý mà không ai muốn bạn đọc ở cuối hợp đồng:

```
<small>Những người tạo ra trang web này sẽ không chịu trách nhiệm pháp lý cho bất kỳ thương tích nào có thể xảy ra do đua xe đạp một bánh không có người giám sát.</small>
```

Văn bản bên trong phần tử `<small>` vẫn được hiển thị như mọi khi, sử dụng cỡ chữ nhỏ hơn, trừ khi bạn ghi đè cài đặt đó bằng biểu định kiểu.

GHI CHÚ

Ý kiến về kỹ thuật `<small>` này khác nhau. Một mặt, nó rất tốt cho khả năng tương thích ngược, vì các trình duyệt cũ đã hỗ trợ phần tử `<small>` và vì vậy chúng sẽ tiếp tục hỗ trợ phần tử này trong trang HTML5. Mặt khác, nó đưa ra một sự thay đổi ý nghĩa có thể gây nhầm lẫn cho các trang cũ. Họ có thể đang sử dụng phần tử `<small>` cho mục đích trình bày mà không muốn đề xuất "bản in nhỏ".

Một yếu tố thay đổi khác là `<hr>` (viết tắt của thuộc kè ngang), vẽ một đường phân cách giữa các phần. Trong HTML5, `<hr>` đại diện cho một ngắt theo chủ đề-ví dụ: chuyển đổi sang một chủ đề khác. Định dạng mặc định vẫn được giữ nguyên, nhưng bây giờ một ý nghĩa mới được áp dụng.

Tương tự như vậy, `<s>` (đối với văn bản bị gạch bỏ), không chỉ là việc gạch bỏ các từ nữa-nó hiện đại diện cho văn bản không còn chính xác hoặc có liên quan nữa và đã bị "gạch" khỏi tài liệu. Cả hai thay đổi này đều tinh tế hơn so với sự thay đổi về ý nghĩa của phần tử `<small>`, bởi vì chúng nắm bắt được những cách mà các phần tử `<hr>` và `<s>` thường được sử dụng trong HTML truyền thống.

ĐỊNH DẠNG ĐẬM VÀ IN nghiêng

Các yếu tố được điều chỉnh quan trọng nhất là các yếu tố dành cho định dạng in đậm và in nghiêng. Hai trong số các phần tử được sử dụng phổ biến nhất của HTML—đó là `` cho chữ đậm và `<i>` cho chữ nghiêng—đã được thay thế một phần khi phiên bản đầu tiên của XHTML giới thiệu các phần tử `` và `` giống nhau. Ý tưởng là ngừng xem xét mọi thứ từ quan điểm định dạng (in đậm và in nghiêng), và thay vào đó là các yếu tố thay thế có ý nghĩa logic thực sự (mức độ quan trọng mạnh hoặc nhẫn mạnh). Ý tưởng này khá hợp lý, nhưng các thẻ `` và `<i>` vẫn tồn tại dưới dạng các lựa chọn thay thế ngắn hơn và quen thuộc hơn cho bản sửa lỗi XHTML.

HTML5 có một vết nứt khác để giải quyết vấn đề. Thay vì cố gắng buộc các nhà phát triển tránh xa `` và `<i>`, nó gán ý nghĩa mới cho cả hai yếu tố. Ý tưởng là cho phép cả bốn yếu tố cùng tồn tại trong một tài liệu HTML5 đáng kính. Kết quả là một bộ hướng dẫn hơi khó hiểu được liệt kê ở đây:

- Sử dụng `` cho văn bản nổi bật **mạnh mẽ** **tầm quan trọng**. Đây là văn bản cần so với môi trường xung quanh.
- Sử dụng `` cho văn bản nên được in đậm nhưng không quan trọng hơn phần còn lại của văn bản. Điều này có thể bao gồm các từ khóa, tên sản phẩm và bất kỳ thứ gì khác sẽ được in đậm.
- Sử dụng `` cho văn bản có **nhấn mạnh** –nói cách khác, văn bản sẽ có một biến dạng khác nếu đọc to.
- Sử dụng `<i>` cho văn bản cần được trình bày bằng chữ nghiêng nhưng không có thêm phần nhấn mạnh. Điều này có thể bao gồm các từ nước ngoài, thuật ngữ kỹ thuật và bất kỳ thứ gì khác mà bạn đặt in nghiêng khi in.

Và đây là một đoạn mã đánh dấu sử dụng cả bốn yếu tố này theo cách thích hợp:

```
<strong>Tin nóng!</strong> Đang có chương trình giảm giá kẹo <i>leche quemada</i> tại nhà hàng El Azul. Đểng chậm trễ, vì khi chiếc kẹo cuối cùng hết, nó sẽ <em>hết</em>.
```

Trong trình duyệt, văn bản trông như thế này:

Tin nóng hổi! Đang có đợt giảm giá leche quemada kẹo tại nhà hàng El Azul . Đểng trì hoãn, vì khi chiếc kẹo cuối cùng đã hết, nó sẽ Không còn.

Một số nhà phát triển web sẽ tuân theo các quy tắc có mục đích tốt của HTML, trong khi những người khác chỉ tuân theo các yếu tố quen thuộc nhất để định dạng in đậm và in nghiêng.

yếu tố tinh chỉnh

HTML5 cũng thay đổi quy tắc của một số yếu tố. Thông thường, những thay đổi này là những chi tiết nhỏ mà chỉ những người giỏi HTML mới nhận thấy, nhưng đôi khi chúng có tác động sâu hơn.

Một ví dụ là phần tử `<address>` hiếm khi được sử dụng, không phù hợp (mặc dù có tên) cho các địa chỉ bưu chính. Thay vào đó, phần tử `<address>` có mục đích hẹp là cung cấp thông tin liên hệ cho người tạo tài liệu HTML, thường là địa chỉ email hoặc liên kết trang web:

```
Trang web của chúng tôi được quản lý
bởi: <address>

<a href="mailto:jsolo@mysite.com">John Solo</a>, <a
href="mailto:lcheng@mysite.com">Lisa Cheng</a> và <a href="mailto:
rpavane@mysite.com">Ryan Pavane</a>. </địa chỉ>
```

Phần tử `<cite>` cũng đã thay đổi. Nó vẫn có thể được sử dụng để trích dẫn một số tác phẩm (ví dụ: một câu chuyện, bài báo hoặc chương trình truyền hình), như sau:

```
<p>Charles Dickens đã viết <cite>Câu chuyện về hai thành phố</cite>.</p>
```

Tuy nhiên, không được phép sử dụng `<cite>` để đánh dấu tên của một người. Sự hạn chế này hóa ra lại gây tranh cãi một cách đáng ngạc nhiên, bởi vì trước đây việc sử dụng này đã được cho phép. Một số nhà phát triển web chuyên gia đang kêu gọi mọi người bỏ qua quy tắc `<cite>` mới, điều này hơi kỳ lạ, bởi vì bạn có thể dành cả đời để chỉnh sửa các trang web mà không bao giờ vấp phải yếu tố `<cite>` trong cuộc sống thực.

Một điều chỉnh quan trọng hơn ảnh hưởng đến phần tử `<a>` để tạo liên kết. Các phiên bản HTML trước đây đã cho phép phần tử `<a>` giữ văn bản có thể nhấp hoặc hình ảnh có thể nhấp. Trong HTML5, phần tử `<a>` cho phép mọi thứ và mọi thứ, có nghĩa là hoàn toàn chấp nhận được việc nhét toàn bộ đoạn văn vào đó, cùng với danh sách, hình ảnh, v.v.

(Nếu làm như vậy, bạn sẽ thấy rằng tất cả văn bản bên trong chuyển sang màu xanh lam và được gạch chân, đồng thời tắt cả hình ảnh bên trong đường viền màu xanh lam nổi bật.) Các trình duyệt web đã hỗ trợ hành vi này trong nhiều năm, nhưng chỉ HTML5 mới biến nó thành chính thức, mặc dù không phải vậy. cực kỳ hữu ích, một phần của tiêu chuẩn HTML.

Ngoài ra còn có một số tính chỉnh chưa hoạt động-trong bất kỳ trình duyệt nào. Ví dụ: phần tử `` (dành cho danh sách có thứ tự) hiện có thuộc tính đảo ngược mà bạn có thể đặt để đếm ngược (về 1 hoặc về bất kỳ giá trị bắt đầu nào bạn đặt với thuộc tính bắt đầu), nhưng hiện tại chỉ có hai các trình duyệt nhận ra cài đặt này-Chrome và Safari.

Bạn sẽ tìm hiểu về một vài chỉnh sửa khác khi bạn tìm hiểu qua cuốn sách này.

Các yếu tố tiêu chuẩn hóa

HTML5 cũng thêm hỗ trợ cho một số thành phần đã được hỗ trợ nhưng không được chào đón chính thức bằng ngôn ngữ HTML hoặc XHTML. Một trong những ví dụ nổi tiếng nhất là `<embed>`, được sử dụng trên toàn bộ Web như một cách đa năng để đưa một trình cắm vào một trang.

Một ví dụ kỳ lạ hơn là `<wbr>`, biểu thị dấu ngắt từ tùy chọn-nói cách khác, một nơi mà trình duyệt có thể tách một dòng nếu từ đó quá dài để vừa trong vùng chứa của nó:

```
<p>Nhiều nhà ngôn ngữ học vẫn không tin rằng
<b>supercali<wbr>fragilistic<wbr>expialidocious</b> thực sự là một từ.</p>
```

Phần tử `<wbr>` rất hữu ích khi bạn có các tên dài (đôi khi được thấy trong thuật ngữ kết hợp chương trình) ở những vị trí nhỏ, chẳng hạn như các ô của bảng hoặc các hộp nhỏ. Ngay cả khi trình duyệt hỗ trợ `<wbr>`, nó sẽ phá vỡ từ chỉ khi nó không vừa với không gian có sẵn. Trong ví dụ trước, điều đó có nghĩa là trình duyệt có thể hiển thị từ theo một trong các cách sau:

The diagram illustrates three different ways a browser might render the provided HTML code, based on its word-wrapping logic:

- Example 1 (Top Box):** Shows the text split into two lines at the first `<wbr>` character, resulting in "Nhiều nhà ngôn ngữ học vẫn không tin rằng" on the first line and "supercalifragilisticexpialidocious thực sự là một từ." on the second line.
- Example 2 (Middle Box):** Shows the text split into three lines. It wraps at the first `<wbr>` character, then at the second, resulting in "Nhiều nhà ngôn ngữ học vẫn không tin rằng" on the first line, "supercalifragilistic" on the second line, and "expialidocious thực sự là một từ." on the third line.
- Example 3 (Bottom Box):** Shows the text as a single long line, indicating that the browser did not perform any word wrapping and instead kept all words together.

Phản tử <wbr> có sự tương đồng tự nhiên với phản tử <nobr>, ngăn không cho văn bản bị ngắt quãng cho dù không gian có sẵn có hép đến đâu. Tuy nhiên, HTML5 coi <nobr> đã lỗi thời và khuyên tất cả các nhà phát triển web có lòng tự trọng nên tránh sử dụng nó. Thay vào đó, bạn có thể nhận được hiệu ứng tương tự bằng cách thêm thuộc tính khoảng trắng vào biểu định kiểu của mình và đặt nó thành nowrap.

Sử dụng HTML5 ngay hôm nay

Trước khi cam kết sử dụng HTML5, bạn cần biết nó hoạt động tốt như thế nào với các trình duyệt mà khách truy cập của bạn có thể sẽ sử dụng. Xét cho cùng, điều cuối cùng mà bất kỳ nhà phát triển web nào muốn là một trang mới sáng bóng sẽ sụp đổ thành một mớ lộn xộn đánh dấu và lỗi tập lệnh khi nó gặp một trình duyệt cổ điển.

Trong giây lát, bạn sẽ học cách nghiên cứu các tính năng HTML5 cụ thể để tìm ra trình duyệt nào hỗ trợ chúng và kiểm tra số liệu thống kê sử dụng trình duyệt để tìm ra phần nào đối tượng của bạn đáp ứng tiêu chuẩn. Nhưng trước khi đi sâu vào các chi tiết nhỏ, đây là tổng quan về trạng thái hỗ trợ HTML5 hiện tại:

- Nếu khách truy cập của bạn sử dụng Google Chrome hoặc Mozilla Firefox phổ biến, họ sẽ ổn thôi. Cả hai trình duyệt không chỉ hỗ trợ phần lớn HTML5 trong vài năm mà còn được thiết kế để tự động cập nhật. Điều đó có nghĩa là bạn khó có thể tìm thấy phiên bản cũ của Chrome hoặc Firefox ngoài tự nhiên.
- Nếu khách truy cập của bạn sử dụng Safari hoặc Opera, có thể bạn vẫn đang ở trên vùng đất an toàn. Một lần nữa, các trình duyệt này đã hỗ trợ HTML5 tốt trong vài năm và hiếm khi thấy các phiên bản cũ.
- Nếu khách truy cập của bạn sử dụng máy tính bảng hoặc điện thoại thông minh, bạn có thể gặp phải một số hạn chế với một số tính năng nhất định, như bạn sẽ tìm hiểu trong suốt cuốn sách này. Tuy nhiên, các trình duyệt di động trên tất cả các tiện ích hỗ trợ web ngày nay đã được tạo ra với HTML5. Điều đó có nghĩa là các trang của bạn có thể có một vài trực tiếp, không phải là một chương trình kinh dị.
- Nếu khách truy cập của bạn sử dụng phiên bản Internet Explorer cũ hơn-tức là bất kỳ phiên bản nào trước IE 10-hầu hết các tính năng HTML5 đều hoạt động. Đây là lý do Windows 8 và phiên bản mới nhất của phông nền Windows bao gồm các phiên bản cũ của Internet Explorer. Tệ hơn nữa, nhiều phiên bản Windows cũ không cho phép người dùng của họ nâng cấp lên phiên bản IE hiện đại, hỗ trợ HTML5. Ví dụ, Windows Vista bị giới hạn ở IE 9. Windows XP cũ kỹ đến khó tin (nhưng vẫn phổ biến) bị mắc kẹt với IE 8.

Không, đó không phải là kế hoạch độc ác của Microsoft nhằm phá vỡ Web-chỉ là các phiên bản IE mới hơn được thiết kế với phần cứng máy tính mới hơn. Phần mềm mới này sẽ không hoạt động trên các máy cũ. Những người có phiên bản Windows cũ có thể sử dụng một trình duyệt thay thế như Firefox, mặc dù họ có thể không biết cách cài đặt nó hoặc có thể không được phép thực hiện những thay đổi như vậy đối với máy tính của công ty.

GHI CHÚ

Mặc dù các phiên bản Internet Explorer thực sự cũ-như IE 6 và IE 7-cuối cùng đã biến mất khỏi hiện trường, nhưng IE 8 và IE 9 có vấn đề vẫn chiếm hơn 10% tổng lưu lượng truy cập Web (tại thời điểm viết bài này).

Và bởi vì không bao giờ ổn khi buộc một trong mười khách truy cập trang web phải chịu thiệt hại, bạn sẽ cần nghĩ về các giải pháp thay thế cho hầu hết các tính năng HTML5-it nhất là trong tương lai gần.

LÊN ĐẾN TỐC ĐỘ

Xử lý các trình duyệt cũ

Trong vài năm tới, một số trình duyệt của khách truy cập sẽ không hỗ trợ tất cả các tính năng HTML5 mà bạn muốn sử dụng. Đó là một thực tế của cuộc sống. Nhưng nó không cần phải ngăn bạn sử dụng các tính năng này, nếu bạn sẵn sàng nỗ lực hơn một chút. Có hai chiến lược cơ bản bạn có thể sử dụng:

- Xuống cấp nhẹ nhàng. Mỗi khi, khi một tính năng không hoạt động, thì đó không phải là trình chiếu. Ví dụ: phần tử `<video>` mới của HTML5 có cơ chế dự phòng cho phép bạn cung cấp thứ gì đó khác cho các trình duyệt cũ hơn, chẳng hạn như trình phát video sử dụng trình cảm Flash. (Việc cung cấp một thông báo lỗi hơi thô lỗ và chắc chắn không phải là một ví dụ về việc xuống cấp một cách duyên dáng.)
- Trang của bạn cũng có thể xuống cấp một cách duyên dáng bằng cách bỏ qua những chi tiết rườm rà không cần thiết, chẳng hạn như một số tính năng của biểu mẫu web (như văn bản giữ chỗ) và một số thuộc tính định dạng từ CSS3 (như góc bo tròn và đổ bóng). Hoặc, bạn có thể viết JavaScript của riêng mình

mã kiểm tra xem trình duyệt hiện tại có hỗ trợ tính năng bạn muốn sử dụng hay không (sử dụng công cụ như Modernizr). Nếu trình duyệt không vượt qua được bài kiểm tra, mã của bạn có thể hiển thị nội dung khác hoặc sử dụng cách tiếp cận kém hấp dẫn hơn.

- Sử dụng giải pháp thay thế JavaScript. Nhiều tính năng mới của HTML5 được lấy cảm hứng từ những thứ mà các nhà phát triển web đang thực hiện một cách khó khăn. Do đó, không có gì ngạc nhiên khi bạn có thể sao chép nhiều tính năng của HTML5 bằng thư viện JavaScript tốt (hoặc, trong trường hợp xấu nhất, bằng cách viết một lượng lớn JavaScript tùy chỉnh của riêng bạn). Tạo các giải pháp thay thế JavaScript có thể tốn nhiều công sức, nhưng có hàng trăm giải pháp tốt (và không tốt lắm) có sẵn miễn phí trên Web mà bạn có thể đưa vào các trang của mình khi cần. Những cái phức tạp hơn được gọi là polyfill (trang 35).

Cách tìm yêu cầu trình duyệt cho bất kỳ HTML5 nào
Tính năng

Những người có quyết định cuối cùng về lượng HTML5 bạn sử dụng là nhà cung cấp trình duyệt. Nếu chúng không hỗ trợ một tính năng nào đó, thì việc cố gắng sử dụng nó cũng chẳng ích gì, bất kể tiêu chuẩn có nói gì đi chăng nữa. Ngày nay, có bốn hoặc năm trình duyệt chính (không bao gồm các biến thể di động chạy trên các thiết bị kết nối web như điện thoại thông minh và máy tính bảng). Một nhà phát triển web đơn lẻ không có cơ hội thử nghiệm từng tính năng tiềm năng trên mọi trình duyệt—chưa kể đến việc đánh giá hỗ trợ trong các phiên bản cũ hơn vẫn được sử dụng rộng rãi.

May mắn thay, có một trang web khéo léo tên là "Tôi có thể sử dụng không" có thể giúp bạn. Nó nêu chi tiết hỗ trợ HTML5 được tìm thấy trong trình duyệt chính. Trên hết, nó cho phép bạn tập trung vào chính xác các tính năng bạn cần. Đây là cách nó hoạt động:

- Trò trình duyệt của bạn tới <http://caniuse.com>.

Trang chính có một loạt các liên kết được nhóm thành các danh mục, như CSS, HTML5, v.v.

2. Chọn tính năng bạn muốn nghiên cứu.

Cách nhanh nhất để tìm một tính năng là nhập tên của nó vào hộp Tìm kiếm gần đầu trang.

Hoặc, bạn có thể duyệt đến tính năng này bằng cách nhấp vào một trong các liên kết trên trang đầu tiên.

Nhóm HTML5 có một tập hợp các liên kết được coi là một phần của tiêu chuẩn HTML5 cốt lõi; nhóm API JS có các liên kết dành cho các tính năng dựa trên JavaScript bắt đầu như một phần của HTML5 nhưng sau đó đã bị tách ra; nhóm CSS có các liên kết cho các tính năng tạo kiểu là một phần của CSS3; và như thế.

TIẾN BOA

Nếu muốn, bạn có thể xem tất cả các bảng hỗ trợ cho mọi tính năng trong một nhóm cùng một lúc. Nhấp vào tiêu đề nhóm (như HTML5 hoặc JS API), bản thân nó là một liên kết.

3. Kiểm tra kết quả của bạn (Hình 1-4).

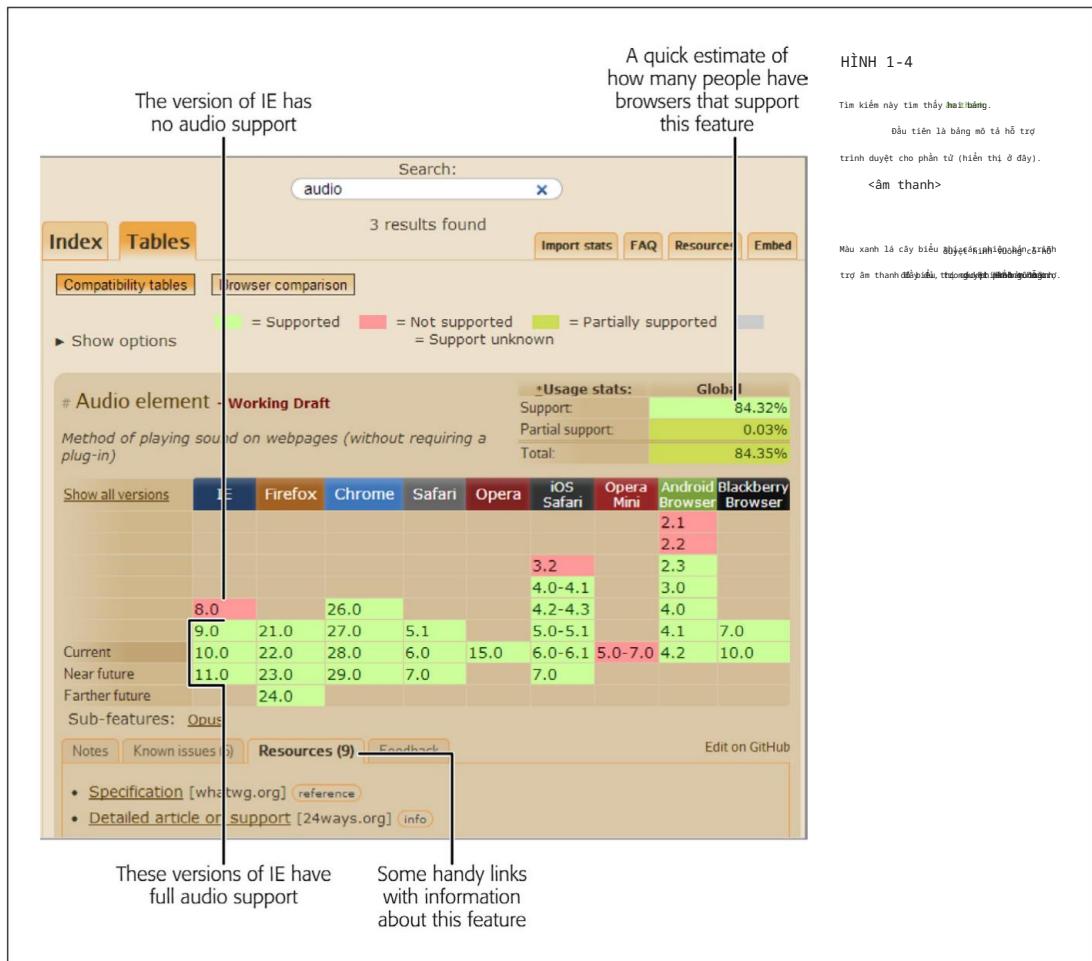
Mỗi bảng tính năng hiển thị một lưới các phiên bản trình duyệt khác nhau. Các bảng biểu thị hỗ trợ bảng màu của ô, có thể là màu đỏ (không hỗ trợ), xanh lục sáng (hỗ trợ đầy đủ), xanh ô liu (hỗ trợ một phần) hoặc xám (không xác định, thường là do phiên bản trình duyệt này vẫn đang được phát triển và tính năng này chưa được thêm vào).

4. Tùy chọn, chọn các trình duyệt khác nhau để đặt dưới kính hiển vi.

Thông thường, bảng hỗ trợ bao gồm các phiên bản mới nhất của các trình duyệt phổ biến nhất. Tuy nhiên, bạn có thể điều chỉnh bảng để nó bao gồm thông tin hỗ trợ cho các trình duyệt khác có thể quan trọng đối với bạn-chẳng hạn như IE 7 cũ hoặc một trình duyệt dành cho thiết bị di động chuyên biệt như Firefox dành cho Android.

Để chọn trình duyệt nào xuất hiện trong bảng, hãy bắt đầu bằng cách nhấp vào liên kết "Hiển thị tùy chọn" phía trên bảng. Một danh sách các trình duyệt xuất hiện và bạn có thể chọn các trình duyệt mình muốn bằng cách thêm dấu kiểm bên cạnh tên của chúng. Bạn cũng có thể điều chỉnh thanh trượt "Phiên bản được hiển thị", thanh trượt này hoạt động như một loại mức độ phổ biến vượt qua giới hạn cũ-hạ thấp thanh trượt này để bao gồm các phiên bản trình duyệt cũ ít được sử dụng hơn.

Ngoài ra, hãy nhấp vào liên kết "Hiển thị tất cả các phiên bản" ở góc trên cùng bên trái của bảng để xem thông tin về khả năng tương thích của trình duyệt với các tính năng sẽ hiện được sử dụng có trong chế độ duyệt những ngày đen tối của Firefox 2 và IE 5.5.



Cách tìm ra trình duyệt nào trên web

Làm thế nào để bạn biết phiên bản trình duyệt mà bạn cần quan tâm? Số liệu thống kê về việc sử dụng trình duyệt có thể cho bạn biết phần nào đối tượng của bạn có trình duyệt hỗ trợ các tính năng mà bạn định sử dụng. Một nơi tốt để có được một bức tranh toàn cảnh về tất cả các trình duyệt trên Web là GlobalStats, một trang web theo dõi phổ biến. Đây là cách sử dụng nó:

1. Duyệt đến <http://gs.statcounter.com>.

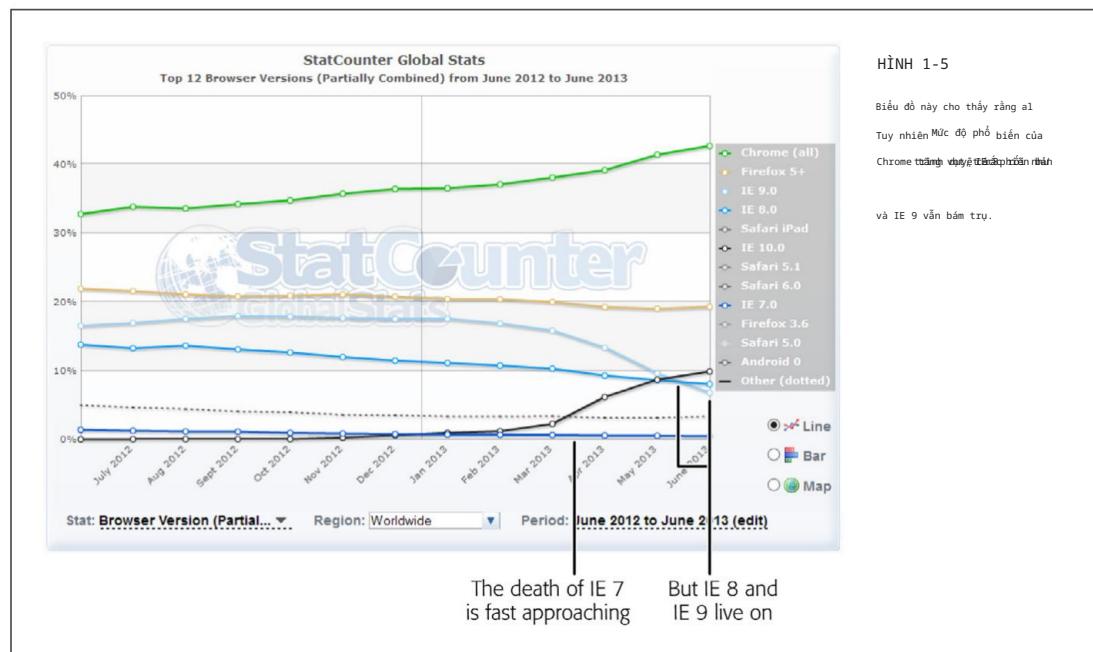
Trên trang GlobalStats, bạn sẽ thấy biểu đồ đường hiển thị các trình duyệt phổ biến nhất trong năm trước. Tuy nhiên, biểu đồ này không bao gồm thông tin phiên bản, vì vậy nó không cho bạn biết có bao nhiêu người đang lướt các phiên bản Internet Explorer có vấn đề (các phiên bản trước IE 10). Để có được thông tin này, bạn cần điều chỉnh một cài đặt khác.

2. Tìm cài đặt Thống kê (dưới biểu đồ) và chọn "Phiên bản trình duyệt (Kết hợp một phần)".

Lựa chọn này cho phép bạn xem xét chỉ trình duyệt nào đang được sử dụng, mà cả phiên bản trình duyệt nào. Việc kết hợp một phần yêu cầu GlobalStats nhóm để tập hợp các trình duyệt được cập nhật nhanh chóng, như Chrome và Firefox (Hình 1-5), vì vậy biểu đồ của bạn không bị lộn xộn với hàng chục dòng bổ sung.

3. Tùy ý thay đổi vùng địa lý trong hộp Vùng.

Cài đặt tiêu chuẩn là Toàn cầu, hiển thị số liệu thống kê trình duyệt được chọn lọc từ khắp nơi trên thế giới. Tuy nhiên, bạn có thể sinh sống tại một quốc gia cụ thể (như Bolivia) hoặc lục địa (như Bắc Mỹ).



4. Tùy chọn, nhấp vào văn bản bên cạnh cài đặt Khoảng thời gian để chọn một ngày khác phạm vi.

Thông thường, bạn sẽ thấy xu hướng sử dụng trình duyệt trong cả năm nhưng bạn có thể chọn tập trung vào một phạm vi nhỏ hơn, chẳng hạn như ba tháng qua.

5. Tùy chọn, thay đổi loại biểu đồ bằng cách sử dụng các nút tùy chọn chỉ để bên phải của hộp biểu đồ.

Chọn tùy chọn Đường để xem biểu đồ đường cho biết xu hướng sử dụng trình duyệt theo thời gian. Chọn Thanh để xem biểu đồ thanh hiển thị ảnh chụp nhanh về tình hình hiện tại. Hoặc, chọn Bản đồ để xem bản đồ được mã hóa màu hiển thị các quốc gia nơi các trình duyệt khác nhau chiếm ưu thế.

GlobalStats tổng hợp số liệu thống kê hàng ngày bằng cách sử dụng mã theo dõi hiện có trên hàng triệu trang web. Và mặc dù đó là một số lượng lớn các trang và một lượng dữ liệu khổng lồ, nhưng nó vẫn chỉ là một phần nhỏ trong tổng số trang Web, điều đó có nghĩa là bạn không nhất thiết phải cho rằng khách truy cập trang web của mình sẽ sử dụng cùng một trình duyệt.

