

André Milani
Softblue



HTML

e-book



Sumário

Apresentação	4
Disclosure	5
1. Introdução	6
2. História da Internet	7
3. O que é HTML?	9
4. O que é o CSS?	11
5. Estrutura básica de páginas HTML.....	12
5.1. Conteúdo hipertexto	12
5.2. Cabeçalho de uma página	14
5.3. Título da janela	14
5.4. Configurando a codificação dos caracteres	15
5.5. Corpo do conteúdo.....	17
5.6. Integrando a página com estilos CSS	18
6. Trabalhando com textos	24
6.1. Parágrafos, quebras de linha e espaços em branco	25
6.2. Agrupamento de elementos (trechos de textos)	27
6.3. Negrito, itálico, sublinhado e riscado	29
6.4. Textos com maior relevância semântica	31
6.5. Texto subscrito e sobrescrito	35
6.6. Títulos em conteúdos de texto	36
7. Trabalhando com artigos e seções.....	38
7.1. Organizando artigos em seções	39
7.2. Trabalhando com links (pontos de acesso).....	42
8. Trabalhando com imagens	44
8.1. Trabalhando com vídeos	47
9. Trabalhando com listas	49
9.1. Listas não numeradas	49
9.2. Listas numeradas.....	50
10. Trabalhando com tabelas.....	51
11. Trabalhando com formulários	54
11.1. Formulário.....	54
11.2. Legendas.....	54

11.3. Pontos de interação entre o formulário e o usuário	55
12. Trabalhando com painéis	60
13. Semântica avançada da estrutura do site	62
14. Exemplo de página macro estruturada semanticamente.....	65
14.1. Estrutura básica da página	65
14.2. Construindo o topo da página	68
14.3. Conteúdo principal	72
14.4. Conteúdo lateral	75
14.5. Rodapé da página.....	77
14.6. Código na íntegra	78
15. Ufa, chegamos ao final de uma jornada!	83
Lista de Figuras	84
Lista de Códigos-Fonte.....	85
Índice Remissivo	86

Apresentação

A Softblue é uma empresa de cursos online na área de programação. Fundada em 2003 na cidade de Curitiba-PR, a empresa conta atualmente com um grande portfólio de cursos e dezenas de milhares de alunos espalhados em dezenas de países pelo mundo.

Este e-book foi criado por André Milani, sócio fundador da Softblue. André Milani é formado em Ciência da Computação pela PUC-PR, pós-graduado em Business Intelligence pela mesma instituição e possui diversas certificações na área de TI. É também autor de vários livros na área de informática, entre eles o *Programando para iPhone e iPad*, *MySQL - Guia do Programador* e *Construindo Aplicações Web com PHP & MySQL*, todos pela editora Novatec. Atua desde 2003 com desenvolvimento web e treinamentos de profissionais. Também é desenvolvedor de aplicativos para o ambiente iOS da Apple, possuindo aplicações que juntas somam mais de 50.000 downloads na AppStore.

Disclosure

Este e-book foi elaborado pela Softblue e é de uso exclusivo de seu destinatário. Seu conteúdo não pode ser reproduzido ou distribuído, no todo ou em parte, a qualquer terceiro sem autorização expressa.

A reprodução indevida, não autorizada, deste e-book ou de qualquer parte dele sujeitará o infrator à multa de até 3 (três) mil vezes o valor do e-book, à apreensão das cópias ilegais, à responsabilidade reparatória civil e persecução criminal, nos termos dos artigos 102 e seguintes da Lei 9.610/98.

1. Introdução

Para construir um site ou uma solução web que possa ser acessada por meio de um navegador (programa de acesso a páginas na internet), é necessário ter conhecimentos em HTML. Fique tranquilo: se você ainda não sabe é que é um site, web ou HTML, este e-book lhe ensinará esses conceitos.

Neste e-book será apresentado inicialmente um breve resumo da história da internet no mundo e vários conceitos importantes sobre essa tecnologia. Em seguida, os conhecimentos serão aprofundados na linguagem HTML e suas principais marcações, organizadas por categorias, com explicações e exemplos de uso na versão 5, a mais atual da linguagem.

O objetivo deste e-book é apresentar a você, programador, as principais marcações da linguagem HTML, a fim de organizar semanticamente o conteúdo de uma página web.

2. História da Internet

A internet foi criada em meados da década de 60 com o objetivo militar de descentralizar os centros de processamento de dados da defesa dos Estados Unidos, pois, fazendo isso, ficaria muito mais difícil para o inimigo destruí-los com poucos ataques. Por exemplo: ao invés de jogar uma única bomba em um único alvo e destruir todos os dados de uma única vez, seria necessário agora destruir dezenas ou até mesmo centenas de instalações em diferentes localidades.

Apesar da demanda militar ser o fator que originou a internet, muitos cientistas também começaram a usufruir dessa invenção. Com o passar do tempo, a demanda científica se tornou maior que a militar e a tecnologia continuou passando por constantes evoluções, sendo subsidiada pela Fundação Nacional de Ciência dos Estados Unidos até 1994. Além dos militares e cientistas, a internet também começou a ser utilizada pela população para fins civis e comerciais, seja para troca de e-mails ou para outros tipos de comunicações de softwares. Com o passar do tempo, observou-se uma falta de padronização nos conteúdos disponibilizados entre quem a utilizava. Foi o suficiente para dar força para alguns projetos, entre eles o do HTML, uma linguagem de marcação de hipertexto, utilizada até os dias de hoje.

As letras HTML representam as palavras em inglês *Hyper Text Markup Language*, termo que, em português, significa linguagem de marcação de hipertexto. Basicamente o HTML é um conjunto de marcações de textos, onde cada marcação apresenta uma funcionalidade específica a ser aplicada no texto delimitado pela marcação em questão. E acredite: essas marcações podem fazer muitas coisas!

Com o passar do tempo, o setor de comércio incorporou a internet ainda mais intensamente em sua realidade. Atualmente o setor com maior demanda de uso da internet é o comercial, fazendo com que a internet continue crescendo de forma exponencial por meio de sites de diversos tipos.

SITE: Local na internet que pode ser acessado a partir de um nome de domínio (endereço virtual), constituído de páginas com conteúdos construídos por meio do uso de conteúdos marcados via hipertexto (HTML).

Com o tempo a internet ganhou cada vez mais força e muitos adeptos pela sua facilidade de utilização, por reduzir distâncias e diversas outras inovações tecnológicas, sendo cada vez mais utilizada por diversas áreas como entretenimento, educação, cultura e negócios.

Como a internet disponibiliza uma forma de comunicação instantânea entre diferentes pontos geográficos no planeta, tornou-se comum o uso de endereços de sites e de e-mails em cartões de visita e, com isso, o crescimento do número de sites no comércio. Com a evolução e com o número de sites crescendo de forma muito rápida, surgiram os primeiros sites de busca, com diferentes propostas de como catalogar os conteúdos existentes na internet a fim de permitirem que buscas fossem realizadas em seus dados coletados. Desta forma, não era mais necessário conhecer uma empresa primeiro para depois acessar o seu site, pois a partir de um termo de busca tornou-se possível descobrir quais empresas estavam relacionadas ao tema, para depois conhecê-las por meio de seus sites.

Cada vez mais popular e informativa, a internet continua evoluindo até hoje. Um conjunto de mudanças significativas tem se propagado no que se chama da nova web 2.0, transformando não o conteúdo da atual web, mas sim, em como ela se relaciona com seus usuários. Alguns especialistas afirmam que a web original (1.0) conectava computadores, e que a web 2.0 conecta pessoas. Redes sociais, de todos os tipos, têm sido a grande marca da web 2.0, aproximando cada vez mais as pessoas, ao invés de apenas computadores.

3. O que é HTML?

Conforme você já viu no resumo da história da internet, a sigla HTML representa as iniciais do termo *Hiper Text Markup Language*, que em português poderia ser traduzido em algo como linguagem de marcação de hipertexto, e que nada mais é do que um conjunto de marcações utilizadas de forma integrada com conteúdo em formato de texto, com o objetivo de apresentar o conteúdo de uma página web de maneira organizada.

De todas as tecnologias existentes na construção de sites, dinâmicos ou não, o HTML é a mais fundamental de todas, pois sem o seu conhecimento, nenhuma outra tecnologia poderá ser empregada, pelo fato da maioria delas se relacionar com esta linguagem. Quanto mais se conhece HTML, melhor será o seu uso na integração com outras tecnologias.

Um exemplo simples de código HTML e o site gerado a partir deste conteúdo pode ser visualizado a seguir:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Softblue</title>
06   </head>
07   <body>
08     <table>
09       <tr>
10         <td>
11           <img src='logo.png'>
12           <br>
13           <br>
14           <h1>Site em construção</h1>
15           <br>Para maiores informações, envie um e-mail para contato@softblue.com.br
16         </td>
17       </tr>
18     </table>
19   </body>
20 </html>
```

Código-Fonte 3:1. exemploHtml.htm

O código anteriormente apresentado resulta na página web apresentada na Figura 3-1, quando interpretada e renderizada por um navegador web (ferramenta de acesso a páginas web). Considere também que a numeração de linhas que ocorre do lado esquerdo do código HTML não faz parte do código, mas servirá para explicações de linhas específicas do código no decorrer deste e-book.



Figura 3-1. Exemplo de página HTML

As marcações HTML podem ser categorizadas de acordo com as suas funcionalidades. As mais comuns são utilizadas para estruturar páginas, textos, listas, tabelas e formulários. Outras, mais avançadas, abordam questões de semântica, que está relacionada ao fato de como o conteúdo é organizado, atribuindo maiores prioridades para as partes mais importantes do conteúdo, visando o melhor aproveitamento dos dados por parte dos sites de buscas, como o Yahoo e o Google. Por meio do bom uso da semântica, é possível dar maior importância a conteúdos específicos de uma página, justamente para que os algoritmos dos sites de buscas fiquem mais atentos a estes conteúdos.

Outro fator importante que deve ser abordado é a possibilidade de formatar graficamente o conteúdo das páginas utilizando as marcações HTML. É possível colorir textos, mudar tipos e tamanhos de fontes, inserir bordas, trabalhar com margens, espaçamentos e diversas outras propriedades utilizando atributos específicos em boa parte das marcações. Contudo, com o avanço da internet, outra tecnologia evoluiu para fazer este serviço: o CSS. Atualmente a prática de programação mais recomendada é o uso do HTML para organizar o conteúdo da página semanticamente, e toda e qualquer formatação visual ser realizada especificamente por meio do uso de CSS. Os próprios algoritmos dos buscadores Yahoo e Google priorizam sites construídos dessa forma. Por este motivo este e-book de HTML5 abordará somente as questões que competem ao bom uso do HTML, introduzindo o leitor também aos conceitos mais fundamentais do CSS.

4. O que é o CSS?

Você provavelmente já deve ter ouvido falar de CSS. Essas letras representam as folhas de estilo, ou do inglês, *Cascading Style Sheets*. Esse recurso não faz parte do HTML. Trata-se de uma tecnologia própria que pode ser utilizada de forma integrada com o HTML ou com outros tipos de formatos, como por exemplo arquivos XML.

Por meio do uso de CSS é possível definir estilos de formatação de conteúdo. Isso implica em poder alterar tipo de fonte, tamanho, cor, margens, bordas e dezenas de outras propriedades por meio dessa tecnologia. A grande pergunta que você pode estar se fazendo neste momento é a seguinte: *“Por qual motivo eu devo optar pelo CSS, se eu posso alterar a cor do fundo da minha página utilizando o próprio HTML?”*. Se você se fez essa pergunta, ou se essa pergunta fez sentido para você, é porque você ainda não conhece os benefícios do uso do CSS.

Suponha que você pretenda construir um site com várias páginas HTML: a página principal, a página de produtos e a página de contato com a empresa. Se todas as suas páginas tiverem o fundo na cor azul, você precisará informar a cor em três arquivos HTML separados, na respectiva marcação HTML responsável por isso. Caso futuramente a cor de fundo de sua página precise ser alterada de azul para vermelho, você precisará realizar essa alteração em três lugares. Considere que existem sites com centenas ou até mesmo milhares de páginas. Já pensou o trabalho para alterar em todas elas? Para facilitar esse tipo de manutenção, você pode fazer uso do CSS.

Arquivos CSS são diferentes dos arquivos HTML. É possível criar um arquivo cuja extensão seja `.css` e, neste, criar os estilos de formatação, incluindo o estilo que definirá a cor de fundo de uma página. Dessa forma, todas as páginas do seu site poderão fazer referência a este arquivo e utilizá-lo livremente, por meio dos nomes dos estilos criados. Se futuramente for necessário alterar a cor de todas as páginas, e considerando que todas as páginas fazem referência ao arquivo CSS criado a parte, basta alterar a cor somente no CSS, para que todas as páginas reflitam essa alteração.

Não se preocupe se todos esses conceitos ainda parecem confusos para você. Eles ficarão mais claros com a construção dos primeiros exemplos deste e-book.

5. Estrutura básica de páginas HTML

Os arquivos HTML possuem um formato básico de estrutura que deve ser respeitado em todas as páginas web, por meio das marcações de definição estrutural das páginas hipertexto. Antes de um navegador interpretar os códigos de diagramação de conteúdo, geralmente existem outros comandos que informam ao navegador dados importantes, que nem sempre ficam expostos aos olhos do visitante do site. Informações estruturais e semânticas sobre o idioma da página, conjunto de caracteres, e diversas outras, podem (e devem) ser definidas, mesmo que não apareçam na janela do navegador, pois são tão importantes quanto todo o resto do conteúdo.

As primeiras marcações HTML que serão apresentadas são as que dizem respeito à estrutura das páginas web, como mostram os próximos tópicos. A partir deste ponto, serão apresentadas as principais marcações do HTML5. Algumas apresentam resultados visíveis na tela, outras não, mas são igualmente importantes. Não se preocupe com o uso isolado de algumas marcações estruturais ou de semântica. Seus usos ficarão mais claros ao serem integrados com outras marcações no decorrer deste e-book.

IMPORTANTE: Para acompanhar e praticar os exemplos descritos neste livro, você pode criar um ou mais arquivos com a extensão *.htm* com os conteúdos que serão apresentados. Lembre-se de sempre salvar o arquivo depois de realizar qualquer alteração. Caso contrário ela não estará visível ao atualizar o navegador (tecla *F5*).

5.1. Conteúdo hipertexto

A primeira marcação HTML de qualquer arquivo deve ser sempre a `<!DOCTYPE html>`. Este comando informa ao navegador que o arquivo que está sendo tratado é um documento de marcação de hipertexto (HTML). Nas versões anteriores do HTML essa propriedade era mais extensa, e foi reduzida a partir da versão 5 para o formato apresentado.

Logo em seguida, a marcação `<html>` deve ser aberta e, antes de seu fechamento, todo o conteúdo da página deve ser organizado e apresentado, como mostra o exemplo a seguir.

```
01 <!DOCTYPE html>
02 <html>
03   <!-- Conteúdo do HTML -->
04 </html>
```

Código-Fonte 5.1. `marcacaoHtml.htm`

A marcação HTML deve aparecer duas vezes no seu arquivo. A primeira é a marcação de abertura (linha 2), definindo o começo do conteúdo hipertexto da página. A segunda delimita o final do conteúdo (linha 4). Observe que a segunda, também chamada de marcação de fechamento, apresenta uma barra (/) antes do nome da marcação, indicando que ela representa o fim da marcação. Esse comportamento será seguido por várias outras marcações que serão abordadas neste e-book. Contudo, existem algumas marcações que não precisam de fechamento, como é o caso da `<!DOCTYPE html>`.

Como é possível notar, este comando utilizado isoladamente não exibe nenhuma informação na tela, nem mesmo na barra de título da janela. Ele apenas define o início e fim do conteúdo que será inserido dentro de suas marcações como conteúdo de uma página HTML. Mesmo que algumas outras marcações funcionem mesmo sem o uso da abertura e fechamento da marcação `<html>`, é importante utilizá-las, pois os algoritmos de buscadores como o Yahoo e o Google avaliam os códigos das páginas e códigos incompletos ou mal construídos são avaliados como menos relevantes, o que pode impactar na posição em que o seu site aparece na relação de busca desses sites.

Outra informação importante sobre as marcações HTML é que as mesmas podem ser escritas em caixa-baixa ou caixa-alta. No entanto, há uma forte preferência popular em utilizá-las em caixa-baixa (letras minúsculas), pelo fato de ser recomendado pelo órgão responsável pela organização da padronização da internet, o W3C.

Ainda no exemplo apresentado no código Código-Fonte 5:1, é possível notar a existência das marcações `<!--` e `-->`. Essas marcações representam comentários livres no conteúdo HTML, que não são formatados nem diagramados e não aparecem na tela. Em outras palavras, essas marcações servem para você inserir informações no seu conteúdo fonte do HTML, mas que são ignoradas pelos navegadores, não aparecendo para os visitantes da página. Por este motivo, o texto `Conteúdo do HTML` não é exibido na tela.

A marcação HTML pode ser complementada com o atributo `lang`, utilizado para informar ao navegador e também aos algoritmos dos buscadores o idioma do conteúdo da página, no seu formato de sigla ISO. No caso do idioma português brasileiro, a sigla utilizada deve ser a `pt-br`. O código-fonte com essa alteração pode ser visualizado a seguir:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <!-- Conteúdo do HTML -->
04 </html>
```

Código-Fonte 5:2. marcacaoHtmlAtributoLang.htm

Novamente, o exemplo construído ainda não apresenta nenhum resultado visível na tela do navegador.

