

CSS3

Tradução da Terceira edição

o manual que faltava®

O livro que devia vir na caixa®

David Sawyer McFarland



ALTA BOOKS
E D I T O R A

Rio de Janeiro, 2015

CSS3: O Manual que Faltava Copyright © 2015 da Starlin Alta Editora e Consultoria Eireli.
ISBN: 978-85-7608-863-9

Translated from original The Missing Manual CSS3, Sixth Edition© 2013 by David Pogue ISBN 978-1-449-32594. This translation is published and sold by permission O'Reilly Media, Inc., the owner of all rights to publish and sell the same. PORTUGUESE language edition published by Starlin Alta Editora e Consultoria Eireli, Copyright © 2014 by Starlin Alta Editora e Consultoria Eireli.

Todos os direitos reservados e protegidos por Lei. Nenhuma parte deste livro, sem autorização prévia por escrito da editora, poderá ser reproduzida ou transmitida.

Erratas: No site da editora relatamos, com a devida correção, qualquer erro encontrado em nossos livros (Procure pelo título do livro).

Marcas Registradas: Todos os termos mencionados e reconhecidos como Marca Registrada e/ou Comercial são de responsabilidade de seus proprietários. A Editora informa não estar associada a nenhum produto e/ou fornecedor apresentado no livro.

Impresso no Brasil — 1ª Edição, 2015

Vedada, nos termos da lei, a reprodução total ou parcial deste livro.

Produção Editorial Editora Alta Books	Supervisão e Qualidade Editorial Angel Cabeza Sergio de Souza	Design Editorial Aurélio Corrêa	Captação e Contratação de Obras Nacionais J. A. Rugeri Marco Pace autoria@altabooks.com.br	Vendas Atacado e Varejo Daniele Fonseca Viviane Paiva comercial@altabooks.com.br
Gerência Editorial Anderson Vieira			Marketing e Promoção Hannah Carriello marketing@altabooks.com.br	Ouvidoria ouvidoria@altabooks.com.br
Produtor Editorial Thiê Alves				
Equipe Editorial	Claudia Braga Jessica Carvalho Juliana de Oliveira	Letícia de Souza Mayara Coelho Mayara Soares	Milena Lepsch Rômulo Lentini Silas Amaro	
Tradução Docware Traduções Técnicas	Copidesque Alessandro Thomé	Revisão Gramatical Priscila Gurgel Iara Zanardo	Diagramação Joyce Matos	Revisão Técnica: Roberto de Assis

Dados Internacionais de Catalogação na Publicação (CIP)

M143c McFarland, David Sawyer.
CSS3 : o manual que faltava : o livro que devia vir na caixa /
David Sawyer McFarland. – Rio de Janeiro, RJ : Alta Books, 2015.
640 p. : il. ; 24 cm – (O manual que faltava)

Inclui bibliografia e índice.
Tradução de: The missing manual CSS3 (3. ed.).
ISBN 978-85-7608-863-9

1. Folhas de estilo. 2. Sites da Web - Desenvolvimento. 3.
HTML5 (Linguagem de marcação de documento). I. Título. II.
Série.

CDU 004.438
CDD 006.74

Índice para catálogo sistemático:

1. Folhas de estilo 004.438

(Bibliotecária responsável: Sabrina Leal Araujo – CRB 10/1507)

Sumário

Créditos	vii
Introdução	1

Parte Um: **Noções Básicas de CSS**

CAPÍTULO 1:	HTML para CSS	17
	HTML: Passado e Presente	17
	Escrevendo HTML para CSS	20
	A Importância do Doctype	30
	Certificando-se de que o Internet Explorer Está Atualizado	32
CAPÍTULO 2:	Criando Estilos e Folhas de Estilos	35
	Anatomia de um Estilo	35
	Entendendo Folhas de Estilos	38
	Folhas de Estilos Internas	40
	Folhas de Estilos Externas	41
	Tutorial: Criando Sua Primeira Folha de Estilos	43
CAPÍTULO 3:	Seletores: Identificando o que Estilizar	55
	Seletores de Tag: Estilo de Página Inteira	56
	Seletores de Classe: Controle Preciso	57
	Seletores de ID: Elementos Específicos de Página	60
	Estilizando Grupos de Tags	62
	Estilizando Tags Dentro de Tags	63
	Pseudoclasses e Pseudoelementos	68
	Seletores de Atributo	72
	Seletores de Filho	74
	Irmãs	79
	O Seletor <code>:not()</code>	80
	Tutorial: Amostra de Seletores	81
CAPÍTULO 4:	Economizando Tempo com Herança de Estilo	93
	O que É Herança?	93
	Como a Herança Otimiza Folhas de Estilos	95
	Os Limites da Herança	95
	Tutorial: Herança	97
CAPÍTULO 5:	Gerenciando Múltiplos Estilos: A Cascata	103
	Como Estilizar Cascatas	104
	Especificidade: Qual Estilo Vence	108
	Controlando a Cascata	111
	Tutorial: A Cascata em Ação	117

Parte Dois: **CSS Aplicada**

CAPÍTULO 6:	Formatando Texto	127
	Usando Fontes	127
	Usando Fontes Web	132
	Descobrimos Fontes Web no Google	146
	Adicionando Cor ao Texto	154
	Alterando o Tamanho da Fonte	157
	Formatando Palavras e Letras	162
	Adicionando Sombra ao Texto	166
	Formatando Parágrafos Inteiros	167
	Estilizando Listas	173
	Tutorial: Formatação de Texto em Ação	178
CAPÍTULO 7:	Margens, Preenchimento e Bordas	193
	Compreendendo o Modelo de Caixa	193
	Controlando Espaço com Margens e Preenchimento	195
	Adicionando Bordas	202
	Colorindo o Fundo	206
	Criando Cantos Arredondados	207
	Adicionando Sombras Projetadas	210
	Determinando Altura e Largura	212
	Fazendo o Conteúdo Contornar Elementos Flutuantes da Página	220
	Tutorial: Margens, Fundos e Bordas	225
CAPÍTULO 8:	Adicionando Imagens Gráficas a Páginas Web	239
	Descobrimos CSS e a Tag 	239
	Adicionando Imagens de Fundo	240
	Controlando a Repetição	244
	Posicionando uma Imagem de Fundo	245
	Usando a Forma Abreviada da Propriedade Background	255
	Usando Múltiplas Imagens de Fundo	257
	Utilizando Fundos Degradês	259
	Aplicando Degradês Fáceis com Colorzilla	269
	Tutorial: Melhorando Imagens	272
	Tutorial: Criando uma Galeria de Fotos	278
	Tutorial: Usando Imagens de Fundo	282
CAPÍTULO 9:	Decorando a Navegação de Seu Site	293
	Selecionando Quais Links Estilizar	293
	Estilizando Links	297
	Construindo Barras de Navegação	304
	Estilos de CSS que Pré-Carregam Rollovers	313
	Estilizando Tipos Específicos de Links	315

	Tutorial: Estilizando Links	318
	Tutorial: Criando uma Barra de Navegação	324
CAPÍTULO 10:	Transformações, Transições e Animações CSS	335
	Transformações	335
	Transições	345
	Animações	354
	Tutorial	366
CAPÍTULO 11:	Formatando Tabelas e Formulários	375
	Usando Tabelas da Maneira Certa	375
	Estilizando Tabelas	378
	Estilizando Formulários	384
	Tutorial: Estilizando uma Tabela	390
	Tutorial: Estilizando um Formulário	395

Parte Três: **Layout de Página CSS**

CAPÍTULO 12:	Introduzindo o Layout de CSS	405
	Tipos de Layouts de Página Web	405
	Como Funciona o Layout de CSS	408
	Estratégias de Layout	412
CAPÍTULO 13:	Construindo Layouts Baseados em Elementos Flutuantes	417
	Aplicando Elementos Flutuantes a Seus Layouts	421
	Superando Problemas de Elementos Flutuantes	425
	Tutorial: Layouts Multicolunas	440
CAPÍTULO 14:	Web Design Responsivo	453
	Princípios Básicos do Web Design Responsivo	454
	Configurando uma Página Web para Web Design Responsivo	455
	Consultas de Mídia	457
	Grades Flexíveis	465
	Imagens Fluidas	470
	Tutorial sobre Web Design Responsivo	474
CAPÍTULO 15:	Posicionando Elementos em uma Página Web	487
	Como as Propriedades de Posicionamento Funcionam	488
	Estratégias Poderosas de Posicionamento	502
	Tutorial: Posicionando Elementos de Página	507

Parte Quatro: **CSS Avançada**

CAPÍTULO 16:	CSS Para Página Impressa	519
	Como Funcionam as Folhas de Estilos de Mídia.....	519
	Como Adicionar Folhas de Estilos de Mídia.....	522
	Criando Folhas de Estilos de Impressão.....	523
	Tutorial: Construindo uma Folha de Estilos de Impressão.....	530
CAPÍTULO 17:	Melhorando Seus Hábitos de CSS	539
	Adicionando Comentários	539
	Organizando Estilos e Folhas de Estilos.....	541
	Eliminando Interferência do Estilo de Navegador.....	548
	Utilizando Seletores Descendentes	553
	Tentando uma CSS Diferente para o Internet Explorer.....	558

Parte Cinco: **Apêndices**

APÊNDICE A:	Referência a Propriedades CSS	563
	Valores CSS	563
	Propriedades de Texto	567
	Propriedades de Lista	572
	Preenchimento, Bordas e Margens	573
	Fundos.....	579
	Propriedades de Layout de Página	582
	Propriedades Animation, Transform e Transition	588
	Propriedades de Tabela.....	593
	Propriedades Diversas.....	595
APÊNDICE B:	Recursos de CSS	599
	Referências.....	599
	Ajuda sobre CSS.....	600
	Dicas, Truques e Conselhos sobre CSS.....	600
	Navegação com CSS	601
	Layout com CSS.....	602
	Sites de Apresentação	603
	Livros sobre CSS.....	604
	Software de CSS	604
	Índice	607

Introdução

Cascading Style Sheets (folhas de estilos em cascata, ou CSS, para abreviar) dão a você o controle criativo sobre o layout e o design de suas páginas web. Usando CSS, você pode aprimorar o texto de seu site com títulos atraentes, capitulares e bordas, exatamente como aqueles vistos em revistas de qualidade. Você também pode dispor imagens com precisão, criar colunas e banners e destacar links com efeitos dinâmicos de rollover. Você pode até mesmo fazer com que elementos apareçam e desapareçam gradualmente, mover objetos pela página ou fazer um botão mudar lentamente de cor quando um visitante passa o mouse sobre ele.

Fazer tudo isso deve ser muito complicado, certo? Ao contrário! A ideia por trás da CSS é otimizar o processo de estilização de páginas web. A seguir, você aprenderá os conceitos básicos da CSS. No Capítulo 1, você começará a criar uma página web usando CSS.

■ Como Funciona a CSS

A CSS funciona com HTML, mas não é HTML. Ela é uma linguagem completamente diferente. Embora a HTML forneça a estrutura para um documento organizando informações em títulos, parágrafos, listas com marcadores etc., a CSS funciona lado a lado com o navegador para fazer a HTML ter uma boa aparência.

Por exemplo, você pode usar a HTML para transformar uma frase em um título de nível superior, indicando que ela apresenta o conteúdo do restante da página. Entretanto, você pode usar CSS para formatar esse título, digamos, com um texto vermelho em negrito e posicioná-lo 50 pixels entre a borda esquerda e a janela. Na CSS, essa formatação de texto envolve um estilo — uma regra que descreve a aparência de uma parte específica de uma página web. Uma folha de estilos é um conjunto desses estilos.

Você também pode criar estilos especificamente para trabalhar com imagens. Por exemplo, um estilo pode alinhar uma imagem ao longo da borda direita da página web, circundar a imagem com uma borda colorida e inserir uma margem de 50 pixels entre a imagem e o texto adjacente.