Hơn nữa, kết quả chia sẻ trình duyệt thay đổi tùy thuộc vào quốc gia của người lượt web và loại trang web. Ví dụ, ở Đức, Firefox là trình duyệt hàng đầu với hơn 40% người lượt web. Và trên trang web TechCrunch (một trang tin tức nổi tiếng dành cho những người mê máy tính), các phiên bản cũ của Internet Explorer rất hiếm. Vì vậy, nếu bạn muốn thiết kế một trang web phù hợp với mục đích của mình, thì bạn nên xem lại các thống kê web do các trang của chính bạn tạo ra. (Và nếu bạn chưa sử dụng dịch vụ theo dõi web cho trang web của mình, hãy xem Google Analytics hàng đầu và hoàn toàn miễn phí tại www.google.com/analytics.)

Phát hiện tính năng với Modernizr

Phát hiện tính năng là một chiến lược để xử lý các tính năng không được hỗ trợ bởi tất cả các trình duyệt truy cập vào trang web của bạn. Mẫu điển hình là như sau: Trang của bạn tải và chạy một đoạn mã JavaScript để kiểm tra xem một tính năng cụ thể có khả dụng hay không.

Sau đó, bạn có thể cảnh báo người dùng (tùy chọn yêu nhất), quay trở lại phiên bản trang kém ứng dụng một chút (tốt hơn) hoặc triển khai giải pháp thay thế sao chép tính năng HTML5 mà bạn muốn sử dụng (tốt nhất).

Thật không may, bởi vì HTML5, về bản chất, là một tập hợp lỏng lẻo các tiêu chuẩn liên quan, nên không có bài kiểm tra hỗ trợ HTML5 nào. Thay vào đó, bạn cần hàng chục thử nghiệm khác nhau để kiểm tra hàng chục tính năng khác nhau—và đôi khi thậm chí để kiểm tra xem một tính năng cụ thể phản của một tính năng được hỗ trợ, tính năng này trở nên xấu xí nhanh chóng.

Kiểm tra hỗ trợ thường liên quan đến việc tìm kiếm một thuộc tính trên một đối tượng lập trình hoặc tạo một đối tượng và cố gắng sử dụng nó theo một cách nhất định. Nhưng hãy suy nghĩ kỹ trước khi bạn viết loại mã kiểm tra tính năng này, vì rất dễ làm sai. Ví dụ: mã kiểm tra tính năng của bạn có thể không hoạt động trên một số trình duyệt nhất định vì một lý do khó hiểu nào đó hoặc lý do khác hoặc nhanh chóng trở nên lỗi thời. Thay vào đó, hãy cân nhắc sử dụng Modernizr (), một công cụ nhỏ, được cấp phép Miễn phí. Để kiểm tra tính năng hỗ trợ một ứng dụng HTML5, hãy dán phím F12 để kiểm tra tính năng hỗ trợ ứng dụng các tính năng CSS3 mới, mà bạn sẽ thấy ở trang 180.

Đây là cách sử dụng Modernizr trên một trong các trang web của bạn:

- Truy cập trang tải xuống Modernizr tại <http://Modernizr.com/download>.

Hãy tìm liên kết "Phiên bản phát triển", trỏ đến tệp JavaScript tất cả trong một mới nhất cho Modernizr.

- Nhấp chuột phải vào liên kết "Phiên bản phát triển" và chọn "Lưu liên kết dưới dạng" hoặc "Lưu mục tiêu dưới dạng".

Cả hai lệnh đều giống nhau-từ ngữ chỉ phụ thuộc vào trình duyệt bạn đang sử dụng.

- Chọn một nơi trên máy tính của bạn để lưu tệp và nhấp vào Lưu.

Tệp JavaScript có tên khác. <Modernizr-latest.js>, trừ khi bạn chọn một cái gì đó

- Khi bạn đã sẵn sàng sử dụng Modernizr, hãy đặt tệp đó vào cùng thư mục với trang web của bạn.

Hoặc, đặt nó trong một thư mục con và sửa đổi đường dẫn trong tham chiếu JavaScript cho phù hợp.

- Thêm tham chiếu đến tệp JavaScript trong phần <head> của trang web của bạn.

Đây là một ví dụ về giao diện đánh dấu của bạn, giả sử tệp nằm trong cùng thư mục với trang web của bạn: [hiện đại hóa mới nhất.js](#)

```
<đầu>
<meta charset="utf-8">
<title>Phát hiện tính năng HTML5</title>
<script src="modernizr-latest.js"></script>
...
</head>
```

Bây giờ, khi trang của bạn tải, tập lệnh Modernizr sẽ chạy. Nó kiểm tra vài chức năng mới chỉ trong một phần nghìn giây, sau đó tạo một đối tượng JavaScript có tên là modernizr chứa các kết quả. Bạn có thể kiểm tra các thuộc tính của đối tượng này để kiểm tra sự hỗ trợ của trình duyệt đối với một tính năng cụ thể.

TIỀN BOA

Để biết danh sách đầy đủ các tính năng mà Modernizr kiểm tra và mã JavaScript mà bạn cần kiểm tra từng tính năng, hãy tham khảo tài liệu tại <http://Modernizr.com/docs>.

- Viết một số mã kịch bản để kiểm tra tính năng bạn muốn và sau đó thực hiện ra các hành động thích hợp.

Ví dụ: đây là cách bạn có thể kiểm tra xem Modernizr có hỗ trợ tính năng kéo và thả HTML5 hay không và hiển thị kết quả trên trang:

```

<!DOCTYPE html>
<html lang="vi">
<head>
    <meta charset="utf-8">
    <title>Phát hiện tính năng HTML5</title>
    <script src="modernizr-latest.js"></script> <!--
head>

<body>
    <p>Phán quyết là... <span id="result"></p>

    <kịch bản>
    // Tìm phần tử trên trang (kết quả được đặt tên) nơi bạn có thể hiển
    thị // kết quả. var result = document.getElementById("kết quả"); if
    (Modernizr.draganddrop) { result.innerHTML = "Hãy vui mừng! Trình
    duyệt của bạn hỗ trợ kéo và thả.";

}

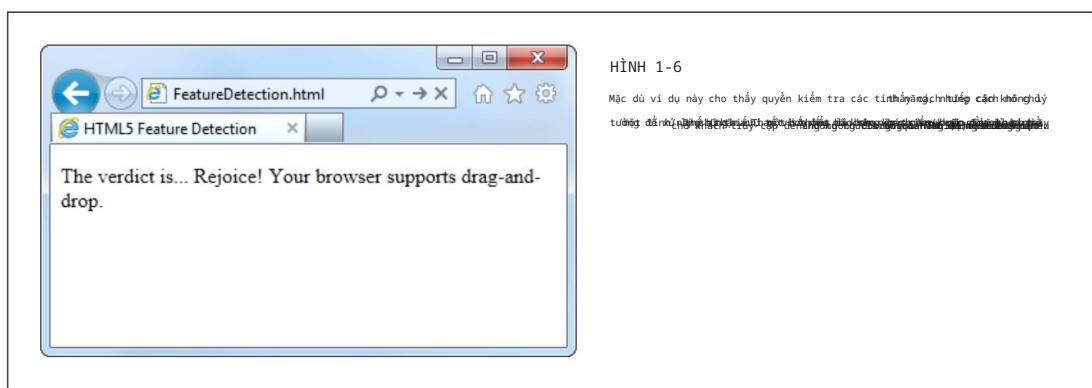
other { result.innerHTML = "Trình duyệt yêu của bạn không hỗ trợ kéo và thả.";
}

</script>
</body>

</html>

```

Hình 1-6 cho thấy kết quả.


TIẾN BỘ

Ví dụ này sử dụng các kỹ thuật JavaScript cơ bản và lâu đời—tìm kiếm một phần tử theo ID và thay đổi nội dung của phần tử đó. Nếu thấy nó hơi khó hiểu, bạn có thể xem lại phần đánh giá JavaScript trong Phụ lục B, "JavaScript: Bộ não của trang của bạn".

Tập lệnh Modernizr đầy đủ hơi cồng kềnh. Nó nhằm mục đích thử nghiệm trong khi bạn vẫn đang làm việc trên trang web của mình. Khi bạn đã hoàn tất quá trình phát triển và sẵn sàng đưa vào hoạt động, bạn có thể tạo một phiên bản rút gọn của tập lệnh Modernizr để chỉ kiểm tra các tính năng bạn cần. Để làm như vậy, hãy truy cập trang tải xuống tại .

<http://Modernizr.com/dload> Nhưng lần này, thay vì sử dụng liên kết "Phiên bản phát triển", hãy kiểm tra các hộp kiểm bên dưới. Nhấp vào những cái tương ứng với các tính năng bạn cần phát hiện. Cuối cùng, nhấp vào nút Tạo để tạo phiên bản Modernizr tùy chỉnh của riêng bạn, sau đó nhấp vào nút Tải xuống để lưu vào máy tính của bạn (Hình 1-7).

HÌNH 1-7

Bạn sáp tài xuống để xây dựng hiện đại a Phong tục nó có thể phát hiện hỗ trợ cho HTML5 và các công nghệ mới

Bản dựng Modernizr này sẽ không bao gồm

Tính năng “Filling” với Polyfills Modernizr

giúp bạn phát hiện các lỗ hổng trong hỗ trợ trình duyệt. Nó thông báo cho bạn khi một tính năng không hoạt động. Tuy nhiên, nó không làm bất cứ điều gì để khắc phục những vấn đề này. Đó là **polyfill** nơi bạn nên đến. Về cơ bản, polyfill là một tập hợp hỗn hợp các kỹ thuật để lấp đầy các khoảng trống trong hỗ trợ HTML5 trên các trình duyệt cũ. Từ này được [mượn từ polyfill](#) một cách chất lượng để lấp đầy các lỗ trên vách thạch cao trước khi sơn (còn được gọi là hò dán spackling). Trong HTML5, polyfill lý tưởng là một polyfill mà bạn có thể thả vào một trang mà không cần thực hiện bất kỳ thao tác nào. Nó xử lý khả năng tương thích ngược một cách liền mạch, kín đáo, vì vậy bạn có thể làm việc với HTML5 thuần túy trong khi người khác lo lắng về các giải pháp thay thế.

Nhưng polyfill không hoàn hảo. Một số dựa vào các công nghệ khác có thể chỉ được hỗ trợ một phần. Ví dụ: một polyfill cho phép bạn mô phỏng canvas HTML5 trên các phiên bản cũ của Internet Explorer bằng cách sử dụng phần bổ trợ Silverlight. Nhưng nếu khách truy cập web không sẵn sàng hoặc không thể cài đặt Silverlight, thì bạn cần phải quay lại với thứ khác. Các polyfill khác có thể có ít tính năng hơn so với tính năng HTML5 thực hoặc hiệu suất kém hơn.

Đôi khi, cuốn sách này sẽ chỉ cho bạn một polyfill tiềm năng. Nếu bạn muốn biết thêm thông tin, bạn có thể tìm thấy thứ gần nhất có một danh mục toàn diện về các polyfill HTML5 trên GitHub tại [. Nhưng lưu ý—các polyfill khác nhau có thể có chất lượng, hiệu suất và hỗ trợ](#).

TIỀN BOA

Hãy nhớ rằng, chỉ biết rằng một polyfill tồn tại cho một tính năng HTML5 nhất định là chưa đủ. Bạn nó và kiểm tra xem nó hoạt động tốt như thế nào trên các trình duyệt cũ khác trước phải kiểm tra rủi ro khi kết hợp tính năng tương ứng nhau vào trang web của bạn.

Với các công cụ như thống kê trình duyệt, phát hiện tính năng và polyfill, bạn đã sẵn sàng suy nghĩ sâu về việc tích hợp các tính năng HTML5 vào các trang web của riêng mình. Trong chương tiếp theo, bạn sẽ thực hiện bước đầu tiên với một số thành phần HTML5 có thể hoạt động trong các trình duyệt cả mới và cũ.

Cấu trúc các trang với Yếu tố ngữ nghĩa

đáng kể. Nhưng ngạc nhiên lớn nhất không phải là Web đã thay đổi bao nhiêu, Trong ~~hàng chục~~ ~~những~~ ~~nhà~~ ~~phát~~ ~~triển~~ ~~web~~ ~~hiện~~ ~~đại~~ ~~đã~~ ~~thay~~ ~~đổi~~. Trên thực tế, các nhà phát triển web sử dụng cùng một tập hợp các yếu tố để xây dựng các trang web hiện đại ngày nay mà họ đã sử dụng để xây dựng các trang web tiền nhiệm 10 năm trước.

Một yếu tố cụ thể—`<div>` (hoặc) khiêm tốn—là nền tảng ~~đa~~~~gắn~~~~như~~ mọi trang web hiện đại. Sử dụng các phần tử `<div>`, bạn có thể khắc tài liệu HTML thành các tiêu đề, bảng điều khiển bên, thanh điều hướng, v.v. Thêm một chút định dạng CSS lành mạnh và bạn có thể biến các phần này thành các hộp có đường viền hoặc cột được tô bóng và đặt từng phần chính xác vào vị trí của nó.

Kỹ thuật `<div>-and-style` này đơn giản, mạnh mẽ và linh hoạt, nhưng không phải vậy. Khi ~~trong~~ ~~suốt~~ bạn xem đánh dấu của người khác, bạn phải nỗ lực tìm hiểu xem mỗi `<div>` đại diện cho điều gì và toàn bộ trang khớp với nhau như thế nào.

Để hiểu được tất cả, bạn cần nhảy qua nhảy lại giữa đánh dấu, biểu định kiểu và trang được hiển thị trong trình duyệt. Và bạn sẽ phải đổi mặt với sự nhầm lẫn này mỗi khi bạn mở trang nữa vời của bất kỳ ai khác, mặc dù bạn có thể đang sử dụng các kỹ thuật thiết kế tương tự trong các trang web của riêng mình.

Tình huống này khiến mọi người phải suy nghĩ. Nếu có cách thay thế `<div>` bằng thứ gì đó tốt hơn thì sao? Một cái gì đó hoạt động giống như `<div>`, nhưng truyền đạt nhiều ý nghĩa hơn một chút. Thứ gì đó có thể giúp tách thanh bên khỏi tiêu đề và thanh quảng cáo khỏi menu. HTML5 hoàn thành ước mơ này với một tập hợp các thành phần mới để cấu trúc các trang.

TIỀN BOA

Nếu kỹ năng CSS của bạn kém đến mức bạn cần tiêm phòng uốn ván trước khi có thể mở một biểu định kiểu, thì bạn chưa hoàn toàn sẵn sàng cho chương này. May mắn thay, Phụ lục A, "CSS cơ bản," có phần giới thiệu cô đọng về các nguyên tắc cơ bản.

Giới thiệu các yếu tố ngữ nghĩa

Để cải thiện cấu trúc của các trang web của bạn, bạn cần có HTML5. Các phần **yếu tố ngữ nghĩa** . từ này cung cấp thêm ý nghĩa cho nội dung mà chúng chưa đựng. Ví dụ: phần tử `<time>` mới đánh dấu ngày hoặc giờ hợp lệ trong trang web của bạn. Đây là một ví dụ về phần tử `<time>` đơn giản nhất:

Bắt đầu đăng ký vào `<time>25-11-2014</time>`.

Và đây là những gì ai đó nhìn thấy khi xem trang:

Đăng ký bắt đầu vào ngày 25-11-2014.

Điều quan trọng cần hiểu là phần tử `<time>` không có bất kỳ định dạng tích hợp sẵn nào. Trên thực tế, người đọc trang web không có cách nào biết rằng có một yếu tố bổ sung bao quanh ngày tháng. Bạn có thể thêm định dạng của riêng mình vào phần tử `<time>` bằng biểu định kiểu, nhưng theo mặc định, vẫn bản bên trong phần tử `<time>` có thể phân biệt được với văn bản thông thường.

Phần tử `<time>` được thiết kế để bọc một phần thông tin. Tuy nhiên, hầu hết các yếu tố ngữ nghĩa của HTML5 được thiết kế để xác định các phần nội dung lớn hơn.

Ví dụ: phần tử `<nav>` xác định một tập hợp các liên kết điều hướng. Phần tử `<footer>` bao bọc chân trang nằm ở cuối trang. Và cứ thế, cho hàng chục (hoặc hơn) phần tử mới.

GHI CHÚ

Mặc dù các yếu tố ngữ nghĩa là ít sắc sỡ nhất trong số các tính năng mới của HTML5, nhưng chúng lại là một trong những tính năng lớn nhất. Trên thực tế, phần lớn các phần tử mới trong HTML5 là các phần tử ngữ nghĩa.

Tất cả các yếu tố ngữ nghĩa đều có chung một đặc điểm: Chúng không thực sự làm gì cả. Ngược lại, ví dụ: phần tử `<video>` nhưng trình phát video có đầy đủ chức năng vào trang của bạn (trang 147). Vậy tại sao phải bận tâm sử dụng tất cả các yếu tố mới không thay đổi giao diện trang web của bạn?

Có một số lý do chính đáng:

- Chính sửa và bảo trì dễ dàng hơn. Có thể khó giải thích đánh dấu trong trang HTML truyền thống. Để hiểu bối cảnh thẻ và tầm quan trọng của các phần khác nhau, bạn thường cần tìm kiếm biểu định kiểu của trang web. Nhưng bằng cách sử dụng các phần tử ngữ nghĩa của HTML5, bạn cung cấp thêm thông tin cấu trúc trong phần đánh dấu. Điều đó làm cho cuộc sống của bạn dễ dàng hơn khi bạn cần chỉnh sửa trang sau nhiều tháng và điều đó thậm chí còn quan trọng hơn nếu người khác cần sửa lại công việc của bạn.

- Khả năng tiếp cận. Một trong những chủ đề chính của thiết kế web hiện đại là làm cho truy cập [trang-nghĩa](#) là các trang mà mọi người có thể điều hướng bằng cách sử dụng trình đọc màn hình và các công cụ hỗ trợ khác. Các công cụ truy cập web HTML5 có thể cung cấp trải nghiệm duyệt web tốt hơn nhiều cho khách truy cập bị khuyết tật. (Chỉ với một ví dụ, hãy tưởng tượng cách trình đọc màn hình có thể truy cập vào các phần <nav> để nhanh chóng tìm thấy các liên kết điều hướng cho một trang web.)

TIỀN BOA

Để tìm hiểu thêm về các phương pháp hay nhất về khả năng truy cập web, bạn có thể truy cập trang web WAI (Sáng kiến khả năng truy cập web) <http://www.w3.org/WAI/standards-guidelines/accessibility-principles/> (đã được sáp nhập vào trang WAI) để tìm hiểu thêm về các tiêu chí <http://tinyurl.com/6bu4pe>.

- Tối ưu hóa công cụ tìm kiếm. Các công cụ tìm kiếm như Google sử dụng mạnh mẽ [Tìm kiếm bot](#) –các chương trình tự động thu thập thông tin trên Web và tìm nạp mọi trang có thể– để quét nội dung của bạn và lập chỉ mục nội dung đó trong cơ sở dữ liệu tìm kiếm của chúng. Google càng hiểu rõ trang web của bạn thì càng có cơ hội khớp truy vấn của người tìm kiếm trên web với nội dung của bạn và cơ hội để trang web của bạn xuất hiện trong kết quả tìm kiếm của ai đó càng cao. Các bot tìm kiếm đã kiểm tra một số yếu tố ngữ nghĩa của HTML5 để thu thập thêm thông tin về các trang mà chúng đang lập chỉ mục.
- Các tính năng trong tương lai. Các trình duyệt và công cụ chỉnh sửa web mới chắc chắn sẽ tận dụng các yếu tố ngữ nghĩa. Ví dụ: một trình duyệt có thể cung cấp một phác thảo cho phép khách truy cập chuyển đến phần thích hợp trong một trang. (Trên thực tế, Chrome đã có một trình cảm ứng chính xác điều đó–xem trang 65.) Tương tự, các công cụ thiết kế web có thể bao gồm các tính năng cho phép bạn tạo hoặc chỉnh sửa menu điều hướng bằng cách quản lý nội dung bạn đã đặt trong <nav> tiết diện.

Điểm mấu chốt là: Nếu bạn có thể áp dụng các yếu tố ngữ nghĩa một cách chính xác, bạn có thể tạo các trang sạch hơn, rõ ràng hơn, sẵn sàng cho làn sóng trình duyệt và công cụ web tiếp theo. Nhưng nếu bộ não của bạn vẫn bị trói buộc với các thực hành lỗi thời của HTML truyền thống, thì tương lai có thể bỏ qua bạn.

Trang bị thêm một trang HTML truyền thống

Cách dễ nhất để tự làm quen với các yếu tố ngữ nghĩa mới–và để học cách sử dụng chúng để cấu trúc một trang –là lấy một tài liệu HTML cổ điển và thêm vào đó một số điểm hay của HTML5. Hình 2-1 cho thấy ví dụ đầu tiên mà bạn sẽ giải quyết. Đó là một trang web độc lập, đơn giản chứa một bài báo, mặc dù các loại nội dung khác (như bài đăng trên blog, mô tả sản phẩm hoặc truyện ngắn) cũng sẽ hoạt động tốt như nhau.

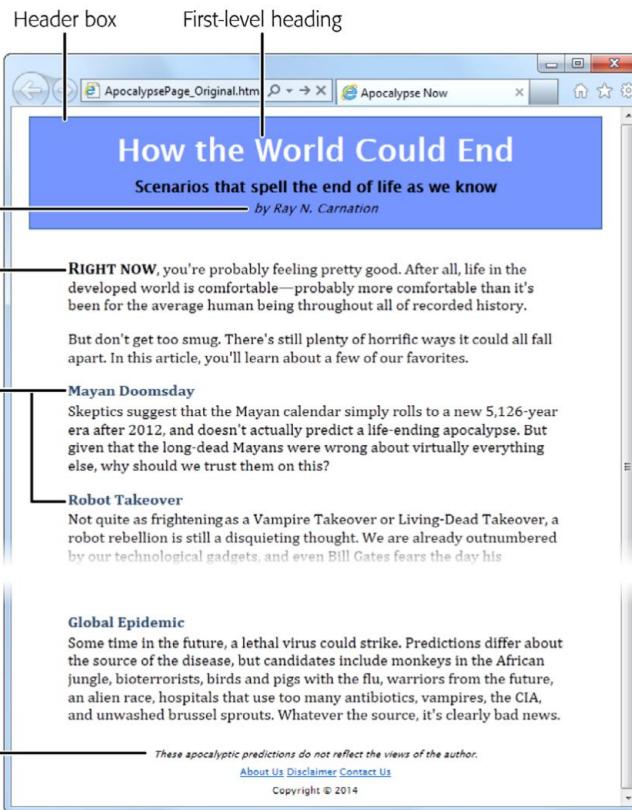
TIỀN BOA

Bạn có thể xem hoặc tải xuống ví dụ trong Hình 2-1 từ trang web dùng thử cùng với <http://prosetech.com/html5>, tất cả các ví dụ cho chương này. Đầu tiên với nếu bạn muốn tự mình sửa sang [Tạo HTML](#) hoặc nếu bạn muốn [chuyển đổi sang](#) sản phẩm cuối cùng được cải tiến HTML5.

trang bị thêm

MỘT TRUYỀN THỐNG

TRANG HTML



HÌNH 2-1

Trang HTML thông thường này có cấu trúc cơ bản giống như tài liệu.

Một biểu định kiểu được liên kết [kiết](#) [đúng](#) cấp tắt cá các

Cấu trúc trang theo cách cũ Có một số

cách để định dạng một trang giống như trong Hình 2-1. Rất may, ví dụ này sử dụng các phương pháp hay nhất về HTML, có nghĩa là phần đánh dấu không có chút logic định dạng nào. Không có phần tử **in đậm** hoặc **in nghiêng**, không có kiểu nội tuyến và chắc chắn không có gì ghê tởm bằng phần tử **** đã lỗi thời. Thay vào đó, nó là một tài liệu được định dạng gọn gàng được liên kết với một biểu định kiểu bên ngoài.

Đây là phiên bản rút gọn của phần đánh dấu, đánh dấu vị trí mà tài liệu được bổ sung vào biểu định kiểu của nó:

```
<div class="Header">
    <h1>Thế giới có thể kết thúc như thế nào</h1>
    <p class="Teaser">Kịch bản đánh dấu sự kết thúc của cuộc sống như chúng ta biết</p> <p class="Byline">của
    Ray N. Carnation</p> <p class="Byline">của Ray N. Carnation</p> </div>

    <div class="Nội dung">
        <p><span class="LeadIn">Ngay bây giờ, có lẽ bạn đang...</span> <p>...</p>
        ...
        <h2>Ngày tận thế của người Maya</h2>
        <h2> <p>Những người hoài nghi đề xuất ...</p>
        ...
        </div>

        <div class="Chân trang">
            <p class="Disclaimer">Những dự đoán về ngày tận thế này...</p> <p> <a
            href="AboutUs.html">Giới thiệu về chúng tôi</a>
            ...
            </p>
            <p>Bản quyền © 2014</p> </
        div>
```

LÊN ĐẾN TỐC ĐỘ

Những dấu chấm này (.) là gì?

Cuốn sách này không thể hiển thị cho bạn đánh dấu đầy đủ cho mọi ví dụ - ít nhất là không thể không mở rộng bản thân nó lên 12.000 trang và quét sạch toàn bộ khu rừng già. Nhưng nó cho bạn thấy cách mà chúng ta đã bắt đầu các yếu tố quan trọng của nó. Để làm điều đó, nhiều ví dụ trong cuốn sách này sử dụng dấu chấm lửng (một loạt ba dấu chấm) để chỉ ra nơi một số nội dung bị bỏ sót.

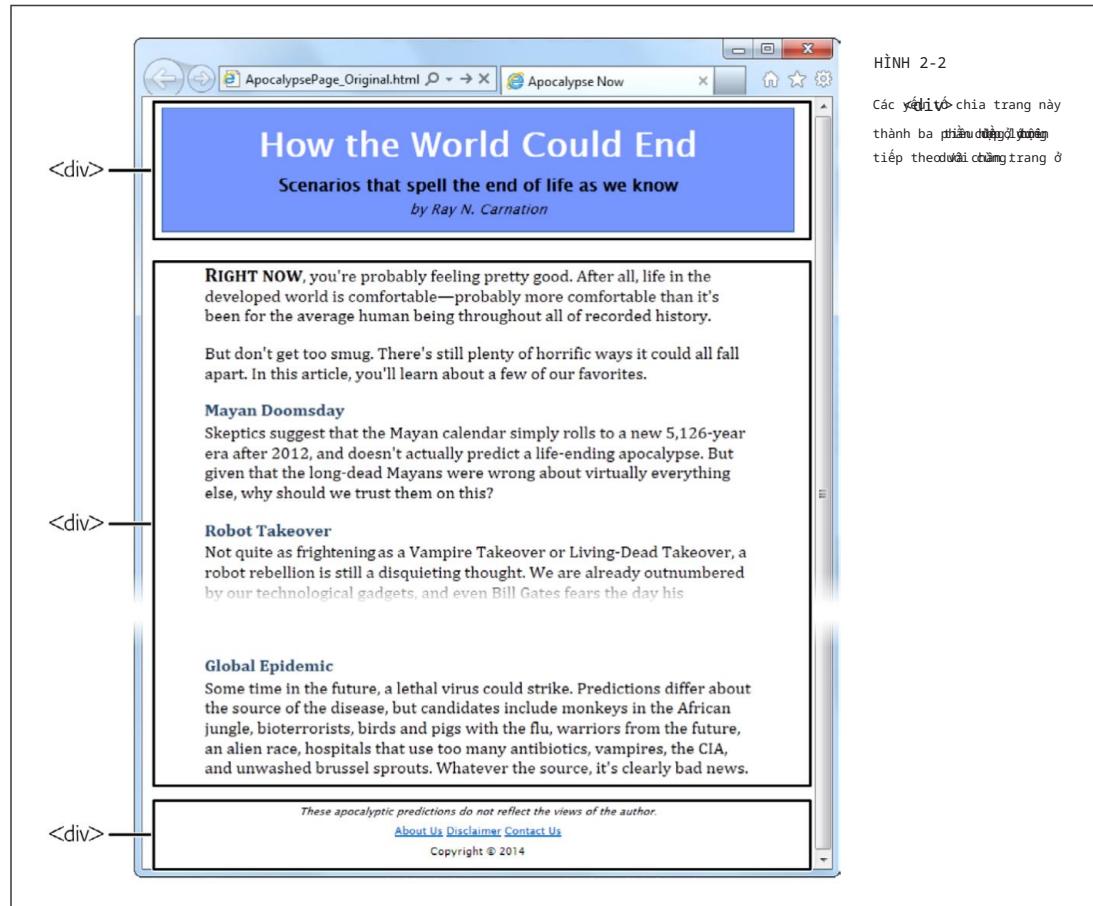
Ví dụ: hãy xem xét đánh dấu được hiển thị ở trên trang này. Nó bao gồm toàn bộ trang được hiển thị trong Hình 2-2, nhưng nó bỏ qua toàn văn của hầu hết các đoạn văn, hầu hết bài báo sau tiêu đề "Ngày tận thế của người Maya" và danh sách đầy đủ các liên kết ở cuối trang. Tuy nhiên, như bạn đã biết, bạn có thể xem kỹ từng chi tiết bằng cách kiểm tra các tệp mẫu cho chương này trên trang web dùng thử (<http://prosetech.com/html5>).

Trong một trang HTML truyền thống, được viết tốt (như trang này), hầu hết công việc được đưa vào biểu định kiểu bằng cách sử dụng các bộ chứa `<div>` và ``. `` cho phép bạn định dạng các đoạn văn bản bên trong một phần tử khác. `<div>` cho phép bạn định dạng toàn bộ các phần nội dung và nó thiết lập cấu trúc tổng thể của trang (Hình 2-2).

trang bị thêm

MỘT TRUYỀN THỐNG

TRANG HTML



Ở đây, các tác vụ định dạng biểu định kiểu rất đơn giản. Toàn bộ trang được cung cấp chiều rộng tối đa (800 pixel) để ngăn các dòng văn bản thực sự dài trên màn hình rộng. Tiêu đề được đặt trong một hộp màu xanh có viền. Nội dung được đệm ở hai bên và chân trang được căn giữa ở cuối trang.