5.2. Cabeçalho de uma página

Dentro da marcação `<html>`, deverá existir a marcação `<head>`, que é responsável por definir e informar ao navegador algumas propriedades sobre o arquivo, compondo o cabeçalho do mesmo.

Dentre essas informações, é possível detalhar características como título da janela, arquivos externos relacionados com o arquivo atual (arquivos JavaScript, CSS ou outros) entre outras informações do tipo `meta`, que serão apresentadas mais adiante, úteis para informar propriedades como o conjunto de caracteres da página ou outras configurações.

O cabeçalho de uma página faz parte do conteúdo hipertexto. Por este motivo, as marcações de abertura e fechamento do cabeçalho `<head>` devem estar dentro das marcações de abertura e fechamento do arquivo `<html>`.

Um exemplo de utilização da marcação `<head>` pode ser visto a seguir:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <!-- Conteúdo do cabeçalho -->
05   </head>
06 </html>
```

Código-Fonte 5.3. marcacaoHead.htm

Novamente, a utilização exclusiva da marcação `<head>` não impacta em nenhuma alteração visual da página. Ainda, novamente as marcações de comentários `<!--` e `-->` foram utilizadas para representar a localização do conteúdo que faz parte do cabeçalho de um arquivo HTML. O texto `Conteúdo do cabeçalho` não é impresso na tela por estar dentro das marcações de comentários, que são ignoradas junto com seus conteúdos pelos navegadores web.

5.3. Título da janela

A primeira informação que apresenta uma diferença na visualização de um arquivo HTML é o conteúdo da marcação `<title>`. O texto que estiver dentro de suas marcações será utilizado para apresentar o título da janela do navegador quando a página for carregada para visualização.

O título de uma janela é uma informação do cabeçalho do arquivo, e por este motivo as marcações de abertura e fechamento do título `<title>` devem estar dentro das

marcações de abertura e fechamento do cabeçalho `<head>`, que por sua vez devem estar dentro das marcações `<html>`.

A seguir é apresentado um exemplo de site cujo título é `Capítulo HTML`:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <title>Capítulo HTML</title>
05   </head>
06 </html>
```

Código-Fonte 5:4. marcacaoTitle.htm

Observe na imagem a seguir o resultado da interpretação do código-fonte HTML apresentado por um navegador web:

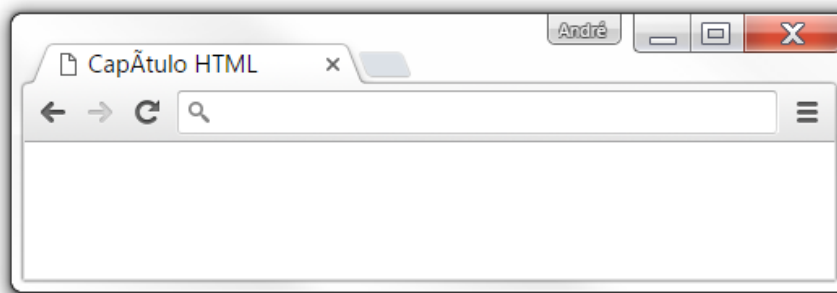


Figura 5-1. Título da página apresentado no topo da janela do navegador

O título foi associado com a janela do navegador. Mas dependendo dos caracteres envolvidos e do tipo da codificação utilizada para salvar o seu arquivo, problemas nos caracteres especiais, como acentuações e cedilhas, podem ser apresentados. Isso ocorre por existirem diferentes codificações que podem ser utilizadas nos arquivos de computador, sendo recomendado que você informe no HTML qual delas você está utilizando. No exemplo apresentado foi utilizada a codificação UTF-8, portanto isso deve ser configurado, como será visto no próximo tópico.

5.4. Configurando a codificação dos caracteres

Para informar no HTML a codificação utilizada pelo seu arquivo, é necessário fazer o uso de uma nova marcação chamada `<meta>`, que define uma configuração no cabeçalho do arquivo, integrada com o atributo `charset`, responsável por informar que a configuração em questão é a de codificação de caracteres. Neste caso, o seu código-fonte deve ficar da seguinte forma:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
```

```
06 </head>
07 </html>
```

Código-Fonte 5.5. marcacaoMeta.htm

Com a inclusão da marcação `<meta>` para a configuração do conjunto de caracteres, a interpretação do código-fonte atualizado pode ser conferida na imagem a seguir:

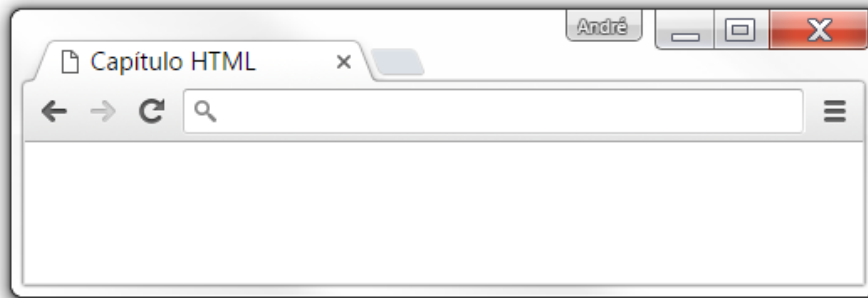


Figura 5-2. Caracteres corretamente interpretados pelo navegador

A marcação `<meta>` também pode ser utilizada para definir outras propriedades importantes do arquivo HTML, para que os algoritmos de busca possam catalogar com mais detalhes sua página, a fim de melhorar o posicionamento do seu site nos sites de busca.

Entre as principais utilizações da marcação `<meta>`, podemos informar uma descrição curta da página, palavras-chave específicas daquela página e o autor da mesma. Para definir essas informações, a marcação `<meta>` precisa informar o atributo `name`, com a propriedade que está definindo; e o atributo `content`, com o valor desejado, como mostra o código-fonte a seguir:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <meta name="description" content="Site de cursos online">
06     <meta name="keywords" content="Java, PHP, HTML, Android, iOS">
07     <meta name="author" content="André Milani (Softblue)">
08     <title>Capítulo HTML</title>
09   </head>
10 </html>
```

Código-Fonte 5.6. marcacaoMetaOutrasConfiguracoes.htm

A interpretação do Código-Fonte 5:6 não apresenta conteúdo na tela ainda, apenas no título da janela do navegador, por meio da linha 8, como já foi visto anteriormente. Mesmo diferenças não sendo apresentadas na tela, as novas marcações informadas nas linhas 5, 6 e 7 possuem alta importância semântica e auxiliam os algoritmos dos sites de busca a catalogar suas páginas com maiores detalhes.

É importante citar também que os editores de textos ou ferramentas de criação e edição de arquivos HTML podem salvar os arquivos em diferentes formatos, entre eles LATIN1, UTF-8 ou outros. Consulte no manual da sua ferramenta sobre como alterar o tipo de codificação e/ou salvar arquivos definindo o formato desejado para garantir o bom funcionamento do uso deste recurso.

Com a estrutura atual do arquivo, estamos quase prontos para começarmos a inserir conteúdos visíveis na tela. Para isso, precisamos conhecer mais uma marcação importante da estrutura básica de um arquivo HTML: a marcação `<body>`, abordada a seguir.

5.5. Corpo do conteúdo

Para finalizar as marcações mais básicas de um site, a marcação `<body>` representa o corpo do arquivo, informando que todo o conteúdo dentro desta marcação é o que deve ser diagramado e apresentado na janela do navegador para o usuário.

O corpo de uma página faz parte do arquivo principal. Por este motivo, as marcações de abertura e fechamento do corpo `<body>` devem estar dentro das marcações de abertura e fechamento do arquivo `<html>`, preferencialmente depois do fechamento do cabeçalho `</head>`.

Todo o conteúdo do site como tabelas, textos, imagens e links, que serão vistos mais adiante, deverão estar obrigatoriamente dentro da marcação `<body>` para terem seu correto funcionamento assegurado. Observe um exemplo de utilização da marcação `<body>` e as demais marcações que compõem a estrutura básica de uma página:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06   </head>
07   <body>
08     Conteúdo da página
09   </body>
10 </html>
```

Código-Fonte 5.7. marcacaoBody.htm

A interpretação do código-fonte atualizado, e que agora contém uma região adequada para inclusão de conteúdo, pode ser conferida na Figura 5-3. Observe também que o texto `Conteúdo da página` agora aparece sem o uso da marcação de comentário, ou seja, este texto é apresentado na tela para o usuário.

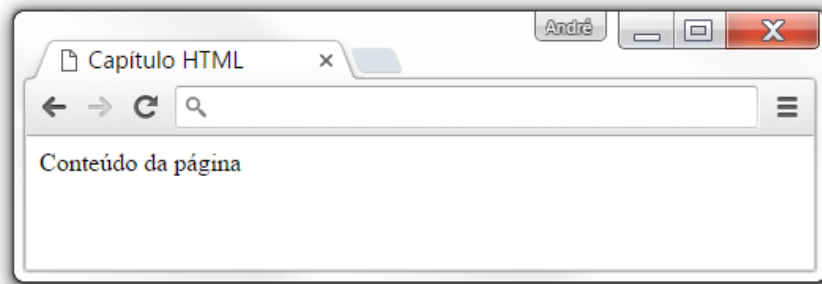


Figura 5-3. Página simples com conteúdo

A estrutura mais básica de um arquivo HTML é formada pelas marcações `<!DOCTYPE html>`, `<html>`, `<head>`, `<title>` e `<body>`. Existem várias outras marcações de estruturas, configurações e semânticas que serão abordadas no decorrer deste e-book. Não se preocupe se neste momento com algumas outras marcações que ainda não foram apresentadas: a ordem em que as mesmas serão abordadas neste e-book foi definida para o melhor aproveitamento do conteúdo, iniciando pela estrutura mais simples do arquivo.

5.6. Integrando a página com estilos CSS

É neste ponto que o CSS pode começar a ser integrado com o HTML para manipular visualmente o conteúdo da sua página.

Vimos até o momento um código básico HTML que cria uma página com um texto simples, mas já existem várias propriedades que podem ser formatadas visualmente, como tipo de fonte, tamanho, cor, entre outras. Portanto vamos trabalhar com alguns conceitos básicos do CSS.

A primeira forma de se utilizar o CSS é aplicando os estilos diretamente nas marcações HTML por meio do atributo `style`, da seguinte forma:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06   </head>
07   <body style="margin-left: 100px; margin-top: 50px;">
08     Conteúdo da página
09   </body>
10 </html>
```

Código-Fonte 5.8. estiloCssFormatoAtributo.htm

Observe na linha 7 o uso do atributo `style` na marcação `<body>`. Entre as aspas que delimitam o início e o fim do atributo `style` deve aparecer a formatação CSS desejada para esta marcação. Existem centenas de formatações CSS que podem ser utilizadas e

você vai conhecer boa parte das principais no decorrer deste e-book, não se preocupe. Por enquanto vamos considerar o uso da formatação de espaçamento para a margem esquerda, que pode ser configurada pela propriedade `margin-left`, neste caso definida com `100px` (`px` é a sigla do CSS para pixels); e da margem superior, configurada pela propriedade `margin-top` e definida com 50 pixels (`50px`). Observe que as propriedades CSS são atribuídas por meio do operador dois pontos (`:`) e com um ponto e vírgula (`;`) para finalizar a operação, podendo apresentar outras formatações na sequência. Este código-fonte, quando interpretado pelo navegador, apresenta o resultado apresentado na Figura 5-4.

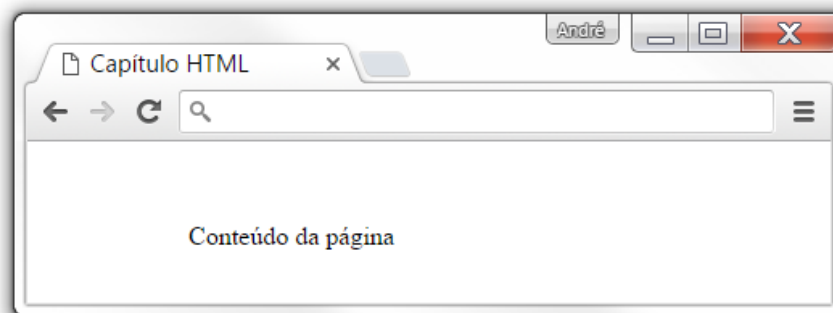


Figura 5-4. Formatando margens com CSS

Essa forma de associação entre marcações HTML e formatações CSS não é a mais apropriada, pois dificulta a manutenção caso a formatação seja utilizada em outras marcações da própria página. Suponha que uma determinada formatação CSS seja utilizada em cinco momentos isolados na página, como em cinco parágrafos ou em diferentes marcações. Seria necessário repetir a definição da formatação cinco vezes e, caso futuramente seja necessário trocar o valor para todas elas, cinco alterações serão necessárias.

Pensando em resolver este problema, o CSS disponibilizou uma forma de criar estilos e associá-los às marcações por meio de um nome de estilo. Para ilustrar essa situação, considere o código-fonte apresentado a seguir:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <style>
07       .meuEstilo
08       {
09         margin-left: 100px;
10         margin-top: 50px;
11       }
12     </style>
13   </head>
14   <body class="meuEstilo">
15     Conteúdo da página
```

```
16 </body>
17 </html>
```

Código-Fonte 5:9. estiloCssFormatoMarcacao.htm

Neste formato de associação de formatações CSS com o HTML, uma marcação `<style>` é aberta na linha 6 e fechada na linha 12, e dentro dessas marcações podem ser definidos estilos de formatação. A linha 7 define um nome para o estilo, que deve iniciar com um caractere ponto (.) por enquanto e, entre as chaves apresentadas nas linhas 8 e 11, as propriedades de formatação são vinculadas ao estilo que está sendo criado.

Por último, é necessário incluir o atributo `class` nas marcações que forem utilizar o estilo criado anteriormente, indicando o nome do estilo, como ocorre na linha 14 para a marcação `<body>`. Neste formato de associação entre HTML e CSS, caso existam várias marcações fazendo referências a um estilo por meio do atributo `class`, basta alterar uma única vez a formatação diretamente entre as linhas 8 e 11 para que todas as marcações HTML que façam uso deste estilo reflitam as alterações. O resultado da interpretação deste conteúdo HTML é o mesmo apresentado na Figura 5-4.

É importante ressaltar que é possível ter um ou mais estilos de formatação dentro da marcação `<style>`. Veremos isso mais adiante neste e-book. Já sobre a nomenclatura dos estilos, eles devem começar com o operador ponto (.). Nomes de estilos que não comecem com o operador ponto devem fazer referência direta aos nomes das marcações HTML. Por exemplo, caso o seu estilo deva ser aplicado somente na marcação `<body>`, o conteúdo CSS poderia ter sido criado como mostra o código-fonte a seguir:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <style>
07       body
08       {
09         margin-left: 100px;
10         margin-top: 50px;
11       }
12     </style>
13   </head>
14   <body>
15     Conteúdo da página
16   </body>
17 </html>
```

Código-Fonte 5:10. estiloCssNomeMarcacao.htm

Observe no Código-Fonte 5:10 que o estilo de formatação é nomeado com o título de uma marcação HTML, no caso a `<body>`, sem os sinais de menor e maior e sem o

operador ponto (.) utilizado no exemplo anterior. Dessa forma a marcação `<body>` não precisa fazer uso do atributo `class`, pois a referência entre a marcação HTML e o estilo CSS ocorre automaticamente. Para estilos de formatação específicos, e que vão se repetir sempre para a mesma marcação, este formato pode ser utilizado.

