



I A LINGUAGEM DA WEB

Capítulo 1: [Introdução a HTML5](#)

Capítulo 2: [Entendendo tags em HTML5](#)

Capítulo 3: [Tags de texto e um pouco de CSS3](#)

Capítulo 4: [Trabalhando com valores de cor](#)



1

INTRODUÇÃO A HTML5

ESTE CAPÍTULO traz um panorama geral sobre o que muda, o que continua igual e o que foi removido da linguagem HTML para criar a linguagem HTML5. Uma das tarefas mais importantes neste momento é descobrir os navegadores que funcionam com HTML5, os que estão em desenvolvimento e prometem compatibilidade e como cada um deles já começou a implementar a compatibilidade com HTML5. Além disso, você vai aprender

sobre alguns dos novos navegadores especificamente desenvolvidos para dispositivos móveis, para que seja possível testar páginas em HTML5 também nesses dispositivos. Para começar, faça o download de todos os navegadores (abordados neste capítulo) para que possa aprender o que os usuários verão quando se depararem com a página Web em HTML5 que você criou.

CRIANDO COM TAGS: UM PANORAMA

A maior parte do conteúdo na Internet é criada com o HTML (Linguagem de Marcação de Hipertexto, do inglês HyperText Markup Language). Você pode se surpreender quando descobrir que diversos aplicativos que utiliza diariamente – seu editor de texto, por exemplo – também foram criadas com linguagens de marcação. Porém, assim como em todas as linguagens de computador, no HTML apenas o conteúdo é visível; a linguagem utilizada permanece oculta. A linguagem funciona como a estrutura de um edifício – sabemos que está lá, sob todas as camadas de argamassa e tinta, mas não podemos vê-la. O que eu faço por meio deste livro é tornar a linguagem HTML claramente visível e mostrar como utilizá-la para construir suas próprias estruturas.

Se você já conhece as versões anteriores de HTML e XHTML, poderá transferir a maior parte dos seus conhecimentos para HTML5. E se estiver começando agora, vai descobrir que trabalhar com HTML5 é bem fácil. Essencialmente, tudo o que você precisa fazer é posicionar seu conteúdo entre uma tag de abertura e uma tag de fechamento, chamadas de container, e esse container irá aplicar um estilo ao seu texto ou exibir seus gráficos e arquivos de mídia em uma página Web. A Figura 1-1 ilustra os containers:

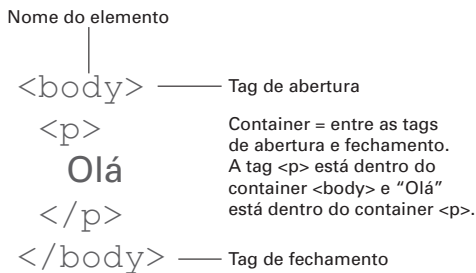


Figura 1-1: Containers em HTML5.

Por exemplo, a linha a seguir

```
<h1>Este texto é grande.</h1>
```

manda o interpretador no seu navegador criar um texto grande, como o que aparece abaixo:

Este texto é grande.

O texto contido entre os sinais de menor e maior `< >` é o código. Neste caso, `h1` é o código para texto grande. Os sinais de menor e maior dizem onde começar o container (`<h1>`) e onde terminar o container (`</h1>`). Tudo o que está dentro do container é configurado no tamanho e estilo da tag, que pode estar incorporado à própria tag ou criado usando CSS3.

Já que estamos começando, você pode se divertir criando e visualizando um pouco de HTML5 com pequenos exemplos. Tudo que você precisa fazer é digitar qualquer um dos códigos neste capítulo em um editor de texto como o Bloco de Notas, se estiver usando Windows, ou TextEdit, se estiver em um Mac. Salve o arquivo com a extensão `.html` e, em seguida, abra o arquivo em um navegador HTML5. Para abrir uma página Web, simplesmente inicie seu navegador, e então, a partir da barra de menu, selecione Arquivo, Abrir (ou Abrir arquivo) e

localize o nome do arquivo. (Você pode simplesmente clicar duas vezes no ícone do arquivo na área de trabalho para abrir a maioria dos arquivos HTML.)

INCORPORANDO OS NOVOS ELEMENTOS DE HTML5

Uma tag é composta por elementos e atributos. A tag é identificada por seu elemento, por exemplo, `<h1>` – em que `<h1>` é o elemento. Quando falamos em tag, em geral nos referimos a seu elemento, mas, na verdade, uma tag possui seus elementos e atributos. Os atributos são as diferentes características ou propriedades de um elemento que você pode programar para mudar as funcionalidades do conteúdo no container da tag. Por hora, vou abordar somente o elemento, então os termos *tag* e *elemento* serão utilizados alternadamente.

Para dar uma ideia dos novos elementos de HTML5, a Tabela 1.1 mostra todos os elementos novos, além de uma breve descrição de cada um. Mais adiante no livro, apresento vários exemplos e explico como utilizá-los.