Nhờ có `<div>`, việc định dạng trở nên dễ dàng. Ví dụ, các CSS biểu định kiểu sử dụng các quy tắc sau để định dạng hộp tiêu đề và nội dung bên trong:

```
/* Định dạng <div> đại diện cho tiêu đề (dưới dạng hộp màu xanh, có viền).
 */ .Header { màu nền: #7695FE; viền: mỏng #336699 đặc; phần đệm: 10px;
```

```

lề: 10px; căn
chỉnh văn bản: trung tâm;
}

/* Định dạng bất kỳ tiêu đề <h1> nào trong tiêu đề <div> (đó là tiêu đề bài viết).
*/ .Header h1 { lề: 0px; màu trắng; cỡ chữ: xx-lớn;

}

/* Định dạng phụ đề trong tiêu đề <div>.
*/ .Header .Teaser { lề: 0px; trọng lượng phông
chữ: đậm;

}

/* Định dạng dòng tên trong tiêu đề <div>.
*/ .Header .Byline { font-style: italic; cỡ
chữ: nhỏ; lề: 0px;

}

```

Bạn sẽ nhận thấy rằng ví dụ này sử dụng tốt các bộ chọn theo ngữ cảnh (trang 441). Ví dụ: nó sử dụng bộ chọn .Header h1 để định dạng tất cả các phần tử <h1> trong hộp tiêu đề.

TIỀN BOA Ví dụ này cũng được mô tả trong bài đánh giá CSS ở Phụ lục A, "CSS cơ bản". Nếu bạn muốn xem chi tiết các quy tắc biểu định kiểu định dạng từng phần, hãy chuyển sang trang 445.

Cấu trúc trang với HTML5 Phần tử <div>

vẫn là một yếu tố chính của thiết kế web. Đó là một bộ chứa đơn giản, đa năng mà bạn có thể sử dụng để áp dụng định dạng ở bất kỳ đâu bạn muốn trong trang web.

Hạn chế của <div> là nó không cung cấp bất kỳ thông tin nào về trang.

Khi bạn (hoặc trình duyệt hoặc công cụ thiết kế hoặc trình đọc màn hình hoặc bot tìm kiếm) bắt gặp một <div>, bạn biết rằng mình đã tìm thấy một phần riêng biệt của trang nhưng bạn không biết mục đích của phần đó.

Để cải thiện tình trạng này trong HTML5, bạn có thể thay thế một số phần tử <div> của mình bằng các phần tử ngữ nghĩa mô tả hơn. Các phần tử ngữ nghĩa hoạt động chính xác như các phần tử <div>: Chúng nhóm một khối đánh dấu, chúng không tự làm bất cứ điều gì và chúng cung cấp cho bạn một móc kiểu dáng cho phép bạn áp dụng định dạng. Tuy nhiên, họ cũng cung cấp cho trang của bạn một chút thông minh ngữ nghĩa hơn.

Đây là bản sửa đổi nhanh của bài báo được hiển thị trong Hình 2-1. Nó loại bỏ hai phần tử <div> và thêm hai phần tử ngữ nghĩa từ HTML5: <header> và <footer>.

trang bị thêm

MỘT TRUYỀN THỐNG

TRANG HTML

```

<header class="Header">
    <h1>Thế giới có thể kết thúc như thế nào</h1>
    <p class="Teaser">Kịch bản đánh dấu sự kết thúc của cuộc sống như chúng ta biết</p> <p class="Byline">của
    Ray N. Carnation</p> <p class="Byline">của Ray N. Carnation</p> </tiêu đề>

<div class="Nội dung">
    <p><span class="LeadIn">Ngay bây giờ, có lẽ bạn đang...</span> <p>...</p>

    <h2>Ngày tận thế của người Maya</h2>
    <p>Những người hoài nghi đề xuất ...</p>
    ...
</div>

<footer class="Chân trang">
    <p class="Disclaimer">Những dự đoán về ngày tận thế này....</p>
    <p>
        <a href="AboutUs.html">Giới thiệu về chúng tôi</a>
        ...
    </p>
    <p>Bản quyền © 2014</p>
</footer>
```

Trong ví dụ này, các phần tử `<header>` và `<footer>` thay thế cho các phần tử `<div>` đã có trước đó. Các nhà phát triển web đang sửa đổi một trang web lớn có thể bắt đầu bằng cách gói các phần tử `<div>` hiện có trong các phần tử ngữ nghĩa HTML5 thích hợp.

Bạn cũng sẽ nhận thấy rằng các phần tử `<header>` và `<footer>` trong ví dụ này vẫn sử dụng các tên lớp giống nhau. Bằng cách này, bạn không cần thay đổi biểu định kiểu ban đầu. Nhờ các tên lớp, các quy tắc biểu định kiểu được sử dụng để định dạng các phần tử `<div>` giờ đây định dạng các phần tử `<header>` và `<footer>`.

Tuy nhiên, bạn có thể cảm thấy tên lớp có vẻ hơi thừa. Nếu vậy, bạn có thể bỏ chúng đi, như thế này:

```

<tiêu đề>
    <h1>Thế giới có thể kết thúc như thế nào</h1>
    <p class="Teaser">Kịch bản đánh dấu sự kết thúc của cuộc sống như chúng ta biết</p> <p class="Byline">của
    Ray N. Carnation</p> <p class="Byline">của Ray N. Carnation</p> </tiêu đề>
```

Để thực hiện công việc này, bạn cần thay đổi quy tắc biểu định kiểu của mình để chúng tự áp dụng theo tên thành phần. Điều này phù hợp với đầu trang và chân trang, bởi vì trang hiện tại chỉ có một phần tử `<header>` và một phần tử `<footer>` duy nhất.

Đây là biểu định kiểu đã sửa đổi áp dụng định dạng của nó cho phần tử <header>:

```
/* Định dạng <header> (dưới dạng hộp màu xanh, có viền.)  
*/ header {  
    ...  
}  
  
/* Định dạng bất kỳ tiêu đề <h1> nào trong <header> (đó là tiêu đề bài viết). */  
tiêu đề h1 {  
    ...  
}  
  
/* Định dạng phụ đề trong <header>. */ tiêu  
đề .Teaser {  
    ...  
}  
  
/* Định dạng dòng tên trong <header>. */ tiêu  
đề .Byline {  
    ...  
}
```

Cả hai cách tiếp cận đều có giá trị như nhau. Cũng như nhiều quyết định thiết kế trong HTML5, có rất nhiều cuộc thảo luận nhưng không có quy tắc cứng nhắc nào.

Bạn sẽ nhận thấy rằng phần <div> dành cho nội dung vẫn còn. Điều này hoàn toàn có thể chấp nhận được, vì các trang web HTML5 thường chứa hỗn hợp các yếu tố ngữ nghĩa và các bộ chứa <div> chung chung hơn. Vì không có phần tử “nội dung” HTML5 nên một <div> thông thường vẫn có ý nghĩa.

GHI CHÚ

Để lại cho các thiết bị của chính nó, trang web này sẽ không hiển thị chính xác trên các phiên bản Internet Explorer trước IE 9. Để khắc phục sự cố này, bạn cần một giải pháp thay thế đơn giản được thảo luận ở trang 51. Nhưng trước tiên, hãy kiểm tra một số yếu tố ngữ nghĩa khác có thể nâng cao các trang của bạn.

Cuối cùng, có thêm một yếu tố đáng để thêm vào ví dụ này. HTML5 bao gồm một phần tử <article> đại diện cho một phần nội dung hoàn chỉnh, độc lập, chẳng hạn như một bài đăng trên blog hoặc một câu chuyện tin tức. Phần tử <article> bao gồm toàn bộ shebang, bao gồm tiêu đề, tên tác giả và nội dung chính. Khi bạn thêm phần tử <article> vào trang, bạn sẽ có cấu trúc sau:

```
<bài viết>  
<tiêu đề>  
    <h1>Thế giới có thể kết thúc như thế nào</h1>  
    ...  
</tiêu đề>
```

trang bị thêm

MỘT TRUYỀN THỐNG

TRANG HTML

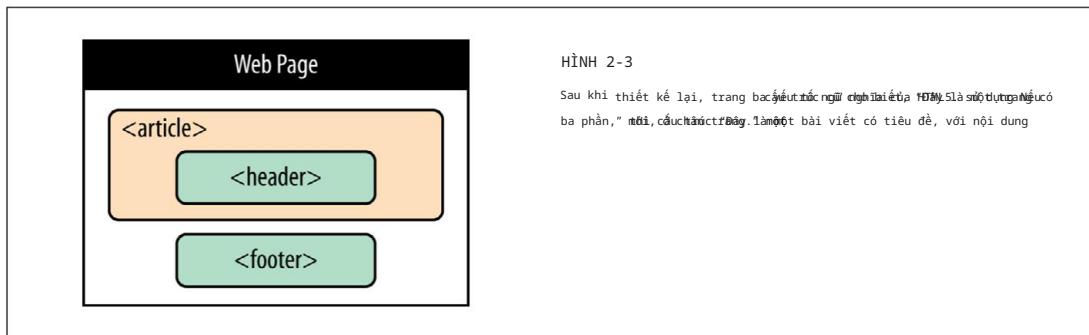
```

<div class="Nội dung">
<p><span class="LeadIn">Ngay bây giờ, có lẽ bạn đang...</p> <p>...</p>
<h2>Ngày tận thế của người Maya</h2> <p>Những người hoài nghi đè nghi ...</p>

...
</div>
</bài viết>

<chân trang>
<p class="Disclaimer">Những dự đoán về ngày tận thế này...</p>
...
</footer>
```

Hình 2-3 cho thấy cấu trúc cuối cùng.



Mặc dù trang web trông vẫn giống nhau trong trình duyệt, nhưng có khá nhiều thông tin bổ sung ẩn sau hậu trường. Ví dụ: một bot tìm kiếm ghé qua trang web của bạn có thể nhanh chóng tìm thấy nội dung trang của bạn (đó là bài viết của bạn) và tiêu đề của nội dung đó (đó là tiêu đề). Nó sẽ không chú ý nhiều đến footer.

GHI CHÚ

Đôi khi các bài báo được chia thành nhiều trang web. Sự đồng thuận hiện tại của các webhead là mỗi phần của bài viết nên được bao bọc trong phần tử `<article>` của riêng nó, mặc dù nó không hoàn chỉnh và độc lập. Sự thỏa hiệp lộn xộn này chỉ là một trong số nhiều điều xảy ra khi ngữ nghĩa đáp ứng các cân nhắc thực tế, trình bày của Web.

Thêm hình với `<figure>` Rất nhiều trang có hình

ảnh. Nhưng khái niệm về một là một chút khác nhau. Đặc tả `HTML5` gợi ý rằng bạn nghĩ về chúng giống như các hình vẽ trong sách-nói cách khác, một hình ảnh tách biệt với văn bản, nhưng được nhắc đến trong văn bản.

Nói chung, bạn để các số liệu `có nghĩa` là bạn đặt chúng ở vị trí thuận tiện gần nhất bên cạnh văn bản của bạn, thay vì khóa chúng ở vị trí bên cạnh một từ hoặc yếu tố cụ thể. Thông thường, các số liệu có chủ thích nổi cùng với chúng.

Ví dụ sau đây cho thấy một số đánh dấu HTML để thêm một con số vào bài báo về ngày tận thế. Nó cũng bao gồm đoạn ngay trước hình và đoạn sau nó, vì vậy bạn có thể thấy chính xác vị trí đặt hình trong phần đánh dấu.

```
<p><span class="LeadIn">Ngay bây giờ, có lẽ bạn đang...</p> <div
class="Floatpicture">
 <p>Bạn
sẽ là người cuối cùng đứng vững nếu một trong những viễn cảnh tận thế này
xảy ra?</p> </div>
```

<p>Nhưng đừng quá tự mãn...</p>

Đánh dấu này giả định rằng bạn đã tạo quy tắc biểu định kiểu định vị hình (và đặt lề, kiểm soát định dạng của văn bản chú thích và tùy ý vẽ đường viền xung quanh hình). Đây là một ví dụ:

```
/* Định dạng hộp hình nổi. */ .FloatHình
{ float: left; lề trái: 0px; lề trên:
  0px; lề phải: 20px; lề dưới: 0px;

}

/* Định dạng văn bản chú thích hình.
 */ .FloatHình p {
  chiều rộng tối đa:
  200px; cỡ chữ:
  nhô; kiểu chữ:
  nghiêng; lề dưới: 5px;
}
```

Hình 2-4 cho thấy ví dụ này tại nơi làm việc.

Nếu bạn đã tạo loại hình này trước đây, bạn sẽ muốn biết rằng HTML5 cung cấp các phần tử ngữ nghĩa mới được thiết kế riêng cho mẫu này. Thay vì sử dụng `<div>` nhằm chán để giữ hộp hình, bạn sử dụng phần tử `<figure>`. Và nếu bạn có bất kỳ văn bản chú thích nào, bạn đặt nó vào phần tử `<figcaption>` bên trong `<figure>`:

```
<figure class="Floatpicture">

<figcaption> Bạn sẽ là người cuối cùng đứng vững nếu một trong những
viễn cảnh tận thế này xảy ra?</figcaption> </figure>
```

trang bị thêm

TRANG HTML

HÌNH 2-4

Bây giờ một con sói bám bão
Trong phần đánh dấu,
nó được xem là hàng ~~nhà~~ ~~nhà~~

Lưu ý rằng chiều rộng của
chú thích ~~hỗn loạn~~ ~~hỗn loạn~~ ~~hỗn loạn~~

How the World Could End

Scenarios that spell the end of life as we know

by Ray N. Carnation

RIGHT NOW, you're probably feeling pretty good. After all, life in the developed world is comfortable—probably more comfortable than it's been for the average human being throughout all of recorded history.

Will you be the last person standing if one of these apocalyptic scenarios plays out?

But don't get too smug. There's still plenty of horrific ways it could all fall apart. In this article, you'll learn about a few of our favorites.

Mayan Doomsday

Skeptics suggest that the Mayan calendar simply rolls to a new 5,126-year era after 2012, and doesn't actually predict a life-ending apocalypse. But given that the long-dead Mayans were wrong about virtually everything else, why should we trust them on this?

Robot Takeover

Not quite as frightening as a Vampire Takeover or Living-Dead Takeover, a robot rebellion is still a disquieting thought. We are already outnumbered by our technological gadgets, and even Bill Gates fears the day his Japanese robot slave turns

Tất nhiên, việc sử dụng biểu định kiểu để định vị và định dạng hộp hình của bạn vẫn tùy thuộc vào bạn. Trong ví dụ này, bạn cần thay đổi bộ chọn quy tắc kiểu nhắm mục tiêu văn bản phụ đề. Ngay bây giờ, nó sử dụng `.Floatpicture p`, nhưng ví dụ sửa đổi yêu cầu chú thích hình `.floatpicture`.

TIẾN BOA Trong ví dụ này, phần tử `<figure>` vẫn có định dạng dựa trên tên lớp của nó (`Floatpicture`), chứ không phải kiểu phân tử của nó. Đó là bởi vì bạn có thể định dạng số liệu theo nhiều cách. Ví dụ: bạn có thể có các hình nổi ở bên trái, các hình nổi ở bên phải, các hình cần cài đặt lề hoặc chú thích khác nhau, v.v. Để duy trì loại tính linh hoạt này, bạn nên định dạng số liệu của mình bằng các lớp.

Trong trình duyệt, con số vẫn giống nhau. Sự khác biệt là mục đích đánh dấu hình của bạn bây giờ hoàn toàn rõ ràng. (Ngẫu nhiên, <figcaption> không bị giới hạn trong việc giữ văn bản-bạn có thể sử dụng bất kỳ phần tử HTML nào có ý nghĩa. Các khả năng khác bao gồm các liên kết và biểu tượng nhỏ.)

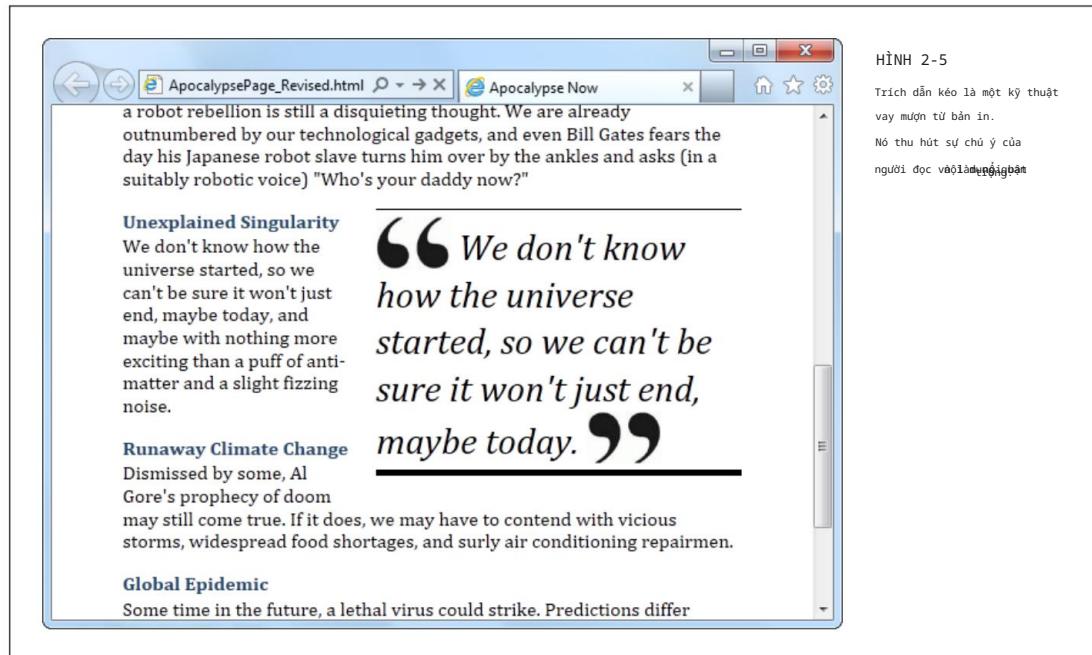
Cuối cùng, cần lưu ý rằng trong một số trường hợp, chú thích hình có thể bao gồm mô tả đầy đủ về hình ảnh, khiến văn bản thay thế trở nên dư thừa. Trong tình huống này, bạn có thể xóa thuộc tính alt khỏi phần tử :

```
<figure class="FloatFigure">
  
  <figcaption>Một hộp sọ người nằm trên cát</figcaption>
</figure>
```

Chỉ cần đảm bảo rằng bạn không đặt văn bản thay thế bằng một chuỗi trống, vì điều đó có nghĩa là hình ảnh của bạn hoàn toàn mang tính trình bày và trình đọc màn hình nên tránh nó cùng nhau.

Thêm Thanh bên với `<aside>` Phần tử `<aside>`

mới đại diện cho nội dung có liên quan tiếp tục với văn bản bao quanh nó. Ví dụ: bạn có thể sử dụng `<aside>` giống như cách bạn sử dụng thanh bên trong bản in để giới thiệu một chủ đề liên quan hoặc để mở rộng một điểm được nêu ra trong tài liệu chính. (Ví dụ: xem hộp ở cuối trang 50.) Phần tử `<aside>` cũng có ý nghĩa nếu bạn cần một nơi nào đó để lưu trữ một khái quát cáo, một số liên kết nội dung có liên quan hoặc thậm chí là một trích dẫn kéo như được hiển thị trong Hình 2-5.



Bạn có thể dễ dàng tạo hiệu ứng này với phần tử `<div>` cũ kỹ, nhưng phần tử `<aside>` cung cấp cho bạn một cách có ý nghĩa hơn để đánh dấu cùng một nội dung:

```
<p>... (bằng một giọng máy phù hợp) "Ai là bố của bạn bây giờ?"</p>

<aside class="PullQuote">
  
```

trang bị thêm

TRANG HTML

Chúng ta không biết vũ trụ bắt đầu như thế nào, vì vậy chúng ta không thể chắc chắn rằng nó sẽ không kết thúc, có thể là ngày hôm nay. 

<h2>Kỳ dị không giải thích được</h2>

Lần này, quy tắc biểu định kiểu thả trích dẫn kéo sang bên phải. Dưới đây là các chi tiết về kiểu dáng, chỉ trong trường hợp bạn tò mò:

```
.PullQuote  
{ float: d\u00fcng;  
chi\u00eau r\u00f6ng t\u00f3i da:  
300px; vi\u00e1n tr\u00e2n c\u00fung: kh\u00f4i m\u00e1u  
den m\u00f3ng; vi\u00e1n-d\u00e1y: đặc den d\u00e1y; c\u00f3  
ch\u00f9: 30px; chi\u00eau cao d\u00f3ng: 130%;  
ki\u00eau ch\u00f9: nghi\u00e9ng; d\u00e9m-top: 5px;  
ph\u00e1n d\u00e9m du\u00f3i c\u00fung: 5px; l\u00e8 tr\u00e1i:  
15px; l\u00e8 du\u00f3i: 10px;  
}  
}
```

```
.PullQuote img  
{ vertical-align: bottom;  
}  
}
```

Làm thế nào các yếu tố ngữ nghĩa được chọn

Trước khi phát minh ra HTML5, những người tạo ra nó đã xem xét rất lâu về các trang web hiện tại. Và họ không chỉ duyệt qua các trang web yêu thích của mình; thay vào đó, họ đã xem xét Google và các trang web khác để thu thập dữ liệu thống kê giờ trong một tý trang web. (Bạn có thể thấy kết quả của cuộc khảo sát đáng chú ý này tại <http://tinyurl.com/trang-thai-cua-web>.)

Cuộc khảo sát của Google đã phân tích đánh dấu và biên soạn một danh sách các tên lớp mà các tác giả web đang sử dụng trong các trang của họ. Ý tưởng của họ là tên lớp có thể phản ánh mục đích của phần tử và đưa ra manh mối có giá trị về cách mọi người cấu trúc các trang. Ví dụ: nếu mọi người đều có phần tử `<div>` sử dụng lớp có tên là `tiêu đề`, thì sẽ hợp lý khi cho rằng mọi người đều đặt tiêu đề ở đầu trang web của họ.

Điều đầu tiên mà Google tìm thấy là phần lớn các trang không sử dụng tên lớp (hoặc biến định kiểu). Tiếp theo, họ

Một số tên phổ biến nhất là footer, header, title, menu, nav-tương ứng với các phần tử ngữ nghĩa mới của HTML5 <footer>, <header> và <nav>. Một số khác để xuất các yếu tố ngữ nghĩa có thể chưa được tạo, như tìm kiếm và bản quyền.

Nói cách khác, Web tràn ngập các dấu hiệu cơ bản giống nhau-ví dụ: các trang có tiêu đề, chân trang, thanh bên và menu điều hướng. Nhưng mọi người đều có một cách hơi khác nhau để làm it nhiều giống nhau. Từ nhận thức này, nó chỉ là một bước nhảy vọt nhỏ để quyết định rằng ngôn ngữ HTML có thể được mở rộng với một vài yếu tố mới nắm bắt được ngữ nghĩa của những gì mọi người đang làm. Và đây chính xác là cái nhìn sâu sắc đã truyền cảm hứng cho các yếu tố ngữ nghĩa của HTML5.

Khả năng tương thích của trình duyệt cho ngữ nghĩa yếu tố

Cho đến nay, bài tập này đã được rất nhiều niềm vui. Nhưng trang có cấu trúc tốt nhất sẽ vô dụng nếu nó không hoạt động trên các trình duyệt cũ hơn.

May mắn thay, các yếu tố ngữ nghĩa của HTML5 được hỗ trợ rộng rãi trên tất cả các trình duyệt hiện đại.

Hầu như không thể tìm thấy phiên bản Chrome, Firefox, Safari hoặc Opera không nhận ra chúng. Trở ngại chính là bất kỳ phiên bản Internet Explorer nào trước IE 9, bao gồm cả IE 8 vẫn đang hoạt động.

May mắn thay, đây là một tính năng còn thiếu rất dễ sửa chữa. Rốt cuộc, các yếu tố ngữ nghĩa không thực sự làm bất cứ điều gì. Để hỗ trợ chúng, một trình duyệt chỉ cần coi chúng như một phần tử `<div>` bình thường. Và để biến điều đó thành hiện thực, tất cả những gì bạn cần làm là tìm hiểu về phong cách của họ, như được mô tả trong các phần sau. Hãy làm điều đó và bạn sẽ nhận được phần thưởng là các yếu tố ngữ nghĩa siêu đáng tin cậy hoạt động với bất kỳ trình duyệt nào được phát hành trong 10 năm qua.

GHI CHÚ

Nếu bạn đang sử dụng Modernizr (trang 31), các trang của bạn sẽ tự động được miễn dịch trước các vấn đề về yếu tố ngữ nghĩa và bạn có thể bỏ qua phần thảo luận sau một cách an toàn. Nhưng nếu bạn không sử dụng Modernizr hoặc nếu bạn tò mò về cách hoạt động của bản sửa lỗi này, hãy đọc tiếp.

Tạo kiểu cho các phần tử ngữ nghĩa Khi một trình duyệt

gặp một phần tử mà nó không nhận ra, nó sẽ coi nó như một phần tử nội tuyến. Hầu hết các phần tử ngữ nghĩa của HTML5 (bao gồm tất cả những phần tử bạn đã thấy trong chương này, ngoại trừ `<time>`) là các phần tử, có nghĩa là trình duyệt phải hiển thị chúng không **chính xác** dưới dạng định **định** trong **trong** và **và** một **một** chút khoảng cách giữa

Các trình duyệt web không nhận ra các phần tử HTML5 sẽ không biết hiển thị một số trong số chúng dưới dạng các phần tử khống, do đó, chúng có khả năng trở thành một mớ hỗn độn.

Để khắc phục sự cố này, bạn chỉ cần thêm quy tắc mới vào biểu định kiểu của mình. Đây là một siêu quy tắc áp dụng định dạng hiển thị khống cho chín thành phần HTML5 cần nó trong một bước:

bài viết, sang một bên, hình, chú thích hình, chân trang, tiêu đề, chính, điều hướng,
phần, tóm tắt { hiển thị: khống;

}

Quy tắc biểu định kiểu này sẽ không có bất kỳ ảnh hưởng nào đối với các trình duyệt đã nhận dạng HTML5 vì thuộc tính hiển thị đã được đặt thành chặn. Và nó sẽ không ảnh hưởng đến bất kỳ quy tắc kiểu dáng nào bạn đã sử dụng để định dạng các phần tử này. Họ vẫn sẽ áp dụng cho quy tắc này.

Trong

phép cộng

Sử dụng HTML5 Shiv Kỹ thuật được mô tả

trong phần trước là đủ để giải quyết các vấn đề tương thích trong hầu hết các trình duyệt, nhưng "hầu hết" không bao gồm Internet Explorer 8 trở lên.

Các phiên bản cũ của IE đưa ra thách thức thứ hai: Họ từ chối áp dụng định dạng biểu định kiểu cho các phần tử mà họ không nhận ra. May mắn thay, có một cách giải quyết: Bạn có thể lừa IE nhận ra một phần tử lạ bằng cách đăng ký phần tử đó bằng một lệnh JavaScript. Ví dụ: đây là khái niệm lệnh cung cấp cho IE khả năng nhận dạng và tạo kiểu cho phần tử `<header>`:

```
<script>
  document.createElement('header') </
<script>
```

Thay vì tự viết loại mã này, bạn có thể tìm một tập lệnh được tạo sẵn để thực hiện việc đó cho bạn. Bạn chỉ cần thêm một tham chiếu đến nó trong phần `<head>` trên trang của mình, như sau:

```
<đầu>
  <title>...</title>
  <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
  ...
<đầu>
```

Mã này lấy tập lệnh từ máy chủ web và chạy <http://html5shim.googlecode.com> bắt đầu xử lý phần còn lại của trang. Tập lệnh này sử dụng JavaScript được mô tả ở trên để tạo tất cả các phần tử HTML5 mới và tiến thêm một bước, bằng cách tự động áp dụng các kiểu được mô tả ở trang 51, để đảm bảo các phần tử mới hiển thị dưới dạng các phần tử khớp phù hợp. Nhiệm vụ còn lại duy nhất là bạn sử dụng các phần tử và thêm các quy tắc biểu định kiểu của riêng bạn để định dạng chúng.

Ngẫu nhiên, mã tập lệnh `html5shim` điều kiện mã này chỉ chạy nếu phát hiện ra rằng bạn đang chạy phiên bản cũ của Internet Explorer. Nhưng nếu bạn muốn tránh chi phí yêu cầu tệp JavaScript, bạn có thể đặt điều kiện tham chiếu tập lệnh, như sau:

```
<!--[nếu là IE 9]>
<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script> <![endif]-->
```

Bằng cách đó, các trình duyệt khác (và IE 9 trở lên) sẽ bỏ qua hướng dẫn này, tiết kiệm thời gian cho trang của bạn vài mili giây.

TIẾN BỘ Ví dụ trước sử dụngshiv HTML5 trực tiếp từ trang web lưu trữ mã của Google. Tuy nhiên, bạn có thể tải xuống bản sao của riêng mình <http://html5shim.googlecode.com> và đặt nó bên cạnh `index.html` và `style.css` để đảm bảo tải lên.

Cuối cùng, đáng để chỉ ra rằng nếu bạn kiểm tra một trang web trên máy tính của mình (thay vì tải nó lên một máy chủ web), Internet Explorer sẽ tự động đặt

trang ở chế độ hạn chế. Điều đó có nghĩa là bạn sẽ thấy thanh bảo mật khét tiếng của IE ở đầu trang, cảnh báo bạn rằng Internet Explorer đã vô hiệu hóa tất cả các tập lệnh của bạn, bao gồm cả shiv HTML5. Để chạy nó, bạn cần nhấp rõ ràng vào thanh bảo mật và chọn cho phép nội dung hoạt động.

Sự cố này sẽ biến mất sau khi bạn tải trang lên một trang web, nhưng nó có thể gây rắc rối khi kiểm tra công việc của bạn. Giải pháp là thêm nhận xét “mark of the Web” vào đầu trang web của bạn, như được mô tả ở trang 14.

Modernizr: Bản sửa lỗi tất cả trong một

Có một giải pháp thay thế tuyệt vời cho vấn đề định kiểu ngữ nghĩa: Sử dụng Modernizr (trang 31). Nó được tích hợp HTML5 shiv, có nghĩa là bạn không cần phải loay hoay với các quy tắc kiểu dáng hoặc bao gồm tham chiếu đến tập lệnh. Vì vậy, bạn Modernizr để kiểm tra tính năng hỗ trợ, hãy xem như sự cố đã được giải quyết.

Thiết kế một trang web với ngữ nghĩa yếu tố

Việc thêm các yếu tố ngữ nghĩa vào một trang đơn giản giống như tài liệu thật dễ dàng. Việc thêm chúng vào một trang web hoàn chỉnh không khó hơn, nhưng nó đặt ra nhiều câu hỏi hơn. Và bởi vì HTML5 về cơ bản là lãnh thổ nguyên vẹn, nên có một số lượng nhỏ các quy ước đã được giải quyết (nhưng một số lượng lớn các bất đồng chính đáng). Điều đó có nghĩa là khi bạn có lựa chọn giữa hai cách tiếp cận đánh dấu và tiêu chuẩn HTML5 cho biết cả hai cách tiếp cận này đều hoàn toàn chấp nhận được, thì bạn phải quyết định cách nào phù hợp nhất với nội dung của mình.