Ainda, se para uma mesma marcação HTML você precise de diferentes estilos e queira utilizar os nomes das marcações HTML para os diferentes estilos, você pode fazer de uma terceira forma no CSS, que é a seguinte:

```
p#grande
{
  font-size: 20pt;
}

p#pequeno
{
  font-size: 10pt;
}
```

Considere no código-fonte apresentado que dois estilos foram criados para a marcação `<p>` do HTML, que trabalha com parágrafos e será abordada mais adiante. O primeiro estilo é o `p#grande`, que define a fonte do texto com tamanho de 20 pontos (20pt), enquanto que o segundo estilo, `p#pequeno`, define o tamanho da fonte para 10 pontos (10pt). Para que as marcações `<p>` do HTML façam uso desses estilos, é necessário que informem o atributo `id` com o valor que aparece depois do operador `#` de cada nome de estilo, da seguinte forma:

```
<p id="grande">Texto grande</p>
<p id="pequeno">Texto pequeno</p>
```

Neste caso, a marcação `<p>` com o atributo `id` definido com `grande`, procurará no arquivo CSS pela formatação da tag `<p>`, seguido do operador `#` e seguido do nome definido pelo atributo `id`, que no caso é `grande`. Ou seja, procurará pelo estilo `p#grande`. O mesmo vale para `p#pequeno`.

Pergunta comum neste momento: “Qual formato devo utilizar?”. Depende de cada caso. Se o seu estilo puder ser utilizado por diferentes marcações HTML, é indicada a nomenclatura com o ponto (.) no início do nome, pois ela não faz relação com nenhuma marcação específica. Já se a formatação em questão for utilizada exclusivamente por apenas um tipo de marcação do HTML, é recomendado utilizar a nomenclatura que faz uso do nome da marcação HTML. E se houver diferentes formatações para a mesma marcação HTML, o terceiro modo é o mais indicado, pois usa o nome da marcação e ainda permite definir o título do estilo após o operador `#`.

Mas ainda temos um problema que pode ser resolvido: a grande maioria dos sites na internet não possui apenas uma página, mas sim dezenas ou até mesmo centenas.

Neste caso será necessário repetir a marcação `<style>` em todos eles e, quando for necessário mudar uma propriedade, alterar em todos os arquivos. Isso não parece uma coisa prática, certo?

Para contornar este problema, o CSS pode (e deve) ser mantido em um ou mais arquivos a parte, separado das páginas HTML. Dessa forma é possível criar referências para os arquivos CSS a partir das páginas, mantendo as formatações em um único lugar, facilitando a manutenção.

Para fazer isso, primeiro vamos separar o CSS do HTML. Crie um novo arquivo, chamado `arquivoDeEstilos.css`, com o conteúdo apresentado no Código-Fonte 5:11.

```
01 .meuEstilo
02 {
03     margin-left: 100px;
04     margin-top: 50px;"
05 }
```

Código-Fonte 5:11. arquivoDeEstilos.css

Em seguida, altere o seu arquivo HTML para o seguinte conteúdo:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03     <head>
04         <meta charset="utf-8">
05         <title>Capítulo HTML</title>
06         <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07     </head>
08     <body class="meuEstilo">
09         Conteúdo da página
10     </body>
11 </html>
```

Código-Fonte 5:12. estiloCssFormatoArquivo.htm

Observe que nesta versão de associação entre o CSS e o HTML o conteúdo de formatação fica isolado no arquivo CSS, e que o mesmo é referenciado pela página HTML por meio da marcação `<link>` dentro do cabeçalho do arquivo, na linha 6. Essa marcação deve informar os atributos `rel`, que deve ser preenchido com `stylesheet`, indicando que é um link que aponta para uma fonte de folhas de estilos de formatação; o atributo `type`, que informa o tipo do conteúdo relacionado, do tipo texto CSS (`text/css`); e o atributo `href`, com o endereço virtual do arquivo de estilos. Caso o arquivo seja salvo no mesmo diretório do arquivo HTML, apenas o nome (com extensão) é necessário. Se o arquivo estiver em outra pasta, é necessário informar o caminho para o arquivo.

O último formato de associação entre CSS e HTML, apresentado no Código-Fonte 5:12, é o mais indicado, por proporcionar a manutenção mais otimizada. No decorrer

deste e-book várias formatações serão apresentadas, relacionadas aos itens e marcações HTML abordadas em cada momento.

Agora que você já tem uma base da estrutura fundamental de um arquivo HTML e da integração com CSS, vamos dar sequência com as principais marcações e suas propriedades de formatação mais populares.

6. Trabalhando com textos

Uma vez apresentadas as marcações fundamentais que formam a estrutura básica de um arquivo HTML, as próximas a serem abordadas são as que fazem a organização de conteúdos de textos da página. É preciso diferenciar a estrutura de um texto de sua formatação. A estrutura é composta por parágrafos, linhas, quebras de linhas, blocos de textos isolados, enquanto a formatação é composta por tipo de fonte, tamanho, cor, espaçamento e outras propriedades.

Conforme mencionado anteriormente neste e-book, formatações como tipo de fonte, tamanho e cor devem ser aplicadas por meio de CSS. Mas existem algumas marcações HTML que gerenciam o uso de negrito, itálico, sublinhado e riscado que devem ser utilizadas em conjunto com o CSS, principalmente por algumas delas apresentarem variações com funções semânticas, dando mais relevância ao texto marcado por elas.

Outra curiosidade importante para quem está tendo seu primeiro contato com o HTML é que abrir um arquivo hipertexto em um editor como o Microsoft Word e colorir seus parágrafos não faz com que esses resultados sejam repassados para o conteúdo hipertexto, nem mesmo serem interpretados no navegador. Isso quer dizer que até mesmo a quebra de linha (tecla `<ENTER>`) deixa de funcionar se não houver a marcação HTML apropriada para ela no seu conteúdo hipertexto, assim como espaços em branco redundantes. Observe o código-fonte apresentado a seguir:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     Conteúdo da          página.
10     Segunda linha de conteúdo.
11     E mais              conteúdo.
12   </body>
13 </html>
```

Código-Fonte 6:1. textosSemMarcacao.htm

A interpretação do Código-Fonte 6:1 pode ser visualizada na Figura 6-1.

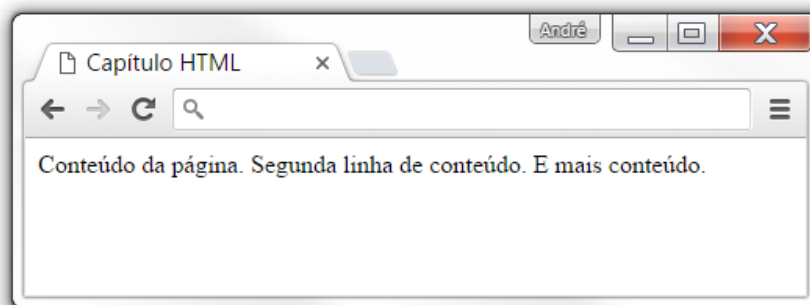


Figura 6-1. Textos com quebras de linhas e espaços em branco sem marcações

Observe na Figura 6-1 que as linhas aparecem todas em uma única linha. Isso se deve ao fato do HTML necessitar de marcações inclusive para representar parágrafos e quebras de linha. Ainda, os espaços em branco redundantes – quando aparecem dois ou mais espaços em branco juntos – nas linhas 9 e 11, que no código-fonte mantêm o ponto final de cada frase alinhados na mesma posição, são ignorados ao serem interpretados pelos navegadores web.

Existem várias marcações específicas para estruturas de texto no HTML, e com elas será possível contornar as situações mencionadas neste tópico. Você vai conhecer as principais a partir de agora.

6.1. Parágrafos, quebras de linha e espaços em branco

A marcação `<p>` representa um parágrafo no documento. Por meio dela é possível organizar e formatar separadamente os parágrafos, desde que cada um tenha suas próprias marcações de abertura e fechamento correspondentes.

Existe ainda a marcação `
` que executa uma quebra de linha, podendo ser utilizada inclusive fora das marcações que definem um parágrafo, ou dentro delas.

Para espaços em branco, é possível utilizar o código ` ` que representa um espaço em branco no HTML. Caso você queira apresentar cinco espaços em branco juntos, será necessário utilizar cinco vezes o código ` `. Dessa forma o HTML não ignorará os espaços em branco redundantes, escritos por meio deste código, na sua página.

A seguir, alguns exemplos da utilização das marcações `<p>` e `
`:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
```



```

01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     <p class="alinhadoDireita">Meu parágrafo alinhado a direita.</p>
10     <p class="alinhadoEsquerda">Meu parágrafo alinhado a esquerda.</p>
11     <p class="alinhadoCentralizado">Meu parágrafo centralizado.</p>
12     <p class="alinhadoJustificado">
13       Meu parágrafo justificado. Ele deve ser grande o suficiente para
14       que quebras de linhas sejam necessárias para que o efeito
15       justificado possa ser visto.
16     </p>
17   </body>
18 </html>

```

Código-Fonte 6:3. textosAlinhamentoParagrafos.htm

A interpretação do arquivo HTML apresentado no Código-Fonte 6:3 pode ser visualizada na Figura 6-3.

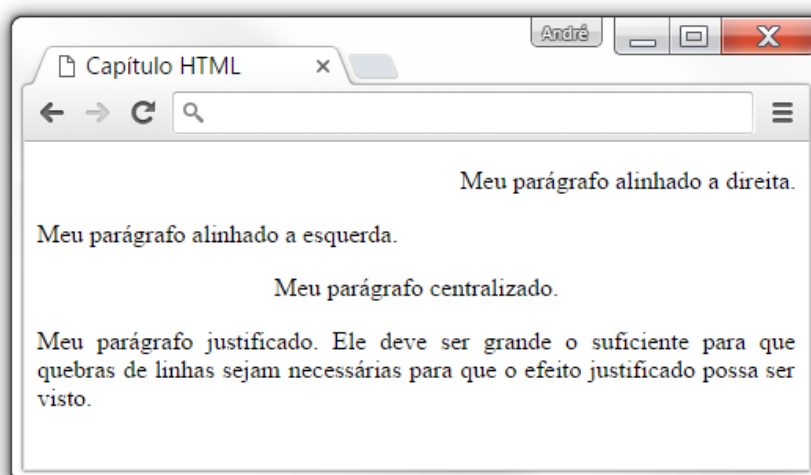


Figura 6-3. Diferentes tipos de alinhamentos de parágrafos via CSS

É importante reforçar que as quebras de linhas que ocorrem no parágrafo alinhado de forma justificada, nas linhas 13 e 14, não tem relação com as quebras de linha apresentadas na Figura 6-3. Quebras de linhas realizadas com a tecla `<ENTER>` no arquivo HTML são ignoradas, sendo as quebras realizadas somente quando o texto do parágrafo atinge a lateral direita da janela do navegador neste exemplo.

6.2. Agrupamento de elementos (trechos de textos)

A marcação `` permite agrupar elementos em uma única marcação, que podem ser formatados de uma única vez, tendo suas características repassadas para todos os elementos do grupo, apresentados entre a abertura e o fechamento da marcação.

Grande parte do uso desta marcação é na formatação de trechos de textos dentro de parágrafos, de forma a diferenciar alguma palavra ou frase como mostra o exemplo do Código-Fonte 6:4.

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     <p>
10       Este parágrafo possui uma
11       <span class='textoEstilo1'>parte de seu texto</span>
12       formatado utilizando a marcação span.
13     </p>
14   </body>
15 </html>
```

Código-Fonte 6:4. textosTrechosMarcados.htm

Para que essa marcação `` funcione adequadamente, é necessário que exista uma formatação CSS no arquivo de estilos chamada `.textoEstilo1`. Observe ainda que o nome do estilo não precisa apresentar o caractere ponto (.) quando vinculado ao atributo `class` nas marcações HTML. Uma opção de formatação para esse estilo pode ser visualizada no código-fonte a seguir, que deve fazer parte do arquivo CSS.

```
.textoEstilo1
{
  color: #FF0000;
  font-size: 16;
  font-family: Arial;
}
```

O resultado da interpretação do Código-Fonte 6:4 integrado ao arquivo CSS pode ser visualizado na Figura 6-4.

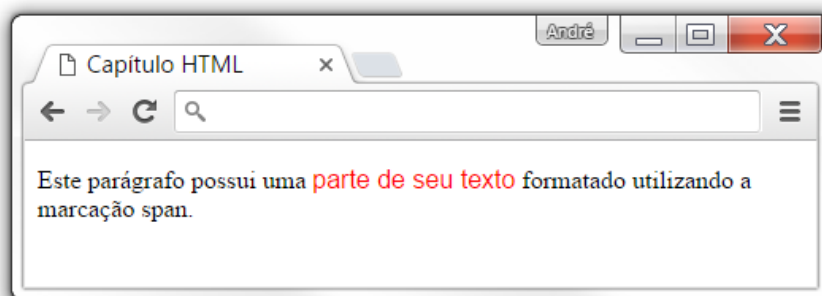


Figura 6-4. Trecho de parágrafo formatado isoladamente

O estilo `.textoEstilo1` no arquivo CSS apresenta algumas novas propriedades do CSS para você. A primeira é o atributo `color`, que por meio de um valor hexadecimal dos canais RGB representa uma cor para ser aplicada nos elementos vinculados ao

estilo. Já o atributo `font-size` representa o tamanho da fonte de texto, e `font-family` o tipo da fonte de texto.

DICA: Os exemplos completos apresentados neste livro podem ser baixados na internet. Acesse o site da Softblue, na área do e-book, para ter acesso aos códigos-fontes.

6.3. Negrito, itálico, sublinhado e riscado

Quando estamos tratando de textos, existem formatações bastante familiares que já conhecemos dos editores de textos. As tradicionais formatações de negrito, itálico, sublinhado e riscado podem ser realizadas também por meio das seguintes marcações HTML:

- ``: formata o texto em negrito.
- `<i>`: formata o texto em itálico.
- `<u>`: formata o texto em sublinhado.
- `<s>`: formata o texto em riscado.

A seguir, alguns exemplos das marcações apresentadas anteriormente:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     <b>Negrito</b>
10     <i>Itálico</i>
11     <u>Sublinhado</u>
12     <s>Riscado</s>
13     Texto <b>misturando <i>estilos</i></b>
14   </body>
15 </html>
```

Código-Fonte 6:5. textosFormatacaoClassicaMarcacaoHtml.htm

O resultado da interpretação do Código-Fonte 6:5 por um navegador web pode ser visualizado na Figura 6-5.



Figura 6-5. Textos com formatações clássicas via marcações HTML

Uma dúvida bastante recorrente entre os estudantes de HTML e CSS é: “*existe a possibilidade de formatar os efeitos de negrito, itálico, sublinhado ou riscado utilizando estilos de formatação?*”. Sim, é possível, e fica a critério de cada desenvolvedor optar pelo uso de marcações HTML ou estilos CSS para fazer isso. Caso o conteúdo formatado seja um texto que deva ter maior relevância, existem marcações específicas para essas formatações, que serão vistas no próximo tópico. Mas por enquanto vamos ilustrar o uso de estilos para realizar formatações deste tipo. Para isso, considere os novos estilos no arquivo CSS apresentados a seguir.

```
.formatoNegrito
{
  font-weight: bold;
}

.formatoItalico
{
  font-style: italic;
}

.formatoSublinhado
{
  text-decoration: underline;
}

.formatoRiscado
{
  text-decoration: line-through;
}
```

O estilo `.formatoNegrito` utiliza a propriedade `font-weight` definida como `bold` para formatar em negrito. Já o estilo `.formatoItalico` utiliza a propriedade `font-style` definida como `italic` para formatar em itálico. Para o efeito sublinhado, o estilo `.formatoSublinhado` define a propriedade `text-decoration` como `underline` para realizar o efeito, e para textos sublinhados, a mesma propriedade `text-decoration` é utilizada, mas desta vez com o valor `line-through` definido. Para fazer o uso desses novos estilos considere o arquivo HTML escrito com as marcações `` da seguinte forma:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
```

```

03 <head>
04   <meta charset="utf-8">
05   <title>Capítulo HTML</title>
06   <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07 </head>
08 <body>
09   <span class="formatoNegrito">Negrito</span>
10   <span class="formatoItalico">Itálico</span>
11   <span class="formatoSublinhado">Sublinhado</span>
12   <span class="formatoRiscado">Riscado</span>
13   Texto <span class="formatoNegrito">misturando
14   <span class="formatoItalico">estilos</span></span>
15 </body>
16 </html>

```

Código-Fonte 6:6. textosFormatacaoClassicaEstiloCss.htm

O resultado da interpretação do Código-Fonte 6:6 por um navegador web é exatamente o mesmo já apresentado na Figura 6-5. A opção entre utilizar as marcações HTML ou estilos CSS para estas propriedades de formatação é sua, não há preferência. Contudo, se o conteúdo marcado apresentar maior relevância em sua página, você vai conhecer no próximo tópico novas marcações com funções semânticas que devem ser utilizadas nessas situações.