Tabela 1.1: Novos elementos em HTML5

Novo Elemento	Descrição
<code><article></code>	Artigo (com começo, meio e fim) dentro do documento
<code><aside></code>	Conteúdo tangencialmente relacionado ao conteúdo do artigo
<code><audio></code>	Container de conteúdo de áudio
<code><canvas></code>	Container de desenvolvimento gráfico
<code><command></code>	Um comando que o usuário pode evocar
<code><datalist></code>	Gerador de lista quando utilizado com o elemento <code><input></code> e seu novo atributo de lista
<code><details></code>	Revela os detalhes de um elemento
<code><embed></code>	Plug-in ou conteúdo externo interativo
<code><figcaption></code>	Tag de legenda para o elemento de figura
<code><figure></code>	Inclui um grupo de conteúdo de mídia e suas legendas
<code><footer></code>	Container para o rodapé de uma seção ou página
<code><header></code>	Container para o cabeçalho de uma seção ou página
<code><hgroup></code>	Cabeçalho de uma seção com múltiplos elementos h1 a h6 dentro de um documento
<code><keygen></code>	Representação do controle do gerador de chaves pares
<code><mark></code>	Uma linha de texto em um documento, marcada ou realçada para referência em outro documento

(continua)

Tabela 1.1

Novo Elemento	Descrição
<code><meter></code>	Container para uma gama de valores conhecida (por exemplo, uso de disco)
<code><nav></code>	Representação de uma seção de um documento que será usada para navegação
<code><output></code>	Define o progresso de uma tarefa de qualquer tipo
<code><progress></code>	Representação do progresso realizado em uma tarefa (como a porcentagem já completada de uma operação de download)
<code><rp></code>	Indicador com anotações em Ruby (uma linguagem de programação) para definir o que deve ser mostrado a navegadores sem suporte ao elemento <code><ruby></code>
<code><rt></code>	Marca o componente de texto ruby em uma anotação em ruby
<code><ruby></code>	Elemento para períodos com anotações em ruby
<code><section></code>	Identificador de tema para agrupamento de conteúdo
<code><source></code>	Container para especificação múltipla de recursos de mídia
<code><summary></code>	Informações sobre um elemento <code><details></code>
<code><time></code>	Container para data/hora
<code><video></code>	Elemento de link para arquivo de vídeo
<code><wbr></code>	Representação de uma oportunidade de quebra de linha para guiar a hifenação de palavras ou linhas de texto longas

Alguns dos elementos novos, como `<video>` e `<audio>`, adicionam recursos multimídia ao HTML e representam uma importante inovação na capacidade da linguagem. Outros, como `<ruby>`, são bastante especializados e, a não ser você que precise de certos caracteres do Leste Asiático, é pouco provável que o utilize.

Uma característica de boa parte das novas tags é a de que elas trabalham em conjunto com CSS3 ou JavaScript. Porém, a maioria dos novos elementos ainda funciona sozinha, sem qualquer ajuda externa. Na hora de aplicar um estilo ou algum dos recursos mais bacanas, você pode acabar usando um pouco de CSS3 ou JavaScript, mas não é necessário aprender toda a linguagem JavaScript ou CSS3 para se divertir bastante.

Por exemplo, o script a seguir utiliza o novo elemento `<datalist>` que não estava disponível nas versões anteriores de HTML. Insira este código em um editor de texto, salve o arquivo como `Datalist.html`, abra-o em seu navegador e você vai ver como isso ajuda o usuário a inserir dados. (Você encontra `Datalist.html` na pasta deste capítulo, na página do livro, em www.bookman.com.br.)

```
<!DOCTYPE HTML>
<html>
<head>
```

```

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Datalist</title>
</head>
<body>
<p>
  <label> Which of the following would you like to learn?<br />
  <input type="text" name="web" list="lang">
  <datalist id="lang">
    <option value="HTML5">
    <option value="JavaScript">
    <option value="jQuery">
    <option value="ActionScript 3.0">
    <option value="Java">
  </datalist>
</label>
<br />
</p>
</body>
</html>

```

Ao abrir o arquivo em um navegador Opera, você verá a lista de opções de entrada que aparece na Figura 1-2.

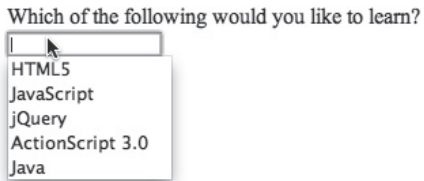


Figura 1-2: Usando a tag `<datalist>` em um navegador Opera.

Ao contrário das versões anteriores de HTML, nas quais uma entrada de texto não exibía uma lista de opções ao usuário, esta exibe.

USANDO TAGS VÁLIDAS DE HTML4

Mesmo que você conheça HTML4 (ou versões anteriores de HTML), pode se surpreender pela quantidade de elementos HTML que talvez não saiba usar ou sequer tenha ouvido falar. Por exemplo, o que é a tag `<q>`? Quando é usada? Se for novato em HTML, não tente decorar todos os elementos da Tabela 1.2, mas dê uma olhada neles para ter uma ideia geral das tags disponíveis e suas descrições.