Hình 2-6 cho thấy một ví dụ tham vọng hơn mà bạn sẽ xem xét tiếp theo.

Tìm hiểu sâu hơn về Tiêu đề Có hai

cách tương tự nhưng khác nhau một cách tinh tế để sử dụng phần tử `<header>`. Đầu tiên, bạn có thể sử dụng nó để đặt tiêu đề cho một số nội dung. Thứ hai, bạn có thể sử dụng nó để đặt tiêu đề cho trang web của mình. Đôi khi, hai nhiệm vụ này chồng chéo lên nhau (như ví dụ về một bài báo trong Hình 2-1). Nhưng những lần khác, bạn sẽ có một trang web có cả tiêu đề trang và một hoặc nhiều phần nội dung có tiêu đề. Hình 2-6 là loại ví dụ này.

Điều khiến tình huống này phức tạp hơn một chút là quy ước sử dụng phần tử `<header>` thay đổi dựa trên vai trò của nó. Nếu bạn đang xử lý nội dung, có thể bạn sẽ không sử dụng tiêu đề trừ khi cần. Vì bạn chỉ cần nó nếu bạn đang tạo tiêu đề "béo".

Đó là, một nội dung bao gồm tiêu đề và một số nội dung khác-ví dụ: tóm tắt, ngày xuất bản, tên tác giả, hình ảnh hoặc liên kết chủ đề phụ. Đây là một ví dụ:

```
<tiêu đề>
<h1>Thế giới có thể kết thúc như thế nào</h1> <p
class="Tagline">Các kịch bản đánh dấu sự kết thúc của cuộc sống như chúng ta biết</p> <p class="Byline">của
Ray N. Carnation</p> <p class="Byline">của Ray N. Carnation</p> p> </tiêu đề>
```

Firefox

Apocalypse Today file:///C/HTML5/Chapter 02/ApocalypseSite.html

ARE YOU READY FOR...
Apocalypse Today



Articles

- How The World Could End
- Would Aliens Enslave or Eradicate Us?
- Great Floods of the Past
- Could Chemotherapy Cause Cancer?
- Why Everything You Know About Zombie Attacks Is Wrong
- More...

About Us

Apocalypse Today is a world leader in conspiracy theories, doomsday predictions, and panic-spreading. Our motto is "be prepared for every possibility (except the good ones)."



Will you be the last person standing if one of these apocalyptic scenarios plays out?

How the World Could End
Scenarios that spell the end of life as we know
by Ray N. Carnation

RIGHT NOW, you're probably feeling pretty good. After all, life in the developed world is comfortable—probably more comfortable than it's been for the average human being throughout all of recorded history.



But don't get too smug. There's still plenty of horrific ways it could all fall apart. In this article, you'll learn about a few of our favorites.

Mayan Doomsday
Skeptics suggest that the Mayan calendar simply rolls to a new 5,126-year era after 2012, and doesn't actually predict a life-ending apocalypse. But given that the long-dead Mayans were wrong about virtually everything else, why should we trust them on this?

Robot Takeover
Not quite as frightening as a Vampire Takeover or Living-Dead Takeover, a robot rebellion is still a disquieting thought. We are already outnumbered by our technological gadgets, and even Bill Gates fears the day his Japanese robot slave turns him over by the ankles and asks (in a suitably robotic voice) "Who's your daddy now?"

Unexplained Singularity
We don't know how the universe

“ We don't know

HÌNH 2-6

Ở đây, bài báo một trang được xem

Một

Tuy nhiên, khi mọi người tạo tiêu đề cho một trang web, hầu như họ luôn bọc nó trong phần tử `<header>`, ngay cả khi không có gì ở đó ngoài tiêu đề trong hộp định dạng CSS lớn. Rốt cuộc, đó là một điểm thiết kế chính của trang web của bạn và ai biết được khi nào bạn có thể cần mở nó ra và thêm một cái gì đó mới?

Đây là điểm rút ra: Các trang có thể có nhiều hơn một phần tử <header> (và chúng thường như vậy), mặc dù các phần tử này đóng các vai trò khác nhau trên trang.

Trang khái huyền (Hình 2-6) sử dụng phần tử `<header>` cho tiêu đề trang web và một phần tử `<header>` khác cho tiêu đề bài viết. `<header>` giới hạn trang web chứa hình ảnh biểu ngữ, kết hợp văn bản đồ họa và hình ảnh:

```
<header class="SiteHeader">  
    
  <h1 style="display:none">Ngày tận thế</h1>  
</header>
```

LÊN ĐẾN TỐC ĐỘ

Biến một trang web thành một trang web

Hình 2-6 hiển thị một trang từ một trang web hư cấu.

Trong một trang web thực, bạn sẽ có cùng một bộ cục (và cùng một bảng điều khiển bên) trên [http://www.trangdikhoa.com](#) khác nhau và cập nhật cho tất cả các trang là nội dung trang chính trong trường hợp này là bài viết.

HTML5 không có phép thuật đặc biệt nào để chuyển trang web thành trang web. Thay vào đó, bạn cần sử dụng các thủ thuật và công nghệ tương tự mà các nhà phát triển web dựa vào trong HTML truyền thống:

- Server-side frameworks. Ý tưởng rất đơn giản: Khi trình duyệt yêu cầu một trang, máy chủ web sẽ lắp ráp các phần, bao gồm các thành phần chung (như thanh điều hướng) và nội dung. Cách tiếp cận này cho đến nay là phổ biến nhất và đó là cách duy nhất để truy cập các trang web lớn, chuyên nghiệp. Vô số công nghệ khác nhau thực hiện cách tiếp cận này theo những cách khác nhau, từ lập trình web

các nền tảng như ASP.NET và PHP cho đến các hệ thống quản lý nội dung như Drupal và WordPress.

- Mẫu trang. Một số trình chỉnh sửa trang web mạnh mẽ (như Adobe Dreamweaver và Microsoft Visual Studio) bao gồm tính năng mẫu trang. Bạn bắt đầu bằng cách tạo một mẫu xác định cấu trúc trang web của mình và bao gồm nội dung lặp lại mà bạn muốn xuất hiện trên mọi trang (như tiêu đề và thanh bên). Sau đó, bạn sử dụng mẫu để tạo tất cả các trang trên trang web của mình. Đây là phần gọn gàng: Khi bạn cập nhật mẫu, trình chỉnh sửa trang web của bạn sẽ tự động cập nhật tất cả các trang sử dụng nó.

Tất nhiên, bạn có thể tự do sử dụng một trong hai kỹ thuật, vì vậy cuốn sách này tập trung vào kết quả cuối cùng: đánh dấu được dán lại với nhau tạo thành một trang hoàn chỉnh và được hiển thị trong trình duyệt web.

Ngay lập tức, bạn sẽ nhận thấy rằng tiêu đề này thêm một chi tiết mà bạn không thấy trên trang: tiêu đề `<h1>` sao chép nội dung trong ảnh. Tuy nhiên, cài đặt kiểu nội tuyến sẽ ẩn tiêu đề này.

Ví dụ này đặt ra một câu hỏi rõ ràng: Thêm một tiêu đề mà bạn không nhìn thấy để làm gì? Thực tế có một số lý do. Đầu tiên, tất cả các phần tử `<header>` yêu cầu một số cấp độ tiêu đề bên trong, chỉ để đáp ứng các quy tắc của HTML5. Thứ hai, thiết kế này làm cho trang dễ truy cập hơn đối với những người đang điều hướng trang bằng trình đọc màn hình, bởi vì họ thường chuyển từ tiêu đề này sang tiêu đề khác mà không nghe nội dung ở giữa. Và thứ ba, nó thiết lập cấu trúc tiêu đề mà bạn có thể sử dụng trong phần còn lại của trang. Đó là một cách nói hoa mỹ rằng nếu bạn bắt đầu bằng `<h1>` cho tiêu đề trang web của mình, bạn có thể quyết định sử dụng các phần tử `<h2>` để đặt tiêu đề cho các phần khác của trang (như “Bài viết” và “Giới thiệu về chúng tôi” trong thanh bên). Để biết thêm về quyết định thiết kế này, hãy xem hộp ở trang 56.

GHI CHÚ

Tất nhiên, bạn có thể đơn giản hóa cuộc sống của mình bằng cách tạo một tiêu đề văn bản thông thường. (Và nếu bạn muốn phông chữ ưa thích, tính năng phông chữ web CSS3, được mô tả ở trang 206, có thể giúp bạn.) Nhưng đối với nhiều trang web đặt tiêu đề trong ảnh, thủ thuật tiêu đề ẩn là giải pháp tốt nhất tiếp theo.

CÂU HỎI THƯỜNG GẶP

Cấu trúc tiêu đề của một trang web

Có chấp nhận được việc có ~~thông~~ tiêu đề cấp 1 trên một trang

Nó là một ý tưởng tốt?

Theo các quy tắc chính thức của HTML, bạn có thể có bao nhiêu tiêu đề cấp 1 tùy thích. Tuy nhiên, người tạo trang web thường cố gắng chỉ có một tiêu đề cấp 1 duy nhất trên mỗi trang vì điều đó giúp trang web dễ truy cập hơn—vì những người sử dụng trình đọc màn hình có thể bỏ lỡ tiêu đề cấp 1 khi họ chuyển từ tiêu đề cấp 2 sang tiêu đề cấp 2 tiếp theo. Cũng có một trường phái quản trị trang web cho rằng mỗi trang nên có chính xác một tiêu đề cấp 1, tiêu đề này là duy nhất trên toàn bộ trang web và cho các công cụ tìm kiếm biết rõ ràng nội dung nào đang chờ đợi.

Ví dụ trong Hình 2-6 sử dụng phong cách này. Tiêu đề “Ngày tận thế” ở đầu trang là `<h1>` duy nhất trên trang. Các phần khác trên trang, như “Bài viết” và “Giới thiệu về chúng tôi” trong thanh bên, sử dụng tiêu đề cấp 2. Tiêu đề bài viết cũng sử dụng tiêu đề cấp 2. (Với một chút lập kế hoạch bổ sung, bạn có thể thay đổi văn bản của tiêu đề cấp 1 để phù hợp với bài viết hiện tại—xét cho cùng, tiêu đề này không thực sự hiển thị,

và nó có thể giúp trang phù hợp với các truy vấn tập trung hơn trong công cụ tìm kiếm như Google.)

Nhưng có những cách tiếp cận khác, hợp lệ như nhau. Ví dụ: bạn có thể sử dụng tiêu đề cấp 1 để đặt tiêu đề cho từng phần chính trên trang của mình, bao gồm thanh bên, bài viết, v.v.

Hoặc, bạn có thể cung cấp cho trang web tiêu đề trang web cấp 1 và đặt tiêu đề cấp 2 vào thanh bên (như trong ví dụ hiện tại) nhưng đặt tiêu đề bài viết thành tiêu đề cấp 1 thứ hai. Điều này hoạt động tốt trong HTML5, vì hệ thống phác thảo mới của nó. Như bạn sẽ tìm hiểu ở trang 65, một số phần tử, bao gồm cả `<article>`, được coi là các phần riêng biệt, với các đường viền riêng biệt của chúng.

Vì vậy, hoàn toàn hợp lý khi các phần này bắt đầu lại hệ thống phân cấp tiêu đề với một `<h1>` hoàn toàn mới. (Tuy nhiên, HTML5 nói rằng bạn cũng có thể bắt đầu với một cấp tiêu đề khác.)

Tóm lại, không có câu trả lời rõ ràng về cách cấu trúc trang web của bạn. Có vẻ như cách tiếp cận “nhiều `<h1>`” sẽ ngày càng trở nên phổ biến khi HTML5 chính phục Web. Nhưng hiện tại, nhiều nhà phát triển web đang gắn bó với phương pháp “đơn `<h1>`” để giữ cho trình đọc màn hình hài lòng.

Liên kết Điều hướng với `<nav>` Tính năng

mới thú vị nhất trong trang web khai huyền là thanh bên ở bên trái, cung cấp điều hướng của trang web, một số thông tin bổ sung và quảng cáo hình ảnh. (Thông thường, bạn sẽ đưa vào một khối JavaScript để tìm nạp một quảng cáo được chọn ngẫu nhiên bằng cách sử dụng một dịch vụ như Google AdSense. Nhưng ví dụ này chỉ mã hóa cứng một hình ảnh để thay thế cho điều đó.)

Trong trang web HTML truyền thống, bạn sẽ bọc toàn bộ thanh bên trong `<div>`. Trong HTML5, hầu như bạn luôn dựa vào hai yếu tố thẻ hơn: `<aside>` và `<nav>`.

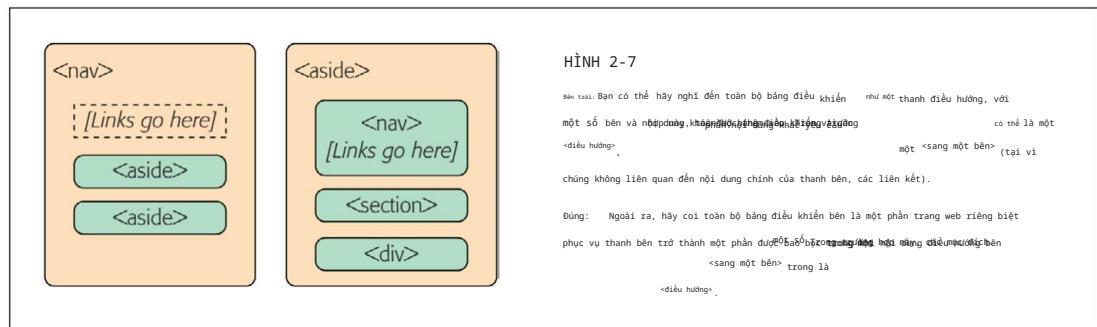
Phần tử `<aside>` hơi giống phần tử `<header>` ở chỗ nó có một ý nghĩa tinh tế, có thể kéo dài một chút. Bạn có thể sử dụng nó để đánh dấu một phần nội dung có liên quan, như bạn đã làm với trích dẫn kéo ở trang 49. Hoặc, bạn cũng có thể sử dụng nó để chỉ định một phần hoàn toàn riêng biệt của trang—một phần khác với luồng chính.

Phần tử `<nav>` bao bọc một khối liên kết. Các liên kết này có thể trỏ đến các chủ đề trên trang hiện tại hoặc các trang khác trên trang web. Hầu hết các trang sẽ có nhiều phần `<nav>` trong đó. Nhưng không phải tất cả các liên kết đều cần có phần `<nav>`—thay vào đó, phần này thường được dành riêng cho các phần điều hướng lớn nhất và quan trọng nhất trên một trang. Vì

ví dụ, nếu bạn có một danh sách các bài báo (như trong Hình 2-6), bạn chắc chắn cần có phần `<nav>`. Nhưng nếu bạn chỉ có một vài liên kết ở cuối trang với giấy phép và thông tin liên hệ, thì không cần thiết phải có `<nav>` đầy đủ.

Với hai yếu tố này, đây là thời điểm tốt để thử một bài tập thực hành. Đầu tiên, xem lại thanh bên trong Hình 2-6. Tiếp theo, hãy phác thảo trên một tờ giấy cách bạn đánh dấu cấu trúc của nội dung này. Sau đó, đọc tiếp để tìm ra giải pháp tốt nhất có thể.

Trên thực tế, có ít nhất hai cách hợp lý để cấu trúc thanh bên này, như trong Hình 2-7.



Địa điểm tận thê sử dụng cách tiếp cận thứ hai (Hình 2-7, bên phải). Đó là bởi vì thanh bên thường như phục vụ một số mục đích, không có mục đích nào rõ ràng là chiếm ưu thế. Nhưng nếu bạn có một phần điều hướng dài và phức tạp (chẳng hạn như menu có thể thu gọn) theo sau là một đoạn nội dung ngắn, thì cách tiếp cận đầu tiên có thể hợp lý hơn.

Đây là phần đánh dấu định hình thanh bên, chia nó thành ba phần:

```

<aside class="NavSidebar">
    <điều hướng>
    <h2>Bài báo</h2>
    <ul>
        <li><a href="...">Thế giới có thể kết thúc như thế nào</a></li>
        <li><a href="...">Người ngoài hành tinh sẽ nô dịch hay tiêu diệt chúng ta?</a> </li>
        ...
    </ul>
    </nav>

    <phần>
    <h2>Giới thiệu</h2>
    <p>Apocalypse Today là công ty hàng đầu thế giới về thuyết âm mưu..." </p> </section>

```

```
<div>
     </
div> </aside>
```

Đây là điểm mấu chốt:

- Phần tiêu đề ("Bài báo" và "Giới thiệu về chúng tôi") là tiêu đề cấp 2. Theo cách đó, chúng rõ ràng là phụ thuộc vào tiêu đề trang web cấp 1, điều này làm cho trình đọc màn hình dễ tiếp cận trang hơn.
- Các liên kết được đánh dấu trong một danh sách không có thứ tự bằng cách sử dụng các phần tử `` và ``. Các nhà thiết kế trang web đồng ý rằng một danh sách là cách tốt nhất, dễ tiếp cận nhất để xử lý một loạt các liên kết. Tuy nhiên, bạn có thể cần sử dụng các quy tắc biểu định kiểu để loại bỏ thực lề (như được thực hiện ở đây) và dấu đầu dòng (không được thực hiện trong ví dụ này).
- Phần "Giới thiệu về chúng tôi" được bao bọc trong phần tử `<section>`. Đó là bởi vì không có yếu tố ngữ nghĩa nào khác phù hợp với nội dung này. `<section>` cụ thể hơn một chút so với `<div>`—nó phù hợp với bất kỳ khối nội dung nào bắt đầu bằng tiêu đề. Nếu có một phần tử cụ thể hơn để sử dụng (ví dụ: một phần tử `<about>` giả định), thì phần tử đó sẽ thích hợp hơn một phần tử `<section>` cơ bản, nhưng không có phần tử đó.
- Quảng cáo hình ảnh được bao bọc trong `<div>`. Phần tử `<section>` chỉ phù hợp với nội dung bắt đầu bằng tiêu đề và phần hình ảnh không có tiêu đề. (Mặc dù nếu đúng như vậy—hãy nói, "Lời từ các nhà tài trợ của chúng tôi"—một phần tử `<section>` sẽ là lựa chọn tốt hơn.) Về mặt kỹ thuật, không cần thiết phải đặt bất kỳ phần tử nào khác xung quanh hình ảnh, nhưng `<div>` giúp việc này trở nên dễ dàng hơn để tách phần này, tạo kiểu cho nó và thêm một số mã JavaScript để thao tác với nó, nếu cần.

Ngoài ra còn có một số chi tiết mà thanh bên này ~~lấy~~ nhưng nhiều thanh bên khác lại có. Ví dụ: các thanh bên phức hợp có thể bắt đầu bằng `<header>` và kết thúc bằng `<footer>`. Chúng cũng có thể bao gồm nhiều phần `<nav>`—ví dụ: một phần dành cho danh sách nội dung được lưu trữ, một phần chứa danh sách tin tức mới nhất, một phần chứa danh sách blog hoặc danh sách các trang web liên quan, v.v. Ví dụ: hãy xem thanh bên của một blog điện hình, có đầy đủ các phần, nhiều phần trong số đó là điều hướng.

Các quy tắc biểu định kiểu bạn sử dụng để định dạng thanh bên `<aside>` giống với những quy tắc bạn sẽ sử dụng để định dạng thanh bên `<div>` truyền thống. Họ đặt thanh bên vào vị trí chính xác, sử dụng định vị tuyệt đối và đặt một số chi tiết định dạng, như `tiếng đậm` và `màu nền`:

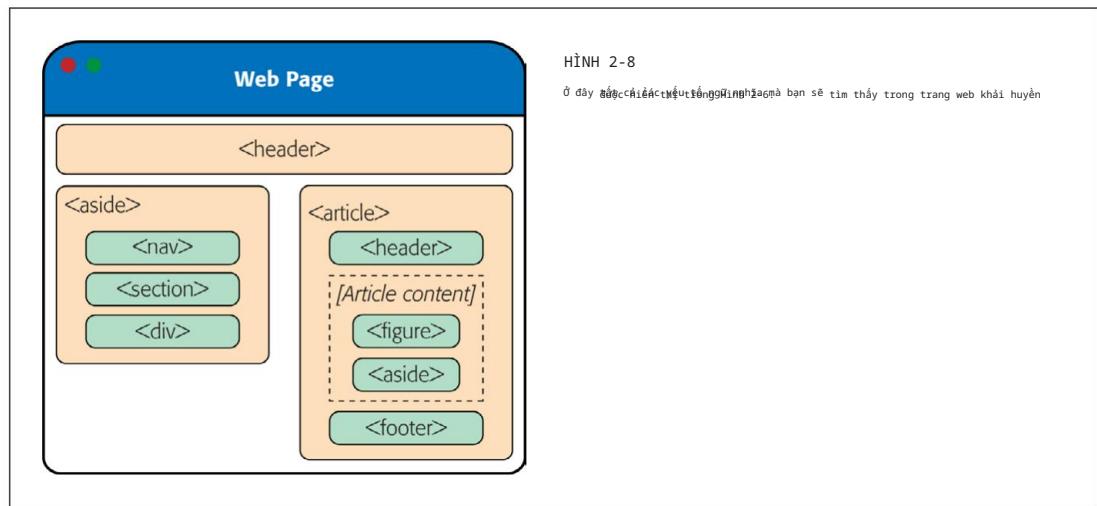
```
sang một bên.NavSidebar
{
    vị trí: tuyệt đối;
    trên cùng: 179px;
    trái: 0px; phần đậm:
    5px 15px 0px 15px; chiều rộng:
    203px;
```

chiều cao tối thiểu:
1500px; màu nền: #eee;
cỡ chữ: nhỏ;

Quy tắc này được tuân theo bởi các quy tắc biểu định kiểu theo ngữ cảnh định dạng các phần tử <h2>, , và trong thanh bên. (Như thường lệ, bạn có thể lấy mã mẫu từ <http://prosetech.com/html5>, và lướt qua biểu định kiểu hoàn chỉnh.)

Bây giờ bạn đã hiểu cách thanh bên được đặt cùng nhau, bạn sẽ hiểu cách nó phù hợp với bố cục của toàn bộ trang, như trong Hình 2-8.

GHI CHÚ Như bạn đã biết, `<nav>` thường được tìm thấy riêng lẻ hoặc trong `<aside>`. Còn một com nữa



Đi sâu hơn vào các phần Như bạn đã biết, `<phản>` là yếu tố ngữ nghĩa cuối cùng. Nếu bạn có một khối nội dung có tiêu đề và các phần tử ngữ nghĩa khác không phù hợp, thì phần tử `<section>` thường là lựa chọn tốt hơn so với `<div>`.

Vì vậy, những gì diễn ra trong một phần điển hình? Tùy thuộc vào quan điểm của bạn, phần tử `<section>` hoặc là một công cụ linh hoạt phù hợp với nhiều nhu cầu, hoặc là một con quái vật lồng lèo và rỗng thùng thình không có danh tính rõ ràng. Đó là bởi vì các phần đóng nhiều vai trò trang web khác nhau. Họ có thể đánh dấu bất kỳ điều nào sau đây:

- Các khái niệm nội dung nhỏ được hiển thị đọc theo trang chính, chẳng hạn như đoạn "Giới thiệu về chúng tôi" trong trang web về ngày tận thế.
 - Nội dung đọc lập không thể thực sự được mô tả dưới dạng một bài báo, chẳng hạn như hồ sơ thành toán của khách hàng hoặc danh sách sản phẩm.

- Các nhóm nội dung-ví dụ: tập hợp các bài báo trên một trang tin tức.
 - Một **phần** liệu dài hơn. Ví dụ: trong bài viết về ngày tận thế, bạn có thể đánh dấu từng kịch bản về ngày tận thế riêng biệt thành một phần riêng biệt.
- Đôi khi, bạn sẽ sử dụng các phần theo cách này để đảm bảo dàn ý chính xác cho tài liệu của mình, như được giải thích trong phần tiếp theo.

Hai mục cuối cùng trong danh sách là đáng ngạc nhiên nhất. Nhiều nhà phát triển web cảm thấy hơi khó khăn khi sử dụng cùng một yếu tố để xử lý một đoạn nhỏ của bài viết và toàn bộ nhóm bài viết. Một số người cho rằng HTML5 nên có ít nhất hai phần tử khác nhau để xử lý các tình huống khác nhau này. Nhưng những người tạo ra HTML5 đã quyết định giữ mọi thứ đơn giản (bằng cách giới hạn số lượng phần tử mới) trong khi làm cho các phần tử mới trở nên linh hoạt và thiết thực nhất có thể.

Có một sự cân nhắc cuối cùng. Phần tử `<section>` cũng có ảnh hưởng đến đường viền của trang web, đây là khái niệm mà bạn sẽ khám phá ở trang 65.

ĐÁ QUÝ TRONG TUYỆT VỜI

Hộp có thẻ gấp lại với `<chi tiết>` và `<tóm tắt>`

Bạn chắc chắn đã nhìn thấy các hộp có thẻ thu gọn trên Web-các phần nội dung mà bạn có thể hiển thị hoặc ẩn bằng cách nhấp vào tiêu đề.

Các hộp có thẻ thu gọn là một trong những kỹ công dễ thực hiện nhất với JavaScript cơ bản. Bạn chỉ cần phản ứng khi tiêu đề được nhấp vào, sau đó thay đổi cài đặt kiểu để ẩn hộp của bạn:

```
hộp var = tài liệu.  
  
getElementById("myBox");  
  
box.style.display = "không"; Và sau đó  
  
quay lại để làm cho nó xuất hiện lại:  
  
hộp var = tài liệu.  
  
getElementById("myBox");  
box.style.display = "khối";
```

Thật thú vị, HTML5 bổ sung thêm hai yếu tố ngữ nghĩa nhằm mục đích làm cho hành vi này trở nên tự động. Ý tưởng là bạn bọc phần có thẻ thu gọn của mình trong phần tử `<details>` và bọc tiêu đề bên trong phần tử `<summary>`. Kết quả cuối cùng là một cái gì đó như thế này:

```
<chi tiết>  
<summary>Phản #1</summary>
```

`<p>Nếu bạn có thẻ thấy nội dung này, phần này
được mở rộng</p> </details>`

Các trình duyệt hỗ trợ các yếu tố này (hiện tại chỉ có Chrome), sẽ chỉ hiển thị tiêu đề, có thẻ với một số loại trang trí trực quan (chẳng hạn như biểu tượng hình tam giác nhỏ bên cạnh tiêu đề).

Sau đó, nếu người dùng nhấp vào tiêu đề, toàn bộ nội dung sẽ mở rộng trong chế độ xem. Các trình duyệt không hỗ trợ các phần tử `<details>` và `<summary>` sẽ hiển thị toàn bộ nội dung ngay từ đầu mà không cung cấp cho người dùng bất kỳ cách nào để thu gọn nội dung đó.

Các phần tử `<details>` và `<summary>` đang gây tranh cãi. Nhiều nhà phát triển web cảm thấy rằng chúng không thực sự có ngữ nghĩa, bởi vì chúng thiên về phong cách trực quan hơn là cấu trúc logic.

Hiện tại, tốt nhất bạn nên tránh các phần tử `<details>` và `<summary>` vì chúng hỗ trợ trình duyệt rất kém. Mặc dù bạn có thể viết một giải pháp thay thế sử dụng JavaScript trên các trình duyệt không hỗ trợ chúng, nhưng việc viết giải pháp thay thế này tốn nhiều công sức hơn là chỉ sử dụng một vài dòng JavaScript để tự thực hiện việc thu gọn trên bất kỳ trình duyệt nào.

Tìm hiểu sâu hơn về

Footers HTML5 và các tiêu đề chất lượng là dành cho nhau. Bạn không chỉ có thể chèn phụ đề và dòng phụ đề mà còn có thể thêm hình ảnh, phần điều hướng (với phần tử `<nav>`) và hầu như bất kỳ thứ gì khác thuộc về đầu trang của bạn.

Thật kỳ lạ, HTML5 không phù hợp khi nói đến chân trang. Phần chân trang được cho là chỉ giới hạn ở một số chi tiết cơ bản về bản quyền của trang web, nguồn gốc tác giả, các hạn chế pháp lý và liên kết. Phần chân trang không được phép chứa danh sách dài các liên kết, các phần nội dung quan trọng hoặc các chi tiết không liên quan như quảng cáo, nút mạng xã hội và tiện ích trang web.

Điều này đặt ra một câu hỏi: Bạn nên làm gì nếu thiết kế trang web của bạn yêu cầu một footer béo? Xét cho cùng, các footer mập đang cực kỳ phổ biến trong thiết kế trang web hiện nay (xem ví dụ trong Hình 2-9). Họ kết hợp một số kỹ thuật ưa thích, đôi khi bao gồm những điều sau đây:

- Vị trí cố định, vì vậy chân trang luôn được gắn vào cuối cửa sổ trình duyệt, bất kể khách truy cập cuộn ở đâu (như ví dụ trong Hình 2-9).
 - Một nút đóng để khách truy cập có thể đóng chân trang và lấy lại không gian sau khi đọc nội dung (như ví dụ trong Hình 2-9). Để thực hiện công việc này, bạn sử dụng một đoạn JavaScript đơn giản để ẩn phần tử bao quanh chân trang (như mã được hiển thị trong hộp ở trang 60).



- Nền trong suốt một phần để bạn có thể xem nội dung trang xuyên qua phần cuối trang. Thiết lập này hoạt động tốt nếu chân trang đang quảng cáo tin nóng hoặc tuyên bố từ chối trách nhiệm quan trọng và nó thường được sử dụng cùng với nút đóng.
- Hoạt hình, để chân trang lò xo hoặc trượt vào dạng xem. (Ví dụ: xem hộp bài viết liên quan đặt lên khi bạn đến cuối bài viết tại) www.nytimes.com.

Nếu trang web của bạn bao gồm loại chân trang này, bạn có một sự lựa chọn. Cách tiếp cận đơn giản là bỏ qua các quy tắc. Cách tiếp cận này không khủng khiếp như nó có vẻ, bởi vì các nhà phát triển trang web khác chắc chắn sẽ phạm phải sai lầm tương tự và theo thời gian, các quy tắc chính thức có thể được nới lỏng, cho phép các phần chân trang đẹp hơn. Nhưng nếu bạn muốn ở phía bên phải của tiêu chuẩn ngay bây giờ, bạn cần điều chỉnh đánh dấu của mình. May mắn thay, nó không quá khó.