6.4. Textos com maior relevância semântica

Para realizar marcações com ênfase semântica, dando maior relevância e importância para determinados trechos de textos, podem ser utilizadas algumas marcações específicas, como a ``. Esta marcação também era utilizada para formatar negrito em textos até a versão 4 do HTML. A partir da versão 5 ela representa importância nos itens marcados. Ela continua aplicando o negrito como antes, mas sua função mais importante é em relação à relevância do trecho de texto marcado.

Por exemplo: considere um site que apresenta um e-book de HTML como produto principal a ser divulgado. Pode ser interessante nesta página ressaltar as palavras **e-book** e **HTML** de alguma forma, para que os algoritmos dos buscadores da web (Yahoo, Google e outros) possam priorizar essas palavras nessa página. Assim, as chances da página aparecer nos primeiros resultados dos buscadores em casos de busca por termos como **e-book de HTML** aumentam potencialmente. Esta página poderia ser construída como mostra o código-fonte a seguir:

```

01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     Este texto é sobre <strong>HTML</strong>,

```

```

10   cujo conteúdo é apresentado em formato de <strong>e-book</strong>.
11 </body>
12 </html>

```

Código-Fonte 6:7. textosFormatacaoClassicaMarcacaoHtmlRelevanciaSemantica.htm

O resultado da interpretação do Código-Fonte 6:7 por um navegador web pode ser visualizado na Figura 6-6.

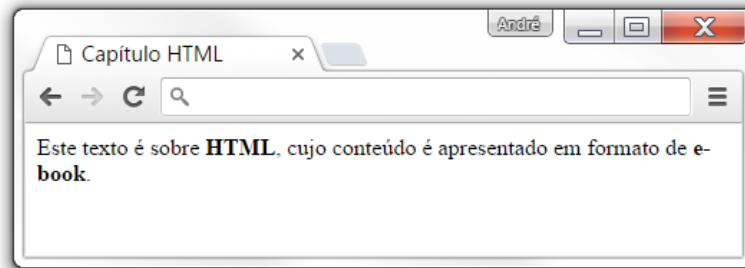


Figura 6-6. Textos com formatações clássicas via marcações HTML com relevância semântica

A marcação `` apresenta funções semânticas. Mas como consequência, torna o texto negrito. Caso exista a necessidade de cancelar o efeito de negrito, mas ainda assim utilizar esta marcação exclusivamente para fins semânticos, é possível criar um estilo que cancele a formatação de negrito desta marcação, por meio da propriedade `font-weight`, definida com o valor `normal`, como mostra o estilo adicionado no arquivo CSS apresentado a seguir:

```

.formatoNeutro
{
    font-weight: normal;
}

```

A utilização do estilo que remove o negrito pode ser visualizado no arquivo HTML apresentado a seguir:

```

01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     Este texto é sobre <strong class="formatoNeutro">HTML</strong>.
10   </body>
11 </html>

```

Código-Fonte 6:8. textosMarcacaoStrongSemFormatacao.htm

O resultado do uso da marcação `` com o estilo `.formatoNeutro` pode ser visualizado na Figura 6-7.

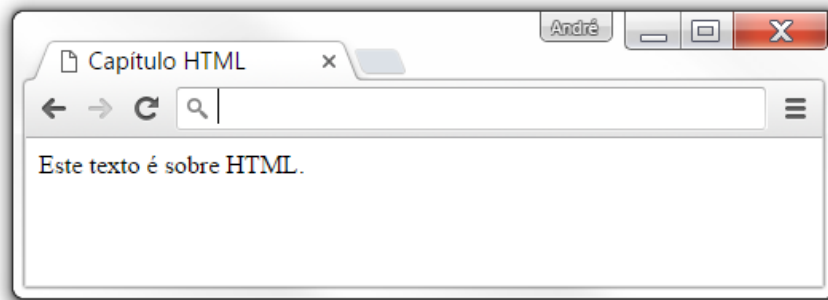


Figura 6-7. Texto marcado com relevância sem formatação

Existem ainda outras marcações HTML voltadas para formatação de textos importantes, com relevâncias semânticas específicas, mas que apresentem outros formatos de citação, populares em páginas web. São elas:

- ``: marcação para textos enfatizados
- `<code>`: marcação para códigos-fonte
- `<samp>`: marcação de resultados apresentados por programas de computador
- `<kbd>`: marcação de entradas de teclado
- `<var>`: marcação de definição de variável

Um exemplo geral com o uso das marcações de textos relevantes, aplicados adequadamente em um exemplo fictício que apresente todos esses recursos, pode ser observado no código-fonte apresentado a seguir:

```

01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     <p>
10       Esta página apresenta trechos de códigos do <strong>algoritmo
11       de busca binária</strong>.
12     </p>
13     <p>
14       O nome da busca vem do inglês <em>binary search algorithm</em>.
15     </p>
16     <code>
17     <pre>
18       public static int buscaBinaria(int[] array, int valor){
19         int inicio = 0;
20         int fim = array.length-1;
21
22         while(inicio <= fim){
23           int meio = (inicio+fim)/2;
24
25           if(array[meio] == valor){
26             return meio;
27           }
28           if(valor > array[meio]){
29             inicio = meio+1;

```

```

30         } else {
31             fim = meio-1;
32         }
33     }
34     return -1;
35 }
36 </pre>
37 </code>
38 <p>
39     O algoritmo inicia criando as variáveis <var>inicio</var>
40     e <var>fim</var>. O programa iniciará quando o usuário
41     pressionar a tecla <kbd>ENTER</kbd>, e apresentará o
42     resultado no final, que pode ser <samp>5 BUSCAS</samp>.
43 </p>
44 </body>
45 </html>

```

Código-Fonte 6:9. textosMarcacoesMaiorRelevancia.htm

O resultado da interpretação do Código-Fonte 6:9 por um navegador web pode ser visualizado na Figura 6-8.

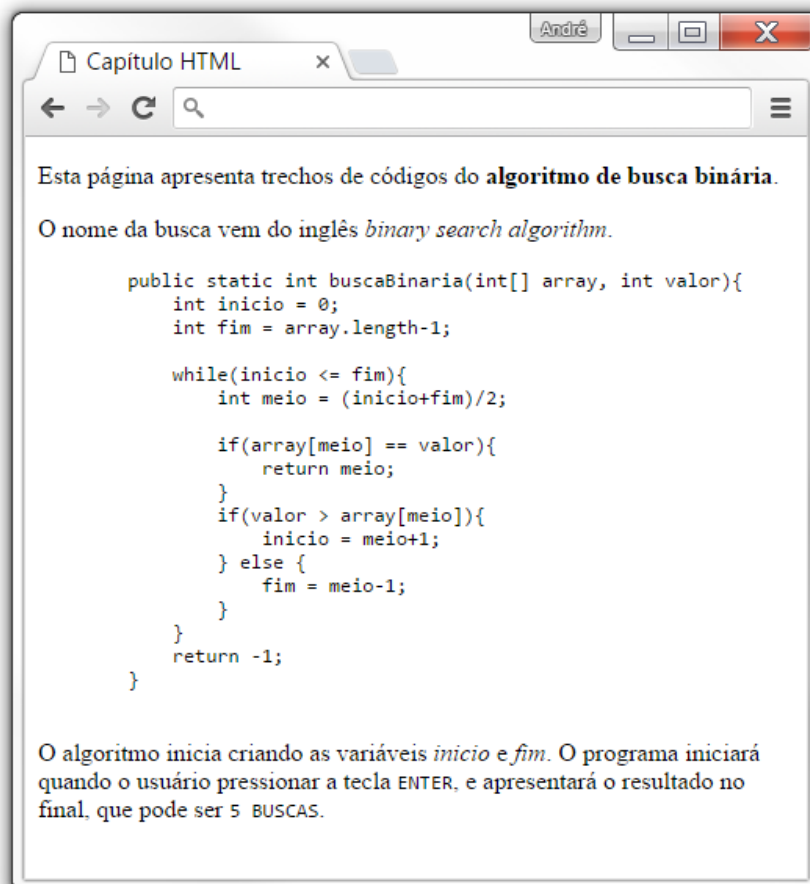


Figura 6-8. Texto marcado com relevância em diferentes situações

Observe no Código-Fonte 6:9 a utilização das marcações. A primeira utilizada foi a marcação `` entre as linhas 10 e 11 fornecendo maior importância ao texto `algoritmo de busca binária`, pois é o tema principal desta página fictícia. Essas palavras

são marcadas para que tenham maior relevância nos sites de busca, quando um estudante de programação, por exemplo, buscar pelo termo `algoritmo de busca binária` em um buscador da web.

Em seguida, uma menção ao nome em inglês do algoritmo é realizada. Essa menção poderia ser marcada com a marcação ``, mas neste caso optou-se pela marcação de citação de texto `` na linha 14. Esse recurso também apresenta funções semânticas, assim como as outras que serão detalhadas ainda neste tópico.

A página apresenta um trecho de código de computador para o usuário. Códigos-fonte devem ser apresentados por meio da marcação `<code>`, aberta na linha 16 e fechada na linha 37. Mas neste ponto um problema surge: como garantir que as quebras de linha do código-fonte sejam interpretadas pelos navegadores, para evitar que o código-fonte apareça em uma única linha? Será necessário incluir a marcação `
` no final de cada linha?

Para forçar a interpretação das quebras de linhas no HTML, foi utilizada a marcação `<pre>`, aberta na linha 17 e fechada na linha 36, dentro da marcação `<code>`. A marcação `<pre>` pode ser utilizada tanto com a marcação `<code>` como em conjunto com outras. Sua funcionalidade é apresentar textos pré-formatados, forçando a impressão de quebras de linhas e espaços em branco existentes nos textos delimitados por esta marcação.

Na sequência, a marcação `<var>` é utilizada duas vezes: na linha 39 para representar a variável `inicio` do algoritmo e na linha 40 para a variável `fim`. A página ainda menciona uma tecla que deve ser pressionada na linha 41, apresentada por meio da marcação `<kbd>`, e o resultado que o programa fictício imprime para o usuário é exibido na linha 42, destacado pela marcação `<samp>`.

Reforçando novamente: todas essas marcações apresentam tanto funções de formatação quanto semânticas, categorizando seus respectivos conteúdos adequadamente junto aos robôs de buscas dos principais buscadores da internet. Caso você queira cancelar suas respectivas formatações, será necessário utilizar o CSS como já visto anteriormente.

6.5. Texto subscrito e sobrescrito

Existem ainda duas marcações de textos bastante especiais, que formatam seus respectivos conteúdos acima ou abaixo do alinhamento de texto padrão. São as marcações `<sub>` e `<sup>`.

A marcação `<sub>` apresenta o texto delimitado por ela alinhado abaixo do texto padrão, enquanto a marcação `<sup>` tem um efeito similar, mas alinhando o texto acima

do alinhamento padrão. Esses recursos são popularmente utilizados em representação de fórmulas científicas e matemáticas, mas podem ser utilizados em diversas outras situações. Um exemplo de uso dessas marcações pode ser visualizado no código-fonte apresentado a seguir:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     <p>
10       A fórmula da água é: H<sub>2</sub>O
11     </p>
12     <p>
13       A fórmula da hipotenusa é: a<sup>2</sup> = b<sup>2</sup> + c<sup>2</sup>
14     </p>
15   </body>
16 </html>
```

Código-Fonte 6:10. textosAlinhamentoAcimaAbaixo.htm

O resultado da interpretação do Código-Fonte 6:10 por um navegador web pode ser visualizado na Figura 6-9.

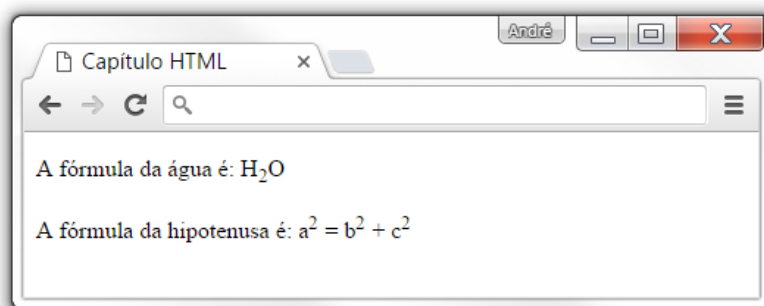


Figura 6-9. Textos com alinhamentos acima e abaixo do padrão

Observe no Código-Fonte 6:10 o uso da marcação `<sub>` na linha 10 para alinhar o número 2 abaixo do alinhamento padrão do texto. Já na linha 13 é utilizada a marcação `<sup>` para alinhar, novamente, o número 2 na fórmula matemática de cálculo da hipotenusa.

6.6. Títulos em conteúdos de texto

Para representar títulos de textos, a marcação `<h>` foi criada, apresentando vários níveis de títulos e subtítulos, iniciando em `<h1>`. Caso existam subtítulos, os mesmos podem ser representados com os níveis posteriores `<h2>`, `<h3>`, `<h4>`, `<h5>` e `<h6>`.

Um exemplo do uso de títulos em textos pode ser observado no código-fonte a seguir:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     <h1>Titanic</h1>
10     <h2>O maior navio do mundo</h2>
11     <p>
12       O Titanic foi construído para ser o maior do mundo.
13     </p>
14     <p>
15       Este navio ficou famoso por causa de seu acidente na viagem inaugural.
16     </p>
17   </body>
18 </html>
```

Código-Fonte 6:11. textosTitulos.htm

O resultado da interpretação do Código-Fonte 6:11 por um navegador web pode ser visualizado na Figura 6-10.

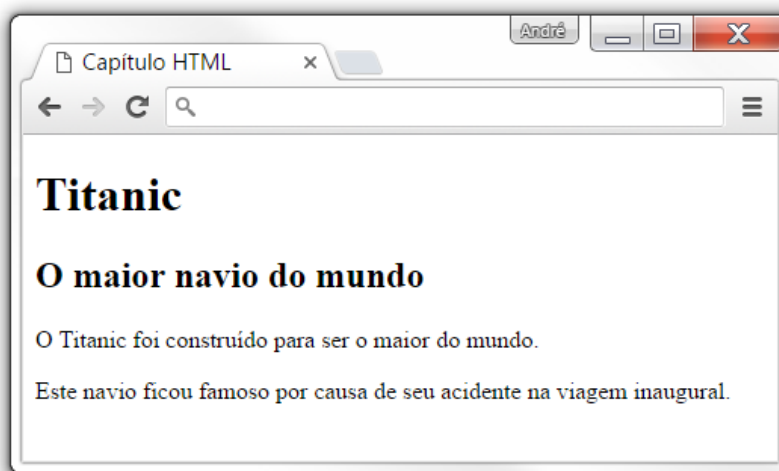


Figura 6-10. Texto com título principal e subtítulo

As marcações de título de textos aumentam a relevância dos textos apresentados no conteúdo da página. Ainda, essas marcações geralmente são utilizadas com outras para compor um artigo, que serão vistas no próximo tópico.

7. Trabalhando com artigos e seções

Os conteúdos das páginas na internet podem (e devem) ser construídos, estruturalmente, como artigos HTML. Isso não quer dizer que o seu conteúdo precise ser um artigo formal, um artigo científico, mas sim que o mesmo faça uso das marcações `<article>` e `<header>` justamente para organizar o texto do seu conteúdo semanticamente.