Tabela 1.2: Tags de versões do HTML anteriores que continuam válidas

Tags válidas	Descrição
<code><!--...--></code>	Um comentário
<code><!DOCTYPE></code>	O tipo de documento (apenas um em HTML5)
<code><a></code>	Hyperlink para uma página ou área da página
<code><abbr></code>	Uma abreviatura
<code><address></code>	Container para endereço
<code><area></code>	Uma área dentro de um mapa de imagem
<code></code>	Texto em negrito
<code><base></code>	Uma URL base para todos os links em uma página
<code><bdo></code>	Direção de exibição do texto
<code><blockquote></code>	Um bloco de texto
<code><body></code>	Começo de um elemento do corpo
<code>
</code>	Uma única quebra de linha
<code><button></code>	Um botão clicável
<code><caption></code>	Uma legenda de tabela
<code><cite></code>	Container para uma citação
<code><code></code>	Formato para texto de código de computador
<code><col></code>	Define atributos para colunas de tabelas
<code><colgroup></code>	Container para grupos de colunas de tabelas
<code><dd></code>	Container para um valor para o elemento <code><dt></code>
<code></code>	Container para texto excluído
<code><dfn></code>	Representação da instância definidora de termo
<code><div></code>	Demarcação de divisão em um documento
<code><dl></code>	Cabeçalho para uma lista de associações
<code><dt></code>	Especificação para um nome em um grupo nome-valor (lista de descrições)
<code></code>	Texto enfatizado
<code><fieldset></code>	Container para um conjunto de controles de formulário
<code><form></code>	Container para um formulário, tipicamente com elementos de entrada
<code><h1></code> a <code><h6></code>	Títulos de texto 1 a 6
<code><head></code>	Container para o primeiro código a ser interpretado pelo navegador
<code><hr></code>	Linha horizontal

(continua)

Tabela 1.2

Tags válidas	Descrição
<html>	Container para um documento HTML
<i>	Texto em itálico
<iframe>	Moldura em uma subjanela interna
	Container de imagem
<input>	Campo de entrada do usuário dentro de um container de formulário
<ins>	Container para texto inserido dentro dos limites implícitos de um parágrafo
<kbd>	Container para entrada de teclado do usuário
<label>	Representação de uma legenda dentro de uma interface de usuário
<legend>	Título em uma borda de elemento fieldset
	Indicador de item em uma lista
<link>	Um recurso de referência (por exemplo, CSS)
<map>	Container para mapa de imagem
<mark>	Texto em um contexto marcado como texto em um contexto diferente
<menu>	Container para uma lista de comandos
<meta>	Container para metainformação
<object>	Container para objeto incorporado (por exemplo, um arquivo SWF)
	Uma lista numerada (ordenada)
<optgroup>	Um cabeçalho de agrupamento de opções em uma lista de opções
<option>	Container para opções individuais em uma lista suspensa
<p>	Parágrafo
<param>	Parâmetros de plug-in
<pre>	Formato de texto pré-formatado
<q>	Texto delimitado por aspas
<samp>	Resultado ou fragmento de código de computador
<script>	Container de script para CSS, JavaScript, ou outro script reconhecido
<select>	Uma lista selecionável
<small>	Texto pequeno
	Seção de linha em um documento
	Texto forte que se parece com negrito
<style>	Container para uma definição de estilo
<sub>	Texto subscrito

(continua)

Tabela 1.2 (continuação)

Tags válidas	Descrição
<sup>	Texto sobrescrito
<table>	Uma definição de tabela
<tbody>	Demarcação para um bloco de linhas no corpo de uma tabela
<td>	Uma célula de tabela
<textarea>	Container para uma área de texto
<tfoot>	Representação para um bloco de linhas nos sumários de coluna de uma tabela
<th>	Formato de cabeçalho de tabela
<thead>	Representação para um bloco de linhas nos sumários de coluna para o cabeçalho tabela
<title>	Título do documento
<tr>	Demarcação de linha de uma tabela
	Uma lista não ordenada (lista com marcadores)
<var>	Estilo variável em fórmula

A maioria dos elementos com os mesmos nomes de HTML4 são exatamente iguais em HTML5, mas alguns possuem significados levemente modificados. Além disso, as regras para algumas tags também mudaram. Por exemplo, ao criar tabelas, a tag para especificar uma linha <tr> não requer mais a tag de fechamento </tr>*. Alguns atributos para elementos também mudaram. À medida que aprender mais sobre as novas funcionalidades de HTML5, você verá que vários dos “velhos” elementos apresentam muitas características novas. O script de tabela em HTML a seguir é um novo exemplo com velhos elementos. Digite este texto no seu editor de texto, salve o arquivo como `NewOldTable.html` e abra o documento em um navegador Opera.

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Table</title>
</head>
<body>
<table>
  <caption>
    =Element Types=
  </caption>
  <thead>
    <tr>
```

* N. de R.T.: Embora as tags de fechamento não precisem ser escritas e especificadas no código, é uma boa prática mantê-las. Isso deixa o código escrito mais organizado e evita possíveis erros de compatibilidade entre navegadores.