Bí quyết là tách các chi tiết chân trang tiêu chuẩn khỏi các chi tiết bổ sung. Trong trình duyệt, chúng có vẻ như là một chân trang duy nhất, nhưng trong phần đánh dấu, chúng sẽ không thuộc về phần tử <footer>. Ví dụ, đây là cấu trúc của fat footer trong Hình 2-9:

```
<div id="FatFooter">
    <!-- Nội dung ở cuối trang béo ở đây. -->
    
    ...
    <chân trang>
        <!-- Nội dung chân trang tiêu chuẩn ở đây. --
        <p>Các quan điểm thẻ hiện trên trang web này *không*</p>
    footer> </div>
```

<div> bên ngoài không có ý nghĩa ngữ nghĩa. Thay vào đó, nó là một gói tiện lợi kết hợp nội dung "béo" bổ sung với các chi tiết chân trang đơn giản. Nó cũng cho phép bạn áp dụng quy tắc định dạng biểu định kiểu để khóa phần cuối trang vào vị trí:

```
#FatFooter
{
    vị trí: đã sửa;
    dưới cùng: 0px;
    chiều cao: 145px;
    chiều rộng: 100%;
    nền: #ECD672; viền
    trên: mỏng màu đen; cỡ chữ: nhỏ;
}
```

GHI CHÚ Trong ví dụ này, quy tắc biểu định kiểu áp dụng định dạng của nó theo tên ID (sử dụng bộ chọn #FatFooter) thay vì theo tên lớp (ví dụ: sử dụng bộ chọn .FatFooter). Đó là vì fat footer đã cần một ID duy nhất để mã JavaScript có thể tìm và ẩn nó khi ai đó nhấp vào nút đóng.

Việc sử dụng ID duy nhất này trong biểu định kiểu sẽ hợp lý hơn là thêm tên lớp cho cùng một mục đích.

Bạn cũng có thể chọn đặt chân trang trong phần tử `<aside>` để chỉ rõ rằng nội dung chân trang là một phần riêng biệt và có liên quan trực tiếp với phần còn lại của nội dung trên trang. Đây là cấu trúc đó trông như thế nào:

```
<div id="FatFooter">
    <sang một bên>
    <!-- Nội dung ở cuối trang béo ở đây. -->
    
    ...
</aside>

<chân trang>
    <!-- Nội dung chân trang tiêu chuẩn ở đây. --> ...
    <p>Các quan điểm thẻ hiện trên trang web này không </p>
    footer> </div>
```

Chi tiết quan trọng ở đây là `<footer>` không được đặt bên trong phần tử `<aside>`. Đó là vì `<footer>` không áp dụng cho `<aside>` mà áp dụng cho toàn bộ trang web. Tương tự, nếu bạn có `<footer>` áp dụng cho một số phần nội dung, thì `<footer>` của bạn cần được đặt bên trong phần tử bao bọc nội dung đó.

GHI CHÚ

Các quy tắc và hướng dẫn sử dụng hợp lý các phần tử ngữ nghĩa của HTML5 vẫn đang phát triển. Các câu hỏi về cách thích hợp để đánh dấu các trang web lớn, phức tạp đã khuấy động cuộc tranh luận gay gắt trong cộng đồng HTML. Lời khuyên tốt nhất là: Nếu điều gì đó có vẻ không đúng với nội dung của bạn, đừng làm điều đó. Hoặc bạn có thể thảo luận trực tuyến, nơi bạn có thể nhận phản hồi từ hàng chục chuyên gia HTML siêu thông minh. (Một trang web đặc biệt tốt là nơi bạn có thể thấy những <http://html5doctor.com>, cuộc tranh luận đang diễn ra này diễn ra trong phần bình luận của hầu hết các bài báo.)

Xác định Nội dung Chính bằng `<main>` HTML5 bao gồm một phần tử `<main>` đôi khi bị bỏ qua để xác định nội dung chính của trang web. Ví dụ: trong trang web khai huyền, nội dung chính là toàn bộ bài viết, không bao gồm tiêu đề, thanh bên hoặc chân trang của trang web. Bạn nên cân nhắc sử dụng nó trên các trang của riêng bạn.

Phần tử `<main>` được áp dụng đúng sẽ bao bọc chính xác phần tử `<article>`. Đây là giao diện của nó trong trang khai huyền:

```
<!DOCTYPE html>
<html lang="vi">
<head>
    ...
</head>

<body>
    <tiêu đề>
        ...
    </tiêu đề>
```

<sang một bên>

```
...
</aside>

<chính>
  <bài viết>
  ...
</article>
</main>
<chân trang>
...
</footer>
</body> </
html>
```

Bạn không thể đặt phần tử `<main>` bên trong phần tử `<article>` (hoặc trong bất kỳ phần tử ngữ nghĩa nào khác). Đó là vì phần tử `<main>` được dùng để chứa toàn bộ nội dung chính của trang. Nó có nghĩa là chỉ ra một phần nội dung quan trọng bên trong tài liệu. ~~Phần tử <main> chỉ có thể chứa nội dung chính và không thể có các phần tử ngôn ngữ nghĩa khác,~~

Một lần

Thoạt nhìn, phần tử `<main>` có vẻ không hữu dụng cho lắm. Tuy nhiên, nó có thể quan trọng đối với trình đọc màn hình vì nó cho phép họ bỏ qua các nội dung không liên quan-như tiêu đề trang web, menu điều hướng, quảng cáo, thanh bên, v.v.—để truy cập nội dung thực. Và mặc dù phần tử `<main>` bám vào phần tử `<article>` trong ví dụ này, điều đó không nhất thiết phải xảy ra trong một trang phức tạp hơn. Ví dụ: nếu bạn đã tạo một trang liệt kê nhiều bản tóm tắt bài viết, mỗi bản tóm tắt được bao bọc trong một phần tử `<article>`, thì phần tử `<main>` sẽ bao bọc danh sách đầy đủ các phần tử `<article>`, như sau:

```
<chính>
  <bài viết>
  ...
</bài viết>

<bài viết>
  ...
</bài viết>

<bài viết>
  ...
</bài viết>

...
</chính>
```

Ở đây, sự khác biệt là rõ ràng-mỗi <bài viết> đại diện cho một phần nội dung độc lập, nhưng nội dung chính của trang là tập hợp đầy đủ các bài viết.

Bạn có thể sử dụng phần tử <main> trên bất kỳ loại trang nào, ngay cả khi trang đó không có bài viết. Ví dụ: nếu bạn tạo trò chơi hoặc ứng dụng, thì nội dung chính là tập hợp các đánh dấu tạo trò chơi hoặc ứng dụng đó. Bạn có thể sử dụng phần tử <main> để bọc toàn bộ shebang, không bao gồm các chi tiết bên ngoài như đầu trang và chân trang.

GHI CHÚ

Phần tử <main> là một phần tử tương đối mới. Nó được giới thiệu trong phiên bản được điều chỉnh một chút của tiêu chuẩn HTML5 có tên là HTML 5.1 (trang xv).

Hệ thống phác thảo HTML5

HTML5 xác định một bộ quy tắc quy định cách bạn có thể tạo một trang cho bất kỳ trang web nào. Phác thảo của một trang web có thể có ích theo nhiều cách. Ví dụ: một trình duyệt có thể cho phép bạn chuyển từ phần này sang phần khác của dàn bài. Một công cụ thiết kế có thể cho phép bạn sắp xếp lại các phần bằng cách kéo và thả chúng vào dạng xem phác thảo. Công cụ tìm kiếm có thể sử dụng đường viền để xây dựng bản xem trước trang tốt hơn và trình đọc màn hình có thể hướng lợi nhuận nhất bằng cách sử dụng đường viền để hướng dẫn người dùng gặp khó khăn về thị lực thông qua hệ thống phân cấp các phần và tiêu mục được lồng sâu vào nhau.

Tuy nhiên, không kích bản nào trong số này là có thật, bởi vì ngoại trừ một tập hợp nhỏ các công cụ dành cho nhà phát triển mà bạn sẽ xem xét trong phần tiếp theo-hầu như không ai sử dụng phác thảo HTML5 ngày nay.

GHI CHÚ

Thật khó để hào hứng với một tính năng không ảnh hưởng đến cách trình bày trang trong trình duyệt và không được các công cụ khác sử dụng. Tuy nhiên, bạn vẫn nên xem lại bối cảnh của các trang web (hoặc ít nhất là bối cảnh của một trang điển hình từ trang web của bạn) để đảm bảo rằng cấu trúc của trang đó hợp lý và bạn không vi phạm bất kỳ quy tắc HTML5 nào.

Làm thế nào để xem một phác thảo

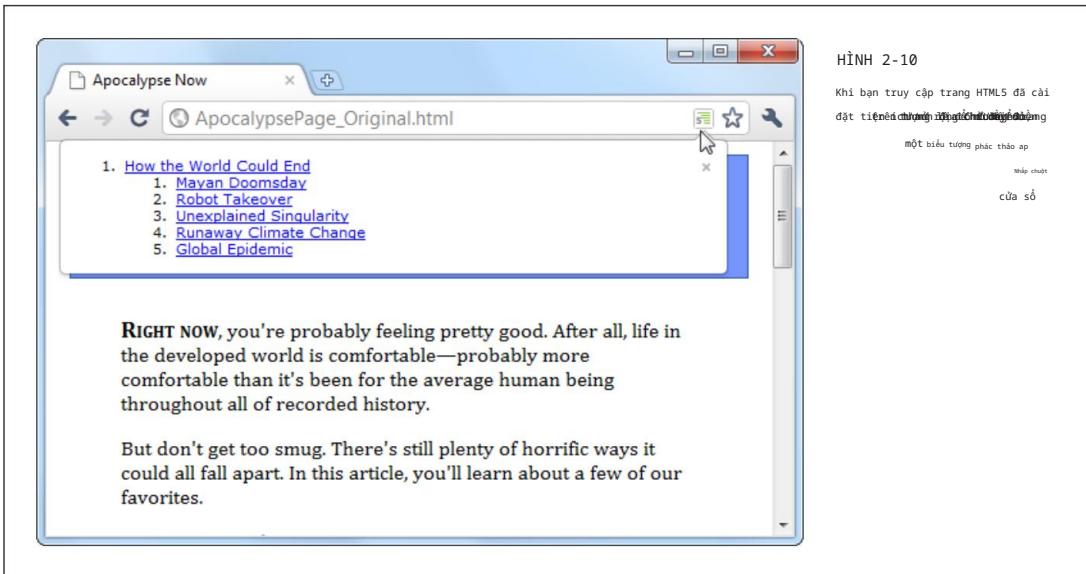
Để hiểu dàn bài, bạn chỉ cần xem dàn bài mà các trang của bạn tạo ra. Ngay bây giờ, không có trình duyệt nào thực hiện các quy tắc của HTML5 phác thảo (hoặc cung cấp cho bạn một cách để xem qua). Tuy nhiên, có một số công cụ lắp đầy khoảng trống:

- Trình soạn thảo HTML trực tuyến. Chuyển thăm <http://gsnedders.html5.org/outliner> và cho người phác thảo biết bạn muốn phác thảo trang nào. Như với trình xác thực HTML5 mà bạn đã sử dụng trong Chương 1 (trang 17), bạn có thể gửi trang mà bạn muốn phác thảo theo bất kỳ cách nào trong ba cách sau: bằng cách tải tệp lên từ máy tính của bạn, bằng cách cung cấp URL hoặc bằng cách dán phần đánh dấu vào tệp đánh dấu. hộp văn bản.
- Tiện ích mở rộng của Chrome. Bạn có thể sử dụng plugin h5o để phân tích đường viền của các trang khi bạn xem chúng trong Chrome. Cài đặt nó tại <http://code.google.com/p/h5o/> và sau đó lướt đến một trang HTML5 ở đâu đó trên Web (đáng tiếc là h5o không

làm việc với các tệp được lưu trữ trên máy tính của bạn). Một biểu tượng phác thảo xuất hiện trên thanh địa chỉ, hiển thị cấu trúc của trang khi được nhấp vào (Hình 2-10).

Trang h5o cũng cung cấp một (một đoạn mã JavaScript mà bạn có thể click chuột để phép bạn hiển thị đường viền trang trong Firefox và Internet Explorer, mặc dù có một số lỗi nhỏ.

- Phần mở rộng Opera. Có phiên bản Opera của tiện ích mở rộng h5o Chrome. Lấy nó ở <http://tinyurl.com/3k3ecdy>.



phác thảo cơ bản

Để trực quan hóa đường viền trang web của bạn, hãy tưởng tượng nó bị tước bỏ tất cả nội dung ngoại trừ văn bản trong thành phần tiêu đề được đánh số (`<h1>`, `<h2>`, `<h3>`, v.v.). Sau đó, thực lè các tiêu đề đó dựa trên vị trí của chúng trong đánh dấu của bạn, để các tiêu đề được lồng sâu hơn sẽ được thực lè nhiều hơn trong dàn bài.

Ví dụ: hãy xem xét bài viết về ngày tận thế ở trạng thái ban đầu, trước HTML5:

```
<body>
  <div class="Header">
    <h1>Thế giới có thể kết thúc như thế nào</h1>
    ...
  </div>
  ...
  <h2>Ngày tận thế của người Maya</h2>
  ...

```

```
<h2>Robot tiép quan</h2>
...
<h2>Ký dí khong giải thích đúợc</h2>
...
<h2>Biến đổi khí hậu đang diễn ra</h2>
...
<h2>Địch bệnh toàn cầu</h2>
...
<div class="Chân trang">
...
</div>
</body>
```

Cáu trúc đơn giản này dẫn đến một phác thảo như thế này:

1. Thế giới có thể kết thúc như thế nào
 1. Ngày tận thế của người Maya
 2. Robot tiép quan
 3. Điểm kí dí khong giải thích đúợc
 4. Biến đổi khí hậu chóng mặt
 5. Dịch bệnh toàn cầu

Hai cấp độ của tiêu đề (`<h1>` và `<h2>`) tạo ra một phác thảo hai cấp độ. Sơ đồ này tương tự như các tính năng phác thảo trong nhiều chương trình xử lý văn bản-ví dụ: bạn có thể thấy nhiều điều tương tự trong ngăn Điều hướng của Microsoft Word.

Mặt khác, đánh dấu như thế này:

```
<h1>Tiêu đề cấp 1</h1> <h2>Tiêu
đề cấp 2</h2> <h2>Tiêu đề cấp
2</h2> <h3>Tiêu đề cấp 3</h3>
<h2>Tiêu đề cấp 2 Tiêu đề</h2>
```

Nhận được một phác thảo như thế này:

1. Tiêu đề cấp 1
 1. Tiêu đề cấp 2
 1. Tiêu đề cấp 3
 2. Tiêu đề cấp 2

Một lần nữa, không có bất ngờ.

Cuối cùng, thuật toán phác thảo đủ thông minh để bỏ qua các cấp độ bị bỏ qua. Ví dụ: nếu bạn viết phần đánh dấu hơi lung tung này, phần đánh dấu này sẽ chuyển trực tiếp từ `<h1>` sang `<h3>`:

```
<h1>Tiêu đề cấp 1</h1>
<h2>Tiêu đề cấp 2</h2>
<h1>Tiêu đề cấp 1</h1>
<h3>Tiêu đề cấp 3</h3>
<h2>Tiêu đề cấp 2 Tiêu đề</h2>
```

Bạn nhận được phác thảo này:

1. Tiêu đề cấp 1
 1. Tiêu đề cấp 2
2. Tiêu đề cấp 1
 1. Tiêu đề cấp 3
2. Tiêu đề cấp 2

Giờ đây, tiêu đề cấp 3 có trạng thái cấp 2 trong dàn bài, dựa trên vị trí của nó trong tài liệu. Điều này có vẻ giống như một trong những trình duyệt sửa lỗi tự động thích thực hiện, nhưng nó thực sự phục vụ một mục đích thực tế. Trong một số trường hợp, một trang web có thể được tập hợp từ các phần riêng biệt-ví dụ: trang web có thể chứa bản sao của một bài báo đã được xuất bản ở nơi khác. Trong trường hợp này, các mức tiêu đề của nội dung được nhúng có thể không khớp hoàn hảo với phần còn lại của trang web. Nhưng vì thuật toán phác thảo làm mịn những khác biệt này, nên nó không có vẻ là một vấn đề.

Các phần tử chia phần Các phần

tử chia phần là những phần tử tạo ra một đường viền mới, được lồng vào bên trong trang của bạn: `<article>`, `<aside>`, `<nav>` và `<section>`. Để hiểu cách hoạt động của các phần tử phân đoạn, hãy tưởng tượng một trang có chứa hai phần tử `<article>`. Bởi vì `<bài viết>` là một thành phần phân đoạn, trang này có (ít nhất) ba dàn bài–đường viền của trang tổng thể và một dàn bài lồng nhau cho mỗi bài viết.

Để hiểu rõ hơn về tình huống này, hãy xem xét cấu trúc của bài báo về ngày tận thế, sau khi nó được sửa đổi bằng HTML5:

```
<nội
dung> <bài viết>
<tiêu đề>
<h1>Thế giới có thể kết thúc như thế nào</h1>
...
</tiêu đề>

<div class="Nội dung">
...
<h2>Ngày tận thế của người Maya</h2>
...

```

```

<h2>Robot tiếp quản</h2>
...
<h2>Kỳ dị không giải thích được</h2>
...
<h2>Biến đổi khí hậu đang diễn ra</h2>
...
<h2>Địch bệnh toàn cầu</h2>
...
</div>
</bài viết>

<chân trang>
...
</footer> </
body>
```

Cắm cái này vào trình xem phác thảo như <http://gsnedders.html5.org/outliner>, và bạn sẽ xem cái này:

1. Phần không có tiêu đề
 1. Thế giới có thể kết thúc như thế nào
 1. Ngày tận thế của người Maya
 2. Robot tiếp quản
 3. Điểm kỳ dị không giải thích được
4. Biến đổi khí hậu chóng mặt
5. Dịch bệnh toàn cầu

Ở đây, phác thảo bắt đầu với một phần không có tiêu đề, đó là phần tử gốc `<body>`.

Phần tử `<article>` bắt đầu một phác thảo mới, lồng nhau, chứa một phần tử `<h1>` và một vài phần tử `<h2>`.

Đôi khi, ghi chú "Phần không có tiêu đề" chỉ ra một lỗi. Mặc dù việc các phần tử `<aside>` và `<nav>` tồn tại mà không có tiêu đề được coi là chấp nhận được, nhưng sự khoan hồng tương tự thường không được trao cho các phần tử `<article>` hoặc `<section>`. Trong ví dụ trước, phần không có tiêu đề là phần chính của trang, thuộc về phần tử `<body>`. Bởi vì trang chứa một bài viết duy nhất, không có lý do gì để trang có một tiêu đề riêng và bạn có thể bỏ qua điều khó hiểu này.

Bây giờ, hãy xem điều gì sẽ xảy ra với một ví dụ phức tạp hơn, chẳng hạn như trang web về ngày tận thế với thanh bên điều hướng (trang 54). Đặt nó thông qua một phác thảo, và bạn sẽ nhận được phác thảo này:

1. tận thế Hôm nay
1. Phần không có tiêu đề
 1. Bài viết
 2. Về chúng tôi
 2. Thế giới có thể kết thúc như thế nào
 1. Ngày tận thế của người Maya
 2. Robot tiếp quản
 3. Phần không có tiêu đề
 4. Điểm kỳ dị không giải thích được
 5. Biển đổi khí hậu chóng mặt
 6. Dịch bệnh toàn cầu

Ở đây, có hai thành phần phân đoạn và hai đường viền lồng nhau: một cho thanh bên và một cho bài viết. Ngoài ra còn có hai phần không có tiêu đề, cả hai đều hợp pháp. Đầu tiên là phần tử `<aside>` cho thanh bên và thứ hai là phần tử `<aside>` đại diện cho trích dẫn kéo trong bài viết.

GHI CHÚ

Ngoài các phần tử phân chia, một số phần tử được gọi là **phần tử**. Những yếu tố này không chỉ là các nhánh của một phác thảo hiện có; họ bắt đầu một dàn bài mới của riêng mình mà không xuất hiện trong dàn bài chính của trang chứa. Phần tử `<body>` chứa nội dung trang web của bạn là phần gốc phân đoạn, điều này có ý nghĩa. Nhưng HTML5 cũng coi các phần tử sau là gốc phân đoạn: `<blockquote>`, `<td>`, `<fieldset>`, `<figure>` và `<details>`.

Giải quyết vấn đề về dàn ý Cho đến

giờ, bạn đã xem xét các ví dụ trong chương này và thấy các dàn ý mà chúng tạo ra. Và cho đến nay, các phác thảo đã có ý nghĩa hoàn hảo. Nhưng đôi khi, một vấn đề có thể xảy ra. Ví dụ, hãy tưởng tượng bạn tạo một tài liệu với cấu trúc này:

```
<nội
dung> <bài viết>
<h1>Những kỳ quan thiên nhiên sẽ ghé thăm trước khi bạn chết</h1>
...
<h2>Ở Bắc Mỹ</h2>
...
<h3>Hẻm núi lớn</h3>
...
<h3>Vườn quốc gia Yellowstone</h3>
...
<h2>Ở các nơi khác trên thế giới</h2>
...
<aside>...</aside>
```

```
...
<h3>Quần đảo Galapagos</h3>
...
<h3>Dãy núi Alps của Thụy Sĩ</h3>
...
</article>
</body>
```

ĐÁ QUÝ TRONG TUYẾT VỜI

Các phần tử trợ giúp như thế nào đối với các trang phức tạp

Phản đoạn là một trợ giúp tuyệt vời cho việc tìm kiếm và phân mềm khác khó xử lý hơn.

với hai ví dụ về nghệ thuật lấy nội dung từ một trang web và đưa nội dung đó vào một trang web khác.

Ví dụ: hãy tưởng tượng bạn có một trang web bao gồm các đoạn trích từ một số bài báo, tất cả đều được lấy từ các trang web khác. Bây giờ, hãy tưởng tượng rằng trang này có cấu trúc các tiêu đề được lồng sâu vào nhau và ở đâu đó bên trong -giả sử dưới tiêu đề <h4>-có một bài viết với nội dung được lấy từ một trang web khác.

Trong HTML truyền thống, bạn muốn tiêu đề đầu tiên trong nội dung này sử dụng phần tử <h5> vì nó được lồng trong <h4>. Nhưng bài viết này ban đầu được phát triển để đặt ở một nơi khác, trên một trang khác, ít lồng nhau hơn, do đó, nó có thể bắt đầu bằng <h2> hoặc <h1>. Trang vẫn hoạt động, nhưng hệ thống phân cấp của nó sẽ bị xáo trộn và trang có thể bị rỗng hơn-khiến bạn khó tự bắn vào chân mình hơn.

Trình đọc màn hình, công cụ tìm kiếm và phần mềm khác khó

Trong HTML5, trang này không phải là vấn đề. Miễn là bạn bọc bài viết lồng nhau trong phần tử <article>, nội dung được trích xuất sẽ trở thành một phần của dàn bài lồng nhau của chính nó. Dàn ý đó có thể bắt đầu với bất kỳ tiêu đề nào-không thành vấn đề. Điều quan trọng là vị trí của nó trong tài liệu chưa. Vì vậy, nếu phần tử <article> nằm sau <h4>, thì cấp đầu tiên của tiêu đề trong bài viết đó sẽ hoạt động giống như một logic <h5>, cấp thứ hai hoạt động giống như một logic <h6>, v.v.

Kết luận là thế này: HTML5 có một hệ thống phác thảo hợp lý giúp kết hợp các tài liệu dễ dàng hơn. Trong hệ thống phác thảo này, vị trí của các tiêu đề trở nên quan trọng hơn và mức độ chính xác của mỗi tiêu đề trở nên ít quan

Bạn có thể mong đợi một phác thảo như thế này:

1. Phần không có tiêu đề cho <body>

1. Những ký quan thiên nhiên đến thăm trước khi bạn chết

1. Ở Bắc Mỹ

1. Hẻm núi lớn

2. Công viên quốc gia Yellowstone

2. Ở phần còn lại của thế giới

3. Phần không có tiêu đề cho <aside>

1. Quần đảo Galapagos

2. Dãy núi Alps của Thụy Sĩ

Nhưng phác thảo bạn thực sự nhận được là thẻ này:

1. Phần không có tiêu đề cho `<body>`
 1. Những kỳ quan thiên nhiên đến thăm trước khi bạn chết
 1. Ở Bắc Mỹ
 1. Hẻm núi lớn
 2. Công viên quốc gia Yellowstone
 2. Ở phần còn lại của thế giới
 3. Phần không có tiêu đề cho `<aside>`
 4. Quần đảo Galapagos
5. Dãy núi Alps của Thụy Sĩ

Bằng cách nào đó, việc thêm `<aside>` sau `<h2>` loại bỏ các phần tử `<h3>` sau, làm cho chúng có cùng mức logic với `<h2>`. Đây rõ ràng không phải là những gì bạn muốn.

Để giải quyết vấn đề này, trước tiên bạn cần hiểu rằng hệ thống dàn bài HTML5 sẽ tự động tạo một phần mới mỗi khi nó tìm thấy phần tử tiêu đề được đánh số (như `<h1>`, `<h2>`, `<h3>`, v.v.), phần tử đó là đã ở trên cùng ~~trên~~^{tên} một phần.

Trong ví dụ này, hệ thống phác thảo không làm gì với phần tử `<h1>` ban đầu, bởi vì nó ở đầu phần `<article>`. Nhưng thuật toán phác thảo sẽ tạo các phần mới cho các phần tử `<h2>` và `<h3>` theo sau. Như thể bạn đã viết đánh dấu này:

```

<nội
dung> <bài viết>
<h1>Những kỳ quan thiên nhiên sẽ ghé thăm trước khi bạn chết</h1>
...
<phần>
<h2>Ở Bắc Mỹ</h2>
...
<phần>
<h3>Hẻm núi lớn</h3>
...
</phần>
<phần>
<h3>Vườn quốc gia Yellowstone</h3>
...
</phần> <
phần>

```

```
<phần>
<h2>Ở các nơi khác trên thế giới</h2>
...
</section>
<aside>...</aside>
...
<phần>
<h3>Quần đảo Galapagos</h3>
...
</phần>
<phần>
<h3>Dãy núi Alps của Thụy Sĩ</h3>
...
</section>
</article> </
body>
```

Hầu hết thời gian, các phần được tạo tự động này không phải là vấn đề. Trên thực tế, chúng thường là một tài sản, bởi vì chúng đảm bảo rằng các tiêu đề được đánh số không chính xác vẫn được đặt ở cấp độ phác thảo bên phải. Cái giá phải trả cho sự thuận tiện này là một trực tiếp không thường xuyên, giống như sự cố được hiển thị ở đây.

Như bạn có thể thấy trong danh sách này, mọi thứ lúc đầu diễn ra suôn sẻ. Phần `<h1>` trên cùng được để yên (vì nó đã ở trong một `<bài viết>`), có một tiêu mục được tạo cho `<h2>` đầu tiên, sau đó là một tiêu mục cho mỗi `<h3>` bên trong, v.v. Sự cố bắt đầu khi thuật toán phác thảo chạy vào phần tử `<aside>`. Nó coi đây là gợi ý để đóng phần hiện tại, nghĩa là khi các phần được tạo cho các phần tử `<h3>` sau, chúng ở cùng mức logic như các phần tử `<h2>` trước đó.

Để khắc phục sự cố này, bạn cần kiểm soát các phần và tiêu mục bằng cách tự xác định một số. Trong ví dụ này, mục tiêu là để ngăn phần thứ hai `<h2>` bị đóng quá sớm, bạn có thể thực hiện điều này bằng cách xác định rõ ràng phần đó trong phần đánh dấu:

```
<nội dung> <bài viết>
<h1>Những kỳ quan thiên nhiên sẽ ghé thăm trước khi bạn chết</h1>
...
<h2>Ở Bắc Mỹ</h2>
...
<h3>Hòn núi lớn</h3>
...
<h3>Vườn quốc gia Yellowstone</h3>
...
<phần>
<h2>Ở các nơi khác trên thế giới</h2>
...
<aside>...</aside>
...
```

```
<h3>Quần đảo Galapagos</h3>
...
<h3>Dãy núi Alps của Thụy Sĩ</h3>
...
</section>
</article> </
body>
```

Bây giờ, thuật toán phác thảo không cần phải tạo một phần tự động cho `<h2>` thứ hai, và do đó, không có nguy cơ nó sẽ đóng phần đó khi nó tình cờ gặp `<aside>`. Mặc dù bạn có thể xác định phần cho mọi tiêu đề trong tài liệu này, nhưng không cần phải làm lộn xộn phần đánh dấu của bạn, vì thay đổi đơn lẻ này sẽ khắc phục được sự cố.

GHI CHÚ

Một giải pháp khác là thay thế `<aside>` bằng `<div>`. `<div>` không phải là một thành phần phân đoạn, vì vậy nó sẽ không khiến một phần bị đóng đột ngột.

Việc sử dụng phần tử `<aside>` không phải lúc nào cũng gây ra sự cố này. Các ví dụ trong bài viết trước đã sử dụng phần tử `<aside>` cho trích dẫn kéo nhưng vẫn hoạt động tốt vì `<aside>` nằm giữa hai phần tử `<h2>`. Nhưng nếu bạn bất cẩn đặt một yếu tố phân chia giữa hai cấp độ tiêu đề khác nhau, thì bạn nên kiểm tra dàn bài của mình để đảm bảo rằng nó vẫn có ý nghĩa.

TIỀN BOA

Nếu toàn bộ khái niệm phác thảo có vẻ quá lý thuyết, đừng lo lắng. Thành thật mà nói, đó là một khái niệm tệ nhỉ mà nhiều nhà phát triển web sẽ bỏ qua (ít nhất là bây giờ). Cách tiếp cận tốt nhất là coi hệ thống phác thảo HTML5 như một công cụ đảm bảo chất lượng có thể giúp bạn. Nếu bạn xem lại các trang của mình trong một trình tạo dàn ý (như một trong những công cụ được liệt kê ở trang 65), bạn có thể phát hiện ra những lỗi có thể chỉ ra các vấn đề khác và đảm bảo rằng bạn đang sử dụng đúng các yếu tố ngữ nghĩa.

Viết thêm

Đánh dấu có ý nghĩa

Trong chương trước, bạn đã gặp các phần tử ngữ nghĩa của HTML5. Với sự giúp đỡ của trình duyệt, công cụ tìm kiếm và thiết bị trợ giúp siêu thông minh.