Para ficar mais claro o uso das marcações `<article>` e `<header>` e os conceitos abordados neste tópico, considere uma página web cujo conteúdo seja sobre o navio Titanic. Essa página geralmente vai conter um título e um texto. Dessa forma, a mesma poderia ser construída como mostra o código-fonte apresentado a seguir:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     <article>
10       <header>
11         <h1>Titanic</h1>
12         <h2>O maior navio do mundo</h2>
13       </header>
14       <p>
15         O Titanic foi construído para ser o maior do mundo.
16       </p>
17       <p>
18         Este navio ficou famoso por causa de seu acidente na viagem inaugural.
19       </p>
20     </article>
21   </body>
22 </html>
```

Código-Fonte 7:1. conteudoOrganizadoComoArtigo.htm

O resultado da interpretação do Código-Fonte 7:1 por um navegador web é o mesmo já apresentado na Figura 6-10. A utilização de marcações semânticas apresentadas neste exemplo não altera o resultado visual do conteúdo em questão.

Analisando o Código-Fonte 7:1, observe que o conteúdo que faz referência ao assunto da página (Titanic) foi separado da estrutura da página. Desta forma, o conteúdo foi apresentado entre a abertura da marcação `<article>` na linha 9, e seu fechamento na linha 20, informando que um artigo será apresentado entre essas linhas. Para os buscadores da internet e outros observadores do arquivo, é importante saber que o conteúdo relevante está entre essas linhas (9 a 20) e que as demais linhas tratam

de questões estruturais e configurações diversas, não necessariamente atreladas ao mesmo assunto.

Outra observação importante é o fato de que o resultado apresentado para o usuário na janela do navegador é exatamente o mesmo visto, anteriormente, na Figura 6-10. Aplicar as boas práticas de marcações semânticas nos conteúdos HTML não necessariamente causa impacto visual na janela do navegador, mas sempre melhora a organização do conteúdo para ser mais bem processado pelos algoritmos de buscas da internet.

Dentro da marcação `<article>` é sugerido abrir um cabeçalho do artigo, por meio da marcação `<header>` aberta na linha 10 e fechada na linha 13. Não confunda a marcação `<header>` (cabeçalho de artigo) com a marcação `<head>` (cabeçalho da página HTML).

É dentro da marcação `<header>` que os títulos do artigo devem ser informados, a fim de que tenham ainda maior relevância pelos algoritmos dos buscadores da internet. Após isso, o cabeçalho `<header>` pode ser encerrado e, entre o fechamento da marcação `</header>` e o fechamento da marcação `</article>`, o conteúdo do artigo (da página) deve ser apresentado.

Dessa forma, eventuais menus do site (tema que será abordado no decorrer deste e-book) ou outras estruturas não terão seus conteúdos misturados com o tema principal da página, além de também possuírem marcações específicas para serem organizadas, que serão abordadas mais adiante.

7.1. Organizando artigos em seções

Uma página pode conter um ou mais artigos. Neste caso, é sugerido que os artigos sejam organizados por seções.

Suponha um site de notícias, que vai apresentar em sua página principal um resumo do dia, com artigos de esporte, negócios e entretenimento. Neste caso é recomendado criar uma seção para cada tema que seja diferente de outro e, dentro de cada seção, apresentar os artigos relacionados àquela seção.

Seções podem ser criadas a partir da marcação `<section>`, sendo que cada seção pode apresentar um ou mais artigos, geralmente de assuntos relacionados entre si de alguma forma. Por exemplo, considere o código-fonte apresentado a seguir:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
```

```

08 <body>
09   <section>
10     <header>
11       <h1>Esportes</h1>
12     </header>
13     <article>
14       <header>
15         <h2>F1</h2>
16       </header>
17       <p>A temporada de Fórmula 1 começou.</p>
18     </article>
19     <article>
20       <header>
21         <h2>Futebol</h2>
22       </header>
23       <p>Veja os gols da rodada.</p>
24     </article>
25   </section>
26   <section>
27     <header>
28       <h1>Economia</h1>
29     </header>
30     <article>
31       <header>
32         <h2>Inflação</h2>
33       </header>
34       <p>A inflação permanece igual ao mês anterior.</p>
35     </article>
36     <article>
37       <header>
38         <h2>Bolsa de Valores</h2>
39       </header>
40       <p>2% foi o aumento na bolsa hoje no Brasil.</p>
41     </article>
42   </section>
43 </body>
44 </html>

```

Código-Fonte 7:2. artigosOrganizadosPorSecao.htm

O resultado da interpretação do Código-Fonte 7:2 por um navegador web pode ser visualizado na Figura 7-1.

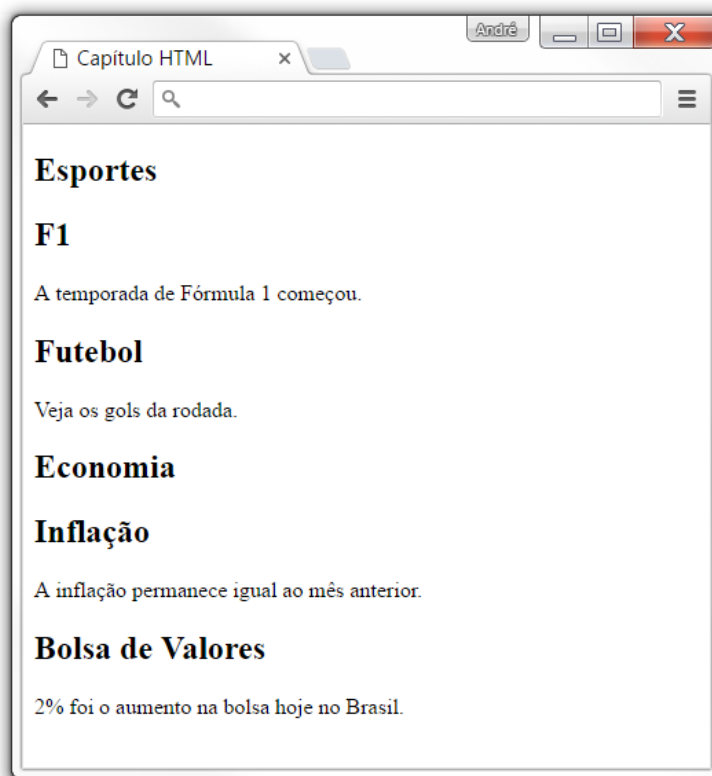


Figura 7-1. Artigos organizados por seção

No Código-Fonte 7:2 foi apresentado um exemplo fictício de página que mostra notícias. Basicamente, as notícias desta página são de duas áreas distintas: esportes e economia. A seção de esportes é aberta na linha 9 e fechada na linha 25, e entre essas linhas é que são apresentados os artigos de notícias relacionados a esportes. Entre as linhas 13 e 18 temos uma notícia em formato de artigo sobre Fórmula 1, e entre as linhas 19 e 24 outra notícia, sobre futebol. Observe que foi incluído um cabeçalho para a seção (`<header>`), entre as linhas 10 e 12, com o título informado na marcação `<h1>`. É perfeitamente possível associar um título a uma seção. Já os cabeçalhos (`<header>`) de cada artigo utilizaram o título na marcação `<h2>`. Essa é uma forma de organizar hierarquicamente o conteúdo e os títulos apresentados entre os diferentes elementos da seção. Não é necessário seguir essa ordem, ela é apenas uma sugestão.

A outra seção do exemplo fictício apresenta notícias do mundo da economia. A seção é apresentada entre as linhas 26 e 42, também apresenta um cabeçalho e título próprio da seção entre as linhas 27 e 29 e duas notícias de economia no formato de artigos, com seus respectivos títulos e conteúdos.

Uma página pode conter quantas seções e artigos o seu autor desejar.

7.2. Trabalhando com links (pontos de acesso)

Agora que você já viu várias marcações de textos, chegou a hora de conhecer a criação de links (pontos de acesso) utilizando o HTML. A marcação `<a>` permite criar links entre páginas na internet. Esta marcação não é restrita a apenas textos. Funciona normalmente com imagens também, mas isto será abordado mais adiante.

A marcação `<a>` informa que o conteúdo entre sua abertura e fechamento, quando clicado, levará o acesso do usuário para algum endereço virtual específico. Já o endereço em questão deve ser indicado pelo atributo `href`, disponível para essa marcação. A seguir, alguns exemplos da utilização da marcação `<a>`.

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     <a href='http://www.google.com.br'>Clique</a> para acessar o Google.
10     <br>
11     <a href='marcacaoBody.htm'>Clique</A> para acessar outra página.
12   </body>
13 </html>
```

Código-Fonte 7.3. marcacaoA.htm

O resultado da interpretação do Código-Fonte 7:3 por um navegador web pode ser visualizado na Figura 7-2.

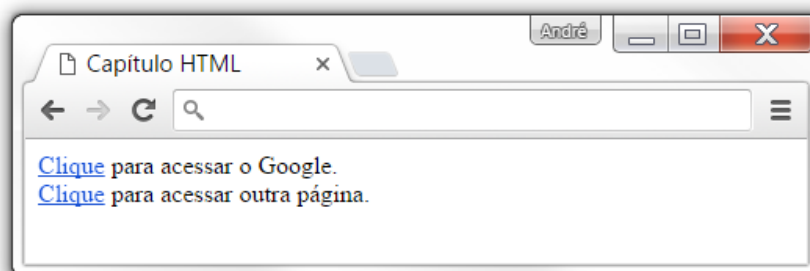


Figura 7-2. Página com links (pontos de acesso)

Observe o Código-Fonte 7:3. A marcação `<a>` é utilizada na linha 9 transformando a palavra `Clique` em um item que pode ser clicado na página para disparar uma ação, sendo a ação, neste caso, o acesso ao endereço da página do Google, informado no atributo `href`, com o valor `http://www.google.com.br`.

Outro exemplo de utilização da marcação `<a>` pode ser conferida na linha 11. Neste caso, o acesso direciona o usuário para outra página na mesma pasta onde o arquivo atual se encontra. Ou seja, para funcionar adequadamente, é necessário constar na

mesma pasta de seu arquivo HTML outro arquivo, chamado de `marcacaoBody.htm`, abordado no começo deste livro. Dessa forma, o acesso em questão direcionará o usuário para este outro arquivo quando ele clicar sobre a palavra `Clique` relacionada.

Quando um endereço virtual for utilizado para apontar para outro site, é importante que o endereço seja completo, começando pelos caracteres `http`. Quando o endereço for para outra página do mesmo site, mesmo domínio de internet, apenas o endereço de pastas é necessário.

8. Trabalhando com imagens

Até o momento muito já foi abordado sobre texto e suas principais marcações. Mas não é apenas de textos que as páginas web vivem e, por este motivo, imagens podem fazer parte do conteúdo dos sites.

A inserção simples de imagens pode ser realizada por meio da marcação ``, que pode apresentar dois atributos: `src`, com o endereço virtual do arquivo de imagem para ser exibido; e `alt`, com um texto que é apresentado na tela quando o usuário mantém o cursor do mouse sobre a imagem por alguns segundos, ou até mesmo em casos onde a imagem não carrega por algum motivo. O código-fonte a seguir apresenta um uso simples desta marcação:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     
11   </body>
12 </html>
```

Código-Fonte 8:1. marcacaoImg.htm

O resultado da interpretação do Código-Fonte 8:1 por um navegador web pode ser visualizado na Figura 8-1.

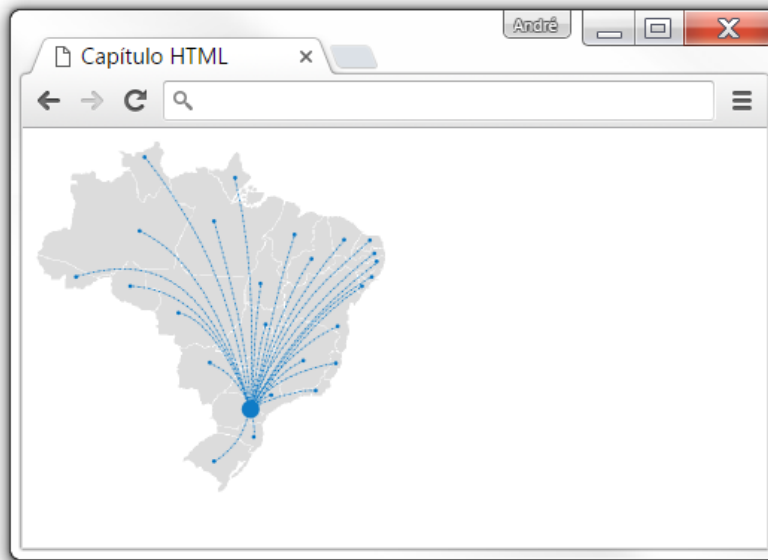


Figura 8-1. Página com imagem

Apesar da inserção de imagens por meio da marcação `` ser bastante simples, ela por si só não atende todos os critérios semânticos atualmente recomendados na internet. Para que atenda, é necessário utilizar essa marcação em conjunto com outras quando sua imagem apresentar conteúdo relevante. As outras marcações são a `<figure>` e `<figcaption>`.

A marcação `<figcaption>` define uma legenda de imagem. Essa legenda geralmente descreve uma imagem e serve para ser analisada pelos algoritmos de buscas para apresentar a sua imagem pelos buscadores, associando-a com as palavras existentes na legenda. Contudo, o uso de `<figcaption>` isoladamente não vincula a legenda com qualquer imagem. Para isso, a marcação `<figure>` deve ser utilizada, devendo as marcações `` e `<figcaption>` serem apresentadas dentro da marcação `<figure>`, como ilustra o exemplo construído a seguir:

```

01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     <figure>
10       
12       <figcaption>Softblue: de Curitiba para todo o Brasil.</figcaption>
13     </figure>
14   </body>
15 </html>

```

Código-Fonte 8.2. marcacaoFigure.htm

O resultado da interpretação do Código-Fonte 8:2 por um navegador web pode ser visualizado na Figura 8-2.



Figura 8-2. Página com imagem marcada de conteúdo relevante e legenda

Observe a marcação `<figure>`, aberta na linha 9 e fechada na linha 13. Entre essas linhas, a marcação `` na linha 10 apresenta uma imagem que fará parte da marcação `<figure>`, pois está entre a abertura e fechamento dessa marcação. Ainda dentro da marcação `<figure>`, na linha 12, a marcação `<figcaption>` apresenta uma legenda para a imagem. Como ambas as marcações `` e `<figcaption>` estão dentro da mesma marcação `<figure>`, isso caracteriza o relacionamento entre elas, associando a legenda em questão, para a imagem informada no conjunto.

Uma dúvida bastante comum sobre este processo é: qual a diferença semântica entre o texto apresentado no atributo `alt` da marcação `` e o texto da legenda da imagem, representado pela marcação `<figcaption>`? A diferença é que o texto apresentado no atributo `alt` da marcação `` deve descrever o conteúdo da imagem em si, enquanto que a legenda da imagem é um texto de livre expressão, construído sobre qualquer assunto relacionado à imagem.

Por exemplo, suponha a pintura Mona Lisa, de Leonardo da Vinci. O nome da pintura, *Mona Lisa*, deve ser informado no atributo `alt` da marcação ``, enquanto que a legenda da imagem, informada na marcação `<figcaption>`, poderia ser *Famosa pintura de Leonardo da Vinci*.