```

    <th> Type
    <th> Text
    <th> Graphics
<tbody>
  <tr>
    <th> UI
    <td> text input
    <td> button
  <tr>
    <th> Links
    <td> underlined
    <td> icon
</table>
</body>
</html>

```

A Figura 1-3 mostra a aparência de sua tabela.

=Element Types=		
Type	Text	Graphics
UI	text input	button
Links	<u>underlined</u>	icon

Figura 1-3: Tabela criada com HTML5.

Geralmente, tabelas não são usadas para formatar texto. Em vez disso, usamos tabelas para formatar dados – por exemplo, dados carregados a partir de um banco de dados ou criados dinamicamente por outro programa, como JavaScript. Porém, em HTML5, as tabelas (usadas em conjunto com CSS3) fazem um pouco mais de formatação do que em versões anteriores de HTML e CSS.

ESQUECENDO OU SUBSTITUINDO TAGS INVÁLIDAS

Este último conjunto de tags (ver Tabela 1.3) é para todos aqueles que já conhecem HTML4 ou versões anteriores de HTML. As tags a seguir não são mais válidas, seja por que apresentavam certos problemas ou por terem sido substituídas por outras estruturas que lidavam melhor com suas antigas funções.

Se você é um novato em HTML, olhe para essas tags para ter uma ideia do que evitar. Na hora de trabalhar com HTML, é possível encontrar muitas amostras na Web, e há grandes chances de que você as incorpore em seu próprio código. Porém, uma vez que HTML5 é uma linguagem tão nova, você verá que a maioria dos documentos em HTML foi criada com versões anteriores, que poderão conter tags obsoletas. Quando isso acontecer, substitua essas tags inválidas por estruturas mais novas.

Tabela 1.3: Tags inválidas

Tags excluídas	Removida ou Substituída
<code><acronym></code>	Substituída por <code><abbr></code>
<code><applet></code>	Substituída por <code><object></code>
<code><basefont></code>	CSS lida melhor com essa função
<code><bgsound></code>	Substituída por <code><audio></code>
<code><big></code>	CSS lida melhor com essa função
<code><blink></code>	Removida de HTML5
<code><center></code>	CSS lida melhor com essa função
<code><dir></code>	Substituída por <code></code>
<code></code>	Removida de HTML5
<code><frame></code>	Removida de HTML5
<code><frameset></code>	Removida de HTML5
<code><isindex></code>	Substituída pelo uso explícito de <code><form></code>
<code><marquee></code>	Removida de HTML5
<code><multicol></code>	Removida de HTML5
<code><nobr></code>	Removida de HTML5
<code><noframes></code>	Removida de HTML5
<code><noscript></code>	Somente em conformidade com a sintaxe HTML
<code><s></code>	CSS lida melhor com essa função
<code><spacer></code>	Removida de HTML5
<code><strike></code>	CSS lida melhor com essa função
<code><tt></code>	CSS lida melhor com essa função
<code><u></code>	CSS lida melhor com essa função

Uma das tags inválidas mais comuns é `<center>`, mas você pode facilmente centralizar textos usando um pouco de CSS, conforme o exemplo a seguir. Digite este texto no seu editor de texto, salve o arquivo como `CenterMe.html` e abra o documento em seu navegador.

```
<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
h1 {
text-align:center;
}
</style>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Centralize com CSS</title>
```