Nhưng bạn vẫn chưa đi đến cuối câu chuyện ngữ nghĩa. Ngữ nghĩa là tất cả về việc thêm vào đánh dấu của bạn và có một số [trang](#) thông tin để bạn bách khoa thư đọc của nó. Khi đó, chúng ta có thể và sẽ làm điều đó bằng cách sử dụng [trang](#) của chúng.

trúc trang cấu

Nhưng ngữ nghĩa cũng có thể bao gồm những [ngôn ngữ](#) khác nhau để giải thích các phần nội dung nhỏ hơn nhiều. Bạn có thể sử dụng ngữ nghĩa ở cấp độ văn bản để chỉ ra các loại thông tin quan trọng có thể bị mất trong biến nội dung trang web, chẳng hạn như tên, địa chỉ, danh sách sự kiện, sản phẩm, công thức nấu ăn, đánh giá hàng, v.v.

Sau đó, nội dung này có thể được trích xuất và sử dụng bởi một loạt các dịch vụ khác nhau—mọi thứ từ phần hỗ trợ tiện lợi của trình duyệt cho đến các công cụ tìm kiếm chuyên dụng.

Trong chương này, bạn sẽ bắt đầu bằng cách quay lại tập hợp nhỏ các phần tử ngữ nghĩa được tích hợp sẵn trong ngôn ngữ HTML5. Bạn sẽ tìm hiểu về một số yếu tố ngữ nghĩa ở cấp độ văn bản mà ngày nay bạn có thể sử dụng một cách dễ dàng. Tiếp theo, bạn sẽ xem xét các tiêu chuẩn đồng hành giải quyết trực tiếp ngữ nghĩa ở cấp độ văn bản. Điều đó có nghĩa là đào sâu vào những gì bắt đầu tồn tại như một phần của [HTML5](#) và [W3C](#) quản lý. Sử dụng ví dụ liệu, bạn sẽ học cách làm phong phú các trang của mình và tăng cường danh sách tìm kiếm trên web của mình.

Các yếu tố ngữ nghĩa được xem xét lại

Có một lý do khiến bạn bắt đầu khám phá ngữ nghĩa với các thành phần cấu trúc trang (xem Bảng 3-1 để biết tóm tắt). Rất đơn giản, cấu trúc trang là một thử thách dễ dàng. Đó là bởi vì phần lớn các trang web sử dụng một tập hợp nhỏ các yếu tố thiết kế phổ biến (đầu trang, chân trang, thanh bên và menu) để tạo bố cục về tất cả sự khác biệt về mặt thẩm mỹ-rất giống nhau.

BẢNG 3-1 Các yếu tố ngữ nghĩa cho cấu trúc trang

YẾU TỐ	SỰ MÔ TẢ
<bài viết>	Đại diện cho bất cứ điều gì bạn coi là một bài viết-một phần nội dung độc lập như một bài báo, một bài đăng trên diễn đàn hoặc một mục blog (không bao gồm những thứ rườm rà như bình luận hoặc tiêu đề tác giả).
<sang một bên>	Đại diện cho một đoạn nội dung hoàn chỉnh tách biệt với nội dung trang chính. Ví dụ: bạn nên sử dụng <aside> để tạo thanh bên có nội dung liên quan hoặc liên kết bên cạnh bài viết chính.
<hình> và <chú thích>	Đại diện cho một con sói. Phần tử <figcaption> bao bọc văn bản chú thích và phần tử <figure> bao bọc phần tử <figcaption> và cho chính ảnh. Mục đích là để chỉ ra mối liên hệ giữa một hình ảnh và chú thích của nó.
<chân trang>	Đại diện cho chân trang ở dưới cùng của trang. Đây là một đoạn nội dung nhỏ có thể bao gồm chữ in nhô, thông báo bản quyền và một bộ liên kết ngắn gọn (ví dụ: "Giới thiệu về chúng tôi" hoặc "Nhận hỗ trợ").
<tiêu đề>	Thể hiện tiêu đề nâng cao bao gồm tiêu đề HTML chuẩn và nội dung bổ sung. Nội dung bổ sung có thể bao gồm biểu trưng, dòng tên người dùng hoặc tập hợp các liên kết diều hướng cho nội dung tiếp theo.
<diều hướng>	Đại diện cho một tập hợp đáng kể các liên kết trên một trang. Các liên kết này có thể trỏ đến các chủ đề trên trang hiện tại hoặc các trang khác trên trang web. Trên thực tế, không có gì lạ khi có một trang có nhiều phần <nay>.
<phần>	Đại diện cho một phần của tài liệu hoặc một nhóm tài liệu. <section> là một vùng chứa đa năng với một quy tắc duy nhất: Nội dung mà nó chứa phải bắt đầu bằng một tiêu đề. Chỉ sử dụng <section> nếu các thành phần ngữ nghĩa khác (ví dụ: <article> và <aside>) không áp dụng.
<chính>	Đại diện cho nội dung chính của trang-tất cả nội dung đó. Ví dụ: <main> có thể bọc một phần tử <article> nhưng bỏ qua các tiêu đề, chân trang và thanh bên trên toàn trang web. Phần tử <main> là phần bổ sung mới cho bản sửa đổi HTML 5.1 của HTML5 (trang xv).

Ngữ nghĩa ở cấp độ văn bản là một vấn đề khó giải quyết hơn. Đó là bởi vì mọi người sử dụng một số lượng lớn các loại nội dung khác nhau. Nếu HTML5 bắt đầu tạo một phần tử cho mọi loại thông tin mà bạn có thể thêm vào một trang, thì ngôn ngữ đó sẽ bị lấn lőn trong một mớ hỗn độn các phần tử. Vấn đề phức tạp hơn là thông tin có cấu trúc cũng được tạo thành từ các phần nhỏ hơn có thể được lắp ráp theo những cách khác nhau.

Ví dụ: ngay cả một địa chỉ bưu chính thường cũng yêu cầu một số phần tử (như `<địa chỉ>`, `<tên>`, `<đường phố>`, `<mã bưu chính>`, `<quốc gia>`, v.v.) trước khi bất kỳ ai có thể sử dụng địa chỉ đó trong một trang.

HTML5 có cách tiếp cận hai hướng. Đầu tiên, nó bổ sung một số lượng rất nhỏ các yếu tố ngữ nghĩa ở cấp độ văn bản. Nhưng thứ hai, và quan trọng hơn, HTML5 hỗ trợ một tiêu chuẩn vi dữ liệu riêng biệt, cung cấp cho mọi người một cách có thể mở rộng để xác định bất kỳ loại thông tin nào họ muốn và sau đó gắn cờ thông tin đó trong các trang của họ. Bạn sẽ đẽ cập đến cả hai chủ đề này trong chương này. Đầu tiên là ba thành phần ngữ nghĩa cấp độ văn bản mới: `<time>`, `<output>` và `<mark>`.

Ngày và Giờ với `<thời gian>`

Thông tin ngày và giờ xuất hiện thường xuyên trong các trang web. Ví dụ, nó xuất hiện ở cuối hầu hết các bài đăng trên blog. Thật không may, không có cách tiêu chuẩn hóa để gắn thẻ ngày, vì vậy không có cách nào dễ dàng để các chương trình khác (như công cụ tìm kiếm) trích xuất chúng mà không cần đoán. Phần tử `<time>` giải quyết vấn đề này. Nó cho phép bạn đánh dấu ngày, giờ hoặc ngày và giờ kết hợp.

Đây là một ví dụ:

Bữa tiệc bắt đầu `<time>21-03-2014</time>`.

GHI CHÚ

Có vẻ hơi phản trực giác khi có một phần tử `<time>` bao bọc một ngày (không có thời gian), nhưng đó chỉ là một trong những điểm đặc biệt của HTML5. Tên thành phần hợp lý hơn sẽ là `<datetime>`, nhưng đó không phải là tên họ đã chọn.

Phần tử `<time>` thực hiện hai vai trò. Đầu tiên, nó cho biết vị trí của giá trị ngày hoặc giờ trong đánh dấu của bạn. Thứ hai, nó cung cấp giá trị ngày hoặc giờ ở dạng mà bất kỳ chương trình phần mềm nào cũng có thể hiểu được. Ví dụ trước đây ứng yêu cầu thứ hai bằng cách sử dụng định dạng ngày chung, bao gồm năm có bốn chữ số, tháng có hai chữ số và ngày có hai chữ số, theo thứ tự đó, với mỗi phần được phân tách bằng dấu gạch ngang. Nói cách khác, định dạng tuân theo mẫu này:

YYYY-MM-DD

Tuy nhiên, hoàn toàn có thể chấp nhận việc hiển thị ngày tháng theo một cách khác đối với người đang đọc trang web của bạn. Trên thực tế, bạn có thể sử dụng bất kỳ văn bản nào bạn muốn, miễn là bạn cung cấp ngày phổ quát mà máy tính có thể đọc được với thuộc tính `datetime`, như sau:

Bữa tiệc bắt đầu `<time datetime="2014-03-21">Ngày 21 tháng 3st</time>`.

Trong như thế này trong trình duyệt:

Bữa tiệc bắt đầu vào ngày 21 tháng 3.

Phần tử `<time>` có các quy tắc tương tự về thời gian mà bạn cung cấp ở định dạng sau:

Giờ: MM

Đó là giờ có hai chữ số (sử dụng đồng hồ 24 giờ), theo sau là số phút có hai chữ số, như sau:

Các bữa tiệc bắt đầu hàng đêm lúc `<time datetime="16:30">4:30 chiều</time>`.

Cuối cùng, bạn có thể chỉ định thời gian vào một ngày cụ thể bằng cách kết hợp hai tiêu chuẩn này. Chỉ cần đặt ngày đầu tiên, theo sau là khoảng trắng và sau đó là thông tin về thời gian.

Bữa tiệc bắt đầu `<time datetime="2014-03-21 16:30">Ngày 21 tháng 3st lúc 4:30 chiều</time>`.

GHI CHÚ

Ban đầu, phần tử `<time>` yêu cầu một định dạng hơi khác để kết hợp thông tin ngày và giờ. Thay vì phân tách hai thành phần bằng dấu cách, bạn phải phân tách chúng bằng chữ hoa (đối với), như trong `2014-03-21T16:30`. Định dạng này vẫn được chấp nhận, vì vậy bạn có thể gặp nó khi duyệt các trang web của người

Khi kết hợp ngày và giờ, bạn có thể chọn giải quyết phần bù múi giờ ở cuối. Ví dụ: New York thuộc múi giờ miền Đông, được gọi là UTC-5:00.

(Bạn có thể tìm ra múi giờ của mình tại .) Để chèn http://en.wikipedia.org/wiki/Time_zone vào đây:

Bữa tiệc bắt đầu `<time datetime="2014-03-21 16:30-05:00">Ngày 21 tháng 3st lúc 4:30 chiều</time>`.

Bằng cách này, những người đọc trang của bạn nhận được thời gian ở định dạng mà họ mong đợi, trong khi các bot tìm kiếm và các phần mềm khác nhận được giá trị ngày giờ rõ ràng mà họ có thể xử lý.

Phần tử `<time>` cũng hỗ trợ thuộc tính `pubdate`. Bạn nên sử dụng điều này nếu ngày của bạn tương ứng với ngày xuất bản của nội dung hiện tại (ví dụ: `<bài viết>` trong đó `<thời gian>` được đặt). Đây là một ví dụ:

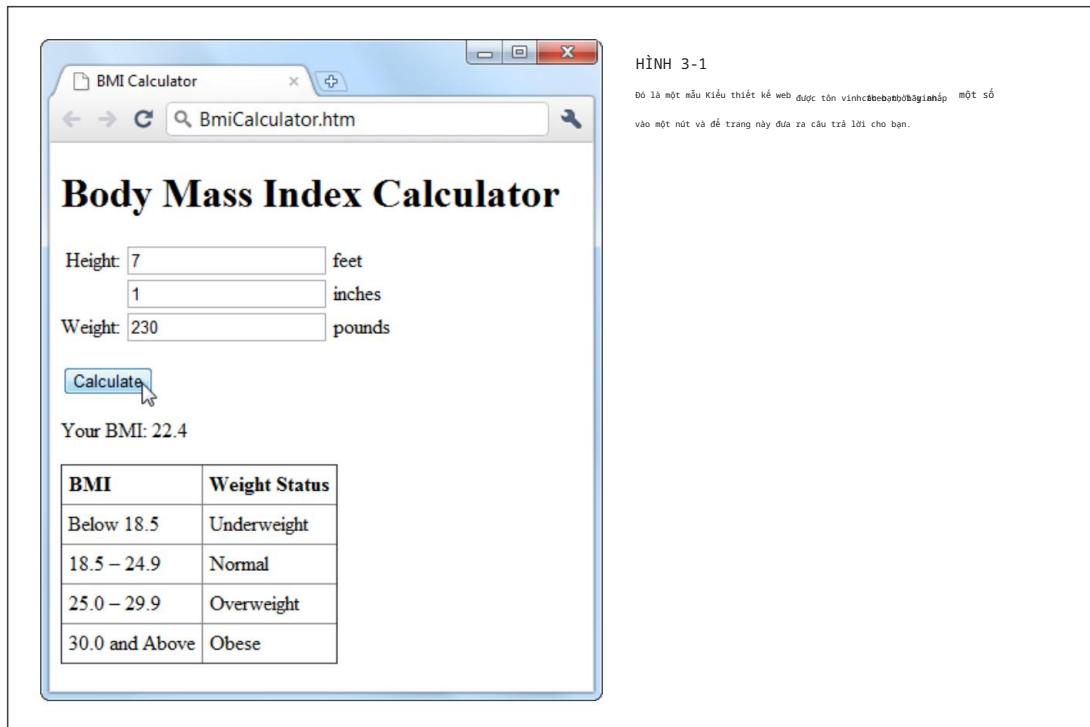
Được đăng trên `<time datetime="2014-03-21" pubdate>ngày 31 tháng 3 năm 2014</time>`.

GHI CHÚ

Vì phần tử `<time>` hoàn toàn là thông tin và không có bất kỳ định dạng liên quan nào nên bạn có thể sử dụng phần tử này với bất kỳ trình duyệt nào. Không có vấn đề tương thích để lo lắng về. Nhưng nếu bạn muốn tạo kiểu cho phần tử `<time>`, bạn cần có giải pháp Internet Explorer được mô tả ở trang 51.

Tính toán JavaScript với `<đầu ra>` HTML5 bao gồm một phần tử ngữ nghĩa được thiết kế để làm cho một số loại trang hỗ trợ JavaScript rõ ràng hơn một chút – phần tử `<đầu ra>`. Nó không gì khác hơn là một trình giữ chỗ mà mã của bạn có thể sử dụng để hiển thị một phần thông tin được tính toán.

Ví dụ, hãy tưởng tượng bạn tạo một trang như trong Hình 3-1. Con số này cho phép người dùng nhập một số thông tin. Sau đó, một tập lệnh lấy thông tin này, thực hiện phép tính và hiển thị kết quả ngay bên dưới.



Cách thông thường để giải quyết vấn đề này là gán một ID duy nhất cho trình giữ chỗ để mã JavaScript có thể tìm thấy nó khi thực hiện phép tính. Thông thường, các nhà phát triển web sử dụng phần tử ``, phần tử hoạt động hoàn hảo nhưng không cung cấp bất kỳ ý nghĩa cụ thể nào:

```
<p>BMI của bạn: <span id="result"></p>
```

Đây là phiên bản có ý nghĩa hơn mà bạn sẽ sử dụng trong HTML5:

```
<p>Chỉ số BMI của bạn: <output id="result"></output></p>
```

Mã JavaScript thực tế không cần bất kỳ thay đổi nào, bởi vì nó tra cứu phần tử theo tên và không quan tâm đến loại phần tử:

```
var resultElement = document.getElementById("kết quả");
```

GHI CHÚ

Trước khi bạn sử dụng `<đầu ra>`, hãy đảm bảo rằng bạn đã bao gồm giải pháp Internet Explorer được mô tả ở trang 51. Nếu không, phần tử này sẽ không thể truy cập được bằng JavaScript trên các phiên bản Internet Explorer cũ (IE 8 trở về trước).

Thông thường, loại trang này có các điều khiển bên trong phần tử `<form>`. Trong ví dụ này, đó là ba hộp văn bản nơi mọi người có thể nhập thông tin:

```
<form action="#" id="bmiCalculator">
  <label for="feet inches">Chiều cao:</label>
  <input name="feet"> feet<br> <input
  name="inches"> inch<br>

  <label for="pounds">Cân nặng:</label>
  <input name="pounds"> pound<br><br>
  ...
</form>
```

Nếu bạn muốn làm cho phần tử `<output>` trông thông minh hơn nữa, bạn có thể thêm thuộc tính biểu mẫu (cho biết ID của biểu mẫu có các điều khiển liên quan) và thuộc tính `for` (liệt kê ID của các điều khiển liên quan, được phân tách bằng khoảng trắng).

Đây là một ví dụ:

```
<p>Chi số BMI của bạn: <output id="result" form="bmiCalculator" for="feet inches pounds"> </
  output></p>
```

Các thuộc tính này không thực sự làm bắt cứ điều gì, ngoài việc truyền tải thông tin về nơi mà phần tử `<output>` của bạn nhận được hàng hóa của nó. Nhưng họ sẽ kiểm cho bạn một số điểm bánh hạnh nhân ngữ nghĩa nghiêm túc. Và nếu những người khác cần chỉnh sửa trang của bạn, những thuộc tính này có thể giúp họ sắp xếp cách thức hoạt động của nó.

TIỀN BOA

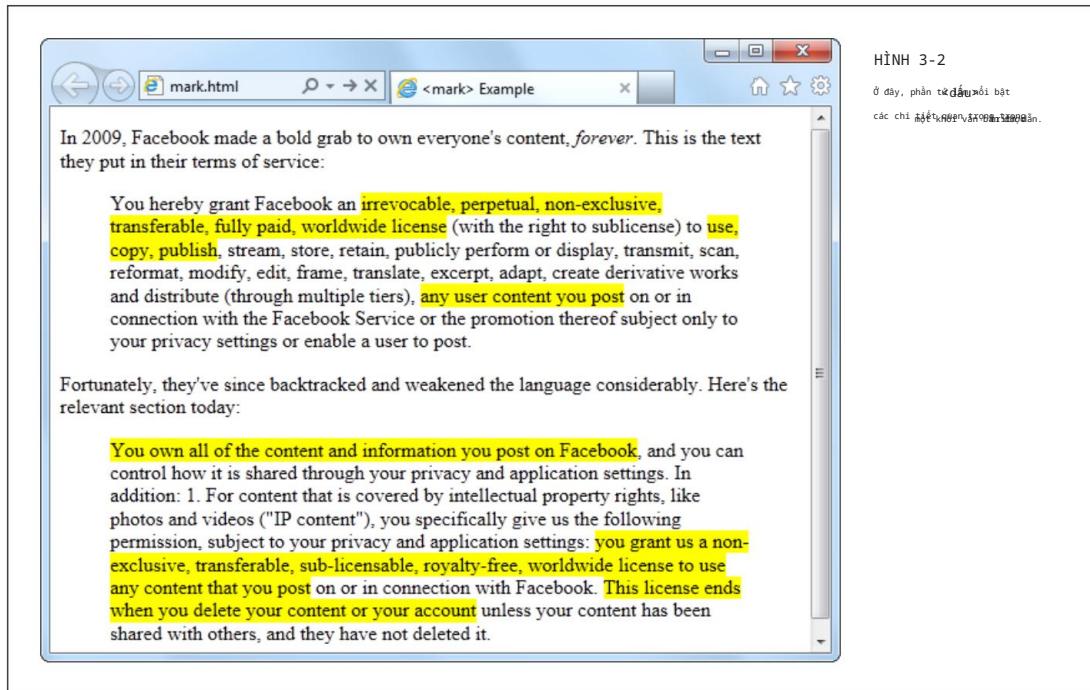
Nếu bạn hơi mơ hồ về các biểu mẫu, bạn sẽ tìm hiểu thêm trong Chương 4. Nếu bạn biết nhiều về Esperanto hơn JavaScript, bạn có thể tìm hiểu thêm về ngôn ngữ lập trình trong Phụ lục B, "JavaScript: Bộ não của trang của bạn". Và nếu bạn muốn tự mình dùng thử trang này, bạn có thể tìm thấy ví dụ hoàn chỉnh tại <http://prosetech.com/html5>.

Văn bản được đánh dấu bằng `<mark>` Phần tử `<mark>`

đại diện cho một phần văn bản được đánh dấu để tham khảo. Nó đặc biệt thích hợp khi bạn đang trích dẫn văn bản của người khác và bạn muốn thu hút sự chú ý vào điều gì đó:

```
<p>Vào năm 2009, Facebook đã thực hiện một bước táo bạo để sở hữu nội dung
  của mọi người, <em>mỗi mài</em>. Đây là văn bản họ đưa vào điều khoản dịch vụ của mình:</
  p> <blockquote>Bạn cấp cho Facebook <mark> giấy phép toàn cầu không thể thu hồi, vĩnh
  viễn, không độc quyền, có thể chuyển nhượng, được thanh toán đầy đủ</mark> (có quyền cung
  cấp phép con) để <mark>sử dụng, sao chép, xuất bản</mark>, phát trực tuyến, lưu trữ, giữ lại,
  thực hiện hoặc hiển thị công khai, truyền, quét, định dạng lại, sửa đổi, chỉnh sửa, đóng
  khung, dịch, trích đoạn, điều chỉnh, tạo các tác phẩm phái sinh và phân phối (through qua
  nhiều tầng), <mark>bất kỳ nội dung người dùng nào bạn đăng</mark>
  ...
</blockquote>
```

Văn bản trong phần tử `<mark>` có nền màu vàng như trong Hình 3-2.



Bạn cũng có thể sử dụng `<mark>` để gắn cờ nội dung hoặc từ khóa quan trọng, như các công cụ tìm kiếm thực hiện khi hiển thị văn bản phù hợp trong kết quả tìm kiếm của bạn hoặc để đánh dấu các thay đổi của tài liệu, kết hợp với `` (đối với văn bản đã xóa) và `<ins>` (cho văn bản được chèn).

Thành thật mà nói, phần tử `<mark>` hơi không phù hợp. Đặc tả HTML5 coi nó là một phần tử ngữ nghĩa, nhưng nó đóng vai trò trình bày được cho là quan trọng hơn. Theo mặc định, văn bản đã đánh dấu được đánh dấu bằng nền màu vàng sáng (Hình 3-2), mặc dù bạn có thể áp dụng các quy tắc biểu định kiểu của riêng mình để sử dụng hiệu ứng định dạng khác.

TIẾN BỘ

Phần tử `<mark>` không thực sự là về định dạng. Xét cho cùng, có rất nhiều cách để làm cho văn bản nổi bật trên một trang web. Thay vào đó, bạn nên sử dụng `<mark>` (kết hợp với bất kỳ định dạng CSS nào bạn thích) khi nó phù hợp về mặt ngữ nghĩa. Một nguyên tắc nhỏ là sử dụng `<mark>` để thu hút sự chú ý đến văn bản thông thường có ý nghĩa quan trọng, do ~~cú pháp~~ luận tạo khung cho nó hoặc do tác vụ mà người dùng đang thực hiện.

Ngay cả khi bạn sử dụng định dạng nền màu vàng mặc định, bạn vẫn nên thêm dự phòng biểu định kiểu cho các trình duyệt không hỗ trợ HTML5. Đây là loại quy tắc phong cách bạn cần:

KHÁC
TIÊU CHUẨN
SỰ TĂNG CƯỜNG
NGỮ NGHĨA

dánh

```
dấu { màu nền: vàng; màu
đen;
}
```

Bạn cũng sẽ cần giải pháp thay thế của Internet Explorer được mô tả ở trang 51 để làm cho phần tử <mark> có thể định kiểu trong các phiên bản IE cũ.

Các tiêu chuẩn khác thúc đẩy ngữ nghĩa

Tại thời điểm này, bạn có thể nhận ra rằng có rất nhiều yếu tố ngữ nghĩa tiềm năng mà HTML có. Chắc chắn, bạn có thể gán cờ ngày tháng và văn bản để xác định rõ ràng tên phần mềm hoặc cá nhân, v.v. thì sao? HTML5 có tình không lấn sân sang lĩnh vực này, bởi vì những người tạo ra nó không muốn làm ngôn ngữ này sa lầy với hàng tá yếu tố chuyên biệt phù hợp với một số người nhưng lại khiến những người khác thấy nhảm chán và không có ảnh hưởng. Để thực sự đạt đến cấp độ tiếp theo với ngữ nghĩa, bạn cần mở rộng tìm kiếm của mình ra ngoài ngôn ngữ HTML5 cốt lõi và xem xét một vài tiêu chuẩn có thể hoạt động với các trang web của bạn.

Dánh dấu thông minh về mặt ngữ nghĩa không phải là một ý tưởng mới. Trên thực tế, khi HTML5 vẫn chỉ là một ảo mộng trong đầu của Ian Hickson, biên tập viên WHATWG, đã có rất nhiều nhà phát triển web kêu gọi tìm cách làm cho phần đánh dấu của họ có ý nghĩa hơn. Mục tiêu của họ không phải lúc nào cũng giống nhau – một số muốn tăng cường khả năng tiếp cận, một số dự định khai thác dữ liệu và những người khác chỉ muốn nhấn mạnh yếu tố thú vị trong sơ yếu lý lịch của họ.

Nhưng không ai trong số họ có thể tìm thấy những gì họ muốn trong ngôn ngữ HTML tiêu chuẩn, đó là lý do tại sao một số thông số kỹ thuật mới xuất hiện để lấp đầy khoảng trống.

Trong các phần sau, bạn sẽ tìm hiểu về không ít hơn Đầu tiên, bạn sẽ nhận bốn của các tiêu chuẩn này. Được thông tin sót đẹo về ARIA, một tiêu chuẩn tập trung vào việc cải thiện khả năng truy cập cho trình đọc màn hình. Sau đó, bạn sẽ xem qua ba cách tiếp cận cạnh tranh để mô tả các loại nội dung khác nhau, cho dù đó là chi tiết liên hệ, địa chỉ, danh sách doanh nghiệp hay bất kỳ thứ gì khác mà bạn có thể đặt giữa các thẻ của trang HTML.

ARIA (Ứng dụng Internet phong phú có thể truy cập)

ARIA là một tiêu chuẩn đang phát triển cho phép bạn cung cấp thêm thông tin cho trình đọc màn hình thông qua các thuộc tính trên bất kỳ phần tử HTML nào. Ví dụ: ARIA giới thiệu thuộc tính vai trò, cho biết mục đích của một phần tử nhất định. Ví dụ: nếu bạn có <div> đại diện cho tiêu đề:

```
<div class="header">
```

Bạn có thể thông báo sự thật đó cho người đọc màn hình bằng cách đặt thuộc tính vai trò ARIA thành biến ngữ:

```
<div class="header" role="banner">
```

Tất nhiên, bạn đã học chương trước rằng HTML5 cũng cung cấp cho bạn một cách có ý nghĩa hơn để đánh dấu các tiêu đề. Vì vậy, những gì bạn thực sự nên sử dụng là một cái gì đó như thế này:

```
<header role="banner">
```

Ví dụ này cho thấy hai sự thật quan trọng. Đầu tiên, ARIA yêu cầu bạn sử dụng một trong danh sách ngắn các tên vai trò được đề xuất. (Để biết danh sách đầy đủ, hãy tham khảo phần thích hợp của thông số kỹ thuật tại .) Thứ hai, các phần của ARIA phủ lên các phần tinyurl.com/mofoes-aria với điều HTML5. Nhưng sự chồng chéo không hoàn thành. Ví dụ: một số tên vai trò trùng lặp với HTML5 (như biểu ngữ và bài viết), trong khi những tên khác đi xa hơn (như thanh công cụ và tìm kiếm).

ARIA cũng thêm hai thuộc tính hoạt động với biểu mẫu HTML. Aria-required at tribution trong hộp văn bản cho biết rằng người dùng cần nhập một giá trị. Thuộc tính aria-invalid trong hộp văn bản cho biết giá trị hiện tại không đúng. Các thuộc tính này rất hữu ích vì trình đọc màn hình có khả năng bỏ lỡ các dấu hiệu trực quan mà người dùng nhìn thấy dựa vào, chẳng hạn như dấu hoa thị bên cạnh trường bị thiếu hoặc biểu tượng lỗi nhập nháy màu đỏ.

Để áp dụng ARIA đúng cách, bạn cần tìm hiểu tiêu chuẩn và dành thời gian xem lại phần đánh dấu của mình. Các nhà phát triển web bị chia rẽ về việc liệu đó có phải là một khoản đầu tư đáng giá hay không, vì tiêu chuẩn này vẫn đang phát triển và HTML5 cung cấp một số lợi ích tương tự với ít rắc rối hơn. Tuy nhiên, nếu bạn muốn tạo một trang web thực sự có thể truy cập ngay hôm nay, bạn cần sử dụng cả hai, vì các trình đọc màn hình mới hơn hỗ trợ ARIA nhưng chưa hỗ trợ HTML5.

GHI CHÚ

Để biết thêm thông tin về ARIA (có tên đầy đủ là WAI-ARIA, vì nó được phát triển bởi Web nhóm Sáng kiến Khả năng Tiếp cận), bạn có thể đọc thông số kỹ thuật tại www.w3.org/TR/wai-aria.

RDFa (Khung mô tả tài nguyên)

RDFa là tiêu chuẩn để nhúng siêu dữ liệu chi tiết vào tài liệu web của bạn bằng các thuộc tính. RDFa có một lợi thế đáng kể: Không giống như các phương pháp tiếp cận khác được thảo luận trong chương này, đây là một tiêu chuẩn ổn định, ổn định. RDFa cũng có hai nhược điểm đáng kể. Đầu tiên, RDFa ban đầu được thiết kế cho XHTML, không phải HTML5. Đó là một vấn đề tranh luận về việc cú pháp RDFa chặt chẽ hơn, phức tạp hơn kết hợp tốt như thế nào với triết lý tự do hơn của HTML5. Thứ hai, RDFa rất phức tạp. Đánh dấu được tăng cường bằng siêu dữ liệu RDFa dài hơn và công kenne hơn đáng kể so với HTML thông thường. Và do tính phức tạp của nó, RDFa cũng có nhiều khả năng chứa lỗi hơn–khả năng cao gấp ba lần, theo một cuộc khảo sát trang web gần đây của Google.