8.1. Trabalhando com vídeos

A inserção de vídeos em páginas HTML também é permitida, podendo ser realizada diretamente por meio da marcação `<video>`, como mostra o código-fonte apresentado a seguir:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     <video width="320" height="180" controls>
10       <source src="seuVideo.mp4" type="video/mp4">
11       Seu navegador não suporta vídeos.
12     </video>
13   </body>
14 </html>
```

Código-Fonte 8.3. marcacaoVideo.htm

O resultado da interpretação do Código-Fonte 8:3 por um navegador web pode ser visualizado na Figura 8-3.

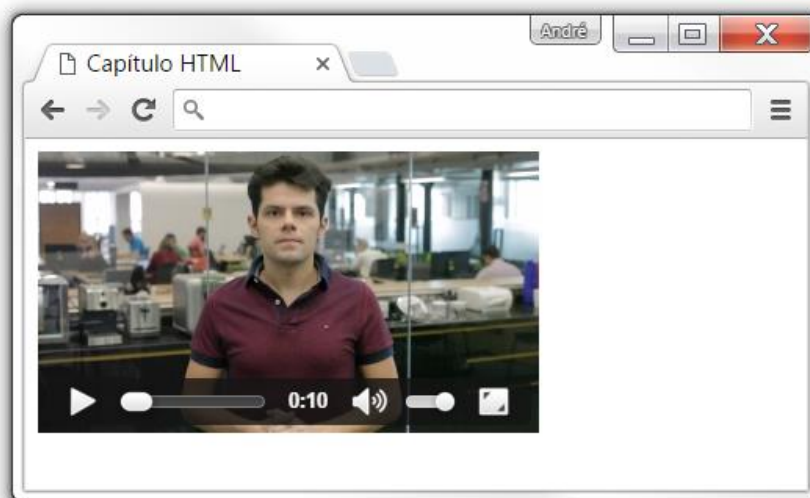


Figura 8-3. Página com vídeo

A utilização da marcação `<video>` no Código-Fonte 8:3 é realizada entre as linhas 9 e 12. Observe que a abertura da marcação apresenta alguns atributos, como `width`, indicando em pixels a largura que o player de vídeo deve ser apresentado na tela; `height`, com a altura em pixels do player; e a indicação `controls`, informando que os botões de interação com o vídeo devem ser apresentados.

Dentro da marcação `<video>`, mais precisamente na linha 10, a marcação `<source>` deve ser inserida, informando em seu atributo `src` o endereço virtual do vídeo a ser

executado, e o atributo `type` com a definição do tipo de vídeo. Observe que é necessário que o caminho do vídeo aponte diretamente para o arquivo do vídeo, e não para uma página que tenha um vídeo, como páginas do YouTube por exemplo.

Caso você queira inserir um vídeo do YouTube, acesse a página do vídeo em questão e obtenha o código-fonte HTML para embutir o vídeo em sua página, fornecido pelo próprio YouTube na seção de compartilhamento do vídeo. Lembrando que esse código só será fornecido caso o proprietário do vídeo autorize essa liberação ao cadastrar o vídeo.

Ainda, é possível definir uma mensagem de texto caso o navegador do usuário não suporte a marcação `<video>`, que é nova no HTML5, ou caso não suporte o tipo de vídeo informado. Neste caso, a frase deve ser inserida entre a abertura e fechamento da marcação `<video>`, como realizado na linha 11.

9. Trabalhando com listas

Outro recurso existente nos principais programas de edição de texto que também pode ser utilizado no HTML é a criação e apresentação de listas de conteúdos, numeradas ou não.

No HTML, existem basicamente dois tipos de listas: as não numeradas e as numeradas.

9.1. Listas não numeradas

O tipo de lista não numerada (*Unordered List*) pode ser construído a partir da marcação ``. Dentro desta marcação, inclua os itens da lista, utilizando para cada item a abertura da marcação ``, sigla do termo *List Item*. A marcação `` não precisa ser fechada. A seguir, um exemplo de código HTML com lista não numerada:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <ul>
09     <li> Item 1
10     <li> Item 2
11     <li> Item 3
12   </ul>
13 </body>
14 </html>
```

Código-Fonte 9:1. marcacaoUL.htm

O resultado da interpretação do Código-Fonte 9:1 por um navegador web pode ser visualizado na Figura 9-1.

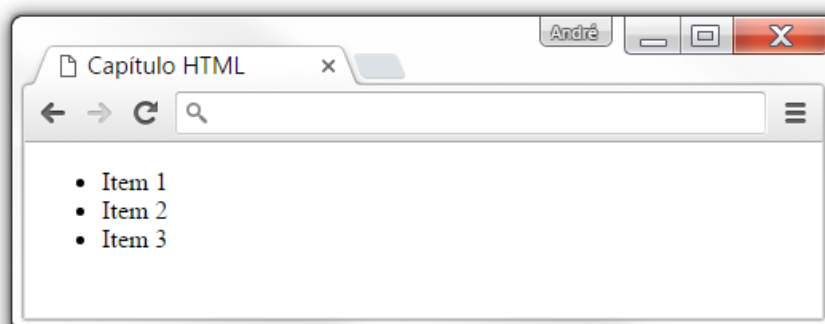


Figura 9-1. Página com lista sem numeração

Observe no Código-Fonte 9:1 que a lista não numerada é aberta na linha 8 por meio da marcação `` e fechada na linha 12. Todo o conteúdo presente entre essas linhas será considerado como conteúdo da lista. Entre essas linhas, ainda, três itens foram apresentados, cada um com sua respectiva marcação ``, que não precisa ser fechada. Como se trata de uma lista de itens, a quebra de linha ocorre automaticamente junto com a marcação ``.

9.2. Listas numeradas

O tipo de lista com numeração (*Ordered List*) pode ser construído a partir da marcação ``. Dentro desta marcação, inclua os itens da lista, utilizando para cada item a abertura da marcação ``, sigla do termo *List Item*. A marcação `` não precisa ser fechada. A seguir, um exemplo HTML com lista numerada:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <ol>
09     <li> Item 1
10     <li> Item 2
11     <li> Item 3
12   </ol>
13 </body>
14 </html>
```

Código-Fonte 9:2.marcacaoOl.htm

O resultado da interpretação do Código-Fonte 9:2 por um navegador web pode ser visualizado na Figura 9-2.

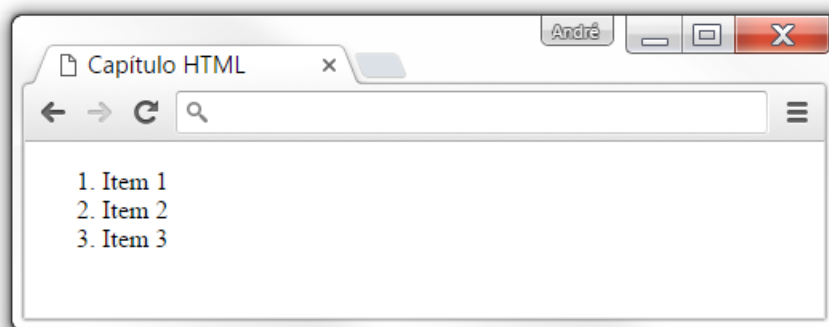


Figura 9-2. Página com lista numerada

Basicamente, o Código-Fonte 9:2 é idêntico ao Código-Fonte 9:1 apresentado anteriormente, com a única diferença na marcação de lista utilizada, trocando `` por ``, tanto em sua abertura quanto no fechamento, nas linhas 8 e 12.

10. Trabalhando com tabelas

Outro recurso que permite organizar dados em massa em um site são as tabelas, recurso que já estamos acostumados a utilizar por causa dos programas de edição de texto e programas de planilhas.

Por meio delas, é possível especificar células de tamanhos específicos e linhas e colunas para exibir listas de dados em formato de tabela. Contudo, no HTML, a construção de tabelas é um tema mais avançado, pois você precisará codificar cada instrução da estrutura (abertura da tabela, de cada linha, de cada coluna).

Para manipular uma tabela em HTML, é necessário ter conhecimento de cinco novas marcações HTML, que serão utilizadas de forma integrada umas com as outras. São elas:

- `<table>`: determina o início e o fim de uma tabela.
- `<caption>`: determina a legenda da tabela.
- `<tr>`: determina o início e o fim de uma linha da tabela.
- `<th>`: determina o início e o fim das células do cabeçalho da tabela.
- `<td>`: determina o início e o fim de uma célula da tabela.

Ainda, existem dois atributos que serão abordados neste tópico, relacionados ao recurso de mesclar células. São eles:

- `colspan`: define quantas colunas a célula deve ocupar, para funções de mesclar células.
- `rowspan`: define quantas linhas a célula deve ocupar, para funções de mesclar células.

Não se preocupe se essas marcações parecem confusas neste momento: muitas marcações e atributos novos ficarão mais claros a partir dos exemplos de uso. Veja a seguir um exemplo completo da utilização destas marcações para montar uma tabela, que será explicado detalhadamente após a apresentação do exemplo:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     <table border=1>
10       <caption>Dados do relatório</caption>
11       <tr>
12         <th>Coluna 1</th>
13         <th>Coluna 2</th>
14         <th>Coluna 3</th>
15       </tr>
16       <tr>
```

```

17     <td rowspan='2'>A</td>
18     <td>B</td>
19     <td>C</td>
20 </tr>
21 <tr>
22     <td>E</td>
23     <td>F</td>
24 </tr>
25 <tr>
26     <td>G</td>
27     <td colspan='2'>H</td>
28 </tr>
29 </table>
30 </body>
31 </html>

```

Código-Fonte 10:1. marcacaoTableCaptionTrThTd.htm

O resultado da interpretação do Código-Fonte 10:1 por um navegador web pode ser visualizado na Figura 10-1.

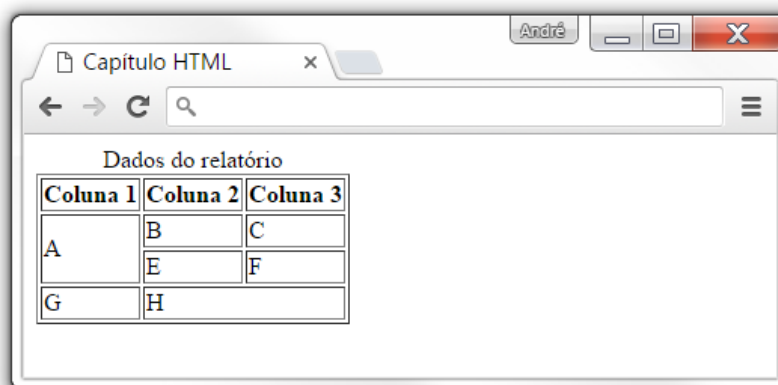


Figura 10-1. Tabela construída em HTML

Para compreender a construção do Código-Fonte 10:1, é necessária uma explicação detalhada sobre suas principais linhas. A construção da tabela inicia na linha 9 e encerra na linha 29, por meio do uso da marcação `<table>`. Todo o conteúdo que estiver entre essas linhas fará parte do conteúdo da tabela. Essa marcação utilizou o atributo `border` na marcação `<table>` com o valor `1` para que as bordas das células da tabela sejam forçadas a aparecer na tela, para fins didáticos.

Dentro da tabela a primeira informação é sobre a legenda da mesma, apresentada na linha 10, informada por meio da marcação `<caption>`. Essa marcação exibe um título para a tabela na tela do navegador e também apresenta funções semânticas, associando a tabela a suas palavras-chave para os algoritmos de busca.

Na sequência, a primeira linha da tabela é construída por meio da marcação `<tr>` apresentada com abertura na linha 11 e fechamento na linha 15. Todo o conteúdo entre essas linhas está relacionado apenas à primeira linha da tabela.

As colunas da primeira linha da tabela podem ser construídas por meio da marcação `<th>`, que representa a marcação de cabeçalhos, de título das colunas. Logicamente, a construção de colunas deve ser realizada dentro de uma linha da tabela, ou seja, entre a abertura e fechamento da marcação `<tr>`, que representa uma linha.

A marcação `<th>`, por considerar o conteúdo da célula como título de uma coluna da tabela, formata o conteúdo com negrito. Mas essa formatação, bem como qualquer outra da tabela, pode ser alterada utilizando CSS, associando um estilo à célula em questão.

A primeira célula da primeira linha é construída na linha 12, com o título `Coluna 1`. Em seguida a segunda célula da primeira linha é construída na linha 13 e a terceira na linha 14.

A segunda linha da tabela é apresentada entre as linhas 16 e 20. Essa linha também apresenta três células, construídas respectivamente nas linhas 17, 18 e 19. As células foram construídas por meio da marcação `<td>`, que é similar a marcação `<th>`, mas é voltada para células que não apresentem o título das colunas, ou seja, é muito comum o uso de `<th>` apenas na primeira linha da tabela e `<td>` nas demais.

Outra novidade ainda na segunda linha é a utilização do atributo `rowspan` na primeira célula desta linha (linha 17). O atributo `rowspan` mescla a célula em questão com as próximas linhas da tabela. Neste caso, ao definir `rowspan` com o valor 2, a célula em questão passa a ocupar duas linhas na tabela.

Seguindo com a construção da tabela, a próxima linha é construída entre as linhas 21 e 24 do código HTML. Observe que essa linha apresenta apenas duas células, apresentadas nas linhas 22 e 23. Isso ocorre por causa do seguinte motivo: se a tabela apresenta três colunas, mas a célula da linha acima ocupa duas linhas por causa do uso do atributo `rowspan` (linha 17), então nesta linha da tabela falta apenas definir as duas células restantes, pois a primeira foi mesclada com a célula da linha anterior.

Por último, a quarta linha da tabela é construída entre as linhas 25 e 28. Essa linha pode definir novamente três colunas, mas define apenas duas, nas linhas 26 e 27. Em particular, a célula apresentada na linha 27 faz o uso do atributo `colspan`, que mescla a célula com outras da mesma linha. Neste caso, a segunda célula (linha 27) é definida para ocupar o espaço de duas colunas, portanto ela aparece mesclada no resultado final da página com o que seria a terceira célula desta linha da tabela.

11. Trabalhando com formulários

A melhor forma de se obter dados de um usuário na internet é por meio de formulários HTML. Geralmente os formulários são enviados para um servidor, após serem preenchidos, a fim de serem processados por alguma linguagem de programação web como PHP, Java ou outra.

O recurso de formulário envolve diversos tipos de interação, algumas de entrada de dados, através dos campos de dados ou de preenchimento, e outras de tomadas de ação, como por exemplo o botão que envia o formulário para processamento.

Existem algumas marcações ao redor deste assunto e vários atributos específicos, portanto vamos com calma: inicialmente serão apresentadas as marcações e atributos e, para esclarecer a forma de utilizá-los, posteriormente será construído um exemplo de formulário com explicações detalhadas de cada passo.

11.1. Formulário

A primeira marcação referente a formulários é a que define um. Todos os campos de um formulário devem estar declarados dentro da marcação `<form>`. Os principais atributos da marcação `<form>` são:

- **method**: define o tipo de envio dos dados do formulário para o servidor. As seguintes opções podem ser utilizadas:
 - **get**: transmite os dados na própria URL de destino do formulário, o que é considerado menos seguro por apresentar os dados de forma desprotegida ao enviar a requisição, mesmo no uso de HTTPS.
 - **post**: transmite os dados na requisição do destino do formulário. Os dados ficam ocultos, sendo mais seguro que o método GET. Se utilizado em conjunto com o protocolo HTTPS, garante a segurança e sigilo dos dados.
- **action**: define o endereço de destino dos dados do formulário para serem processados.

Todas as demais marcações que serão apresentadas sobre formulários devem, preferencialmente, ser apresentadas entre a marcação de abertura e fechamento da marcação `<form>`.

11.2. Legendas

A maioria dos formulários apresentam campos para o usuário preencher com valores e, para a maior parte desses campos, é necessário informar ao usuário o conteúdo que precisa ser preenchido, como o nome do usuário, idade e outros. Existe uma marcação

no HTML recomendada para apresentar o título do campo para o usuário. Essa marcação é a `<label>`. Veremos no exemplo deste tópico o uso dessa marcação.

11.3. Pontos de interação entre o formulário e o usuário

Dentro de um formulário, diversos mecanismos de interação com o mesmo podem ser construídos, entre eles os campos para entrada de dados e botões para tomadas de ações, como o botão de envio do formulário, por exemplo. Por meio da marcação `<input>` é possível construir os três primeiros tipos.

Os principais atributos da marcação `<input>` são:

- **name**: indica o nome da interação, geralmente utilizado junto com campos de entrada de dados.
- **type**: indica o tipo de interação, podendo ser dos seguintes tipos:
 - **text**: cria o campo de entrada para texto livre.
 - **radio**: cria um campo do tipo rádio.
 - **checkbox**: cria um campo do tipo de checagem.
 - **number**: cria um campo de entrada numérico.
 - **reset**: limpa todas as informações preenchidas no formulário.
 - **submit**: envia o formulário com os campos preenchidos para o endereço definido no atributo **action** da marcação `<form>`.
- **value**: para campos de entrada de dados, informa um valor inicial de exibição para o usuário. Já para interações do tipo **reset** ou **submit**, que são apresentadas em formato de botões, definem o texto que aparecerá impresso no botão em questão.
- **disabled**: permite desabilitar a interação de um elemento do formulário, o que é geralmente utilizado para apresentar algum valor para o usuário, mas sem permitir a sua edição.

Para completar o time inicial das marcações voltadas para formulários no HTML, existe a marcação `<select>`, que também apresenta um campo de interação com o usuário. Neste caso, a marcação `<select>` apresenta para o usuário uma lista de valores pré-definidos, para que o usuário possa selecionar um deles. Cada opção apresentada por esta marcação deverá ser informada por meio de uma marcação `<option>`.