```

</head>
<body>
<h1>É Possível Centralizar Cabeçalhos</h1>
</body>
</html>

```

Tudo que você verá quando testar o código em seu navegador vai ser:

É Possível Centralizar Cabeçalhos

Isso pode parecer muito trabalhoso para centralizar um simples cabeçalho, mas as páginas são geralmente curtas, e agora você pode centralizar qualquer cabeçalho com uma tag `<h1>`, porque já mudou o comportamento da tag. Você pode modificar a tag que quiser usando CSS. (Você vai aprender sobre CSS3 no Capítulo 3, mas, se consegue ver o cabeçalho no centro de sua página, você já usou a linguagem.)

ESCOLHENDO UM NAVEGADOR PARA INTERPRETAR HTML5

Se quiser começar uma animada discussão com outros desenvolvedores HTML5, basta perguntar "Qual é o melhor navegador?". Você deveria estar mais preocupado com o navegador usado por quem irá visualizar seu site na Web – não com o navegador usado por outros desenvolvedores. Os desenvolvedores, em geral, usam o melhor navegador até que surja um melhor, por isso eles podem usar navegadores mais avançados e melhores do que o usuário médio da Web. Se você quer que os visitantes de seu site tenham a melhor experiência possível, tente descobrir qual navegador eles têm maior chance de usar. Uma ideia melhor ainda, na hora de desenvolver software para si mesmo ou para um cliente, é testar suas páginas Web em todos os principais navegadores e em pelo menos duas das principais plataformas – Macintosh e Windows. Os principais desenvolvedores de navegadores também fazem navegadores para o Linux OS, mas pouquíssimas pessoas usam sua caixa de Linux para navegar na Web.

Olhando para os principais navegadores que suportam HTML5, a maioria pode ser usada tanto em sistemas operacionais Windows quanto Macintosh, mas às vezes o navegador requer um OS mais novo. Por isso, se você tem um sistema operacional mais antigo, certifique-se de que os requisitos para o navegador que você utiliza estejam de acordo com seu sistema operacional.

Há vários anos, a Microsoft parou de produzir o Internet Explorer (IE) para Macintosh. Porém, a Apple produz uma versão para Windows de seu navegador, o Safari. Três desenvolvedores de navegadores – Google, Mozilla e Opera – não produzem sistemas operacionais para computadores, mas sim navegadores. Nesta seção, apresento resenhas do Firefox, Chrome, Opera, Safari e IE9.

Tenha em mente que as funcionalidades dos navegadores mudam constantemente. As informações estão atualizadas no momento da escrita deste livro, mas podem já ter mudado no momento da sua leitura.

MOZILLA FIREFOX

O Mozilla tem suas raízes no navegador original da Netscape, chamado de Netscape Navigator, que foi lançado no início dos anos 1990. O navegador tinha um mascote que lembrava a criatura do filme Godzilla. O Mosaic foi um navegador desenvolvido pela Universidade de Illinois, que mais tarde se tornou o Netscape Navigator. A combinação entre Mosaic e Godzilla resultou em Mozilla, que hoje é uma empresa sem fins lucrativos, a Fundação Mozilla. O Firefox é o principal navegador da Mozilla com suporte a HTML5.

Além de suportar os sistemas operacionais Windows e Macintosh, o Firefox também suporta o sistema operacional Linux. O Linux não é considerado um sistema operacional primário para computadores domésticos, mas é um dos sistemas operacionais primários para servidores. O Firefox é disponibilizado gratuitamente para todos os sistemas operacionais suportados. A Figura 1-4 mostra a tela de um aplicativo HTML5 no Firefox.