Depois de criar um estilo, você pode aplicá-lo a textos, imagens, títulos ou outros elementos em uma página. Por exemplo, você pode selecionar um parágrafo do texto e aplicar um estilo para mudar instantaneamente o tamanho, a cor e a fonte do texto. Você também pode criar estilos para tags HTML específicas. Por exemplo, todos os cabeçalhos de primeiro nível `<h1>` em seu site são exibidos no mesmo estilo, independentemente de onde eles apareçam.

■ Os Benefícios da CSS

Antes da CSS, web designers estavam *limitados* às opções de layout e formatação da HTML. Se você navegasse pela web em 1995, entenderia perfeitamente a ênfase em *limitados*. A HTML ainda é a base de todas as páginas na World Wide Web, mas ela não é simplesmente uma ferramenta de design. Claro, a HTML fornece opções básicas de formatação de texto, imagens, tabelas e outros elementos da página web, e web masters pacientes e meticulosos podem dar uma boa aparência a páginas utilizando apenas HTML. No entanto, o resultado muitas vezes são páginas lentas, cheias de código deselegante.

A CSS, em contrapartida, oferece as seguintes vantagens:

- Folhas de estilos oferecem muito mais opções de formatação do que HTML. Com CSS, você pode formatar os parágrafos da maneira como eles aparecem em uma revista ou em um jornal (a primeira linha recuada e sem nenhum espaço entre cada parágrafo, por exemplo) e controlar a *entrelinha* (o espaço entre as linhas do texto em um parágrafo).
- Ao usar CSS para adicionar uma imagem de fundo a uma página, você precisa decidir se e como ela é *ladrilhada* (isto é, como ela se repete vertical e horizontalmente na forma de ladrilhos). Com HTML, você não pode nem começar a fazer isso.
- Melhor ainda, os estilos CSS ocupam muito menos espaço do que as opções de formatação da HTML, como a muito odiada tag ``. Geralmente, você pode eliminar vários kilobytes de páginas web cheias de texto usando CSS. Como resultado, suas páginas terão uma excelente aparência e carregarão mais rapidamente.
- Folhas de estilos também tornam a atualização de seu site mais fácil. Você pode agrupar todos os seus estilos em uma única folha de estilos externa vinculada a todas as páginas do seu site. Então, ao editar um estilo, essa alteração é imediatamente aplicada a todo o site sempre que o estilo aparece. Você pode mudar completamente a aparência de um site editando apenas uma única folha de estilos.

■ O que Você Precisa Conhecer

Este livro supõe que você já tenha algum conhecimento em HTML (e talvez também alguma experiência em CSS). Talvez você já tenha construído um ou dois

sites (ou pelo menos uma ou duas páginas) e tenha um pouco de familiaridade com o mar de tags — `<html>`, `<p>`, `<h1>`, `<table>` etc. — que compõem a Hypertext Markup Language. A CSS não faz nada sem a HTML, portanto, para avançar, você precisa saber como criar uma página web usando HTML básica.

Se você já usou HTML para criar páginas web e acha que aquilo que conhece está um pouco enferrujado, a próxima seção apresenta uma reciclagem básica.

DICA

Se você for iniciante em HTML, confira estes tutoriais online gratuitos: HTML Dog (www.htmldog.com/guides/htmlbeginner — conteúdo em inglês) e W3Schools (www.w3schools.com/html — conteúdo em inglês). Se for fã de páginas impressas, você pode obter uma cópia de *Creating a Website: The Missing Manual, Third Edition*, ou *Head First HTML e CSS, Second Edition* (ambos da O'Reilly).

■ HTML: A Estrutura Básica

HTML (Hypertext Markup Language) usa comandos simples chamados *tags* (marcas, etiquetas) para definir as diversas partes de uma página web. Por exemplo, este código HTML cria uma página web simples:

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Oi, eu sou o título desta página web</title>
  </head>
  <body>
    <p>Oi, eu sou um parágrafo nesta página web.</p>
  </body>
</html>
```

Talvez não seja empolgante, mas esse exemplo tem todos os elementos básicos do que uma página web precisa. Você perceberá algo chamado declaração *doctype* no início do código, seguido por `<html>` (entre colchetes), um título, um corpo e algum material — o conteúdo real da página — dentro do corpo, terminando em um `</html>` final.

■ Tipos de Documento

Todas as páginas começam com *doctype* — uma linha de código que identifica a versão da HTML utilizada para escrever a página. Dois doctypes são usados há anos — HTML 4.01 e XHTML 1.0 — e cada um desses doctypes tem dois estilos: *estrito* e *transicional*. Por exemplo, o doctype transicional da HTML 4.01 se parece com isto (os outros doctypes para HTML 4.01 e XHTML 1.0 são semelhantes):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

NOTA

Para exemplos de todos os doctypes, visite www.webstandards.org/learn/reference/templates (conteúdo em inglês).

Se analisar o código da página HTML citado como exemplo nesta seção, você verá que ele usa um doctype muito mais sucinto:

```
<!doctype html>
```

Esse é o doctype novo e mais simples da HTML5. O objetivo da HTML5 é ser mais fácil de usar e mais simples do que suas antecessoras. Este livro usa o doctype da HTML5, que é suportado por todos os navegadores populares (até mesmo pelo antigo Internet Explorer 6). E como a HTML5 é o futuro da web, não há razão para usar os doctypes mais antigos.

NOTA

Simplesmente porque o doctype HTML funciona em navegadores mais antigos não significa que os navegadores entendem todas as tags ou recursos da HTML5. O Internet Explorer 8 e versões anteriores, por exemplo, não reconhecem as novas tags HTML5. Para estilizar tags com a CSS para essas versões do IE, você tem que empregar um pouco de JavaScript. Você aprenderá como atualizar os navegadores mais antigos na página 30.

Independentemente de qual doctype você prefira, o importante é sempre usar um. Sem ele, as páginas terão uma aparência diferente dependendo do navegador utilizado, uma vez que navegadores exibem CSS de uma maneira diferente se não houver um doctype para orientá-los.

Cada doctype exige que você escreva a HTML de uma determinada maneira. Por exemplo, na HTML 4.01, a tag para uma quebra de linha é assim:

```
<br>
```

Mas em XHTML, ela é a seguinte:

```
<br />
```

E há outra vantagem da HTML5: ela aceita qualquer uma das duas.

■ Como as Tags HTML Funcionam

No exemplo simples de HTML na página 3, como no código HTML de qualquer página web, a maioria dos comandos aparece em pares que envolvem um bloco de texto ou outros comandos. Colocadas entre colchetes, essas *tags* instruem um navegador de web sobre como exibir a página da web. Tags são a parte da “marcação” da Hypertext Markup Language.

A tag à esquerda (de abertura) de cada par informa ao navegador onde a instrução começa e a tag à direita indica onde a instrução termina. Tags de fechamento, ou finais, sempre incluem uma barra (/) depois do primeiro símbolo de menor que (<).

Em qualquer página web, você normalmente encontrará pelo menos estes quatro elementos.

- A primeira linha de uma página web é a **declaração DOCTYPE**, discutida na seção anterior.
- A tag **<html>** aparece uma vez no início de uma página web e novamente no final (com uma barra adicionada): **</html>**. Essa tag informa um navegador web que a informação contida nesse documento está escrita em HTML, ao

contrário de outras linguagens. Todo o conteúdo de uma página, incluindo outras tags, aparece entre as tags de abertura e fechamento `<html>`.

Se você imaginar que uma página web é como uma árvore, a tag `<html>` seria as raízes. A partir do tronco há dois galhos que representam as duas partes principais de qualquer página web: a *cabeça* e o *corpo*.

- A *cabeça* de uma página web contém o título da página (“Loja de Lápis da Izzie”). Ela também poderia incluir outras informações invisíveis, como uma descrição da página, que os navegadores e os mecanismos de busca utilizam. A seção de cabeçalho é inserida entre as tags `<head>` de abertura e fechamento.

Além disso, a seção de cabeçalho pode incluir informações que os navegadores usam para formatar a HTML da página e adicionar interatividade. Como veremos, a seção `<head>` pode conter código CSS (como o tipo que você aprenderá a escrever neste livro) ou um link para outro arquivo contendo informações CSS.

- O *corpo*, separado pelas tags `<body>` circundantes, contém todo o conteúdo que aparece dentro de uma janela do navegador — títulos, texto, imagens etc.

Dentro da tag `<body>`, você normalmente encontra tags como estas:

- Você instrui um navegador web onde um parágrafo de texto começa com um `<p>` (tag de abertura de parágrafo) e onde ele termina com um `</p>` (tag de fechamento de parágrafo).
- A tag `` enfatiza o texto. Ao envolver parte do texto com essa tag e sua tag parceira ``, você obtém negrito. O trecho de HTML `Atenção!` instrui um navegador web a enfatizar fortemente a palavra “Atenção!”
- A tag `<a>`, ou tag âncora, cria um hiperlink em uma página web. Quando clicado, um hiperlink — ou link — pode levar a qualquer lugar na web. Você informa ao navegador para onde o link aponta inserindo um endereço web dentro das tags `<a>`. Por exemplo, você pode digitar `Clique aqui!`.

O navegador sabe que, quando o visitante clicar nas palavras “Clique aqui!”, ele deve acessar o site Missing Manual. A parte `href` da tag é chamada atributo e a URL (Uniform Resource Locator, ou endereço web) é o *valor*. Nesse exemplo, `http://www.missingmanuals.com` (conteúdo em inglês) é o *valor* do atributo `href`.

■ XHTML Também

Como qualquer tecnologia, a HTML evoluiu ao longo do tempo. Embora a HTML padrão cumprisse bem sua finalidade, ela sempre foi uma linguagem um pouco desleixada, sem rigor. Entre outras coisas, ela permite usar letras maiúsculas, minúsculas e maiúsculas e minúsculas nas tags (`<body>` e `<BODY>` estão corretos, por exemplo) e permite tags não fechadas (assim, você pode usar uma tag `<p>` sem uma tag `</p>` de fechamento para criar um parágrafo). Embora essa flexibilidade possa tornar mais fácil escrever uma página, ela também torna mais difícil a vida dos navegadores, smartphones e outras tecnologias que interagem com dados na web.

Além disso, a HTML não funciona com uma das linguagens mais importantes da internet: XML, ou Extensible Markup Language.

Para se manter atualizada, uma versão melhorada da HTML, chamada XHTML, foi introduzida em 2000, e você descobrirá que ela é usada com frequência em muitos sites (na verdade, XHTML é uma versão “XML-zada” da HTML). A XHTML era vista como o futuro (em 2000), mas desde então foi suplantada pela HTML5. Embora os navegadores web ainda entendam XHTML (e provavelmente entenderão por muito tempo), este livro não a inclui.

■ HTML5: A Roda Gira de Novo

Quando o W3C — o grupo responsável por muitas tecnologias da internet — introduziu a XHTML como um passo intermediário no processo de transição para a XML como a linguagem principal da web, a comunidade de desenvolvimento web a anunciou como o próximo grande avanço. A história mostrou que essa previsão era um pouco grandiosa demais. Como vimos, a complexidade da migração para a XML evitou que os fabricantes de navegadores seguissem o caminho da XML que o World Wide Web Consortium estabeleceu. Em vez disso, o W3C encerrou o grupo de trabalho da XHTML em favor da HTML5, uma nova versão da HTML que se afasta da XML e volta à HTML, embora uma versão melhorada. Com o suporte de todos os fabricantes de navegadores, a HTML5 é o novo próximo avanço. Alguns navegadores, incluindo o Google Chrome e o Firefox, já reconhecem partes da HTML5 e até mesmo as principais agências de notícias, como Time, Newsweek e CNN, usam HTML5 em reportagens sobre tecnologia.

A HTML5 não é uma tecnologia radicalmente nova. Na verdade, ao contrário da XHTML, cujo objetivo era promover uma nova maneira de construir páginas web, a HTML5 assegura que a web continue a funcionar da maneira como sempre funcionou. A maioria dos conceitos básicos de HTML ainda é a mesma. Além disso, a HTML5 adiciona alguns elementos novos que visam a dar suporte à maneira como web designers constroem sites atualmente. Por exemplo, na HTML5, a tag `<header>` pode conter o conteúdo normalmente encontrado no topo de uma página, como um logotipo e links de navegação por todo um site; a nova tag `<nav>` inclui o conjunto de links usados para navegar por um site; e a tag `<footer>` hospeda o material geralmente inserido na parte inferior de uma página, como avisos legais, contatos por e-mail etc.

Além disso, a HTML5 adiciona novas tags que permitem inserir áudio e vídeo em uma página, e novas tags de formulário que agregam elementos sofisticados, como controles deslizantes e seletores pop-up de datas, bem como suporte embutido a navegadores para validação de formulário (o que garante que os visitantes preencham corretamente os formulários). Infelizmente, o suporte a navegadores desses novos recursos não é consistente, então é difícil usar as novas tags sem algumas soluções alternativas muito elaboradas.

Embora a HTML5 talvez não esteja totalmente pronta para estrear, não há nenhuma razão para não começar a usar agora o doctype da HTML5 e até mesmo alguns dos recursos básicos da HTML5. (Mas você precisará ajudar um pouco o Internet Explorer 8 e as versões anteriores; veja o quadro na página 24.) Discutiremos HTML5 em todo este livro, especialmente no próximo capítulo.

■ Software para CSS

Para criar páginas web compostas de HTML e CSS, você só precisa de um editor de texto simples, como o Notepad (Windows) ou o TextEdit (Mac). No entanto, depois de digitar centenas de linhas de código HTML e CSS, é recomendável experimentar um programa mais adequado para trabalhar com páginas web. Esta seção lista alguns programas comuns, alguns gratuitos e outros que você tem que comprar.

NOTA Existem literalmente centenas de ferramentas que podem ajudá-lo a criar páginas web, portanto a lista a seguir não é completa. Pense nela como uma turnê pelos grandes sucessos dos programas mais populares que os fãs de CSS utilizam hoje em dia.

Programas Gratuitos

Há uma grande quantidade de programas gratuitos disponíveis para editar páginas web e folhas de estilo. Se você ainda estiver usando o Notepad ou o TextEdit, experimente um destes. Eis uma pequena lista para você começar:

- **jEdit** (Windows, Mac, Linux; <http://jedit.org> — conteúdo em inglês). Esse editor de texto gratuito baseado em Java funciona em praticamente qualquer computador e inclui muitos recursos que você encontraria em editores de texto comerciais, como destaque de sintaxe para CSS.
- **Notepad++** (Windows; <http://notepad-plus.sourceforge.net> — conteúdo em inglês). Muitas pessoas confiam nesse editor de texto rápido. Ele tem até mesmo recursos embutidos que o tornam ideal para escrever HTML e CSS, como destaque de sintaxe, tags de codificação em cores e palavras-chave especiais para tornar mais fácil identificar os elementos de uma página HTML e CSS.
- **TextWrangler** (Mac; www.barebones.com/products/textwrangler — conteúdo em inglês). Esse software livre é na verdade uma versão mais simples do BBEdit, o famoso e sofisticado editor de texto para Mac. O TextWrangler não tem todas as ferramentas de HTML integradas do BBEdit, mas inclui destaque de sintaxe, FTP (para que você possa fazer upload de arquivos para um servidor web) e muito mais.

Softwares Comerciais

Programas comerciais de desenvolvimento de site variam de editores de texto de baixo custo a ferramentas de construção de sites completas com todos os recursos:

- **EditPlus** (Windows; www.editplus.com — conteúdo em inglês) é um editor de texto barato que inclui destaque de sintaxe, FTP, autocompletar e outros recursos que poupam esforço de digitação.
- **skEdit** (Mac; www.skedit.com — conteúdo em inglês) é um editor barato e completo de página web, com FTP/SFTP, dicas de códigos e outros recursos úteis.
- **Coda2** (Mac; www.panic.com/coda — conteúdo em inglês) é um kit completo de ferramentas para desenvolvimento web. Ele inclui um editor de texto, visualização de página, FTP/SFTP e ferramentas gráficas de construção de CSS para criar CSS.
- **Sublime Text** (Mac; www.sublimetext.com — conteúdo em inglês) é um editor de texto relativamente novo. Ele é adorado por codificadores no Mac.

- **Dreamweaver** (Mac e Windows; www.adobe.com/products/dreamweaver — conteúdo em inglês) é um editor visual de páginas web. Ele permite ver como a página aparecerá em um navegador web. O programa também inclui um poderoso editor de texto e excelentes ferramentas de gerenciamento e criação de CSS. Confira o *Dreamweaver CS6: The Missing Manual* para detalhes sobre como usar esse programa poderoso.
- **Expression Web 2** (Windows; www.microsoft.com/expression — conteúdo em inglês) é a entrada da Microsoft na área de web design. Ele inclui muitas ferramentas profissionais de web design, incluindo excelentes ferramentas de CSS.

NOTA

Os vários softwares discutidos nesta seção são programas de uso geral que permitem editar HTML/XHTML e CSS. Com eles, você só precisa aprender um único programa para suas necessidades de desenvolvimento web. Mas se você já possui um editor HTML/XHTML preferido que não tem recursos de CSS, então talvez você queira verificar um dos programas de edição específicos de CSS abordados no Apêndice B.

■ Sobre Este Livro

A World Wide Web é muito fácil de usar. Afinal de contas, vovós de pequenas cidades do interior e crianças aprendendo a ler no meio da selva amazônica se conectam à web todos os dias. Infelizmente, as regras que governam como a web *funciona* não são tão fáceis de entender. Analistas de sistemas e outros tipos de tecnólogos que escrevem a documentação oficial não estão interessados em explicar seus conceitos para o João da Silva. Visite www.w3.org/TR/css3-transforms (conteúdo em inglês) para ter uma ideia dos jargões que esses geeks usam.

Pessoas aprendendo CSS muitas vezes não sabem por onde começar. E as questões mais refinadas da CSS podem enganar até mesmo profissionais web experientes. O propósito deste livro, portanto, é servir como o manual que deveria ter vindo com a CSS. Nas páginas deste livro, você encontrará instruções passo a passo sobre como usar a CSS para criar belas páginas web.

CSS3: O Manual que Faltava é projetado para ajudar leitores em todos os níveis técnicos. Para tirar o máximo proveito deste livro, você deve conhecer um pouco de HTML e talvez até mesmo de CSS. Portanto, se você nunca construiu uma página web, verifique o tutorial que começa na página 43. As discussões primárias nestes capítulos foram escritas para iniciantes avançados ou intermediários. Se você estiver começando a construir páginas web, quadros especiais chamados “Atualizando” fornecem as informações introdutórias de que você precisa para entender o tema. Se, por outro lado, você tiver experiência em páginas web, preste atenção às caixas semelhantes, chamadas “Clínica do Usuário Avançado”. Elas oferecem mais dicas técnicas, truques e atalhos para o fã experiente de informática.

ATUALIZANDO

As Diferentes Versões da CSS

Como ocorre com sistemas operacionais e iPods, continuamente são lançadas novas versões da CSS (bem, não tão frequentemente quanto os modelos de iPod). A CSS 1, introduzida em 1996, estabeleceu as bases para as folhas de estilo em cascata. A estrutura básica de um estilo, o conceito de seletor (Capítulo 3) e a maioria das propriedades de CSS neste livro estavam nessa primeira versão.

A CSS 2 adicionou novas funcionalidades, incluindo a capacidade de direcionar a CSS para diferentes impressoras, monitores e outros dispositivos. A CSS 2 também adicionou novos seletores e a capacidade de posicionar precisamente elementos em uma página web.

Este livro cobre completamente a CSS 2.1, que é o padrão atualmente aceito. Ela incorpora toda a CSS 1, adiciona várias novas propriedades e corrige alguns problemas com as diretrizes da CSS 2.

A CSS 2.1 inclui algumas adições e revisões. Ela não foi uma mudança radical em relação à versão 2, e a maioria dos navegadores web se adaptou muito bem às novas regras.

Mais recentemente, a CSS3 vem recebendo muita atenção da imprensa (junto com sua parente HTML5). Ao contrário

das versões anteriores da CSS, a CSS3 não é um único padrão. À medida que a CSS se tornou mais complexa, o W3C dividiu a CSS em módulos separados — o módulo Seletores, o módulo Values e Units, o módulo Box Alignment etc. Como cada módulo pode ser desenvolvido independentemente dos outros, não há nenhum padrão chamado “CSS3”. Na verdade, o nível 3 do módulo Seletores está completo, e o trabalho no nível 4 está em andamento.

Em outras palavras, o que é conhecido como CSS3 é apenas um conjunto disperso de diferentes módulos em diferentes fases de finalização. Os fabricantes de navegadores já incorporaram partes dos módulos mais recentes, mas outros módulos não são suportados em nenhum navegador. No futuro, não haverá uma CSS4; haverá apenas novas versões dos diferentes módulos, cada um em um nível diferente de trabalho.

Por essas razões, este livro discute a essência da CSS 2.1 (que simplesmente foi agregada aos vários módulos da CSS3), bem como as novas propriedades CSS mais empolgantes, populares e amplamente suportadas.

NOTA Este livro recomenda ao longo do texto outros livros sobre CSS que abordam temas muito especializados ou tangenciais para um manual. Às vezes, os títulos recomendados são da série *Missing Manual* da editora O'Reilly, mas nem sempre. Se houver um livro excelente publicado que não seja parte da família O'Reilly, faremos com que você saiba sobre ele.

Sobre a Estrutura de Tópicos

CSS3: O Manual que Faltava é dividido em cinco partes; as quatro primeiras contêm vários capítulos, e a última parte contém os apêndices:

- **Parte Um: Noções Básicas de CSS**, mostra como criar folhas de estilos e fornece uma visão geral dos principais conceitos de CSS, como herança, seletores e a cascata. Ao longo do caminho, você aprenderá as melhores práticas para escrever HTML ao trabalhar com a CSS. Os tutoriais reforçam os principais conceitos de cada parte e introduzem o poder da CSS.