RDFa không được thảo luận trong chương này, mặc dù bạn sẽ tìm hiểu sâu hơn về vi dữ liệu, vi dữ liệu, họ hàng HTML5, ở trang 85. Nhưng nếu bạn muốn tìm hiểu thêm về RDFa, bạn có thể xem phần giới thiệu chi tiết trên Wikipedia tại hoặc bạn [kết nối truy cập wikipedia Trang Google RDFa](http://tinyurl.com/mofoes-aria) Snippets mô tả sau (trang 94), có các phiên bản RDFa của tất cả các ví dụ của nó.

vi định dạng

Vi định dạng là một cách tiếp cận đơn giản, được sắp xếp hợp lý để đưa siêu dữ liệu vào các trang của bạn. Microformats không cố gắng trở thành bất kỳ loại tiêu chuẩn chính thức nào. Thay vào đó, chúng là một tập hợp lỏng lẻo các quy ước đã thỏa thuận cho phép các trang chia sẻ thông tin có cấu trúc mà không yêu cầu sự phức tạp của thứ gì đó như RDFa. Cách tiếp cận này đã mang lại thành công to lớn cho vi định dạng và một cuộc khảo sát trên web gần đây đã phát hiện ra rằng khi một trang có một số loại siêu dữ liệu phong phú, thì 70% thời gian đó là vi định dạng.

Các định dạng vi mô hoạt động theo một cách thú vị—chúng hỗ trợ thuộc tính lớp thường được sử dụng để tạo kiểu. Bạn đánh dấu dữ liệu của mình bằng cách sử dụng các tên kiểu chuẩn nhất định, tùy thuộc vào loại dữ liệu. Sau đó, một chương trình khác có thể đọc đánh dấu của bạn, trích xuất dữ liệu và kiểm tra các thuộc tính để tìm ra ý nghĩa của mọi thứ.

Ví dụ: bạn có thể sử dụng vi định dạng hCard để thể hiện chi tiết liên hệ của một người, công ty, tổ chức hoặc địa điểm. Bước đầu tiên là thêm một phần tử gốc có tên lớp phù hợp. Đối với hCard, tên lớp là vcard. (Thông thường, tên lớp khớp với tên của vi định dạng. Tên vcard được chọn vì lý do lịch sử, vì hCards dựa trên một định dạng cũ hơn nhiều có tên là Versitcard.)

Dưới đây là ví dụ về một <div> sẵn sàng lưu giữ chi tiết liên hệ bằng cách sử dụng vi định dạng hCard:

```
<div class="vcard">
</div>
```

Bên trong phần tử gốc này, bạn cung cấp thông tin liên hệ. Mỗi chi tiết phải được bọc trong một phần tử riêng biệt và được đánh dấu bằng tên lớp chính xác, như được xác định bởi vi định dạng mà bạn đang sử dụng. Ví dụ: trong hCard, bạn có thể sử dụng lớp fn để gắn cờ tên đầy đủ của một người và lớp url cho trang chủ của người đó:

```
<div class="vcard">
  <h3 class="fn">Mike Rowe</h3>
  Bạn có thể xem trang web của Mike Rowe tại
  <a class="url" href="http://www.magicsemantics.com">www.magicsemantics.com
  </a> </div>
```

Khi bạn sử dụng tên lớp cho một vi định dạng, bạn không cần tạo các kiểu phù hợp trong biểu định kiểu của mình. Trong ví dụ trên, điều đó có nghĩa là bạn không cần viết quy tắc kiểu cho các lớp vcard, fn hoặc url. Thay vào đó, các tên lớp được sử dụng cho mục đích khác—quảng cáo dữ liệu của bạn dưới dạng một đoạn nội dung có ý nghĩa, được cấu trúc đặc đáo.

GHI CHÚ

Trước khi bạn có thể đánh dấu bất kỳ dữ liệu nào, bạn cần chọn vi định dạng mà bạn muốn sử dụng. Chỉ có vài chục vi định dạng được sử dụng rộng rãi và hầu hết vẫn đang được điều chỉnh và sửa đổi. Bạn có thể xem những gì có sẵn và đọc thông tin sử dụng chi tiết về từng vi định dạng tại [Để tìm hiểu thêm về hCard](http://microformats.org/wiki/hCard) . <http://microformats.org/wiki/hCard> .

Khi bạn đã tìm hiểu về hCard, bạn sẽ dễ dàng hiểu được hCalendar, vì định dạng phổ biến thứ hai trên thế giới. Sử dụng hCalendar, bạn có thể đánh dấu các cuộc hẹn, cuộc họp, ngày lễ, phát hành sản phẩm, khai trương cửa hàng, v.v. Chỉ cần bọc danh sách sự kiện trong một thành phần có tên lớp là vevent.

Bên trong, bạn cần ít nhất hai thông tin: ngày bắt đầu (được đánh dấu bằng lớp dt-start) và mô tả (được đánh dấu bằng lớp tóm tắt). Bạn cũng có thể chọn từ nhiều thuộc tính tùy chọn được mô tả tại , bao gồm ngày hoặc thời lượng kết thúc, vị trí và URL <http://micformats.org/wiki/hLich> ví dụ:

```
<div class="vevent">
  <h2 class="summary">Nhà phát triển web Clam Bake</h2>
  <p>Tôi sẽ tổ chức một bữa tiệc!</p> <p>Đó là
  <span class="dtstart" title="25/10/2014 13 :30">Thứ
  Ba, ngày 25 tháng 10, 1:30 chiều tại <span class="location">Deep Sea
  Hotel, San Francisco, CA</p> </div>
```

Dựa trên sự phổ biến của các vi định dạng, bạn có thể cho rằng cuộc chiến giành Semantic Web đã được giải quyết. Nhưng không quá nhanh - có một số lưu ý. Đầu tiên, phần lớn các trang không có dữ liệu ngữ nghĩa phong phú nào cả. Thứ hai, hầu hết các trang sử dụng vi định dạng chỉ sử dụng chúng cho hai mục đích: thông tin liên hệ và danh sách sự kiện. Vì vậy, mặc dù các vi định dạng không sớm xuất hiện, nhưng vẫn còn nhiều không gian cho sự cạnh tranh. Thứ ba, mỗi trường đang bắt đầu chuyển sang đặc điểm kỹ thuật linh hoạt hơn nhưng vẫn ít được biết đến hơn. Dường như ngày càng có nhiều khả năng [vi định dạng](#) là điều dùng ngay tại thời điểm hiện nay vì nó đơn giản và hiệu quả hơn.

vi dữ liệu

Dữ liệu vi mô là bước thứ ba trong việc giải quyết thách thức của đánh dấu ngữ nghĩa. Nó bắt đầu tồn tại như một phần của đặc tả HTML5 và sau đó tách thành tiêu chuẩn phát triển của riêng nó tại . Microsoft sử dụng cách tiếp cận tương ứng với của RDFa, nhưng cũng cần thiết Không riêng của nó vẫn không có nguy cơ xung đột với các quy tắc biểu định kiểu (hoặc gây nhầm lẫn cho các nhà phát triển web khác).

Thiết kế này có nghĩa là dữ liệu vi mô hợp lý hơn cũng như dễ dàng điều chỉnh hơn cho các ngôn ngữ tự nhiên của riêng bạn. Nhưng nó cũng phải trả giá bằng sự ngắn gọn–đánh dấu được làm giàu bằng vi dữ liệu có thể phình to hơn một chút so với đánh dấu được làm giàu bằng vi định dạng.

Gần đây, dữ liệu vi mô đã nhận được một sự thúc đẩy lớn khi Microsoft, Google, Yahoo và Yandex (công cụ tìm kiếm lớn nhất của Nga) hợp tác để tạo ra một trang web phân loại dữ liệu vi mô có tên <http://micromodels.net> nhằm mục đích cung cấp cho các công ty khác hCard và hEvent phổ biến) và các loại chuyên biệt hơn để đánh dấu doanh nghiệp, nhà hàng, bài đánh giá, sản phẩm, sách, phim, công thức nấu ăn, TV các buổi biểu diễn, điểm dừng xe buýt, điểm tham quan du lịch, tình trạng y tế, thuốc men, v.v. Ngay bây giờ, chỉ có các công cụ tìm kiếm trả kết quả.

chú ý đến thông tin này, nhưng không thể phủ nhận ảnh hưởng định hình trang web, thúc đẩy lưu lượng truy cập của họ. (Bạn sẽ thấy cách các công cụ tìm kiếm sử dụng loại thông tin này bắt đầu từ trang 94.)

GHI CHÚ

Giờ đây, có vẻ như vi dữ liệu có thể trở thành tiêu chuẩn Goldilocks cho siêu dữ liệu—một đặc điểm kỹ thuật linh hoạt hơn vi định dạng nhưng không hoàn toàn phức tạp như RDFa.

để bắt đầu một phần dữ liệu vi mô, bạn thêm các thuộc tính itemscope và itemtype vào bất kỳ phần tử nào (mặc dù <div> tạo một vùng chứa logic, nếu bạn chưa có).

Thuộc tính `itemscope` chỉ ra rằng bạn đang bắt đầu một đoạn nội dung ngữ nghĩa mới. Thuộc tính `itemtype` cho biết loại dữ liệu cụ thể mà bạn đang mã hóa:

Để xác định loại dữ liệu, bạn sử dụng một đoạn văn bản duy nhất được xác định trước được gọi là **không gian tên XML**. Trong ví dụ này, không gian tên XML là <http://schema.org/Person>, là định dạng vi dữ liệu được tiêu chuẩn hóa để mã hóa chi tiết liên hệ, như được thảo luận trong hộp bên dưới.

Hiểu không gian tên Microdata

Mọi định dạng vi dữ liệu đều cần một không gian tên. Về mặt kỹ thuật, không gian tên xác định **więc** sử dụng **vì** **kết nối** **giao** **nhau** **lên** **chi** **ra** **ràng** phần đánh dấu này **sử** **đụng** **tới** **chung** **Perso** **nir** **Bạn** có **phép** **Hán** **tá** **tự** **vựng** **dữ** **liệu** **vi** **mô** **tại** (xem Hình 3-3).

<http://schema.org>

Không gian tên XML thường là URL. Đôi khi, bạn thậm chí có thể tìm thấy mô tả về loại dữ liệu tương ứng bằng cách nhập URL vào trình duyệt web của mình (như bạn có thể làm với định dạng <http://schema.org/Người>). Tuy nhiên, các không gian tên XML không tương ứng phần CSDL XML, trỏ về khái niệm chung hơn là ngôn ngữ phát triển đã chọn khi tạo định dạng. Ưu điểm của URL là

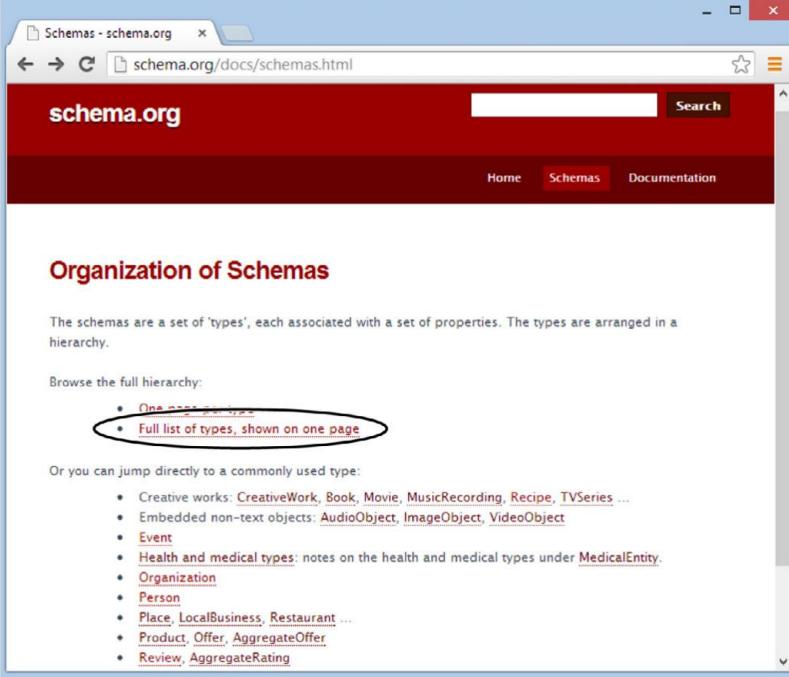
trong đó có thể kết hợp một tên miền thuộc về một người hoặc tổ chức. Bằng cách này, không gian tên có nhiều khả năng là duy nhất - không ai khác sẽ tạo một định dạng dữ liệu khác có cùng tên không gian tên và gây nhầm lẫn cho mọi người.

Nếu một khống gian tên bắt đầu bằng từ vựng được xác nhận bởi nhóm công cụ tìm kiếm trong mơ của Microsoft, Google, Yahoo và Yandex. Vì vậy, nếu bạn sử dụng từ vựng đó, bạn có thể tự tin rằng các công cụ tìm kiếm trên thế giới sẽ hiểu những gì bạn đang làm. Nếu một khống gian tên bắt đầu bằng [nội dung](#) / [sử dụng](#) một [tên](#) [vùng](#), [và](#) [đó](#). [là](#) [kết](#) [hoặc](#) [một](#) công cụ tìm kiếm sẽ vẫn hiểu đánh dấu của bạn, nhưng tốt hơn hết là bạn nên kiên trì theo thời gian và tìm một từ vựng tương đương tại <http://schema.org>.

Khi bạn có phần tử vùng chứa, bạn đã sẵn sàng chuyển sang bước tiếp theo. Bên trong phần tử vùng chứa của bạn, bạn sử dụng thuộc tính `itemprop` để nắm bắt các bit thông tin quan trọng. Cách tiếp cận cơ bản cũng giống như đối với định dạng `data`-`value`, bạn sử dụng tên `itemprop` được công nhận và các phần mềm khác có thể lấy thông tin từ các thành phần liên quan.

Đây là phiên bản vi dữ liệu của vi định dạng hCard mà bạn đã thấy trước đó:

```
<div itemscope itemtype="http://schema.org/Person">
  <h3 itemprop="name">Mike Rowe</h3>
  Bạn có thể xem trang web của Mike Rowe tại
  <a itemprop="url" href="http://www.magicsemantics.com">www.magicsemantics.
  com</a>
</div>
```



HÌNH 3-3

Để tìm một từ vựng dữ liệu vi mô phù hợp với bạn, có một vài điểm khác nhau tốt hơn so với cách tìm kiếm thông thường.

http://luợc.org/docs/schemas.html

để nhảy thẳng vào trang định nghĩa. Nhấp vào từ và sau đó nhấp vào định nghĩa. Nếu từ không có định nghĩa chung,

hoặc

Hoặc, duyệt qua danh mục tệp tin lớp, bằng cách nhấp vào

Sự khác biệt đáng kể nhất giữa dữ liệu vi mô và vi định dạng là dữ liệu vi mô sử dụng thuộc tính `itemprop` để đánh dấu các phần tử thay vì thuộc tính lớp.

GHI CHÚ

Vì vi dữ liệu sử dụng các thuộc tính `itemscope`, `itemtype` và `itemprop` của chính nó, thay vì `class`, sẽ không có khả năng bạn nhầm lẫn đánh dấu ngữ nghĩa của mình với định dạng biểu định kiểu của bạn.

Có rất nhiều chi tiết bổ sung mà bạn có thể đánh dấu bằng cách sử dụng từ vựng Person. Các lựa chọn phổ biến bao gồm địa chỉ bưu chính và email, số điện thoại, ngày sinh, ảnh, chức danh công việc, tên tổ chức, giới tính, quốc tịch, v.v. Để biết danh sách đầy đủ các thuộc tính có thể, hãy tham khảo <http://schema.org/Person>.

MỘT THỰC TẾ
THÍ DỤ:
trang bị thêm
MỘT "GIỚI THIỆU
TÔI" TRANG

GHI CHÚ

Ba tiêu chuẩn cho dữ liệu ngữ nghĩa phong phú-RDFa, ví dữ liệu và vi định dạng-tất cả đều có những điểm tương đồng lớn. Chúng không hoàn toàn tương thích, nhưng cách đánh dấu tương tự nhau đến mức các kỹ năng bạn học được trong một hệ thống hầu hết có thể áp dụng cho các hệ thống khác.

Một ví dụ thực tế: Trang bị thêm phần "Giới thiệu Tôi" Trang

Cho đến giờ, bạn đã học về cấu trúc cơ bản của hai yếu tố chính ngữ nghĩa: vi định dạng và dữ liệu vi mô. Được trang bị kiến thức này, bạn có thể tra cứu một từ vựng vi định dạng mới [\(từ /\) và hoặc với dữ liệu](#) (từ) và bắt đầu viết đánh dấu giàu ngữ nghĩa [@NghiaSchema.org](#)

Tuy nhiên, cuộc sống thường không diễn ra theo cách này-ít nhất là không dành cho hầu hết các nhà phát triển web. Thay vào đó, bạn thường cần lấy một trang web đã có tất cả dữ liệu cần thiết và trang bị thêm dữ liệu ngữ nghĩa cho trang. Nhiệm vụ này khá dễ dàng nếu bạn lưu ý một số điểm sau:

- Thông thường, bạn sẽ có dữ liệu quan trọng lẩn lộn với nội dung mà bạn muốn bỏ qua. Trong trường hợp này, bạn có thể thêm các yếu tố mới xung quanh mỗi phần thông tin bạn muốn nắm bắt. Sử dụng `<div>` nếu bạn muốn một phần tử cấp khối hoặc `` nếu bạn muốn lấy một phần nội dung nội tuyến.
- Đừng lo lắng về thứ tự thông tin của bạn. Miễn là bạn sử dụng đúng lớp (đối với vi định dạng) hoặc tên thuộc tính (đối với dữ liệu vi mô), bạn có thể sắp xếp đánh dấu của mình theo ý muốn.
- Nếu bạn đang cung cấp hình ảnh, bạn có thể sử dụng phần tử ``. Nếu bạn đang cung cấp một liên kết, bạn có thể sử dụng phần tử `<a>`. Thời gian còn lại, bạn thường sẽ đánh dấu văn bản thông thường.

Đây là một ví dụ điển hình. Hãy tưởng tượng bạn bắt đầu với trang "Giới thiệu về tôi" (Hình 3-4) có nội dung như sau:

```

<h1>Giới thiệu về tôi</h1>

 <p>Trang
web này là tác phẩm của <b>Mike Rowe Formatte</b>.
Bạn bè của anh ấy gọi anh ấy là <b>The Big M</b>.</p>

<p>Bạn có thể liên hệ với anh ấy tại nơi anh ấy làm
việc, tại The Magic Semantic Company (diện thoại
641-545-0234 và hỏi tên Mike) hoặc gửi email đến mike-f@magicsemantics.com.</p>

<p>Hoặc, hãy ghé thăm nơi làm việc của Mike tại:<br>
42 Phố Jordan Gordon, Tầng 6<br>
San Francisco, CA 94105<br>
Hoa Kỳ<br>

```

www.magicsemantics.com</p>

HÌNH 3-4

Trang "Giới thiệu về tôi" này bao gồm chi tiết liên hệ của tác giả trang web, trước khi thêm vi dữ liệu. không tí nào

About Me



This website is the work of **Mike Rowe Formatte**. His friends know him as **The Big M**.

You can contact him where he works, at The Magic Semantic Company (phone 641-545-0234 and ask for Mike) or email mike-f@magicsemantics.com.

Or, visit Mike on the job at:
 42 Jordan Gordon Street, 6th Floor
 San Francisco, CA 94105
 USA
www.magicsemantics.com

Rõ ràng, điều này rất phù hợp với từ vựng Người quen thuộc (chi tiết tại <http://schema.org/Person>). Đây là nỗ lực đầu tiên trong việc dệt vi dữ liệu xung quanh các chi tiết chính trong trang "Giới thiệu về tôi". Các bit vi dữ liệu mới được chèn được nhấn mạnh bằng chữ in đậm:

```
<h1>Giới thiệu về tôi</h1>

<div itemscope itemtype="http://schema.org/Person">
    
    <p>Trang web này là sản phẩm của <span itemprop="jobTitle"
    style="display:none">Nhà phát triển web</span> <b itemprop="name">Định
    dạng Mike Rowe</b>.
    Bạn bè của anh ấy gọi anh ấy là <b itemprop="additionalName">The Big M</b>.</p>
```

MỘT THỰC TẾ
THÍ DỤ:
trang bị thêm
MỘT "GIỚI THIỆU
TÔI" TRANG

<p>Bạn có thể liên hệ với anh ấy nơi anh ấy làm việc, tại The Magic Semantic Company (điện thoại 641-545-0234 và hỏi Mike) hoặc gửi email mike-f@magicsemantics.com.</p>

<p>Hoặc, hãy ghé thăm nơi làm việc của Mike tại:
42 Phố Jordan Gordon, Tầng 6
San Francisco, CA 94105
Hoa Kỳ
www.magicsemantics.com</p> </div>

Ví dụ này sử dụng một vài kỹ thuật tiện dụng:

- Nó thêm các phần tử mới để bọc các bit nội dung bạn cần cho vi dữ liệu.
- Nó thêm thuộc tính `itemprop` vào các phần tử hiện có, khi làm như vậy có ý nghĩa. Ví dụ: phần tử bao bọc thông tin tên, vì vậy không cần thêm phần bổ sung . (Tất nhiên là bạn làm như vậy. Ví dụ: bạn có thể muốn viết Mike Rowe Formatte.
- Nó sử dụng ẩn để biểu thị chức danh công việc của người đó. (Văn bản được ẩn bằng quy tắc kiểu nội tuyến đặt thuộc tính hiển thị thành không, như bạn có thể thấy trong phần đánh dấu ở trên.) Kỹ thuật này cho phép bạn ẩn thông tin dư thừa, trong khi vẫn bảo quản thông tin đó cho các công cụ tìm kiếm và các công cụ khác. Điều đó nói rằng, kỹ thuật ẩn nội dung là một chút tranh cãi, bởi vì một số công cụ (như Google) bỏ qua thông tin không hiển thị cho người xem trang web.

Thông thường vi dữ liệu có cấu trúc đặt một `vi-dữ` liệu vo cabulary bên trong một vi dữ liệu khác. Ví dụ: trong từ vựng Người, bạn có thể có một bộ thông tin địa chỉ nằm bên trong các chi tiết cá nhân. Về mặt kỹ thuật, tất cả thông tin địa chỉ thuộc về một từ vựng riêng biệt, được gọi là `PostalAddress`.

Để đánh dấu thông tin địa chỉ, bạn cần thêm phần tử <div> hoặc mới sử dụng thuộc tính `itemprop`, `itemscope` hoặc `itemtype`. Thuộc tính `itemprop` có tên thuộc tính, thuộc tính `itemscope` cho biết rằng bạn đang bắt đầu một từ vựng mới để cung cấp dữ liệu thuộc tính và thuộc tính `itemtype` xác định từ vựng theo không gian tên XML của nó (trong trường hợp này là).

<http://schema.org/PostalAddress>

Đây là cách kết hợp với nhau:

```
<div itemscope itemtype="http://schema.org/Person">
  
  <p>Trang web này là tác phẩm của
  ...

```

MỘT THỰC TẾ
THÍ DỤ:
trang bị thêm
MỘT "GIỚI THIỆU
TÔI" TRANG

```
<p>Hoặc, ghé thăm nơi làm việc của Mike
tại:<br> <span itemprop="address"
itemscope itemtype="http://schema.org/PostalAddress">
...
</div>
```

Sau đó, bạn có thể điền chi tiết địa chỉ bên trong phần mới:

```
<div itemscope itemtype="http://schema.org/Person">

<p>Trang web này là tác phẩm của
...
<p>Hoặc, ghé thăm nơi làm việc của Mike
tại:<br> <span itemprop="address"
itemscope itemtype="http://schema.org/
PostalAddress"><span itemprop="streetAddress">42 Phó Jordan
Gordon, Tầng 6<br> <span itemprop="addressLocality">San
Francisco, <span itemprop="addressRegion">CA <span
itemprop="postalCode">94105 <br> <span
itemprop="addressCountry">Hoa Kỳ<br>
...
</div>
```

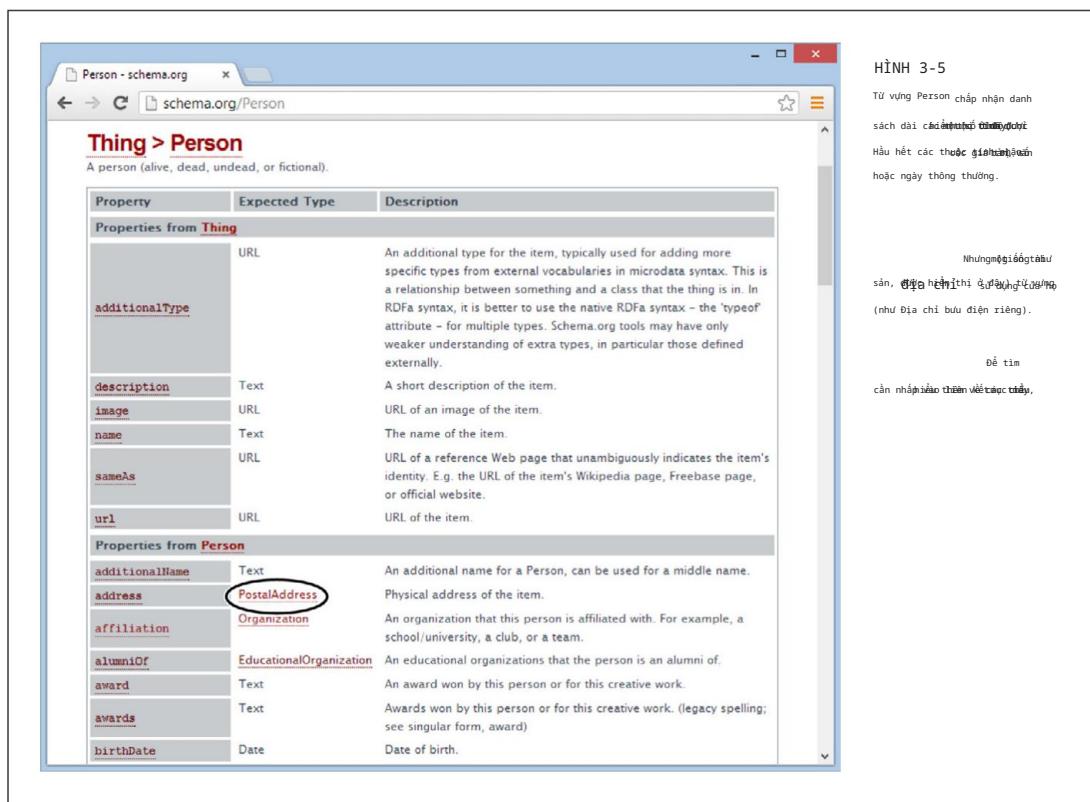
Tất cả điều này hoàn toàn hợp lý, nhưng bạn có thể tự hỏi làm thế nào bạn biết xác định **khi nào** đến phần dữ liệu vi mô mới bên trong phần dữ liệu vi mô đầu tiên của mình. May mắn thay, trang tham khảo trên <http://schema.org> làm cho nó rõ ràng (Hình 3-5).

Một thủ thuật vi dữ liệu bên trong vi dữ liệu tương tự diễn ra khi bạn đánh dấu tên công ty. Tại đây, bạn cần đặt thuộc tính liên kết của người đó bằng cách sử dụng từ vựng Tổ chức:

```
<p>Bạn có thể liên hệ với anh ấy tại nơi
anh ấy làm việc, tại <span
itemprop="affiliation" itemscope
itemtype="http://schema.org/Organization"> <span
itemprop="name">The Magic Semantic Company
```

TIỀN BOA Nếu bạn không muốn tự mình điền vào tất cả các loại vật phẩm và đạo cụ vật phẩm, thì có những công cụ trực tuyến mà bạn có thể sử dụng để tạo đánh dấu làm giàu vi dữ liệu được định dạng đúng. Hai ví dụ là và <http://schema-creator.org> www.microdatagenerator.com. Với cả hai trang web, ý tưởng là giống nhau—you chọn từ vựng của mình, nhập dữ liệu vào các hộp văn bản được cung cấp, sau đó sao chép phần đánh dấu đã hoàn thành.

MỘT THỰC TẾ
THÍ DỤ:
trang bị thêm
MỘT "GIỚI THIỆU
TÔI" TRANG



The screenshot shows the schema.org Person page with two main sections: 'Properties from Thing' and 'Properties from Person'. The 'PostalAddress' property under 'Properties from Person' is circled in red.

Property	Expected Type	Description
<u>additionalType</u>	URL	An additional type for the item, typically used for adding more specific types from external vocabularies in microdata syntax. This is a relationship between something and a class that the thing is in. In RDFa syntax, it is better to use the native RDFa syntax – the 'typeof' attribute – for multiple types. Schema.org tools may have only weaker understanding of extra types, in particular those defined externally.
<u>description</u>	Text	A short description of the item.
<u>image</u>	URL	URL of an image of the item.
<u>name</u>	Text	The name of the item.
<u>sameAs</u>	URL	URL of a reference Web page that unambiguously indicates the item's identity. E.g. the URL of the item's Wikipedia page, Freebase page, or official website.
<u>url</u>	URL	URL of the item.
Properties from Person		
<u>additionalName</u>	Text	An additional name for a Person, can be used for a middle name.
<u>address</u>	PostalAddress	Physical address of the item.
<u>affiliation</u>	Organization	An organization that this person is affiliated with. For example, a school/university, a club, or a team.
<u>alumniOf</u>	EducationalOrganization	An educational organizations that the person is an alumni of.
<u>award</u>	Text	An award won by this person or for this creative work.
<u>awards</u>	Text	Awards won by this person or for this creative work. (legacy spelling; see singular form, award)
<u>birthDate</u>	Date	Date of birth.

HÌNH 3-5

Từ vựng Person chấp nhận danh

sách dài của [schema.org](#) không giới

Hầu hết các thuộc [giai](#) hoặc [bản](#)
hoặc ngày thông thường.

Những [địa chỉ](#) và [tên](#) của

sản, [địa chỉ](#) [hỗn](#) thi [địa chỉ](#) [vô](#) [vô](#)
(như Địa chỉ bưu điện riêng).