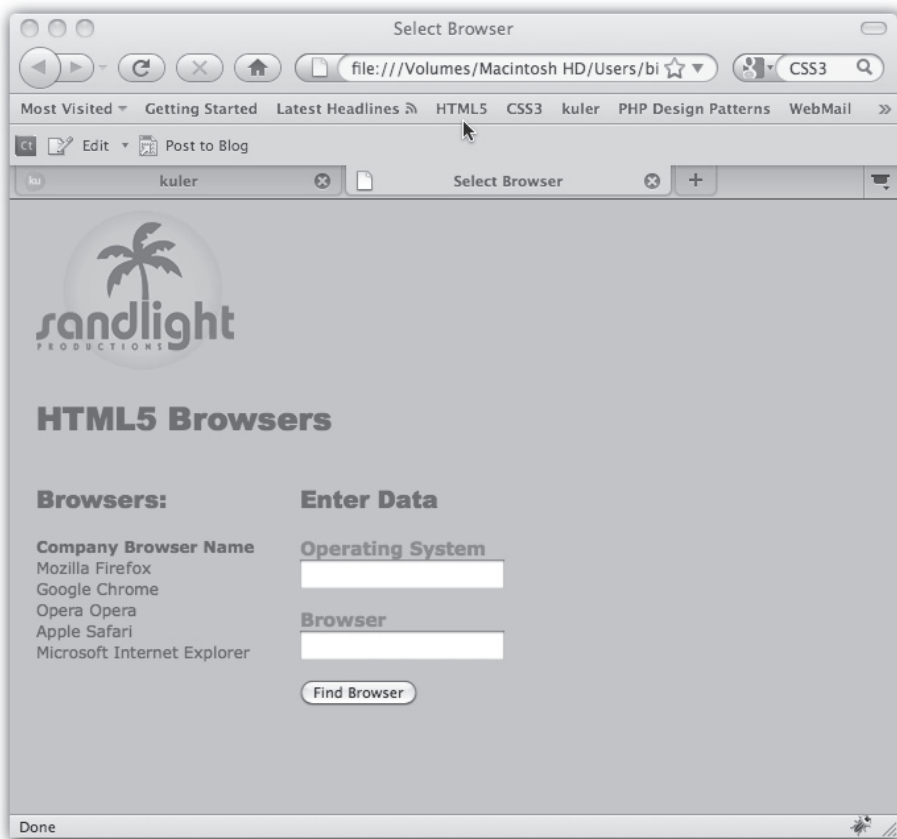


Figura 1-4: O navegador Mozilla Firefox.

Observe que na janela da URL (a janela onde você coloca o endereço em HTML) a referência é para `file:///Volumes/MacintoshHD/` no lugar de um endereço `http://`. Isso acontece porque a página está armazenada na área de trabalho do computador. Além disso, você verá que a aparência das coisas é diferente se são exibidas em um ambiente Windows ou em um Macintosh – ainda que seja usado o mesmo navegador. (A página de exemplo é meramente ilustrativa e não seleciona os navegadores para você.)

GOOGLE CHROME

O Google, famoso por seu mecanismo de busca e mapas, criou seu navegador, o Chrome, a partir do zero com HTML5 em mente. Possui navegadores para sistemas operacionais Apple, Windows e Linux – todos disponibilizados gratuitamente. A Figura 1-5 mostra exatamente a mesma página Web, no mesmo computador que a Figura 1-4 – veja se você consegue detectar as diferenças.

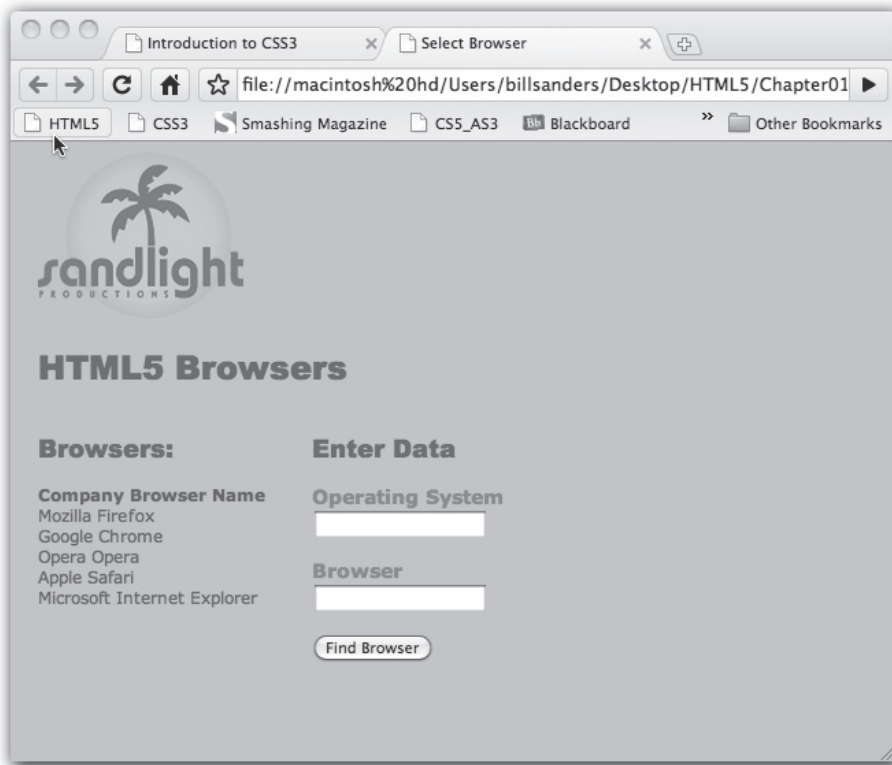


Figura 1-5: Google Chrome exibindo a mesma página em HTML5 mostrada na Figura 1-4.

Fora os diferentes estilos dos dois navegadores, pode ser difícil enxergar as diferenças na página. Com uma página simples, as diferenças sutis não afetarão a aparência de sua página Web. No entanto, à medida que suas páginas ficarem maiores e mais complexas, essas pequenas diferenças podem aumentar.

O Adobe BrowserLab (<https://browserlab.adobe.com/>), uma ferramenta para desenvolvimento de páginas Web, permite visualizar a aparência de uma página Web em diferentes navegadores ao mesmo tempo. O aplicativo BrowserLab pode ser executado diretamente a partir do Adobe Dreamweaver CS5, ou você pode visitar a página do Adobe BrowserLab. Para obter uma diferença um pouco mais impressionante, vamos comparar a página-exemplo no Firefox em um Macintosh, e no Windows 7 rodando o Google Chrome. A Figura 1-6 mostra a comparação lado a lado. (Os gráficos não são exibidos.)

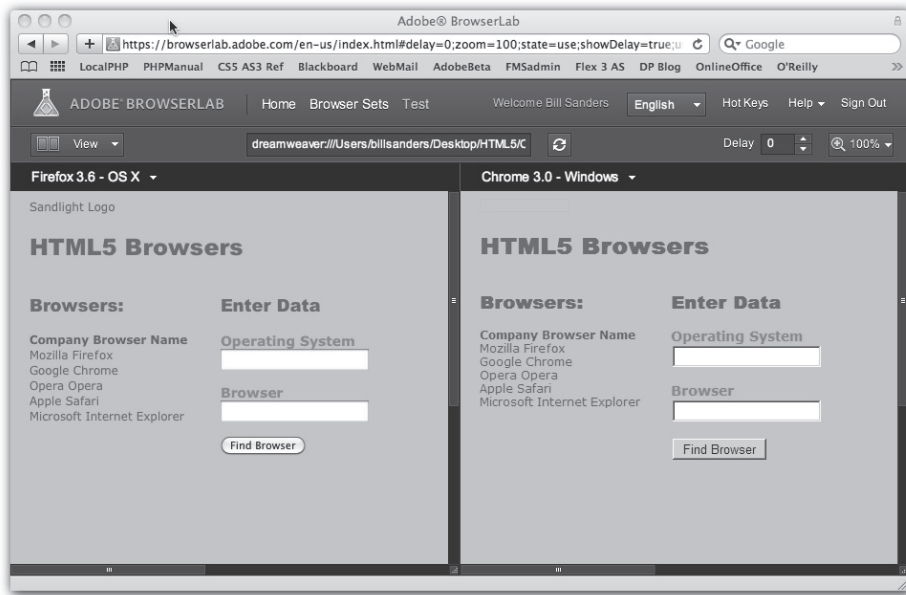


Figura 1-6: Comparando navegadores no Adobe BrowserLab.

Parte da diferença deve-se às formas como os sistemas operacionais Windows e Macintosh exibem texto e interfaces de usuário (UIs). Outra visão que o BrowserLab oferece é chamada de onionskin (casca de cebola), ou seja, quando sobrepõe as duas páginas para que você possa ver mais precisamente onde texto e UIs aparecem. A Figura 1-7 mostra essa diferença.

Quanto mais borrada a aparência da onionskin, maiores são as diferenças na forma como os materiais página Web são processados. Na Figura 1-6, você pode observar que a visualização é extremamente borrada, indicando que há algumas importantes diferenças entre navegadores e sistemas operacionais.

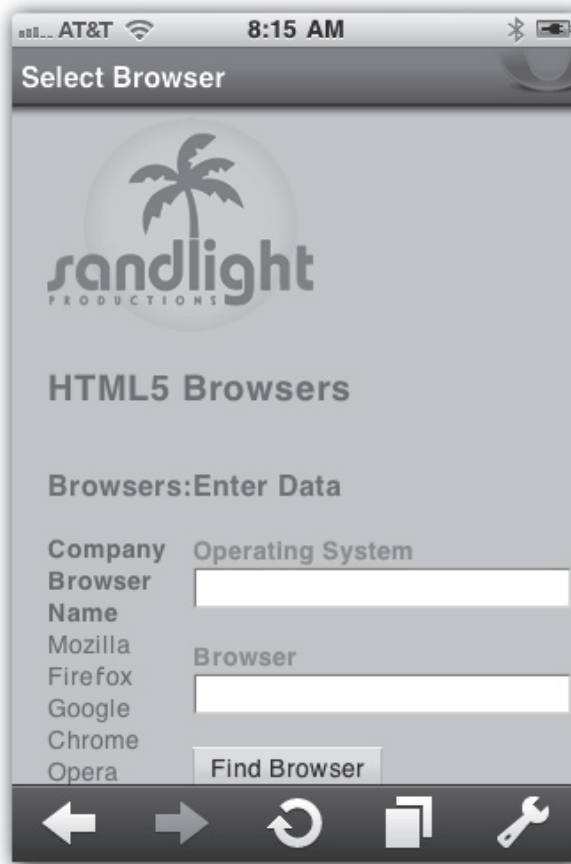


Figura 1-8: Navegador Opera Mini 5.

Os navegadores Opera completos estão disponíveis também para Windows, Macintosh e Linux. Na hora de criar páginas Web, você deve planejá-las para dispositivos de diferentes tamanhos. Como você pode ver, a amostra de aplicativo que temos utilizado cabe perfeitamente tanto em um dispositivo móvel quanto em telas grandes.

APPLE SAFARI