- **Parte Dois: CSS Aplicada**, leva você ao mundo real do web design. Você aprenderá as propriedades de CSS mais importantes e como usá-las para formatar texto, criar ferramentas de navegação úteis e melhorar a página com imagens gráficas. Você aprenderá a criar animações simples com a CSS. Essa seção também fornece conselhos sobre como criar tabelas e formulários atraentes.
- **Parte Três: Layout de Página CSS**, ajuda você em um dos aspectos mais confusos e gratificantes da CSS — controlar o posicionamento dos elementos em uma página. Você aprenderá a criar designs comuns (como layouts de duas e três colunas) e a adicionar barras laterais, além de abordar elementos flutuantes e posicionamento — duas técnicas CSS comuns para controlar o layout de página. Você também aprenderá a criar e adaptar sites para que eles tenham uma boa aparência em navegadores desktop, tablets e celulares.
- **Parte Quatro: CSS Avançada**, ensina como dar a páginas web uma boa aparência quando impressas e discute as técnicas avançadas para usar a CSS de forma mais eficaz e eficiente.
- **Parte Cinco: Apêndices**, fornece dois conjuntos de recursos. A referência a propriedades CSS resume cada propriedade CSS em pequenos fragmentos fáceis de digerir para que você possa entender rapidamente as propriedades CSS úteis que talvez ainda não tenha visto ou aprimorar o que você já conhece. O último apêndice discute as ferramentas e os recursos para criar e utilizar a CSS.


■ Os Conceitos Mais Básicos

Para usar este livro, e de fato usar um computador, você precisa entender alguns conceitos básicos. Você deve conhecer estes termos e conceitos:

- **Clicar.** Este livro fornece três tipos de instruções que exigem o uso do mouse ou trackpad. *Clicar* significa apontar o cursor de seta para algo na tela e então — sem mover o cursor — pressionar e soltar o botão esquerdo do mouse (ou trackpad do laptop). Um clique com o botão direito do mouse é a mesma coisa que usar o botão direito do mouse. (Em um Mac, pressione Control ao clicar se o botão direito do mouse não existir.)

Dar um *clique duplo* significa clicar duas vezes em rápida sucessão, sem mover o cursor. E *arrastar* significa mover o cursor *enquanto* pressiona o botão.

Ao ser instruído a *clicar* em  no Mac, ou *clicar* com a tecla Ctrl pressionada em algo em um PC, clique enquanto pressiona a tecla  ou Ctrl (ambas estão ao lado da barra de espaço).

- **Menus.** Os *menus* são as palavras no topo da tela ou janela: Arquivo, Editar etc. Clique em um para que uma lista de comandos apareça, como se eles estivessem escritos em uma persiana que você acabou de fechar. Este livro supõe que você sabe como abrir um programa, navegar na web e baixar arquivos. Você deve saber como usar o menu Iniciar (Windows) ou o Dock ou menu  (Mac), bem como o Painel de Controle (Windows) ou System Preferences (Mac OS X).

- **Atalhos de teclado.** Toda vez que você tira a mão do teclado para mover o mouse, você perde tempo e potencialmente interrompe seu fluxo criativo. É por isso que muitos usuários de computador experientes, sempre que possível, usam combinações de teclas em vez dos comandos de menu. Ao ver um atalho como Ctrl+S (⌘-S, que salva as alterações no documento atual), ele instrui a manter pressionada a tecla Ctrl ou ⌘ e, enquanto ela está pressionada, digitar a letra S e, então, soltar as duas teclas.

Sobre→Essas→Setas

Ao longo deste livro, e ao longo da série *Missing Manual*, você encontrará frases como esta: “Abra a pasta System→Library→Fonts”. Isso é um atalho para uma instrução muito mais longa, que solicita que você abra três pastas aninhadas na sequência, desta maneira: “No disco rígido, você encontrará uma pasta chamada System. Abra essa pasta. Dentro da janela da pasta System há uma pasta chamada Library; clique duas vezes para abri-la. Dentro dessa pasta há ainda outra chamada Fonts. Também clique duas vezes para abri-la.”

Do mesmo modo, esse tipo de forma abreviada de seta ajuda a simplificar a maneira de escolher comandos nos menus, como mostrado na Figura I-1.



FIGURA I-1

Neste livro, anotações de seta ajudam a simplificar as instruções de menu. Por exemplo, View→Text Size→Increase é uma forma mais compacta de dizer: “No menu View, escolha Text Size; no submenu que aparece, escolha Increase”. (A propósito, quando você vê uma combinação de teclas depois de um comando de menu, como Ctrl++ aqui, isso significa que você pode pressionar essas teclas como um atalho.)

■ Sobre os Recursos Online

Como o proprietário de um Missing Manual (Manual que Faltava, em nossa versão traduzida), você tem mais do que apenas um livro para ler. Online, você encontrará arquivos de exemplos para obter uma experiência prática, bem como dicas, artigos e talvez até mesmo alguns vídeos. Você também pode entrar em contato com a equipe do *Missing Manual* e comentar o que você ama (ou odeia) no livro. Visite www.missingmanuals.com (conteúdo em inglês), ou acesse diretamente uma das seguintes seções.

Exemplos Práticos

Este livro foi projetado para que você coloque seu trabalho na web da maneira mais rápida e profissional possível. É natural, então, que metade da importância deste livro esteja na web.

Ao ler os capítulos do livro, você encontrará uma série de exemplos práticos — tutoriais passo a passo que você mesmo pode construir, utilizando matérias-primas (como imagens gráficas e páginas web semiconcluídas) que você pode baixar em www.sawmac.com/css3/ (conteúdo em inglês ou em www.altabooks.com.br (procure pelo título do livro). Talvez você não aprenda muito simplesmente lendo essas lições passo a passo enquanto relaxa na rede da varanda. Mas se analisá-las no computador, você descobrirá que esses tutoriais fornecem um insight sobre como designers profissionais constroem páginas web.

Você também encontrará, nas lições deste livro, as URLs das páginas finais para que possa comparar seu trabalho com o resultado final. Em outras palavras, você não apenas verá imagens com a aparência que as páginas web devem ter, como também encontrará as páginas web reais funcionando na internet.

Sobre o *MissingManuals.com*

Em www.missingmanuals.com (conteúdo em inglês), você encontrará artigos, dicas e atualizações para *CSS3: O Manual que Faltava*. Na verdade, o convidamos e o incentivamos a sugerir correções e atualizações. Em um esforço para manter o livro o mais atualizado e preciso possível, sempre mais cópias deste livro são impressas, e faremos quaisquer correções confirmadas que você sugerir. Também assinalaremos essas alterações no site para que você possa marcar as correções importantes em sua cópia do livro, se quiser. (Acesse www.missingmanuals.com/feedback — conteúdo em inglês, escolha o nome do livro no menu pop-up, e então clique em Go para ver as alterações.)

Também na nossa página Feedback, ao ler este livro, você pode obter respostas de especialistas a perguntas que surgirem, escrever uma resenha do livro e encontrar grupos de pessoas que compartilham seu interesse em CSS.

Queremos muito ouvir suas sugestões para novos livros na série *Missing Manual*. Também há um lugar para elas em missingmanuals.com. E enquanto você está online, também pode registrar este livro em www.oreilly.com (conteúdo em inglês; você pode acessar diretamente a página de cadastro em: www.oreilly.com/register). O cadastro permite que você envie atualizações sobre este livro, e você também poderá receber ofertas especiais, como descontos nas edições futuras de *CSS3: O Manual que Faltava*.

Noções Básicas de CSS

CAPÍTULO 1:
HTML para CSS

CAPÍTULO 2:
Criando Estilos e Folhas de Estilos

CAPÍTULO 3:
Seletores: Identificando o que Estilizar

CAPÍTULO 4:
Economizando Tempo com Herança de Estilo

CAPÍTULO 5:
Gerenciando Múltiplos Estilos: A Cascata

HTML para CSS

Para tirar o máximo proveito da CSS, o código HTML precisa fornecer uma base sólida e bem construída. Este capítulo mostra como escrever HTML melhor e de maneira mais adequada à CSS. A boa notícia é que, quando você usa CSS em seu site, na verdade torna-se mais fácil escrever HTML. Você não precisa mais se preocupar em tentar transformar HTML em uma ferramenta de design, algo para o que ela nunca foi projetada. Em vez disso, a CSS oferece a maioria dos recursos de design gráfico que você provavelmente vai querer, e páginas HTML escritas para funcionar com CSS são mais fáceis de criar, uma vez que requerem menos código e menos digitação. O download delas também será mais rápido — um bônus que os visitantes de seu site certamente apreciarão (ver Figura 1-1).

■ HTML: Passado e Presente

Como discutido na introdução, a HTML fornece a base para todas as páginas que você encontra na World Wide Web. Ao adicionar CSS ao mix, a maneira como você usa a HTML muda. Diga adeus ao reaproveitamento desajeitado de tags HTML apenas para alcançar certos efeitos visuais. Esqueça completamente algumas tags e atributos HTML — como a velha tag ``.

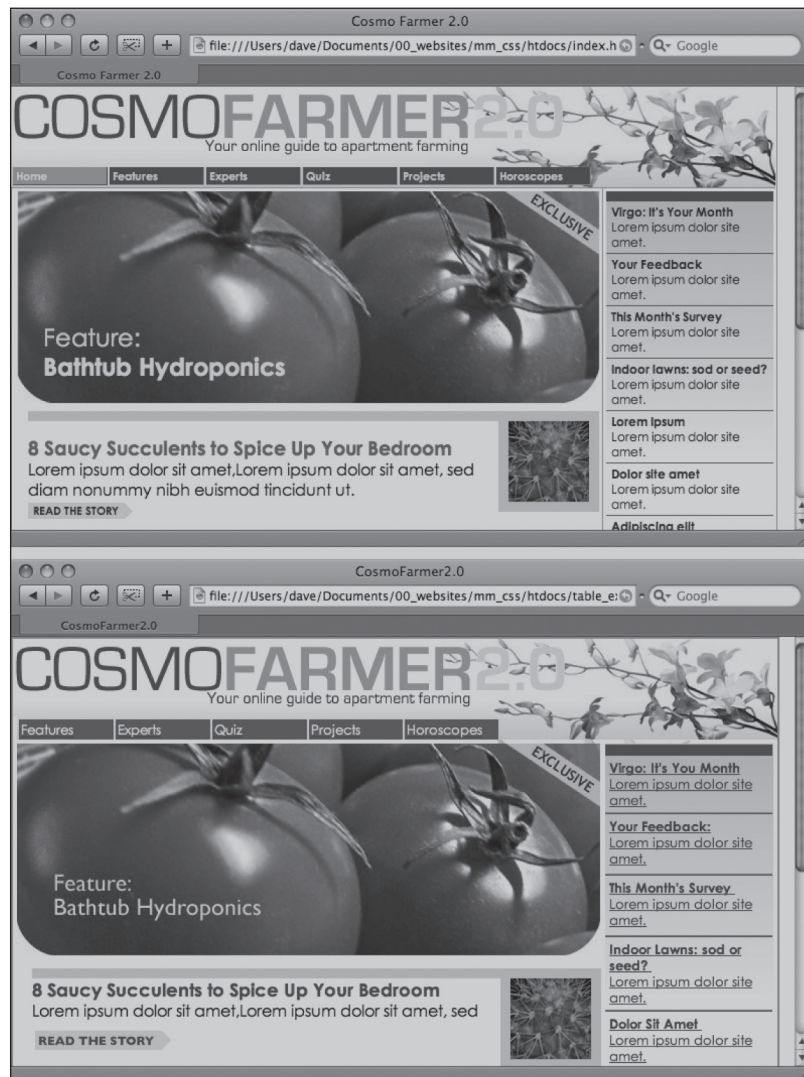


FIGURA 1-1

Web design baseado em CSS torna a escrita HTML mais fácil. Os dois designs apresentados aqui parecem semelhantes, mas a página de cima está totalmente estilizada com CSS, enquanto a página de baixo usa apenas HTML. O tamanho do arquivo HTML para a página no topo é de apenas 4k, enquanto a página somente com HTML é quase quatro vezes mais pesada, com 14k. A abordagem somente com HTML requer muito mais código para alcançar praticamente os mesmos efeitos visuais: 213 linhas de código HTML, em comparação com 71 linhas para a versão CSS.

A HTML do Passado: Qualquer Coisa Parecia Boa

Quando um grupo de cientistas criou a web para ajudar a compartilhar e monitorar documentação técnica, ninguém chamou os designers gráficos. Tudo do que os cientistas precisavam era HTML para estruturar informações a fim de facilitar a compreensão. Por exemplo, a tag `<h1>` indica um título importante, enquanto a tag `<h2>` representa um título menos importante, geralmente um subtítulo da tag

`<h1>`. Outro favorito, a tag `` (lista ordenada) cria uma lista numerada para coisas como “As 10 principais razões para não brincar com águas-vivas”.

Mas assim que as pessoas, além dos cientistas, começaram a usar HTML, elas queriam que suas páginas web tivessem uma boa aparência. Assim, web designers começaram a usar tags para controlar a aparência, em vez de estruturar informações. Por exemplo, você pode usar a tag `<blockquote>` (destinado a material citado de outra fonte) para qualquer texto que você queira recuar um pouco. Você pode utilizar tags de título para tornar o texto maior e mais escuro — independentemente de ele funcionar como um título.

Em uma solução alternativa ainda mais elaborada, os designers aprenderam a usar a tag `<table>` para criar colunas de texto e posicionar com precisão imagens e texto em uma página. Infelizmente, como o objetivo dessa tag era mostrar dados no estilo de planilha — resultados de pesquisas, horários de trens etc. —, os designers tiveram de ser criativos, usando a tag `<table>` de uma maneira incomum, às vezes aninhando outra tabela dentro de uma tabela dentro de uma tabela para que a aparência das páginas fosse boa.

Enquanto isso, os fabricantes de navegadores introduziram novas tags e atributos com a finalidade específica de melhorar a aparência de uma página. A tag ``, por exemplo, permite especificar uma cor de fonte, tipo de letra e um entre sete tamanhos diferentes. (Se estiver fazendo a contagem em casa, isso é aproximadamente 100 vezes menos tamanhos do que é possível obter com o Microsoft Word, por exemplo.)

Por fim, quando os designers não conseguiam exatamente aquilo que queriam, eles costumavam recorrer ao uso de imagens. Por exemplo, eles criavam uma grande imagem para capturar a fonte e o layout exatos dos elementos da página web e então dividiam os arquivos do Photoshop em arquivos menores e os reagrupavam em tabelas para recriar o design original.

Embora todas as técnicas anteriores — usar tags de forma criativa, tirar vantagem dos atributos de tags específicos de design e fazer uso extensivo de imagens — forneçam controle de design sobre as páginas, elas também adicionam muito código HTML extra (e mais rugas à testa).

A HTML do Presente: Andaimos para a CSS

Independentemente do conteúdo que a página web contém — o calendário da temporada de pesca, instruções para localizar a loja de departamentos mais próxima ou fotos da festa de aniversário de seu filho —, é o design da página que faz com que ela pareça um trabalho profissional ou um hobby esporádico. Um bom design aprimora a mensagem de seu site, ajuda os visitantes a encontrar o que estão procurando e determina como o resto do mundo vê seu site. É por isso que os web designers passaram pelas dificuldades descritas na seção anterior para fazer com que a HTML tivesse uma boa aparência. Ao assumir essas funções de design, a CSS deixa a HTML voltar a fazer o que ela faz de melhor — estruturar conteúdo.

Usar HTML para controlar a aparência do texto e outros elementos da página web tornou-se obsoleto. Não se preocupe se o tamanho da tag `<h1>` da HTML é muito grande ou se as listas com marcadores não estão espaçadas da maneira certa. Você pode cuidar dessas listas usando CSS. Em vez disso, pense em HTML como um método para adicionar estrutura ao conteúdo que você quer inserir na web. Use HTML para organizar o conteúdo e CSS para fazer com que o conteúdo tenha uma ótima aparência.

■ Escrevendo HTML para CSS

Se for iniciante em web design, talvez você precise de algumas dicas úteis para orientar suas incursões em HTML (e para se manter longe das técnicas de HTML bem-intencionadas, mas ultrapassadas). Ou, se já construiu páginas web por um tempo, talvez você tenha incorporado alguns hábitos ruins ao escrever HTML, e então é melhor esquecê-los. O restante deste capítulo apresenta alguns hábitos para escrever HTML que deixarão sua mãe orgulhosa e o ajudarão a tirar o máximo proveito da CSS.

Pense em Estrutura

A HTML adiciona significado ao texto dividindo-o logicamente e identificando o papel que o texto desempenha na página: por exemplo, a tag `<h1>` é a introdução mais importante para o conteúdo de uma página. Outros cabeçalhos permitem dividir o conteúdo em outras seções menos importantes, mas relacionadas. Assim como o livro que você tem em mãos, por exemplo, uma página web deve ter uma estrutura lógica. Cada capítulo neste livro tem um título (pense em `<h1>`) e várias seções (pense em `<h2>`), que, por sua vez, contêm subseções menores. Imagine como seria muito mais difícil ler essas páginas se cada palavra simplesmente fosse escrita como um único longo parágrafo.

NOTA

Para um tutorial sobre HTML, visite www.w3schools.com/html/html_intro.asp (conteúdo em inglês). Para uma lista rápida de todas as tags HTML disponíveis, visite <http://reference.sitepoint.com/html> (conteúdo em inglês).

A HTML fornece muitas outras tags, além de cabeçalhos para marcar o conteúdo a fim de identificar seu papel na página. (Afinal de contas, o M em HTML significa *markup*, isto é, marcação.) Entre as mais populares estão a tag `<p>` para parágrafos de texto e a tag `` para criar listas com marcadores (não numeradas). Tags menos conhecidas podem indicar tipos muito específicos de conteúdo, como `<abbr>` para abreviaturas e `<code>` para código de computador.

Ao escrever HTML para CSS, use uma tag que corresponda bem ao papel que o conteúdo desempenha na página, não à aparência (veja a Figura 1-2). Por exemplo, uma grande quantidade de links em uma barra de navegação, na verdade, não é um título, e também não é um parágrafo normal de texto. É mais como uma lista com marcadores de opções, assim a tag `` é uma boa escolha. Se você estiver dizendo, “Mas itens em uma lista com marcadores são empilhados verticalmente uns sobre os outros, e quero uma barra horizontal de navegação em que cada link permaneça ao lado do link anterior”, não se preocupe. Com a mágica da CSS, você pode transformar uma lista vertical de links em uma barra horizontal elegante de navegação, como descrito no Capítulo 9.