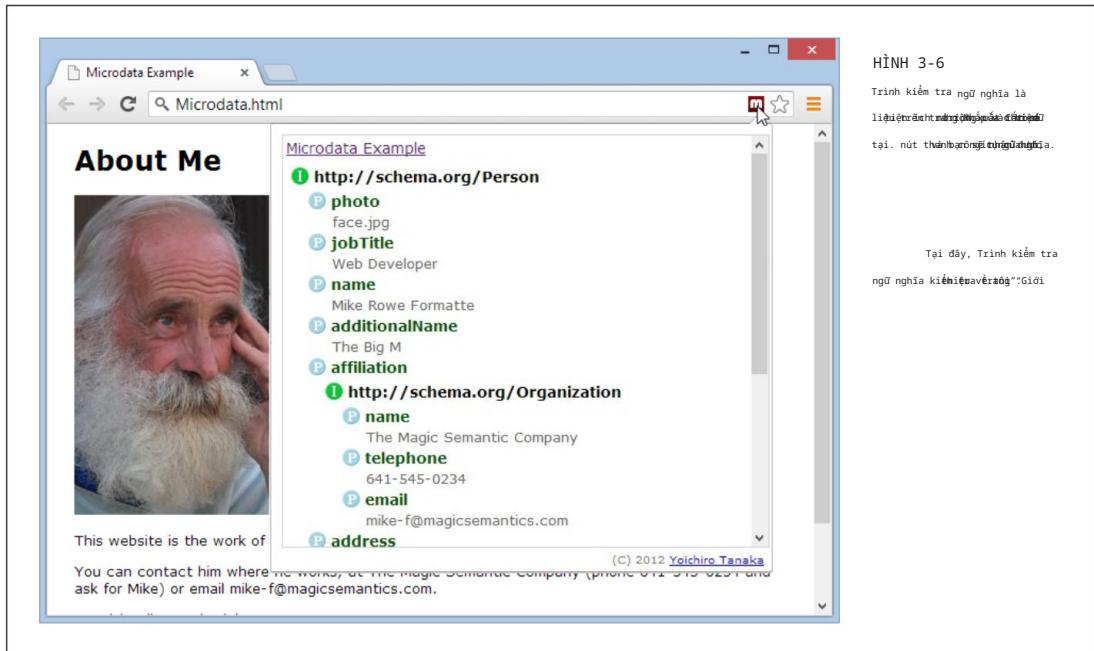
Để tìm

cần nhấp vào thêm [về](#) [tác](#) [tác](#).

Trích xuất dữ liệu ngữ nghĩa trong trình duyệt của bạn Bây giờ

bạn đã trải qua tất cả những rắc rối này, đã đến lúc xem bạn có thể thu được những lợi ích gì. Mặc dù không có trình duyệt nào tự nhận dạng vi định dạng (ít nhất là tại thời điểm viết bài này), nhưng có nhiều plug-in và tập lệnh có thể cung cấp cho trình duyệt những khả năng này. Và không khó để tưởng tượng các kịch bản hữu ích. Ví dụ: trình duyệt có thể phát hiện thông tin liên hệ trên một trang, liệt kê thông tin đó trong bảng điều khiển bên và cung cấp cho bạn các lệnh cho phép bạn thêm một người vào sổ địa chỉ của mình ngay khi bạn đánh dấu trang. Một thủ thuật tương tự có thể phát hiện thông tin sự kiện và cho phép bạn thêm thông tin đó vào lịch của mình chỉ bằng một cú nhấp chuột hoặc tìm vị trí và tự động vé chúng trên bản đồ.

Ngay bây giờ, không có plug-in nào đi xa đến thế. Tuy nhiên, một số nhà phát triển web chuyên nghiệp đã tạo ra nhiều quy trình JavaScript có thể tìm kiếm vi định dạng hoặc siêu dữ liệu, hiển thị nó trong hộp bật lên hoặc sử dụng nó trong một tác vụ khác. (Một ví dụ là Công cụ vi dữ liệu do JavaScript cung cấp <http://krofdrakula.github.io/microdata-tool> Và một số trình duyệt có plug-in có thể phát hiện các loại siêu dữ liệu khác nhau trên một trang web (Hình 3-6)).



Đặt cược tốt nhất cho tương lai dữ liệu vì mô có thể là các trình duyệt chính kết hợp hỗ trợ trực tiếp, giống như Internet Explorer và Firefox làm cho nguồn cấp dữ liệu. A là một loại tài liệu [tạo danh sách cập nhật](#) nội dung cập nhật, chẳng hạn như danh sách các bài báo được xuất bản gần đây. Ví dụ: nếu bạn truy cập một blog bằng Firefox, nó sẽ tự động phát hiện nguồn cấp RSS của blog đó và cho phép bạn tạo một dấu trang "trực tiếp" tự động tìm nạp nội dung mới. Đây chính xác là loại tính năng giá trị gia tăng có thể làm cho vi định dạng thực sự hữu ích.

Cách Công cụ Tìm kiếm Sử dụng Siêu dữ liệu

Nhài trang của bạn với các chi tiết ngữ nghĩa là một cách tuyệt vời để giành được cho mình một số tín nhiệm về web nghiêm túc. Nhưng ngay cả những nhà phát triển web khó tính cũng cần một số khoản hoàn trả để làm cho công việc bổ sung (và đánh dấu lõi xon hơn) trở nên đáng giá. Thật tuyệt khi nghĩ về một thế giới của các trình duyệt siêu thông minh, có nhận thức về mặt ngữ nghĩa, nhưng hiện tại, một thực tế phũ phàng và phũ phàng là những người lướt web chỉ có một vài plug-in trình duyệt thử nghiệm và ít được biết đến.

May mắn thay, có một lý do khác để nắm lấy ngữ nghĩa phong phú: [máy tìm kiếm tối ưu hóa](#) (SEO). SEO là nghệ thuật làm cho trang web của bạn hiển thị nhiều hơn trong công cụ tìm kiếm—nói cách khác, làm cho trang web xuất hiện thường xuyên hơn trong trang kết quả, giúp trang web có thứ hạng tốt hơn cho một số từ khóa nhất định và khiến trang web có nhiều khả năng thu hút khách hàng hơn.

khách truy cập để nhấp qua trang web của bạn. Siêu dữ liệu tốt có thể giúp ích cho phần cuối cùng của phương trình này. Tất cả những gì bạn làm là đặt dữ liệu ngữ nghĩa phù hợp trên trang của mình và công cụ tìm kiếm như Google sẽ sử dụng dữ liệu đó để trình bày danh sách tìm kiếm hấp dẫn hơn, có thể giúp trang web của bạn nổi bật giữa đám đông.

Google Rich Snippets Ngày nay,

hầu hết các công cụ tìm kiếm đều có thẻ hiệu được siêu dữ liệu trong các trang mà chúng lập danh mục. Trong phần còn lại của chương này, bạn sẽ tập trung vào những gì Google làm với siêu dữ liệu mà Google tìm thấy. Có hai lý do để tập trung vào Google. Đầu tiên, Google là công cụ tìm kiếm phổ biến nhất trên Trái đất, chiếm 2/3 thị phần tìm kiếm trên toàn thế giới. Thứ hai, Google đã sử dụng và quảng bá siêu dữ liệu trong nhiều năm. Cách nó sử dụng siêu dữ liệu hôm nay là cách các công cụ tìm kiếm khác sẽ làm như vậy vào ngày mai.

Google sử dụng thuật ngữ ~~rich snippets~~ RDFa, vi định dạng và mi crodata lại với nhau. Như bạn đã biết, những cách tiếp cận này có những điểm tương đồng đáng kể và giải quyết cùng một vấn đề. Google hiểu tất cả họ và cố gắng đổi xử bình đẳng với tất cả họ, vì vậy, việc bạn ứng hộ cách tiếp cận nào không quan trọng. (Các ví dụ sau sử dụng vi dữ liệu, với mục đích giúp bạn làm quen với tiêu chuẩn ngữ nghĩa mới nhất của HTML5.)

Để tìm hiểu thêm về siêu dữ liệu mà Google hỗ trợ, bạn có thể xem tài liệu của Google tại . Nó không chỉ bao gồm một <http://tinyurl.com/googlerichsnippets> và ~~rich snippets~~ và ~~microdata~~ khác nhau (như thông tin liên hệ, sự kiện, sản phẩm, bài đánh giá, công thức nấu ăn, v.v.). Trên hết, Google bao gồm một phiên bản RDFa, vi định dạng và vi dữ liệu của mỗi ví dụ, phiên bản này có thể giúp bạn dịch các kỹ năng ngữ nghĩa của mình từ tiêu chuẩn này sang tiêu chuẩn khác, nếu có nhu cầu.

Kết quả tìm kiếm nâng cao

Để xem tính năng đoạn mã chi tiết của Google hoạt động như thế nào, bạn có thể sử dụng Công cụ kiểm tra dữ liệu có cấu trúc của Google. Công cụ này kiểm tra trang bạn cung cấp, hiển thị cho bạn dữ liệu ngữ nghĩa mà Google có thể trích xuất từ trang và sau đó cho bạn biết cách Google có thể sử dụng thông tin đó để tùy chỉnh cách Google trình bày trang đó trong kết quả tìm kiếm của ai đó.

GHI CHÚ

Công cụ kiểm tra dữ liệu có cấu trúc hữu ích vì hai lý do. Đầu tiên, nó giúp xác thực đánh dấu ngữ nghĩa của bạn. Nếu Google không thể trích xuất tất cả thông tin bạn đưa vào trang hoặc nếu một số thông tin được gán cho thuộc tính sai, thì bạn biết mình đã làm sai điều gì đó. Thứ hai, nó cho bạn thấy cách dữ liệu ngữ nghĩa có thể thay đổi giao diện trang của bạn trong kết quả tìm kiếm của Google.

Để sử dụng Công cụ kiểm tra dữ liệu có cấu trúc, hãy làm theo các bước đơn giản sau:

- Chuyển đến www.google.com/webmasters/tools/richsnippets

Trang đơn giản này bao gồm một hộp văn bản duy nhất (xem Hình 3-7).

HÌNH 3-7

The page you want Google to examine: <http://prosetech.com/html5/Chapter%2003/Microdata.html>

Google uses some of the semantic data in this line

Google found the contact details on the page

Google also found the address details

Preview

Extracted structured data

Item	type:	http://data-vocabulary.org/person
property:		
photo:		http://prosetech.com/html5/Chapter%2003/face.jpg
title:		Web Developer
name:		Mike Rowe Formatte
nickname:		The Big M
affiliation:		The Magic Semantic Company
tel:		641-545-0234
address:		<i>Item 1</i>
url:		www.magicsemantics.com

Item 1	type:	http://data-vocabulary.org/address
property:		
street-address:		42 Jordan Gordon Street, 6th Floor
locality:		San Francisco
region:		CA
postal-code:		94105
country-name:		USA

Tại đây, Google đã tìm thấy chi tiết liên hệ và địa chỉ của trang web. Nó đã sử dụng thông tin này để hiển thị với một số chi tiết cá nhân.

2. Nếu bạn muốn dán đánh dấu của mình, hãy nhấp vào tab HTML.

Có hai cách để sử dụng Công cụ kiểm tra dữ liệu có cấu trúc, được thể hiện bằng hai tab trên trang.

- Tab URL yêu cầu Google phân tích một trang đã trực tuyến. Bạn đơn giản dán vào địa chỉ trang web đã dù của nó.
- Tab HTML cho phép bạn dán đoạn mã đánh dấu mà bạn muốn phân tích (không cần thiết phải có toàn bộ trang) vào một hộp văn bản lớn. Nếu bạn chưa tải tác phẩm của mình lên, đây là cách thuận tiện nhất.

3. Nhập URL của bạn hoặc dán vào đánh dấu của bạn. Sau đó nhấp vào Xem trước.

Bây giờ bạn có thể xem lại kết quả (xem Hình 3-7). Có hai phần quan trọng để xem xét. Phần “Xem trước tìm kiếm của Google” hiển thị cách trang có thể xuất hiện trong kết quả tìm kiếm. “Dữ liệu đoạn mã chi tiết được trích xuất từ trang” hiển thị tất cả dữ liệu ngữ nghĩa mà Google có thể lấy ra khỏi đánh dấu của bạn.

TIẾN BỘ

Nếu bạn thấy thông báo lỗi đáng sợ “Không đủ dữ liệu để tạo bản xem trước”, thì có ba nguyên nhân có thể xảy ra. Đầu tiên, đánh dấu của bạn có thể bị lỗi. Xem lại dữ liệu thô mà Google đã trích xuất và đảm bảo rằng Google đã tìm thấy mọi thứ bạn đặt ở đó. Nếu bạn không tìm thấy sự cố ở đây, có thể bạn đang có sử dụng loại dữ liệu mà Google chưa hỗ trợ hoặc bạn chưa bao gồm tập hợp thuộc tính tối thiểu mà Google cần. Để biết ván đe là gì, hãy so sánh đánh dấu của bạn với một trong các ví dụ của Google <http://tinyurl.com/GoogleRichSnippets>.

Phương pháp Google sử dụng để nhấn mạnh các chi tiết liên hệ (Hình 3-7) khá hạn chế.

Tuy nhiên, chi tiết liên hệ chỉ là một trong những loại dữ liệu phong phú mà Google công nhận.

Trước đó trong chương này (trang 85), bạn đã thấy cách xác định các sự kiện bằng vi định dạng.

Thêm danh sách các sự kiện vào trang của bạn và Google có thể đưa chúng vào cuối kết quả tìm kiếm của bạn, như trong Hình 3-8.

[The Fillmore New York at Irving Plaza Concert Tickets, Schedule ...](#)

Buy The Fillmore New York at Irving Plaza tickets and find concert schedules, venue information, and seating charts for The Fillmore New York at Irving ...

Led Zeppelin 2	Sat, Jan 23
Cheap Trick with Jason Falkner ...	Mon, Jan 25
Hip Hop Karaoke Championship	Fri, Jan 29

www.livenation.com/.../the-fillmore-new-york-at-irving-plaza-new-york-ny-tickets -
Cached - Similar -   

HÌNH 3-8

Ví dụ này có ba sự kiện có URL trang web ([http://tinyurl.com/GoogleRichSnippets](#)) cung cấp “tại đây”, Google mỗi [văn bản](#) thành một liên kết có thể nhấp

Google cũng quan tâm đến danh sách doanh nghiệp (được xử lý theo cách tương tự như chi tiết liên hệ cá nhân), công thức nấu ăn (mà bạn sẽ xem qua trong phần tiếp theo) và đánh giá (mà bạn sẽ xem xét tiếp theo).

Ví dụ sau đây cho thấy phần đánh dấu mà bạn cần để biến một số văn bản đánh giá thành vi dữ liệu đánh giá dễ nhận biết. Tiêu chuẩn dữ liệu được xác định tại . Các thuộc tính [http://schema.org/Kiểm tra lại itemReviewed](#) (trong trường hợp này là nhà hàng), tác giả (người thực hiện đánh giá) và reviewBody (tài khoản đầy đủ của đánh giá). Bạn cũng có thể cung cấp thông tin tổng quan trong một câu (mô tả), ngày thực hiện đánh giá (datePublished, hỗ trợ phần tử `<time>` của HTML5) và điểm thường được thực hiện trên thang điểm từ 0 đến 5 (đánh giá).

Đây là một ví dụ, với tất cả các chi tiết vi dữ liệu được làm nổi bật:

```
<div itemscope itemtype="http://schema.org/Review">
<p itemprop="description">Khá tệ, và sau đó Bộ Y tế đã chỉ ra
lên. </p>
```

```
<div itemprop="itemReviewed" itemscope itemtype="http://schema.org/Thing"> <span
    itemprop="name">Jan's Pizza House</div>
```

```
<div>
    <span itemprop="author" itemscope itemtype="http://schema.org/Person">
        Được đánh giá bởi: <span itemprop="name">Jared Elberadi vào
        <time itemprop="datePublished" datetime="2014-06-26">ngày 26
        tháng 1</time> </div>
```

<div itemprop="reviewBody">Tôi rất muốn ăn một ít bánh pizza và nơi này là điểm bán duy nhất xung quanh. Nó trông giống như một chuyên lận, nhưng tôi đã đi với hy vọng tìm thấy một viên ngọc quý chưa được khám phá. Thay vào đó, tôi theo dõi một thanh tra của Bộ Y tế đóng cửa nơi này. Bản án? Tôi đã không ăn hết chiếc bánh pizza của mình và thanh tra đề nghị tiêm phòng viêm gan C.</div>

```
<div itemprop="reviewRating" itemscope itemtype="http://schema.org/Rating">
    Xếp hạng:<span itemprop="ratingValue">1,5</div> </div>
```

Nếu bạn đưa bài đánh giá có định dạng vi dữ liệu này vào một trang web, Google sẽ dành cho nó sự đối xử thực sự đặc biệt (Hình 3-9).

Structured Data Testing Tool

[URL](#) [HTML](#)

```
<div itemscope itemtype="http://schema.org/Review">
    <p itemprop="description">Pretty bad, and then the Health Department showed up.</p>

    <div itemprop="itemReviewed" itemscope itemtype="http://schema.org/Thing">
        <span itemprop="name">Jan's Pizza House</span>
    </div>

    <div>
        <span itemprop="author" itemscope itemtype="http://schema.org/Person">
            Reviewed by: <span itemprop="name">Jared Elberadi</span>
    </div>

```

Google search results **Google Custom Search**

Preview

Once you add a page title, it will appear here
www.example.com/
★★★★★ Rating: 1.5 - Review by Jared Elberadi - Jun 26, 2014
The excerpt from the page will show up here. The reason we can't show text from your webpage is because the text depends on the query the user types.

HÌNH 3-9

Đánh giá thực sự nổi bật trong kết quả tìm kiếm.
Các ngôi sao xếp hạng **bấp bênh**, và thu hút là quan tâm ngay

Đôi mắt tinh tường sẽ nhận thấy rằng thực sự có **bốn** các định dạng vi dữ liệu đang hoạt động những đánh giá đơn lẻ. Có một định dạng cho bản thân bài đánh <http://schema.org/Review> giá (một định dạng dành cho nội dung bài đánh <http://schema.org/Rating> trong này),), một cho người giá (<http://schema.org/Person> đó làm) và một cho hệ thống xếp hạng (<http://schema.org/Rating>). Bạn có thể sử dụng các tiêu chuẩn này để cung cấp thêm chi tiết cho bài đánh giá của mình. Ví dụ: bạn có thể thêm thực đơn của nhà hàng, địa chỉ email của người đánh giá hoặc giá trị tối thiểu và tối đa của hệ thống xếp hạng tùy chỉnh.

Định dạng dữ liệu dùng /Sử dụng bởi thuộc tính itemReviewed, hơi khác so với các định dạng khác. Thoạt nhìn, từ vựng về “đồ vật” nghe có vẻ khá mơ hồ. Thiết kế này là có chủ ý-nó mang lại cho bạn sự linh hoạt để viết bài đánh giá về hầu như mọi thứ, bởi vì Sự vật là một từ vựng cơ bản làm cơ sở cho nhiều danh mục chuyên biệt hơn. Chúng bao gồm các sản phẩm, địa điểm, sự kiện, sách, bản ghi âm nhạc, v.v. (xem danh sách đầy đủ các loại). Trong ví dụ này, bạn có thể dễ dàng chuyển từ vựng Thing sang từ vựng Nhà hàng chuyên biệt hơn (<http://schema.org/LocalBusiness>) vì trang này không bao gồm bất kỳ thông tin bổ sung nào về nhà hàng nên bạn không cần thực hiện bước này.

<http://schema.org/>

Quán ăn

GHI CHÚ

Bạn có thể xem tương tự cách Bing (công cụ tìm kiếm của Microsoft) xử lý siêu dữ liệu bằng cách sử dụng Trình xác thực đánh dấu Bing <http://webmaster.yandex.ru/microtest.xml>.

Danh sách tìm kiếm nâng cao của Công cụ

Tìm kiếm Công thức là một thủ thuật gọn gàng và chúng có thể hướng lưu lượng truy cập mới vào trang web của bạn. Tuy nhiên, thật khó để không muốn một cái gì đó thậm chí còn ấn tượng hơn để biện minh cho các kỹ năng ngữ nghĩa mới tìm thấy của bạn. May mắn thay, những thiên tài tại Google đang bận rộn mơ ước về tương lai của tìm kiếm, và nó có ngữ nghĩa ở khắp mọi nơi.

Một ý tưởng tuyệt vời là sử dụng thông tin ngữ nghĩa không phải để điều chỉnh cách trình bày một mục trong tìm kiếm mà để cho phép lọc tìm kiếm thông minh hơn. Ví dụ: nếu mọi người đánh dấu sơ yếu lý lịch của họ bằng cách sử dụng RDFa, vi định dạng hoặc vi dữ liệu, Google có thể cung cấp tính năng tìm kiếm sơ yếu lý lịch chuyên biệt để xem xét dữ liệu này, xem xét sơ yếu lý lịch từ mọi trang web nghề nghiệp phổ biến và bỏ qua mọi loại nội dung web khác. Công cụ tìm kiếm sơ yếu lý lịch này cũng có thể cung cấp các tùy chọn lọc nâng cao-ví dụ: cho phép các công ty tìm ứng viên có bằng cấp cụ thể hoặc đã từng làm việc cho các công ty cụ thể.

THỜI ĐIỂM KHẮC PHỤC SỰ CỐ

Phải làm gì khi Google bỏ qua dữ liệu ngữ nghĩa của bạn

Chỉ vì Google có một [cách hiển thị](#) một trang giàu ngữ nghĩa trong cách đặc biệt nào đó không có nghĩa là nó [sẽ](#) Google sử dụng của riêng mình tập hợp các quy tắc bán bí mật để xác định xem ngữ nghĩa thông tin có giá trị cho người tìm kiếm. Nhưng đây là một số các cách chắc chắn để đảm bảo Google bỏ qua dữ liệu của bạn:

- Dữ liệu ngữ nghĩa không thể hiện nội dung chính.
- Nói cách khác, nếu bạn điền chi tiết liên hệ của mình trên một trang về câu cá bằng ruồi, Google sẽ không sử dụng thông tin liên hệ của bạn. (Xét cho cùng, tỷ lệ cược là khi những người tìm kiếm trên web tìm thấy trang này, họ đang tìm kiếm thứ gì đó liên quan đến câu cá, và chẳng có ý nghĩa gì khi thấy một

ghi tên địa chỉ và doanh nghiệp của bạn bên dưới.) Mặt khác, nếu bạn đưa chi tiết liên hệ của mình vào trang sơ yếu lý lịch, thi nhiều khả năng chúng sẽ được sử dụng hơn.

- Ẩn dữ liệu ngữ nghĩa. Google sẽ không sử dụng bất kỳ nội dung nào được ẩn qua CSS.
- Trang web của bạn chỉ sử dụng một ít dữ liệu ngữ nghĩa. Nếu trang web của bạn có tương đối ít trang sử dụng dữ liệu ngữ nghĩa, Google có thể vô tình bỏ qua những trang sử dụng dữ liệu đó.

Tránh những sai lầm này và bạn có cơ hội tốt để nhận được một danh sách nâng cao.

Ngay bây giờ, Google không có công cụ tìm kiếm sơ yếu lý lịch. Tuy nhiên, Google đã thử nghiệm công nghệ tìm kiếm việc làm dành cho cựu chiến binh () <http://tinyurl.com/vetjobsearch> www.google.ca/ nhằm thử nghiệm một trang tìm kiếm hàng triệu công thức nấu ăn.

Đến giờ, bạn có thể đoán dữ liệu công thức trông như thế nào khi được đánh dấu bằng vi dữ liệu hoặc vi định dạng. Toàn bộ công thức nằm bên trong một vùng chứa sử dụng định dạng dữ liệu Công thức (đó là). Có các thuộc tính riêng cho tên công thức [tên công thức](#) và [tên công thức](#) để thêm tóm tắt công thức và [tóm tắt công thức](#).

Đây là một phần của đánh dấu công thức:

```
<div itemscope itemtype="http://data-vocabulary.org/Recipe"> <h1
  itemprop="name"> Súp cà chua tao nhã</h1> <p>Tác giả <span
  itemprop="author">Michael Chiarello</p> <p itemprop="summary">Cà chua
  rang là chìa khóa để phát triển hương vị đậm đà của món này súp cà chua.</p>
```

...

Sau đó, bạn có thể bao gồm các chi tiết chính về công thức, bao gồm thời gian chuẩn bị, thời gian nấu và năng suất. Bạn cũng có thể thêm một phần lồng nhau để biết thông tin dinh dưỡng (với thông tin chi tiết về khẩu phần, lượng calo, chất béo, v.v.):

...

```
<p>Thời gian chuẩn bị: <time itemprop="prepTime" datetime="PT30M">30 phút</
  time></p> <p>Thời gian nấu: <time itemprop="cookTime" datetime="PT1H">40 phút</
  time></p> <p>Năng suất: <span itemprop="yield">4 phần ăn</p>
```

```
<div itemprop="nutrition" itemscope
      itemtype="http://data-vocabulary.org/Nutrition">
    Khẩu phần: <span itemprop="servingSize">1 bát lớn Lượng calo
    mỗi khẩu phần: <span itemprop="calo">250 Chất béo mỗi khẩu
    phần: <span itemprop="fat">3g </div>
```

...

GHI CHÚ

Các thuộc tính `prepTime` và `cookTime` có nghĩa là đại diện cho một khoảng thời gian ngắn và vì vậy chúng không thể sử dụng cùng định dạng với phần tử `<time>` HTML5. Thay vào đó, họ sử dụng định dạng ISO chi tiết tại <http://tinyurl.com/ISOdurations>.

Sau đây là danh sách thành phần của công thức. Mỗi thành phần là một phần lồng nhau riêng biệt, thường bao gồm thông tin như tên và số lượng thành phần:

```
...
<ul>
  <li itemprop="ingredient" itemscope
      itemtype="http://data-vocabulary.org/RecipeIngredient">
    <span itemprop="amount">1 <span itemprop="name"> củ
    hành vàng</span> (thái hạt lựu) </li> <li
    itemprop="ingredient" itemscope itemtype="http://data-
    vocabulary.org/RecipeIngredient"> <span
    itemprop="amount">Can 14 ounce <span itemprop="name">
    cà chua thái hạt lựu</li>
```

```
...
</ul>
...
```

Viết phần này của đánh dấu là tẻ nhạt. Nhưng đừng dừng lại—phần thường còn ở phía trước.

Cuối cùng, hướng dẫn là một loạt các đoạn văn hoặc danh sách các bước. Chúng được gói gọn trong một thuộc tính duy nhất, như thế này:

```
...
<div itemprop="instructions">
  <ol>
    <li>Làm nóng lò nướng ở nhiệt độ 450 độ
    F.</li> <li>Lọc cà chua đóng hộp đã cắt nhỏ, để riêng phần nước ép.</li>
    ...
  </ol>
</div>
```

Để biết ví dụ về công thức đầy đủ, hãy xem <http://tinyurl.com/RichSnippetsRecipe>.

GHI CHÚ

Các công thức nấu ăn có xu hướng dài và khá chi tiết, vì vậy việc đánh dấu chúng là một dự án lâu dài và phức tạp.

Đây là một trường hợp rõ ràng khi một công cụ soạn thảo tốt có thể tạo ra sự khác biệt đáng kể. Lý tưởng nhất là công cụ này sẽ cho phép các tác giả web nhập chi tiết công thức vào các hộp văn bản của một cửa sổ được sắp xếp đẹp mắt. Sau đó, nó sẽ tạo đánh dấu chính xác về mặt ngữ nghĩa mà bạn có thể đặt trong trang web của mình.

Sau khi Google lập chỉ mục trang công thức đã đánh dấu của bạn, nó sẽ cung cấp công thức đó thông qua tính năng tìm kiếm Chế độ xem công thức. Đây là cách dùng thử Chế độ xem công thức:

1. Lướt đến www.google.com/landing/recipes.

Bạn đến trang chủ tính năng Chế độ xem công thức. Nó bao gồm nhiều thông tin về cách hoạt động của Chế độ xem Công thức, bao gồm một video hiển thị hoạt động tìm kiếm công thức.

2. Nhấp vào nút “Dùng thử Google với Chế độ xem công thức”.

Nút này đưa bạn đến trang tìm kiếm Google quen thuộc. Tuy nhiên, có một cái gì đó khác biệt tinh tế. Trong hộp tìm kiếm, tab Công thức được tô sáng màu đỏ, cho biết bạn đang thực hiện tìm kiếm công thức.

3. Nhập tên công thức vào hộp tìm kiếm và nhấp vào nút tìm kiếm.

Google bắt đầu với bạn bằng cách tìm kiếm **mì Ý gà**, nhưng bạn có thể làm tốt hơn.

4. Nhấp vào nút “Công cụ tìm kiếm” (xuất hiện bên dưới bên phải của hộp tìm kiếm).

Vì Google có thể sắp xếp **hầu hết** của mọi công thức nên Google có thể bao gồm các tùy chọn lọc thông minh hơn. Khi bạn nhấp vào “Công cụ tìm kiếm”, Google sẽ gọi ra ba tính năng lọc theo công thức cụ thể, xuất hiện trong danh sách thả xuống, ngay phía trên kết quả tìm kiếm (xem Hình 3-10).

- Thành phần cho phép bạn chọn chỉ xem các công thức nấu ăn bao gồm hoặc bỏ qua một số thành phần nhất định. Bạn chọn bằng cách nhấp vào hộp kiểm Có hoặc Không nhỏ bên cạnh thành phần tương ứng. (Để tạo danh sách thành phần, Google lấy các thành phần được sử dụng phổ biến nhất từ kết quả tìm kiếm cho tìm kiếm của bạn.)

- Bất kỳ thời gian nấu nào cũng cho phép bạn xác định công thức nấu ăn nhanh-ví dụ: những công thức mất dưới một giờ hoặc dưới 10 phút (cảnh báo gà sống).

- Lượng calo bất kỳ cho phép bạn lọc ra các công thức nấu ăn có lượng calo cụ thể相符 mỗi phần, rất tiện cho người ăn kiêng.

CÁCH TÌM KIẾM
ĐỘNG CƠ SỬ DỤNG
METADATA

The screenshot shows a Google search results page for the query "tomato soup". The search bar at the top contains "tomato soup". Below the search bar, there are several tabs: Web, Images, Maps, Shopping, Recipes (which is highlighted in red), More, and Search tools. A sidebar on the left is titled "Ingredients" and includes dropdown menus for "Any cook time" and "Any calories", and a "Clear" button. Under "Ingredients", there are two sections: "Yes" and "No". The "Yes" section lists: Basil, Tomato juice, Mozzarella, Garlic, Eggs, Ciabatta, Tabasco pepper, and Baguette. The "No" section lists: Tomato soup. Below this, there is a link to a recipe from BBC Good Food: "es : Tomato soup co.uk/food/recipes/tomatosoup_13560". The main search results area shows a thumbnail image of a bowl of tomato soup with croutons, followed by the title "Roasted Tomato Soup with Garlic Recipe at Epicurious.com" and a snippet of the recipe description.

HÌNH 3-10

Sau khi bạn thực hiện tìm kiếm, Google sẽ phân loại kết quả theo các tiêu chí sau:

Ở đây, một cuộc tìm kiếm theo từ khóa "tomato soup" nhưng tránh nước ép cà chua.

Dữ liệu ngữ nghĩa mà bạn đã học trong chương này cung cấp cho người lướt web một công cụ tìm kiếm thông tin mạnh mẽ và một cách hiệu quả hơn để tìm các trang web của bạn.