A Apple produz navegadores Safari para Macintosh e Windows, assim como para dispositivos móveis. Para fins comparativos, a Figura 1-9 mostra a aparência da amostra de aplicativo no Mobile Safari, da Apple, desenvolvido para o iPhone. Compare esse exemplo com o Opera Mini 5 na Figura 1-8.

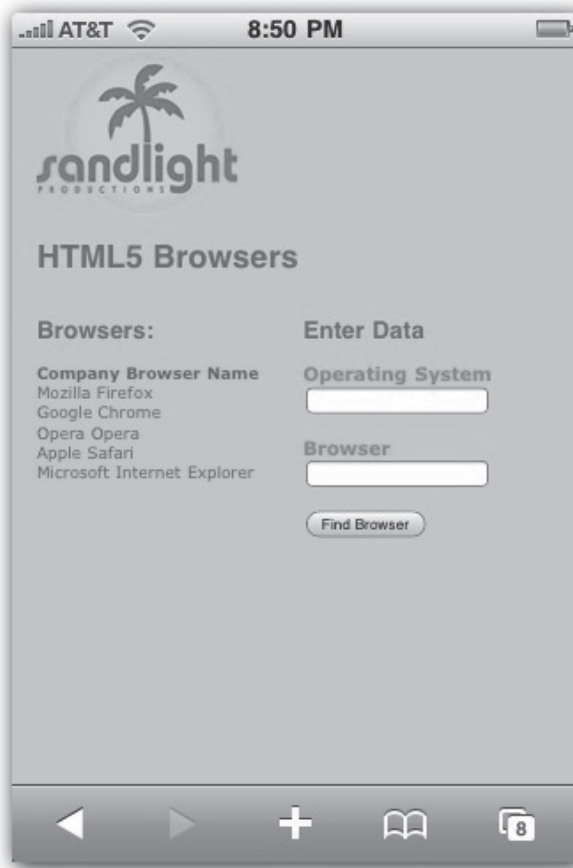


Figura 1-9: Navegador Mobile Safari.

Assim como existem poucas diferenças entre a aparência das páginas Web visualizadas em um computador tipo desktop ou em um laptop, você não deve encontrar muitas diferenças entre o que é mostrado nos diferentes navegadores para dispositivos móveis. Isso é ótimo! Os desenvolvedores Web perdem muito tempo tentando garantir que todas as suas páginas tenham a mesma aparência em diferentes navegadores e plataformas. Com uma implementação comum de HTML5, isso não deve ser um problema. Outras funcionalidades únicas dos navegadores, tais como a presença de abas ou outras características que facilitam a navegação na Web, não apresentam problemas, desde que a implementação de HTML5 seja feita de acordo com as especificações definidas pelo World Wide Web Consortium (W3C).

MICROSOFT INTERNET EXPLORER 9

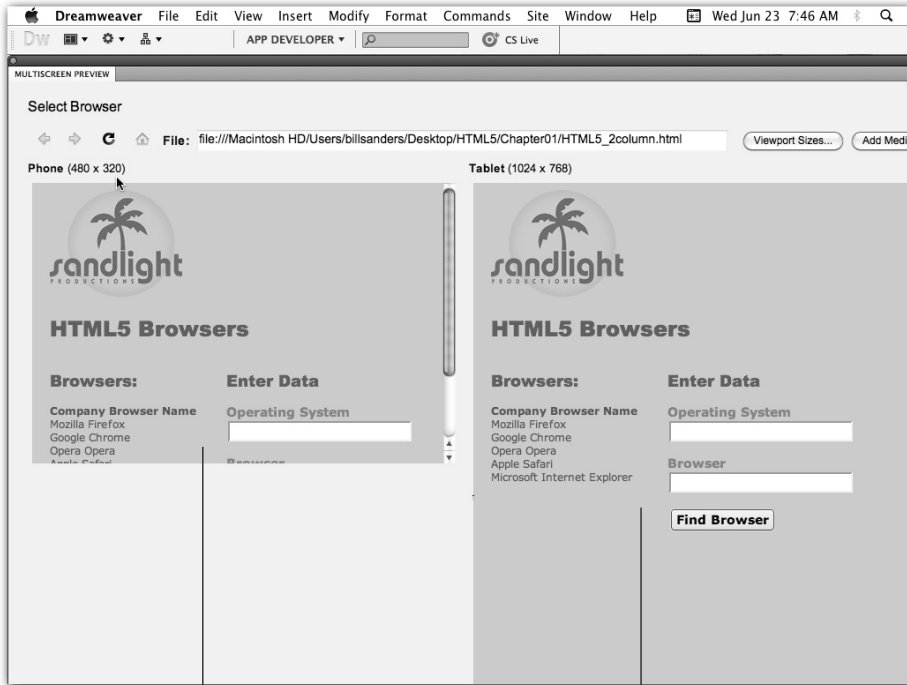
No momento da escrita deste livro, o Internet Explorer 9 (IE9) ainda estava em fase beta. Segundo a Microsoft, seu navegador IE9 será totalmente compatível com os padrões HTML5. Sempre que possível, ao longo deste livro, incluímos exemplos que mostram o IE9 em ação com páginas Web em HTML5, usando o navegador IE9 beta. A Figura 1-10 mostra a página de teste no navegador IE9 beta.



Figura 1-10: Internet Explorer 9.

VISUALIZANDO DIFERENTES MODOS DE EXIBIÇÃO

Como você já viu, as páginas Web podem ser visualizadas em diferentes navegadores e sistemas operacionais. Os desenvolvedores Web devem levar em conta as características dos dispositivos em que suas páginas serão visualizadas, como um computador tipo desktop ou um telefone celular. Suponha que você esteja desenvolvendo para iPhone e iPad (ou algum outro dispositivo móvel e tablet); se conseguir visualizar seu trabalho lado a lado, será mais fácil fazer comparações. O Adobe Dreamweaver, uma ferramenta para desenvolvimento de páginas Web, permite ao desenvolvedor visualizar múltiplas dimensões simultaneamente, conforme mostra a Figura 1-11.



Telefone definido para iPhone: 480 x 320

Tablet definida para iPad: 1024 x 768

Figura 1-11: Visualização “multitelas” no Adobe Dreamweaver.

Você pode modificar as dimensões do dispositivo. Um Motorola Droid, por exemplo, tem tela de 854 x 480 e um Sony VAIO UX tem tela de 1024 x 600. A pré-visualização em múltiplas telas ajuda você a decidir a melhor forma de configurar a página para seus espectadores. Encontrar a melhor solução é uma verdadeira arte, e pode ser mais fácil se você sabe o máximo possível sobre seu público e os dispositivos que eles têm mais chance de utilizar para visualizar seu material.

ASSUMA O VOLANTE

Para começar, este primeiro exemplo permite adicionar algumas informações sobre você mesmo. Não se preocupe em entender tudo (ou nada!) a menos que tenha alguma experiência em HTML. Basta substituir seus dados nas áreas marcadas com sinais de igual duplos. Salve a página em seu computador usando o nome `wheel1.html`. (Você encontra `wheel1.html` na pasta deste capítulo, na página do livro, em `www.bookman.com.br`.)