```
<p>
<strong>
<font color="#0066FF" size="5" face="Verdana,
Arial, Helvetica, sans-serif">Urban Agrarian
Lifestyle</font></strong>
<br />
<font color="#FF3300" size="4" face="Georgia,
Times New Roman, Times, serif">
<em>
<strong>A Revolution in Indoor Agriculture
<br /></strong></em></font>
Lorem ipsum dolor sit amet...</p>
```

The Urban Agrarian Lifestyle

A Revolution in Indoor Agriculture

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exercitation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure.

```
<h1>The Urban Agrarian Lifestyle</h1>
<h2>A Revolution in Indoor Agriculture</h2>
<p>Lorem ipsum dolor sit amet...</p>
```

FIGURA 1-2

Velha escola, nova escola. Antes da CSS, designers tinham de recorrer à tag e a outras tags de HTML para alcançar determinados efeitos visuais (acima). Você pode alcançar o mesmo visual (e muitas vezes um melhor) com bem menos código HTML (abaixo). Além disso, usar CSS para a formatação o libera para escrever HTML que segue a estrutura lógica do conteúdo da página.

Duas Tags HTML a Ter em Mente

O sortimento heterogêneo de tags que a HTML disponibiliza não abrange a ampla variedade de conteúdo que você provavelmente adicionará a uma página. Claro, <code> é ótima para marcar código de programa de computador, mas a maioria das pessoas acharia a tag <recipe> mais prática. Pena que não exista uma. Felizmente, a HTML apresenta duas tags genéricas que permitem identificar melhor o conteúdo e, no processo, fornece um jeito de anexar estilos CSS a diferentes elementos em uma página.

ATUALIZANDO

A HTML Simples É Amigável a Mecanismos de Busca

Depois de dar o salto mental para imaginar HTML como uma maneira de estruturar o conteúdo de um documento e CSS como a ferramenta para fazer o conteúdo ter uma boa aparência, você descobrirá os benefícios adicionais para escrever HTML concisa e significativa. Antes de tudo, você pode melhorar bastante seu ranking nos mecanismos de busca conforme determinado por sites como Google, Yahoo e Bing. Isso ocorre porque, quando os mecanismos de busca varrem a web, indexando o conteúdo nos sites, eles precisam analisar toda a HTML em uma página para descobrir o conteúdo real. A velha maneira da HTML de usar tags especiais (como ``) e uma grande quantidade de tabelas para criar uma página interfere no desempenho de um mecanismo de busca. Na verdade, alguns mecanismos de busca param de ler HTML em uma pági-

na depois de certo número de caracteres. Ao usar muita HTML apenas para design, o mecanismo de busca poderia não perceber conteúdo importante na página ou até mesmo deixar de classificá-lo.

Em contraposição, a HTML estruturada facilita a leitura e indexação por um mecanismo de busca. Usar uma tag `<h1>` para indicar o tema mais importante da página (em vez de simplesmente tornar o texto grande e negrito) é uma estratégia inteligente: os mecanismos de busca dão mais peso aos conteúdos dentro dessa tag durante a indexação da página.

Para ver as sugestões do Google para construir sites amigáveis a mecanismos de busca, visite <http://support.google.com/webmasters/bin/answer.py?hl=en&answer=35769> (conteúdo em inglês).

■ COMPREENDENDO AS TAGS `<DIV>` E ``

A tag `<div>` e a tag `` são como recipientes vazios que são preenchidos com conteúdo. Um `div` é um bloco, o que significa que ele tem uma quebra de linha antes e depois dele, enquanto um `span` aparece inline, como parte de um parágrafo. Do contrário, as tags `div` e `span` não têm propriedades visuais inerentes, portanto você pode usar CSS para fazer com que elas tenham a aparência que você quiser. A tag `<div>` (significando *divisão*) indica qualquer bloco discreto de conteúdo, quase como um parágrafo ou um título. Porém, ela é mais frequentemente usada para agrupar quaisquer outros elementos e, portanto, você pode inserir um título, alguns parágrafos e uma lista com marcadores dentro de um único bloco `<div>`. A tag `<div>` é uma ótima maneira de subdividir uma página em áreas lógicas, como banner, rodapé, quadro lateral etc. Usando CSS, mais tarde você pode posicionar cada área para criar layouts de página sofisticados (um tema discutido na Parte III deste livro).

A tag `` é usada para elementos *inline*, ou seja, palavras ou frases que aparecem dentro de um parágrafo maior ou um título. Trate-a como outras tags HTML inline, como a tag `<a>` (para adicionar um link ao texto em um parágrafo) ou a tag `` (para enfatizar uma palavra em um parágrafo). Por exemplo, você poderia usar uma tag `` para indicar o nome de uma empresa e, então, usar CSS para destacar o nome usando uma fonte diferente, uma cor diferente etc. Eis um exemplo dessas tags em ação, complementado com uma prévia de alguns atributos — `id` e `class` — frequentemente usados para anexar estilos a partes de uma página.