```
<!DOCTYPE HTML>
<html>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<head>
```

```
<style type="text/css">
body {
    background-color:blanchedAlmond;
    color:saddleBrown;
    font-family:Verdana, Geneva, sans-serif;
    font-size:12px;
    margin-left:20px;
}
h1, h2 {
    font-family:"Arial Black", Gadget, sans-serif;
    color:midnightBlue;
}
h1 {
    text-align:center;
}
h3 {
    background-color:goldenrod;
    color:ghostwhite;
    font-size:11px;
    font-family:"Arial";
}
</style>
<title>The Wheel</title>
</head>
<body>
<h1>==Your Name== : The Mighty HTML5 Web Developer</h1>
<h2>==Your Company Name== provides full Web services</h2>
<ul>
    <li>==Service 1==</li>
    <li>==Service 2== </li>
    <li>==Service 3== </li>
    <li>==Service 4== </li>
    <li>==Service 5== </li>
</ul>
<h3>&nbsp; All services guaranteed. Our complaint department is located at: ==URL
    where complains can be sent== &nbsp;. </h3>
</body>
</html>
```

Quando você testar a página em um navegador, observe se ela tem a aparência que você esperava. Além disso, você pode querer saber quais as diferenças de aparência em outros navegadores e no seu dispositivo móvel. (**Lembre-se:** navegadores da Web são gratuitos.) Se quiser fazer mais algumas alterações, vá para www.w3.org/TR/css3-color/#svg-color. Lá você encontrará uma lista com todos os nomes das cores que podem ser utilizadas com HTML5. Tente alterar os nomes das cores no código para cores lhe agradam.