```
<div id="footer">
  <p>Copyright 2006, <span class="bizName">CosmoFarmer.com</span></p>
  <p>Ligue para o atendimento ao cliente em 555-555-5501 para mais infor-
  mações</p>
</div>
```

Essa breve introdução não é a última vez que você verá essas tags. Elas são usadas com frequência em páginas web cheias de código CSS e, neste livro, você aprenderá a usá-las em combinação com CSS para obter controle criativo sobre suas páginas web.

■ TAGS ADICIONAIS NA HTML5

A tag `<div>` é bastante genérica — ela é simplesmente um elemento no nível de bloco usado para dividir uma página em seções. Um dos objetivos da HTML5 é fornecer outras tags mais semânticas, as quais web designers podem escolher. Tornar a HTML mais semântica significa simplesmente usar tags que descrevem com precisão o conteúdo que elas contêm. Como mencionado anteriormente nesta seção, você deve usar a tag `<h1>` (heading 1) ao posicionar um texto que descreve o conteúdo principal da página. Da mesma forma, a tag `<code>` diz claramente quais são as informações posicionadas dentro dela — código de programação.

A HTML5 inclui muitas tags diferentes, cujos nomes refletem o tipo de conteúdo que elas contêm e podem ser usados no lugar da tag `<div>`. A tag `<article>`, por exemplo, é usada para marcar uma seção de uma página que contém uma composição completa, independente. Em outras palavras, um “artigo”, como em um post de blog, um artigo de uma revista online ou simplesmente o texto principal da página. Da mesma forma, a tag `<header>` indica um título ou banner, a parte superior de uma página geralmente contém um logotipo, navegação por todo o site, título de página, tagline etc.

NOTA Para saber mais sobre as novas tags HTML, visite HTML5 Doctor (<http://html5doctor.com> — conteúdo em inglês) e www.w3schools.com/html/html5_intro.asp (conteúdo em inglês).

Muitas das novas tags HTML5 são projetadas para expandir a tag `<div>` genérica. Eis algumas outras tags HTML5 frequentemente usadas para estruturar o conteúdo em uma página:

- A tag **`<section>`** contém um agrupamento de conteúdo relacionado, como o capítulo de um livro. Por exemplo, você poderia dividir o conteúdo de uma página inicial em três seções: uma para a introdução ao site, uma para informações de contato e outra para notícias mais recentes.
- A tag **`<aside>`** armazena o conteúdo que está relacionado ao conteúdo em torno dele. Uma nota lateral em uma revista impressa, por exemplo.
- A tag **`<footer>`** contém informações que você normalmente posicionaria no rodapé de uma página, como um aviso de direitos autorais, informações legais, alguns links de navegação do site etc. Você não está limitado, porém, apenas a uma única `<footer>` por página; você pode colocar um rodapé dentro de uma `<article>`, por exemplo, para armazenar informações relacionadas a esse artigo, como notas de rodapé, referências ou citações.
- O elemento **`<nav>`** contém os principais links navegacionais.
- A tag **`<figure>`** é usada para uma figura ilustrativa. Você pode posicionar uma tag `` dentro dela, bem como uma nova tag HTML5 — a tag `<figcaption>`, que é usada para exibir uma legenda explicando a foto ou ilustração dentro da `<figure>`.

DICA Compreender qual tag HTML5 usar — se o texto for um `<article>` ou `<section>`? — pode ser complicado. Para obter um fluxograma prático que faz sentido para novos elementos de seccionamento da HTML5, baixe o PDF de HTML5 Doctor em <http://html5doctor.com/downloads/h5d-sectioning-flowchart.pdf> (conteúdo em inglês).

Existem outros elementos HTML5, e muitos deles simplesmente fornecem uma alternativa mais descritiva para a tag `<div>`. Este livro usa tanto a tag `<div>` como as novas tags HTML5 para ajudar a organizar o conteúdo da página web. A desvantagem da HTML5 é que o Internet Explorer 8 e versões anteriores não reconhecem as novas tags sem um pouco de ajuda (veja o quadro a seguir).

Além disso, exceto pelo fato de fazer você se sentir atualizado em relação às últimas tendências de web design, não há realmente nenhum benefício tangível para usar algumas dessas tags HTML5. Por exemplo, simplesmente usar a tag `<article>` para armazenar a matéria principal em uma página web não faz com que o Google goste mais de você. Você pode continuar usando a tag `<div>` à vontade e evitar os elementos de seccionamento HTML5, se preferir.

OFICINA DE SOLUÇÕES ALTERNATIVAS

Fazendo o IE 8 Entender HTML5

A HTML5 fornece várias tags HTML novas para brincar. Desde tags que descrevem claramente o tipo de conteúdo que armazenam, como a tag `<nav>`, até aquelas que fornecem funcionalidades adicionais, como a tag `<video>` para incorporar vídeos e a tag `<audio>` para incorporar som e música. Depois de aprender mais sobre a HTML5, você provavelmente começará a aplicar essas novas tags às suas páginas web.

Infelizmente, o Internet Explorer 8 e versões anteriores não reconhecem essas novas tags e não responderão à CSS que você aplicar a elas. Isso mesmo, se você estiver usando HTML5 e visualizando páginas web no IE 8, este livro não será útil para você. Bem, não exatamente. Existe uma maneira de fazer essas versões antigas do IE funcionar, assim elas entenderão toda a CSS que se aplica a tags HTML5.

Simplesmente posicione o código a seguir antes da tag `</head>` de fechamento no topo do arquivo HTML:

```
<!--[if lt IE 9]>
<script src="//html5shiv.googlecode.com/
```

```
svn/trunk/html5.js"></script>
<![endif]-->
```

Esse pequeno trecho ardiloso de código usa o que é chamado “comentário condicional do Internet Explorer” (Internet Explorer Conditional Comment, IECC) para incorporar parte do código JavaScript que só é visível para as versões do Internet Explorer anteriores ao IE 9. Em outras palavras, somente os IE 6, 7 e 8 respondem a esse código, e todos os outros navegadores (incluindo as versões mais recentes do IE) simplesmente o ignoram. Esse código faz com que as versões anteriores do IE carreguem um pequeno programa JavaScript que força o navegador a reconhecer as tags HTML5 e insere a CSS que se aplica a essas tags.

O código só afeta a maneira como o navegador exibe e imprime tags HTML5; ele não faz o navegador “entender” uma tag HTML5 que realmente faz alguma coisa. Por exemplo, o IE 8 e as versões anteriores não entendem a tag `<video>` e não podem reproduzir vídeo em HTML5 (mesmo com o código JavaScript adicional).

Tenha o Layout em Mente

Embora a tag `<h1>` seja utilizada para identificar o tema principal da página, e a tag `<p>`, para adicionar um parágrafo de texto, você irá querer organizar o conteúdo de uma página em um layout agradável. Quando você aprender a usar a CSS para formatar uma página na Parte Três deste livro, é importante ter o design em mente ao escrever a HTML da página.

Você pode pensar no layout de páginas web como um arranjo habilidoso de caixas (veja um exemplo na Figura 1-3). Afinal de contas, um design de duas colunas consistindo em duas colunas verticais de texto é, na verdade, duas únicas caixas retangulares posicionadas lado a lado. Um título consistindo em um logotipo, slogan, caixa de pesquisa e navegação pelo site é, na verdade, uma única grande caixa retangular posicionada ao longo da parte superior da janela do navegador. Em outras palavras, se você imaginar os agrupamentos e o layout do conteúdo em uma página, você verá caixas posicionadas umas sobre outras, umas ao lado de outras e umas abaixo de outras.



FIGURA 1-3

Esse layout básico de duas colunas inclui um banner (acima), uma coluna do conteúdo principal (no meio, à esquerda), um quadro lateral (no meio, à direita) e um rodapé (abaixo). Essas são as principais caixas estruturais que compõem o layout da página.

Na HTML, você cria essas caixas, ou unidades estruturais, usando a tag `<div>`. Basta colocar as tags HTML que compõem a área do banner, por exemplo, dentro de uma div, o valor de uma coluna da HTML em outro etc. Se for experiente em HTML5, você poderá criar o design mostrado na Figura 1-3, com uma tag `<header>` para o banner na parte superior, uma tag `<article>` para o texto principal, uma tag `<aside>` ou `<section>` para o quadro lateral e uma tag `<footer>` para o rodapé da página. Em outras palavras, se planeja posicionar um grupo de tags HTML juntas em algum lugar em uma página, então você precisará colocar essas tags dentro de um elemento de seccionamento como uma `<div>`, `<article>`, `<section>` ou `<aside>`.

HTML para Esquecer

A CSS permite escrever código HTML mais simples por uma grande razão: você para de usar uma grande quantidade de tags e atributos que apenas melhoram a aparência de uma página. A tag `` é o exemplo mais gritante. Seu único objetivo é adicionar cor, tamanho e fonte ao texto. Ela não torna a estrutura da página mais compreensível.

Eis uma lista de tags e atributos que você pode facilmente substituir pela CSS:

- **Livre-se de `` para controlar a exibição do texto.** A CSS faz um trabalho muito melhor com texto. (Consulte o Capítulo 6 para ver as técnicas de formatação de texto.)
- **Não use as tags `` e `<i>` para enfatizar o texto.** Se você quer enfatizar o texto, use a tag ``, que os navegadores normalmente exibem como negrito. Para obter um ponto um pouco menos enfático, use a tag ``, que os navegadores exibem como itálico.

Embora a HTML4 tenha tentado desativar as tags `` e `<i>`, a HTML5 as trouxe de volta. Na HTML5, a função da tag `` é simplesmente formatar o texto em negrito, sem adicionar nenhum significado a esse texto (ou seja, você apenas quer que o texto seja formatado em negrito, mas não quer que as pessoas tratem o texto como se você estivesse gritando com elas). Do mesmo modo, a tag `<i>` é utilizada para formatar o texto em itálico, mas não para enfatizar o significado.

NOTA Para formatar em itálico o título de uma publicação, a tag `<cite>` mata dois coelhos com uma cajadada só. Ela formata o título em itálico e o marca como uma obra citada para o benefício dos mecanismos de busca. Essa é uma tag para se guardar e continuar usando.

- **Não use a tag `<table>` para layout de página.** Use-a apenas para exibir informações tabulares, como planilhas, tabelas e gráficos. Como veremos na Parte Três deste livro, todo o layout pode ser construído com CSS muito mais rapidamente e com menos código do que a miscelânea das tags de tabela.
- **Evite os atributos desajeitados da tag `<body>`** que só aprimoram a apresentação do conteúdo: `background`, `bgcolor`, `text`, `link`, `alink` e `vlink` configuram cores e imagens para página, texto e links. A CSS faz melhor esse trabalho (veja no Capítulo 7 e no Capítulo 8 os equivalentes CSS desses atributos).
- **Não abuse da tag `
`.** Se você está acostumado a usar a tag `
` (`
` na XHTML) para inserir uma quebra de linha sem criar um parágrafo, então você tem um desafio. (Navegadores inserem automaticamente e, às vezes, de uma maneira irritante, um pequeno espaço entre os parágrafos, inclusive entre os títulos e as tags `<p>`. No passado, designers utilizavam soluções alternativas elaboradas para evitar o espaçamento entre parágrafos que eles não queriam, como substituir uma única tag `<p>` por várias quebras de linha e usar uma tag `` para fazer com que a primeira linha do parágrafo parecesse um título.) Utilizando os controles de margem da CSS, você pode facilmente configurar a quantidade de espaço que quer ver entre os parágrafos, títulos e outros elementos de nível de bloco.

NOTA No próximo capítulo você aprenderá uma técnica chamada de “CSS Reset”, que elimina os espaços que os navegadores normalmente inserem entre os parágrafos e outras tags (veja a página 115).

Como regra geral, adicionar atributos a tags que configuram cores, bordas, imagens de fundo ou alinhamento — incluindo atributos que permitem formatar cores, fundos e bordas de uma tabela — é uma HTML da velha escola. Assim como o é usar propriedades de alinhamento para posicionar imagens e centralizar texto em parágrafos e células de tabela. Em vez disso, use CSS para controlar posicionamento do texto (página 169), bordas (página 202), fundos (página 206) e alinhamento de imagem (página 239).

ATUALIZANDO

Valide Suas Páginas Web

A HTML segue certas regras: por exemplo, a tag `<html>` envolve outras tags em uma página, e a tag `<title>` precisa aparecer dentro da tag `<head>`. A XHTML fornece um conjunto ainda mais estrito de regras que devem ser seguidas, enquanto a HTML5 é um pouco mais complacente. É fácil se esquecer dessas regras ou simplesmente cometer um erro de digitação. HTML incorreta (ou inválida, como os geeks diriam) resulta em problemas como uma aparência diferente da mesma página em diferentes navegadores web. Mais importante, mesmo uma CSS válida talvez não funcione como esperado com uma HTML inválida. Felizmente, existem ferramentas para verificar se a HTML está escrita corretamente nas páginas web.

A maneira mais fácil de verificar — ou seja, validar — suas páginas é no site do W3C, em <http://validator.w3.org> (conteúdo em inglês; veja a Figura 1-4). Obtenha a extensão Web Developer para Firefox (<http://chrispederick.com/work/web-developer> — conteúdo em inglês); ela fornece uma maneira rápida de testar uma página no validador do W3C.

O W3C, ou World Wide Web Consortium, é a organização responsável por determinar os padrões para muitas das tecnologias e linguagens da web, incluindo HTML e CSS.

Se o validador do W3C encontrar erros na página, ele informa quais são esses erros. Se você utiliza o Firefox, baixe uma extensão que permite validar uma página web diretamente nesse navegador, sem ter de visitar o site do W3C. Ela pode até mesmo tentar corrigir quaisquer problemas que encontrar. Obtenha a extensão aqui: <http://users.skynet.be/mgueury/mozilla/> (conteúdo em inglês). Usuários do Chrome podem baixar a extensão HTML Tidy para o Chrome em <http://bit.ly/SCONRY> (conteúdo em inglês). Uma ferramenta semelhante também está disponível para o navegador Safari: www.zappatic.net/safaritidy (conteúdo em inglês).

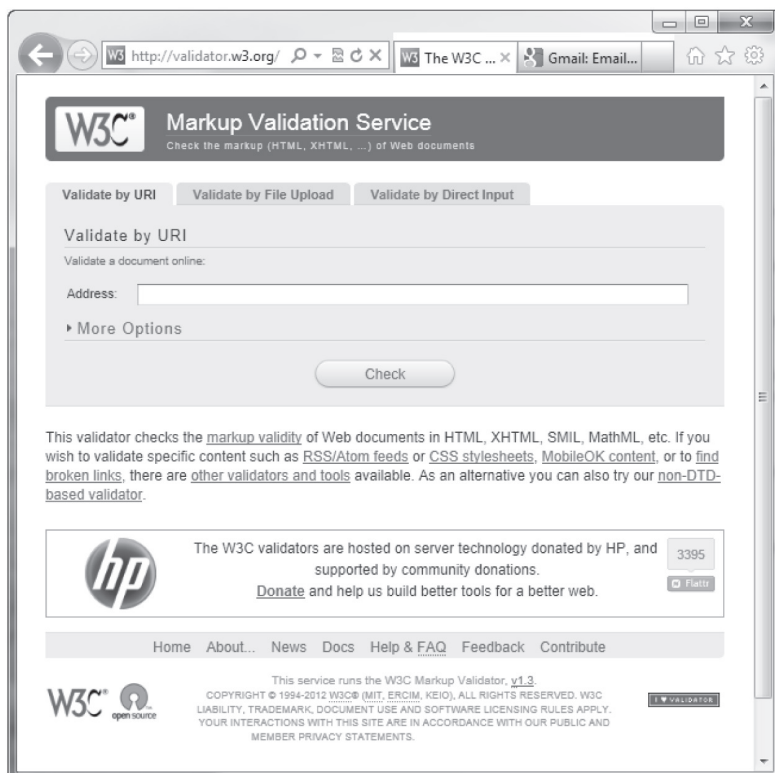


FIGURA 1-4

O validador W3C HTML, localizado em <http://validator.w3.org> (conteúdo em inglês), permite assegurar rapidamente se a HTML em uma página é sólida. Aponte o validador para uma página existente na web, faça o upload de um arquivo HTML do seu computador ou apenas cole o código HTML de uma página web na caixa de formulário e então clique no botão Check.

Dicas para Orientar Seu Caminho

Sempre é bom ter um mapa para entender o terreno. Se você ainda não sabe como usar HTML para criar páginas web bem estruturadas, eis algumas dicas para você começar:

- Use títulos para indicar a importância relativa do texto. Mais uma vez, pense na estrutura. Quando dois títulos são igualmente importantes no tema da página, use o mesmo título de nível em ambos. Se um for menos importante ou for um subtema do outro, use então o próximo nível de título. Por exemplo, depois de um `<h2>`, use uma tag `<h3>` (veja a Figura 1-5). Em geral, uma boa ideia é usar títulos na ordem sem tentar pular números de títulos. Por exemplo, não coloque uma tag `<h5>` depois de uma tag `<h2>`.

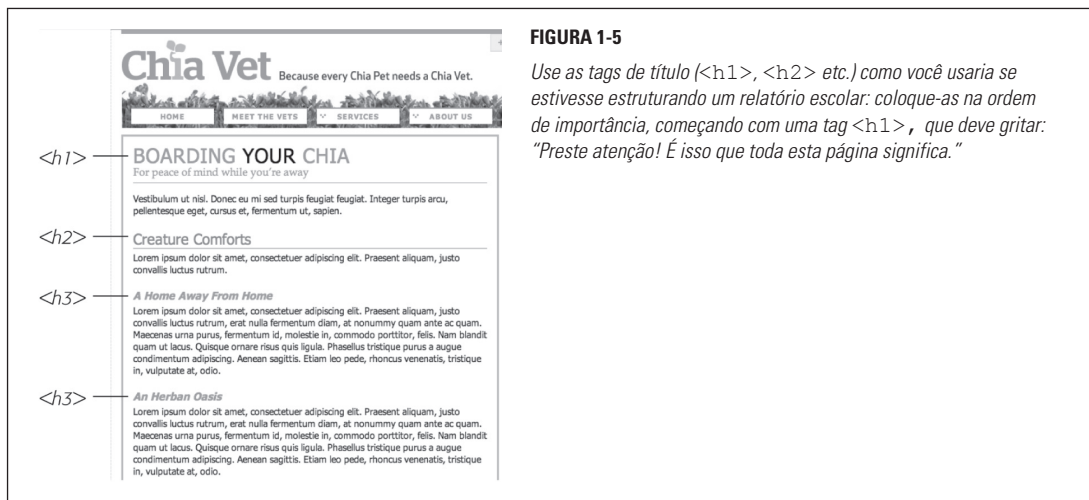


FIGURA 1-5

Use as tags de título (`<h1>`, `<h2>` etc.) como você usaria se estivesse estruturando um relatório escolar: coloque-as na ordem de importância, começando com uma tag `<h1>`, que deve gritar: “Preste atenção! É isso que toda esta página significa.”

- Use a tag `<p>` para parágrafos de texto.
- Use listas não ordenadas (``) quando há uma lista de vários itens relacionados, como links de navegação, títulos ou um conjunto de dicas como essas.
- Use listas numeradas (``) para indicar os passos em um processo ou definir a ordem de um conjunto de itens. Os tutoriais neste livro são um bom exemplo, assim como uma lista de rankings, como “Os dez sites mais populares entre monges”.
- Para criar um glossário dos termos e suas definições ou descrições, use a tag `<dl>` (lista de definição) em conjunto com as tags `<dt>` (termo da definição) e `<dd>` (descrição da definição). (Para um exemplo de como usar essa combinação, visite www.w3schools.com/tags/tryit.asp?filename=tryhtml_list_definition — conteúdo em inglês.)
- Se você quiser incluir uma citação, como um fragmento de texto de outro site, uma resenha de filme ou algum conselho sábio de seu avô, experimente a tag `<blockquote>` para longas passagens ou a tag `<q>` para inserir uma pequena citação dentro de um parágrafo mais longo, desta maneira:


```
<p>Dizem que Mark Twain escreveu <q>O inverno mais frio que já passei foi um verão em São Francisco</q>. Infelizmente, ele nunca escreveu essa famosa frase.</p>
```
- Tire proveito das tags obscuras, como a tag `<cite>` para fazer referência ao título de um livro, artigo de jornal ou site, e a tag `<address>` para identificar e fornecer informações de contato para o autor de uma página (ótima para um aviso de direitos autorais).

- Como explicado na íntegra na página 26, mantenha-se longe de qualquer tag ou atributo cuja única função seja alterar a aparência de um texto ou imagem. A CSS, como veremos, pode fazer tudo.
- Quando simplesmente não há uma tag HTML apropriada, mas você quer identificar um elemento em uma página ou um grupo de elementos em uma página para que possa aplicar uma aparência distinta, utilize as tags `<div>` e `` (veja a página 21). Veremos mais recomendações sobre como usá-las nos capítulos posteriores.
- Não use as tags `<div>` exageradamente. Alguns web designers acham que só precisam de tags `<div>`, ignorando tags que poderiam ser mais apropriadas. Por exemplo, para criar uma barra de navegação, adicionam a tag `<div>` a uma página e a preenchem com um punhado de links. Uma abordagem melhor seria utilizar uma lista com marcadores (tag ``). Afinal de contas, uma barra de navegação é apenas uma lista de links. Como discutido na página 23, a HTML5 fornece várias novas tags que podem substituir a tag `<div>`, como as tags `<article>`, `<section>` e `<footer>`. Para uma barra de navegação, use a tag `<nav>` HTML5.
- Lembre-se de fechar as tags. A tag `<p>` de abertura precisa de um parceiro (a tag `</p>` de fechamento), assim como todas as outras tags, exceto as poucas que se fecham automaticamente, como `
` e `` (`
` e `` na XHTML).
- Valide suas páginas com o validador do W3C (veja a Figura 1-4 e o quadro na página 27). HTML mal escrita ou cheia de erros de digitação resulta em muitos erros estranhos de navegador.

■ A Importância do Doctype

A HTML segue certas regras — essas regras estão contidas em um arquivo Document Type Definition, também conhecido como DTD. Um DTD é um arquivo de texto que explica quais tags, atributos e valores são válidos para um determinado tipo de HTML. E para cada versão da HTML há um DTD correspondente. Agora você poderia perguntar: “Mas o que tudo isso tem a ver com CSS?”.

Tudo — se você quer que suas páginas web apareçam de uma maneira correta e consistente em navegadores web. Você informa a um navegador web a versão da HTML ou XHTML que você está usando, incluindo o que é chamado de *declaração doctype*, no início de uma página web. Essa declaração doctype é a primeira linha no arquivo HTML e define qual versão da HTML você utiliza (como HTML5 ou HTML 4.01 Transitional). Ao digitar erroneamente a declaração doctype ou não incluí-la, você faz com que a maioria dos navegadores entre em um estado alterado chamado *modo de compatibilidade* (*quirks mode*).

O modo de compatibilidade é a tentativa dos fabricantes de navegadores para que os softwares se comportem como os navegadores se comportavam por volta de 1999 (na época do Netscape 4 e Internet Explorer 5). Se um navegador moderno encontrar uma página que não tenha o doctype correto, ele pensa: “Puxa, essa página deve ter sido escrita há muito tempo, em um editor HTML muito, muito antigo. Vou fingir que sou um navegador muito antigo e exibir a página exatamente

como um daqueles navegadores antigos e falhos a exibiriam”. É por isso que, sem um doctype correto, suas páginas web cuidadosamente estilizadas com CSS talvez não tenham a aparência que elas deveriam ter, de acordo com os padrões atuais. Se você vir involuntariamente sua página web no modo de compatibilidade ao verificá-la em um navegador, você pode acabar tentando corrigir problemas de exibição que estão relacionados a um doctype incorreto, e não ao uso incorreto de HTML ou CSS.

NOTA Para informações adicionais (leia-se: técnicas) sobre o modo de compatibilidade, visite www.quirksmode.org/css/quirksmode.html (conteúdo em inglês) e https://developer.mozilla.org/en/Mozilla%2527s_Quirks_Mode (conteúdo em inglês).

Felizmente, é fácil obter o doctype correto. Tudo o que você precisa saber é qual versão da HTML você está usando. Se estiver usando HTML5, tudo fica mais fácil. O doctype é simplesmente:

```
<!doctype html>
```

Coloque isso no topo do arquivo HTML e você estará pronto para prosseguir. Se você ainda estiver usando versões mais antigas da HTML ou XHTML, como HTML 4.01 Transitional e XHTML 1.0 Transitional, então o doctype é muito mais complicado.

Se você estiver usando HTML 4.01 Transitional, por exemplo, digite a seguinte declaração doctype no início de cada página que você cria:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

A declaração doctype para XHTML 1.0 Transitional é semelhante. Também é necessário adicionar um pouco de código à tag <html> de abertura, que é usada para identificar o tipo de XML do arquivo — neste caso, é XHTML —, assim:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

Se toda essa discussão está provocando dor de cabeça e seus olhos estão se fechando lentamente, mantenha a vida simples usando o doctype HTML5. Ele é curto, fácil de lembrar e funciona em todos os navegadores. Use esse doctype mesmo se você não tocar em nenhuma das novas tags HTML5.

NOTA A maioria dos editores visuais de página web, como o Dreamweaver e o Expression Web, adiciona automaticamente uma declaração doctype sempre que você cria uma nova página web, e muitos editores de texto cientes de HTML têm atalhos para adicionar doctypes.

PERGUNTAS FEITAS COM FREQUÊNCIA

Devo me Preocupar com o IE 6, 7 ou 8?

Sempre ouço que o Internet Explorer 6 está morto e não mais precisamos nos preocupar com ele. Isso é verdade? E sobre as outras versões do IE?

Se você for web designer, provavelmente já tem a versão mais recente do Internet Explorer, Firefox, Safari, Chrome ou Opera no computador. As versões anteriores deste livro discutiram bastante o Internet Explorer 6 e até mesmo forneceram um capítulo sobre como lidar com bugs do IE 6. Isso tudo acabou, uma vez que o IE 6 está desaparecendo rapidamente.

Se você estiver construindo sites com um público norte-americano em mente, a partir de agosto de 2012, menos de 0,6% dos internautas nos EUA usa esse navegador desatualizado; no Reino Unido, a taxa é 1,1%. O número exato varia dependendo da fonte da informação (por exemplo, segundo <http://gs.statcounter.com> — conteúdo em inglês — a taxa mundial de uso do IE 6 é de 0,57%, e www.ie6countdown.com — conteúdo em inglês — informa que ela era de 6% em agosto de 2012). Na China, o IE 6 ainda é amplamente utilizado.

Mas mesmo estatísticas que incluem a região geográfica do público do seu site não refletem verdadeiramente o que os visitantes do seu site usam. Se você construir um site voltado para web designers com experiência em tecnologia, é provável que o IE 6 não seja usado para visualizar seu site há muito tempo. Entretanto, se o seu site está direcionado a pessoas na China, talvez seja necessário lidar com o IE 6. A melhor maneira de descobrir quanto tráfego vem via o IE 6 é analisar os arquivos de log do seu servidor

web ou se inscrever no Google Analytics (www.google.com/analytics) para que você possa monitorar os navegadores dos visitantes (entre várias outras coisas).

Para a grande maioria dos projetos de web design, o IE 6 efetivamente desapareceu. Além disso, o IE 7 também está desaparecendo rapidamente, com apenas 1,16% da fatia mundial de mercado em agosto de 2012 (http://gs.statcounter.com/#browser_version-wwwmonthly-201.108-201.208 — conteúdo em inglês).

Mas você ainda precisa lidar com o IE 8. O Windows XP continua a ser um sistema operacional popular, e a Microsoft optou por não incorporar o Internet Explorer 9 ou versão posterior nesse sistema operacional. Assim, como o XP ainda está disponível, o Internet Explorer 8 também está disponível. (Para se manter atualizado em relação aos avanços no desenvolvimento web, usuários do XP podem mudar para o Chrome ou Firefox.) De acordo com a StatCounter, em agosto de 2012, o IE 8 era responsável por 13,65% de todos os navegadores web (http://gs.statcounter.com/#browser_version-wwwmonthly-201108-201208 — conteúdo em inglês). De acordo com a NetMarketShare, o IE 8 é responsável por colossais 26% dos navegadores web (<http://marketshare.hitslink.com/browser-market-share.aspx?qprid=2&qpcustom=0> — conteúdo em inglês). Como o IE 8 não entende tags HTML5, você não pode formatá-las diretamente com CSS; você tem que recorrer a uma solução JavaScript alternativa, como descrito no quadro na página 24.

■ Certificando-se de que o Internet Explorer Está Atualizado

Graças ao recurso de atualização automática da Microsoft, PCs Windows agora atualizam o Internet Explorer para a versão mais recente. Usuários do Windows 7 e 8 tiveram a versão mais recente do Internet Explorer instalada — o IE 10 no momento em que este texto era escrito. O IE 10 suporta muitas das novas propriedades HTML5 e CSS3 empolgantes e poderosas. Como você vai aprender neste livro, a CSS3 fornece várias possibilidades estimulantes de design, como sombras projetadas (página 210), fundos em degradê (página 259) e cantos arredondados (página 207), para citar algumas.

Infelizmente, nem todos os usuários do Windows serão capazes de tirar proveito desses avanços no web design. Como explicado no quadro da página 32, o sistema operacional Windows XP, amplamente utilizado, só pode executar o Internet Explorer 8 ou versões anteriores. Na verdade, o IE 8 ainda é a versão mais comum do Internet Explorer usada na web. Este livro apontará as propriedades CSS que não funcionam no IE 8, bem como possíveis soluções alternativas.

Como o IE 8 ainda é muito popular, você precisa ter algumas coisas em mente. O IE 8 é como uma espécie de camaleão: ele pode assumir a aparência de uma versão diferente. Se você não tiver cuidado, talvez ele não exiba as páginas web da maneira como você quer. Por exemplo, e mais importante, você deve incluir um doctype adequado. Como mencionado na seção anterior, sem doctype, navegadores alternam para o modo de compatibilidade. Bem, quando o IE 8 entra no modo de compatibilidade, ele tenta reproduzir a aparência do IE 5 (!?).

Mas espere — há mais! O IE 8 também pode fingir ser o IE 7. Quando alguém visualizando seu site no IE 8 clica em um botão “exibir compatibilidade”, o IE 8 entra no modo do IE 7, exibindo as páginas sem todas as funcionalidades CSS 2.1 do IE 8. A mesma coisa acontece se a Microsoft colocar o site na Lista de Exibição de Compatibilidade — uma lista de sites que a Microsoft determinou que têm uma aparência melhor no IE 7 do que no IE 8. Se você estiver criando um site usando as diretrizes presentes neste livro, você não vai querer que o IE 8 funcione como o IE 7... *nunca*.

Felizmente, há uma maneira de instruir o IE 8 a parar com todo esse absurdo e simplesmente ser o IE 8. Adicionar uma única tag META a uma página web instrui o IE 8 a ignorar a Lista de Exibição de Compatibilidade e sempre mostrar a página usando o modo mais compatível com os padrões:

```
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
```

Insira essa instrução na seção <head> da página (abaixo da tag <title> é um bom lugar). Essa tag também funciona para as versões atuais do IE: a parte “IE=edge” da tag instruirá as versões do Internet Explorer além do IE 8 a também exibir páginas web no modo padrão. Infelizmente, você deve fazer isso em todas as páginas de seu site.

Agora que a HTML está na direção certa, é hora de saltar para a parte divertida (e a razão pela qual você comprou este livro): Folhas de Estilos em Cascata.

ATUALIZANDO

Teste para Múltiplos Navegadores

Há vários navegadores disponíveis. Se usar o Windows, você receberá automaticamente o Internet Explorer e poderá instalar navegadores adicionais, como Firefox, Safari, Opera ou Chrome do Google. Em um Mac, fique com o navegador Safari já instalado, ou rode o Firefox ou o Chrome. Embora os navegadores mais recentes sejam comparáveis ao exibir páginas web baseadas em CSS, não é possível dizer o mesmo para o IE 8, que ainda está em uso generalizado.

Para realmente certificar-se de que seus sites funcionam para um público mais amplo, você precisa testar seus designs na maior variedade possível de navegadores. Eis algumas técnicas:

Windows. Normalmente, computadores Windows só podem executar uma versão do IE — você não pode testar no IE 6, IE 7 e IE 8 no mesmo computador. Bem, normalmente você não pode, mas *você* pode, se executar máquinas virtuais no computador Windows para criar versões completamente diferentes do Internet Explorer em que é possível testar. É necessária uma quantidade razoável de trabalho, mas fornece a maneira mais precisa de testar uma página em múltiplas versões do IE. Visite www.howtogeek.com/102261/how-to-run-internet-explorer-7-8-and-9-at-the-same-time-using-virtual-machines (conteúdo em inglês) para saber mais sobre como fazer isso.

Você também deve instalar os outros principais navegadores no computador: Firefox, Opera e Chrome. A Apple desativou o Safari para Windows, assim, você pode contar com o Chrome (que tem um mecanismo de renderização subjacente similar) ou utilizar um dos serviços listados posteriormente neste quadro (como o Adobe BrowserLab) para testar seus designs no Safari.

Mac. O teste é um pouco mais complicado para usuários do Mac. Você tem que testar no Internet Explorer — ele ainda é um dos navegadores mais usados no mundo, e a falta da maioria das propriedades CSS3 no IE 8 significa que seu design meticulosamente trabalhado pode parecer fantástico para você, mas se desintegra no IE 8. Você tem algumas opções: primeiro, você pode comprar (ou pedir emprestado) um computador com Windows; segundo, se você tem um Intel Mac, instale o Windows usando o software Boot Camp da Apple (www.apple.com/br/support/bootcamp/); uma terceira opção é instalar um software de virtualização como o VMware Fusion ou Parallels Desktop no Mac. Esse software permite executar uma máquina virtual Windows lado a lado com o Mac OS. Você pode saltar entre o Mac e o Windows para testar em vários navegadores nos dois sistemas opera-

cionais. É a técnica de teste mais eficiente para usuários do Mac. Tanto o Boot Camp quanto o software de virtualização exigem uma cópia do sistema operacional Windows.

Todo o mundo. Outra opção que funciona para o Windows e o Mac e não requer que se instale um software extra é um dos muitos serviços de teste em múltiplos navegadores que permitem ver qual a aparência de suas páginas em vários sistemas operacionais e navegadores diferentes.

- **CrossBrowserTesting.com** (<http://CrossBrowserTesting.com> — conteúdo em inglês) custa mais de 60 reais por mês (ai!) por 150 minutos de uso e oferece um benefício adicional — testes interativos. Você vê a página funcionando remotamente em um PC sob seu controle — você pode testar recursos que uma captura de tela não identifica, como reprodução de filme, animação Flash e interações JavaScript.
- **BrowserStack** (www.browserstack.com — conteúdo em inglês) é outro serviço que permite testar as páginas de forma interativa via navegador web. Por 40 e poucos reais mensais, é como alugar várias máquinas Windows e Mac com múltiplas versões do IE, Chrome, Firefox e Safari instaladas.
- **Browsershots** (www.browsershots.org — conteúdo em inglês) é uma alternativa gratuita que fornece capturas de tela para uma ampla variedade de navegadores no Windows e Linux.
- Mas você também pode ver gratuitamente qual a aparência de uma página no IE 9, 8, 7, 6 e mesmo no 5.5 com o **NetRenderer** (<http://ipinfo.info/netrenderer> — conteúdo em inglês). As páginas que você quer testar precisam estar disponíveis em um site acessível ao público. Visite <http://ipinfo.info/netrenderer>, digite a URL da página e rapidamente você verá uma captura de tela. Infelizmente, ele não cria uma imagem de uma página web completa — apenas da parte superior normalmente vista em uma tela. Por outro lado, ele é gratuito.
- Por fim, use a ferramenta **BrowserLab** da Adobe (<https://browserlab.adobe.com> — conteúdo em inglês) para obter capturas de tela de uma página web em todos os principais navegadores. Por exemplo, se você usa o Dreamweaver para criar sites, abra o BrowserLab de dentro do programa e pré-visualize a página em que você está trabalhando atualmente.