Não se preocupe se todas essas informações ainda parecerem confusas. Vamos a partir de agora construir um exemplo de formulário com todas as informações apresentadas até o momento e certamente este exemplo tornará tudo mais claro.

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     <form method="post" action="">
```

```

10    <label>Nome:</label>
11    <input type="text" name="campoNome"><br>
12    <label>Idade:</label>
13    <input type="number" name="campoIdade" min="1" max="95"/><br>
14    <label>Sexo:</label>
15    <input type="radio" name="campoSexo" value="M"> Masculino
16    <input type="radio" name="campoSexo" value="F"> Feminino<br>
17    <label>Deseja receber e-mails:</label>
18    <input type="checkbox" name="campoReceber"><br>
19    <label>Estado civil:</label>
20    <select name="campoCivil">
21        <option value="solteiro">Solteiro(a)</option>
22        <option value="casado">Casado(a)</option>
23        <option value="divorciado">Divorciado(a)</option>
24        <option value="viuvo">Viúvo(a)</option>
25    </select><br>
26    <input type="reset" value="Limpar dados">
27    <input type="submit" value="Enviar formulário">
28 </form>
29 </body>
30 </html>

```

Código-Fonte 11:1. marcacaoFormLabelInputSelect.htm

O resultado da interpretação do Código-Fonte 11:1 por um navegador web pode ser visualizado na Figura 11-1.

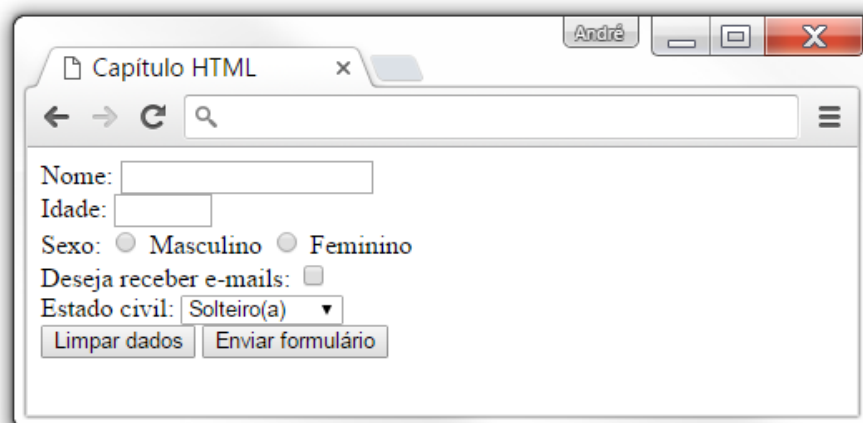


Figura 11-1. Formulário construído em HTML

Vamos compreender melhor o uso das marcações de formulários e de seus elementos, aplicadas no Código-Fonte 11:1. O formulário tem seu início por meio da marcação `<form>` aberta na linha 9, que é fechada somente na linha 29. Todo o conteúdo presente entre essas linhas faz parte do formulário.

A primeira informação que o usuário deve preencher é o seu nome, que é informado por meio da marcação `<label>` apresentada na linha 10. O primeiro campo construído no formulário, depois da apresentação `<label>` que apresenta o campo para o usuário, está na linha 11, por meio da marcação `<input>` do tipo `text` (atributo `type`). Este campo representa o nome do usuário para preenchimento. Ainda, um nome é atrelado ao

campo por meio do atributo `name`, definindo o nome do campo como `campoNome`. Essa informação não aparece na tela para o usuário, mas é utilizada internamente para enviar os dados para processamento por alguma linguagem de programação web posteriormente. Observe que a marcação `<input>` não apresenta uma marcação específica de fechamento. Ela deve ser utilizada como já foi apresentado para a marcação `<link>` e `<!DOCTYPE html>`, ou seja, sem fechamento.

Para que o usuário possa informar o campo idade foi utilizada a marcação `<input>` do tipo `number` na linha 13, previamente apresentada na tela pelo texto `Idade:` que utilizou a marcação `<label>` para informar o nome do campo para o usuário. Este campo recebeu o nome `campoIdade` para ser referenciado posteriormente. Campos do tipo `number` restringem os valores de preenchimento para o campo, ou seja, apenas números poderão ser utilizados. Ainda, é possível definir o valor mínimo e máximo que possa ser preenchido por meio dos atributos `min` e `max`. Neste caso, somente valores entre 1 e 95 serão aceitos por este campo.

Em seguida um campo do tipo rádio com duas opções de resposta é disponibilizado nas linhas 15 e 16, representando a informação descrita pela marcação `<label>` na linha anterior, que é o sexo do usuário. Neste caso, cada opção apresentada deve ser construída por meio de uma marcação `<input>` própria, do tipo `radio`. A opção `Masculina` é construída na linha 15 e a opção feminina é construída na linha 16, mas utilizando o mesmo nome de campo da opção anterior (atributo `name` com o valor `campoSexo`). É dessa forma, por meio do nome do campo, que o HTML consegue desmarcar uma opção quando outra é marcada, ou seja, caso o usuário marque `Masculino` e depois `Feminino`, automaticamente a opção `Masculino` será desmarcada pelo relacionamento que existe entre as duas opções, construídas com o mesmo nome de campo.

Observe que o valor a ser enviado para ser processado posteriormente pelo formulário é definido por meio do atributo `value` para este tipo de campo, para cada uma das opções. Se o usuário marcar `Masculino`, o servidor web receberá `M` para este campo; ou então `F` para a opção `Feminino`.

Existem ainda os campos de checagem, que podem ser construídos por meio da marcação `<input>` do tipo `checkbox`, como mostra a linha 18. Neste caso, nenhum valor precisa ser definido por meio do atributo `value`, pois cada campo de checagem deve apresentar um nome de campo próprio, para cada opção existente no formulário, sendo enviado ao servidor caso seja marcado ou não enviado caso não seja marcado. Dessa forma, a linguagem de programação web, no servidor que irá processar o formulário, avalia o recebimento ou não do campo, para saber se o mesmo foi marcado (quando é enviado) ou não. Maiores detalhes sobre este processo podem ser

consultados diretamente nas linguagens de programação web, pois cada uma apresenta suas características próprias.

Aumentando um pouco a complexidade do formulário, um campo de seleção é construído por meio da marcação `<select>`, aberta na linha 20 e fechada na linha 25, para representar as opções que o usuário pode escolher para a informação de estado civil. Cada opção deve ser construída isoladamente, com sua própria abertura e fechamento da marcação `<option>`, devendo todas elas ser apresentadas entre a abertura e fechamento da marcação `<select>`, criando assim o relacionamento das opções com o campo.

Ainda, opcionalmente, é possível apresentar um atributo `value` para cada marcação `option`, informando qual valor deve ser enviado para o servidor, caso aquela opção seja a selecionada pelo usuário. Caso o atributo `value` não seja informado, é o texto da própria marcação `<option>` que será enviado, caso a opção seja a selecionada.

Para concluir o formulário de exemplo, é construído um botão para limpar os dados do formulário na linha 26, por meio de uma marcação `<input>` do tipo `reset`, e construído o botão de envio dos dados na linha 27 por meio de uma marcação `<input>` do tipo `submit`. Em ambos os casos, o atributo `value` serve para definir o texto a ser apresentado para cada botão quando o formulário for apresentado na tela do navegador para o usuário.

Complementando o assunto de formulário, os atributos `checked`, `selected` e `disabled` serão apresentados no código-fonte disponibilizado a seguir.

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     <form method="post" action="">
10       Nome: <input type="text" name="campoNome" value="André" disabled><br>
11       Idade: <input type="number" name="campoIdade" min="1" max="95"/><br>
12       Sexo:
13       <input type="radio" name="campoSexo" value="M" checked> Masculino
14       <input type="radio" name="campoSexo" value="F"> Feminino<br>
15       Deseja receber e-mails:
16       <input type="checkbox" name="campoReceber" checked><br>
17       Estado civil:
18       <select name="campoCivil">
19         <option value="solteiro">Solteiro(a)</option>
20         <option value="casado" selected>Casado(a)</option>
21         <option value="divorciado">Divorciado(a)</option>
22         <option value="viuvo">Viúvo(a)</option>
23       </select><br>
24       <input type="reset" value="Limpar dados">
25       <input type="submit" value="Enviar formulário">
```

```

26     </form>
27 </body>
28 </html>

```

Código-Fonte 11.2. atributosCheckedSelectedDisabled.htm

O resultado da interpretação do Código-Fonte 11:2 por um navegador web pode ser visualizado na Figura 11-2.

Figura 11-2. Preenchendo campos e selecionando valores via HTML

Observe que o formulário apresentado no Código-Fonte 11:2 é basicamente o mesmo já utilizado anteriormente, com pequenas alterações, para apresentar mais algumas funcionalidades interessantes para você.

A primeira delas está na linha 10, por meio do uso do atributo `disabled`, que desabilita o campo de nome para preenchimento do usuário. Observe ainda que o atributo `value` foi utilizado nesta mesma linha para preencher automaticamente o campo com um valor pré-definido, no caso o nome `André`.

Outra novidade nesta versão do formulário está na linha 13, utilizando o atributo `checked` no campo do tipo rádio para a opção `Masculino`. O atributo `checked` informa que o campo deve já aparecer marcado quando o formulário for carregado pelo navegador. Esse mesmo atributo pode ser utilizado também na marcação `<input>` do tipo `checkbox`, como ocorre na linha 16, já marcando esta opção como preenchida ao exibir o formulário para o usuário.

Finalizando as novidades desta versão do formulário, o atributo `selected` é utilizado na linha 20 para fazer com que a opção `Casado(a)` apareça já selecionada assim que o formulário for exibido. Este atributo tem a mesma funcionalidade do atributo `checked`, mas no caso da marcação `<select>` ele se chama `selected`.

O processamento de formulários não será abordado neste e-book, pois não faz parte do HTML, mas sim das linguagens de programação web utilizadas por casa site.

12. Trabalhando com painéis

Existem diferentes formas de organizar visualmente conteúdo em páginas web, e um recurso bastante útil para esta tarefa, talvez o mais popular inclusive, é o uso de painéis, por meio da marcação `<div>` do HTML.

Painéis do tipo `<div>` são áreas que organizam conteúdos dentro de si, com formatações visuais específicas, separando diferentes áreas do site em painéis distintos. A grande vantagem do uso da marcação `<div>` é justamente a facilidade de formatação visual de seus elementos por meio do CSS. É muito mais prático utilizar painéis do tipo `<div>` e formatá-los via CSS para diagramar os conteúdos de sua página, deixando um conteúdo posicionado acima, do lado, em colunas, ou da forma que você achar mais interessante.

Considere o código-fonte apresentado a seguir:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06     <link rel="stylesheet" type="text/css" href="arquivoDeEstilos.css">
07   </head>
08   <body>
09     <div class="meuDiv1">
10       <h1>Softblue</h1>
11       <p>Qualidade em cursos online</p>
12     </div>
13   </body>
14 </html>
```

Código-Fonte 12:1. marcacaoDiv.htm

O resultado da interpretação do Código-Fonte 12:1 por um navegador web pode ser visualizado na Figura 12-1.

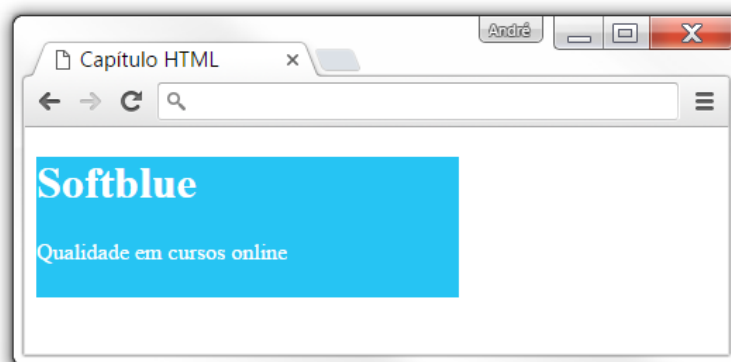


Figura 12-1. Página com painel

A construção do painel inicia na linha 9 com a abertura da marcação `<div>`, e encerra na linha 12 com o fechamento da marcação em questão. Todo o conteúdo apresentado entre essas linhas será considerado conteúdo do painel. Ainda, é possível informar o atributo `class` na marcação `<div>` para vincular a estrutura com um estilo no arquivo CSS. Neste caso, o seguinte estilo foi adicionado no arquivo CSS:

```
.meuDiv1
{
  background-color: #26c4f3;
  color: #ffffff;
  width: 300px;
  height: 100px;
}
```

Basicamente, o estilo `.meuDiv1` define a cor do fundo do painel por meio da propriedade `background-color`, a cor do texto por meio da propriedade `color` e a largura e a altura em pixels do painel por meio das propriedades `width` e `height`, respectivamente.

Conforme dito neste tópico, painéis do tipo `<div>` facilitam bastante a diagramação de um site. Você verá o uso dessas estruturas em ação no exemplo existente no final deste livro, não se preocupe. Como é importante que o uso da marcação `<div>` seja realizado de forma adequada com outras marcações ainda não abordadas, o exemplo será construído após conhecer essas outras marcações, que serão vistas nos próximos tópicos.

13. Semântica avançada da estrutura do site

Agora que você conhece um pouco mais do mundo de marcações HTML para criar o seu site, é importante que seja abordado o tema de semântica estrutural do site de forma mais aprofundada. Você já viu como organizar o conteúdo da sua página utilizando a marcação `<article>`, `<section>` e várias outras, mas ainda existem outros detalhes e marcações que devem ser considerados.

Analisando a maior parte dos sites na internet, com a evolução do tempo constatou-se que eles possuíam algumas estruturas em comum, como por exemplo: topo da página, menu de opções, conteúdo lateral secundário e rodapé.

Baseado nessas estruturas, os algoritmos dos buscadores começaram a priorizar de maneira diferente o conteúdo de cada uma dessas partes, mas não havia um padrão para estabelecer o que era considerado topo, menu de opções, conteúdo lateral ou rodapé. Alguns buscadores até tentaram sugerir padrões, mas como cada um apresentava uma forma diferente, o HTML5 aproveitou para colocar ordem na casa.

A partir do HTML5, existem algumas marcações semânticas para organizar cada parte de sua página, fazendo com que os algoritmos dos buscadores compreendam adequadamente cada estrutura do seu site, apresentando melhores resultados na hora de avaliar a sua página para aparecer nos resultados das buscas. Vamos categorizar essas marcações como macro estruturais, ou seja, elas organizam as regiões de sua página.

A primeira das marcações macro estruturais com funções semânticas é a que define onde começa e termina o topo do seu site, geralmente com o logotipo da empresa e outras informações comuns entre as diferentes páginas do seu site. Essa marcação é a `<header>`, que você já utilizou em conjunto com a marcação `<article>` anteriormente neste e-book. Contudo, para representar o topo da página, a marcação `<header>` deve estar apresentada separadamente de qualquer marcação `<article>` que o seu site venha a ter, e depois da abertura da marcação `<body>`. Um exemplo prático será construído logo mais adiante para ilustrar essa situação, assim que as demais marcações estruturais forem apresentadas.

Outro ponto em comum entre diversos sites na internet são seus menus de navegação. Eles podem aparecer no topo, na lateral ou em outras áreas. O importante mesmo é que o menu de opções seja construído entre a abertura e fechamento da marcação `<nav>`, que organiza semanticamente o conteúdo, avaliando-o como menu de opções do site. Portanto os links do seu site que são comuns entre diversas páginas, tais

como a página principal do site, página de contato e outras, devem ser apresentados entre a abertura e fechamento da marcação `<nav>`.

Em seguida, o conteúdo principal de sua página pode ser apresentado em algum momento no corpo da página, devendo fazer o uso da marcação `<section>` e `<article>`, que já foram abordadas anteriormente.

Caso a sua página venha a apresentar algum conteúdo lateral com menor relevância do que o assunto principal da página, o mesmo deve ser estruturado entre a abertura e fechamento da marcação `<aside>`.

Por último, a marcação que define estruturalmente o rodapé de uma página é a `<footer>`.

Várias marcações macro estruturais com funções semânticas foram vistas neste tópico. Para ilustrar o uso das mesmas, considere o código-fonte apresentado a seguir:

```
01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <title>Capítulo HTML</title>
06   </head>
07   <body>
08
09     <header>
10       <!-- Aqui vai o topo da página -->
11     </header>
12
13     <nav>
14       <!-- Aqui vai o menu da página -->
15     </nav>
16
17     <section>
18       <article>
19         <!-- Aqui vão os conteúdos da página -->
20       </article>
21     </section>
22
23     <aside>
24       <!-- Aqui vai o conteúdo lateral da página -->
25     </aside>
26
27     <footer>
28       <!-- Aqui vai o rodapé da página -->
29     </footer>
30   </body>
31 </html>
```

Código-Fonte 13:1. paginaEstruturadaCrua.htm

O Código-Fonte 13:1 apresentado não produz resultados na tela, pois o conteúdo de cada uma das partes do site ainda não foi criado. Este código serve apenas para

representar o uso das marcações estruturais de uma página em um exemplo fictício, respeitando a hierarquia existente entre as mesmas.

Observe que a marcação `<body>` é aberta na linha 7 e fechada na linha 30 e, entre essas linhas, o site deve ser construído. O topo `<header>`, definido entre as linhas 9 e 11, geralmente é a primeira parte a ser informada, seguido do menu de opções `<nav>`, definido entre as linhas 13 e 15, mas nada impede que o seu site apresente o menu em outra posição ou ordem ou até mesmo antes do próprio topo, ou até contido dentro das marcações que definem o topo ou lateral. O importante mesmo é que os links comuns de navegação do seu site estejam organizados dentro da abertura e fechamento da marcação `<nav>` e que o conteúdo do topo seja organizado dentro da marcação `<header>`.

Após a definição do topo, e geralmente do menu de opções também, inicia-se o bloco de conteúdo da página. Este bloco pode ser formado por uma ou mais seções (marcação `<section>`) e cada seção pode conter um ou mais artigos (marcação `<article>`). No exemplo apresentado, apenas uma seção e um artigo foram ilustrados, entre as linhas 17 e 21.

Caso o seu site apresente conteúdo lateral, a marcação `<aside>` deve ser utilizada, como mostra o exemplo apresentado entre as linhas 23 e 25. Essa marcação por si só não realiza a diagramação do seu site apresentando o conteúdo na lateral para o usuário. Isso deve ser feito via CSS e veremos como realizar esta formatação ainda neste e-book, mais adiante.

Por último, a estrutura do rodapé da página, que geralmente também é bastante comum entre diferentes páginas de um site, deve ser informada entre a abertura e o fechamento da marcação `<footer>`, como organizado entre as linhas 27 e 29.

Calma, muita calma nessa hora. Foram muitas marcações abordadas neste e-book, e várias delas neste último tópico. Então por que não buscar um cafezinho para recarregar as suas energias e visualizarmos estas marcações sendo utilizadas em conjunto em um exemplo maior? É exatamente isso que será feito no próximo tópico deste e-book.

14. Exemplo de página macro estruturada semanticamente

Neste tópico será abordado um exemplo de página de internet, construída com boa parte das marcações abordadas neste livro, organizadas semanticamente. O exemplo fictício que será apresentado neste tópico é a construção de uma página do site da Softblue. O objetivo principal deste exemplo é abordar o uso das marcações HTML e os conhecimentos ensinados neste e-book, podendo o exemplo ser aprofundado e adaptado para outras versões por você, posteriormente.

O exemplo de página que será construída pode ser visualizado na Figura 14-1.



Figura 14-1. Exemplo fictício de página da Softblue para ser construída

14.1. Estrutura básica da página

A primeira parte da página que será construída é o seu cabeçalho, com as principais configurações da página. Para isso as marcações `<!DOCTYPE html>`, `<html>`, `<head>`, `<meta>`, `<title>`, `<body>` e `<div>` serão utilizadas, como mostra o código-fonte apresentado a seguir:

```

01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <meta name="description" content="Site de cursos online">
06     <meta name="keywords" content="Java, PHP, HTML, Android, iOS">
07     <meta name="author" content="André Milani (Softblue)">
08     <title>Softblue</title>
09     <link rel="stylesheet" type="text/css" href="css/site.css">
10   </head>
11   <body>
12     <div id="raiz">
13       <!-- Aqui serão construídas as estruturas do site -->
14     </div>
15   </body>
16 </html>

```

A página apresentada inicia informando que o seu conteúdo é HTML por meio da marcação `<!DOCTYPE html>` apresentada na linha 1, posteriormente abrindo o bloco `<html>` com o atributo `lang="pt-br"` representando o idioma português do Brasil para o conteúdo que será construído.

Observe na estrutura básica da página que o cabeçalho é construído entre as linhas 3 e 10 e, dentro dele, algumas configurações são realizadas. A primeira delas, na linha 4, indicando que o arquivo utiliza a codificação `utf-8` para seus caracteres e, depois dela, três configurações para os algoritmos dos robôs dos sites buscadores: a marcação `<meta>` com o atributo `name` definido com `description` na linha 5, com uma breve descrição do site; marcação `<meta>` com o atributo `name` definido com `keywords` para apresentar algumas palavras chaves da página na linha 6; e a marcação `<meta>` com o atributo `name` definido como `author` para informar o autor da página, na linha 7.

Na sequência, a marcação `<title>` informa o título que deve ser apresentado na janela do navegador, na linha 8.

Para finalizar o cabeçalho do arquivo, na linha 9 é utilizada a marcação `<link>` para relacionar o arquivo HTML com um arquivo CSS, que será construído em paralelo a esta página. Observe que o arquivo CSS utilizado neste exemplo foi chamado de `site.css` e deve ser criado em uma subpasta do seu site, chamada `css`. Essa é uma forma opcional de organizar seus arquivos, criando uma subpasta exclusiva para os conteúdos CSS.

Ainda nessa primeira estrutura apresentada para a página, a marcação `<body>` teve sua abertura e fechamento definidos nas linhas 11 e 15 e, dentro dela, foi criada uma marcação `<div>` cuja propriedade `id` foi definida com o nome `raiz`. Isso foi feito porque no arquivo CSS será criado um estilo chamado `div#raiz` para definir algumas propriedades desta marcação.

Para complementar a primeira parte construída da página, vamos dar uma olhada no arquivo CSS sugerido para utilização neste exemplo. Crie a subpasta `css` e, dentro dela, crie o arquivo `site.css`. Inicialmente vamos configurar o arquivo CSS para que o fundo da página seja na cor azul, tema da Softblue. Para isso, o estilo `body` pode ser definido da seguinte forma:

```
body
{
  background-color: #26c4f3;
  font-family: Century Gothic;
}
```

Além da definição da cor do fundo da página por meio da propriedade `background-color` no estilo `body`, foi informado também que todo o texto do site seja apresentado utilizando a fonte `Century Gothic`, por meio da propriedade `font-family` aplicada no estilo `body`. Neste ponto, uma dúvida é bastante comum: *“E se a fonte que eu quero utilizar não estiver instalada no computador do usuário que vai navegar no meu site?”*. Para resolver essa questão, você pode informar em seu arquivo CSS o endereço para o navegador web baixar automaticamente a fonte, caso seja necessário. Essa instrução é feita da seguinte forma no arquivo CSS:

```
@font-face
{
  font-family: Century Gothic;
  src: url('/fonts/GOTHIC.ttf');
}
```

A informação do arquivo de fonte deve ser realizada por meio de estilos chamados `@font-face`. Dentro desses estilos, é possível informar o parâmetro `src`, com o endereço virtual do arquivo da fonte (geralmente são arquivos com extensão `.ttf` ou `.otf`) que neste caso foram alocadas dentro de uma subpasta do projeto chamada `fonts`. Observe que para informar um caminho virtual no arquivo CSS foi utilizada uma função chamada `url`, passando como parâmetro o caminho em questão.

Ainda, ao abrir o arquivo da fonte, o mesmo apresenta para você o nome oficial da fonte, e este nome é o que deve ser definido na propriedade `font-family` do estilo `@font-face`. Se o nome for digitado incorretamente, essa instrução não vai funcionar. Portanto atenção redobrada neste passo, pois em boa parte dos casos o nome do arquivo (atributo `src` do estilo) é diferente de como se escreve por completo o nome da fonte (atributo `font-family`).

Algumas fontes apresentam diferentes arquivos para representar suas formatações de negrito e itálico. Neste caso, cada arquivo deve ser informado em um estilo `@font-face` próprio. Observe a inclusão do arquivo da fonte `Century Gothic` em negrito, logo depois da inclusão anteriormente apresentada:

```
@font-face
{
    font-family: Century Gothic;
    src: url('/fonts/GOTHICB.ttf');
    font-weight: bold;
}
```

No caso de adicionar uma fonte com alguma formatação específica, é necessário informar no estilo qual é essa formatação. Neste caso foi adicionada a propriedade `font-weight` com o valor `bold`, indicando que esse arquivo representa a fonte `Century Gothic` em negrito. Observe atentamente que o arquivo de fonte é outro: para a fonte regular foi informado o arquivo `GOTHIC.ttf`, enquanto que para a fonte em negrito foi informado o arquivo `GOTHICB.ttf`, com a letra **B** no final (de *bold*, negrito em inglês), antes da extensão. Os arquivos podem ter qualquer nome. O importante é fazer corretamente o relacionamento do nome do arquivo com o nome da fonte no estilo `@font-face`.

Para finalizar a construção da primeira parte da estrutura da página, depois de resolvidas as questões das fontes, o estilo `div#raiz` foi criado no arquivo CSS, como é possível visualizar no código apresentado a seguir:

```
div#raiz
{
    width: 610px;
    margin: 0 auto;
}
```

O estilo `div#raiz` foi criado com dois objetivos. O primeiro é definir a largura total do site em 610 pixels e, por este motivo, uma marcação `<div>` foi criada no arquivo HTML, com a propriedade `id` definida como `raiz`, para fazer uso deste estilo de formatação somente neste painel, na linha 12 do código HTML. O segundo é centralizar esse bloco de 610 pixels horizontalmente na tela. Para isso a propriedade `margin` foi definida com `0 auto`, ou seja, zero pixels de margem; e `auto`, para centralizar horizontalmente.

Essa foi a primeira parte da página! Deste momento para frente, construiremos o topo, menu, corpo, conteúdo lateral e rodapé, todos contidos dentro da marcação `<div>` que limita a largura do site em 610 pixels.

14.2. Construindo o topo da página

Uma vez que a estrutura fundamental da página está pronta, podemos nos preocupar com suas outras partes estruturais, como o topo da página e menu de opções. Geralmente, o menu de opções de um site é apresentado no topo ou em alguma de suas laterais. Neste exemplo fictício, o menu de opções será incluído no topo da página

e, por este motivo, a marcação `<nav>`, que representa o menu, estará contida na marcação `<header>`, que define o topo da página.

O código-fonte apresentado a seguir ilustra uma alternativa da construção deste código. Ainda, o código que será apresentado deverá ser inserido dentro da marcação `<div id="raiz">` do site, criado anteriormente, para que siga as orientações de largura e alinhamento previamente definidas no CSS para o painel principal. Você pode consultar no tópico 14.6 o código-fonte completo para verificar o posicionamento deste trecho de código e dos próximos que serão apresentados.

```

01 <div id="cabecalho">
02   <header>
03     <figure class="imagemMesmaLinhaLogotipo">
04       
06     </figure>
07
08     <nav>
09       <a href="paginaEmpresa.htm">
10         <figure class="imagemMesmaLinhaMenu">
11           
12         </figure>
13       </a>
14       
16       <a href="paginaCursos.htm">
17         <figure class="imagemMesmaLinhaMenu">
18           
19         </figure>
20       </a>
21       
23       <a href="paginaContato.htm">
24         <figure class="imagemMesmaLinhaMenu">
25           
26         </figure>
27       </a>
28     </nav>
29   </header>
30 </div>

```

Calma, não fique preocupado, pois cada passo da construção do topo será detalhado agora para você.

Inicialmente, para facilitar a formatação visual do topo, foi criada uma marcação `<div>`, aberta na linha 1 e fechada na linha 30. Será dentro dela que todo o topo será construído. Para formatar esta marcação, trabalharemos mais para frente com o estilo `div#cabecalho`, relacionado com a marcação `<div>` existente na linha 1, mas por enquanto vamos deixá-lo de lado.

Dentro da marcação `<div id="cabecalho">`, é definida a estrutura semântica do topo da página, por meio da marcação `<header>`, aberta na linha 2 e fechada na linha 28.

Iniciando a estrutura do topo, optou-se por apresentar o logotipo da Softblue. Para isso, a marcação `<figure>` foi definida entre as linhas 3 e 6, incluindo uma imagem por meio da marcação ``, que começa na linha 4 e termina na linha 5. As imagens utilizadas neste exemplo serão organizadas dentro de uma subpasta do projeto chamada `images`. Novamente, observe que para a formatação visual foi atrelado o estilo `.imagemMesmaLinhaLogotipo` para a marcação `<figure>`, que será abordado mais adiante quando o tema CSS do topo da página for detalhado.

Após o logotipo da empresa, foi definido que o menu de opções do site seria apresentado, ainda no topo. O menu deve conter três opções: *EMPRESA*, *CURSOS* e *CONTATO*. Para cada uma dessas opções foi informada uma imagem. O menu poderia ser construído utilizando texto, mas, para fins didáticos, optou-se neste momento em utilizar imagens. Fique tranquilo, haverá outro menu de opções no conteúdo lateral do site, onde serão abordadas questões voltadas ao uso de textos em menus.

Para informar o início e o fim semântico do menu da página, a marcação `<nav>` foi aberta na linha 8 e fechada na linha 28. Dentro do menu, temos basicamente cinco elementos principais. O primeiro deles é a imagem da opção *EMPRESA* do menu, acessada pela marcação `` na linha 11, semanticamente organizada pela marcação `<figure>` entre as linhas 10 e 12. Por último, ainda sobre a imagem *EMPRESA*, ela foi devidamente preparada para direcionar o usuário para a página em questão no caso de clique do mouse, por meio da marcação `<a>` definida entre as linhas 9 e 13. Observe que a marcação `<figcaption>` não foi utilizada, pois, neste momento, para esta imagem de menu, é desejado que nenhum texto seja apresentado na tela detalhando a imagem.

Entre as opções de menu, foi definido que uma barra seria apresentada separando as opções. Essa barra também foi fornecida em imagem e é o segundo elemento do menu, apresentado por meio de uma marcação `` nas linhas 14 e 15. Observe que esta imagem em particular não fez o uso da marcação `<figure>`. O vínculo com uma marcação `<figure>` não foi realizado neste caso porque a imagem da barra não apresenta nenhuma informação relevante no conteúdo da página para os algoritmos de busca. Por este motivo, optou-se em simplesmente apresentar a imagem sem nenhum vínculo semântico.

Após a barra que separa as opções do menu ser apresentada, a segunda opção do menu é construída. Essa opção é a *CURSOS* e foi construída entre as linhas 16 e 20 da mesma forma que a opção *EMPRESA* foi construída anteriormente: organizando semanticamente a marcação `` com a marcação `<figure>` e criando um ponto de acesso por meio da marcação `<a>`, apontando para o que pode vir a ser a página de cursos do site.

Em seguida, novamente uma imagem de barra é apresentada, sem fins semânticos, nas linhas 21 e 22 e, por último, a opção de menu *CONTATO* é construída entre as linhas 23 e 27, exatamente como as demais opções de menu anteriormente detalhadas.

Observe que as marcações `<figure>` das opções do menu e as marcações `` das barras têm suas propriedades `class` definidas com o valor `imagemMesmaLinhaMenu`, enquanto a marcação `<figure>` do logotipo do site atrela o valor `imagemMesmaLinhaLogotipo` para a propriedade `class` de sua marcação `<figure>`. Esses estilos CSS devem ser criados. Se você visualizar o arquivo HTML neste momento sem criar estes estilos, você vai observar que as imagens aparecerão uma abaixo da outra, em uma formatação não muito indicada para o objetivo deste exemplo.

Para formatar visualmente as imagens e seus alinhamentos, será necessário criar alguns estilos em seu arquivo CSS. O primeiro deles é o estilo `div#cabecalho`, que vai definir algumas propriedades para a marcação `<div id="cabecalho">`, que contém o topo da página. Este estilo pode ser conferido a seguir:

```
div#cabecalho
{
  float: left;
  width: 600px;
  margin: 5px;
  display: block;
  clear: both;
  background-color: #ffffff;
  border-radius: 15px;
}
```

Várias propriedades foram definidas no estilo `div#cabecalho`. São elas:

- `float: left`: define o alinhamento do elemento em relação aos outros do site. Neste caso, o valor `left` representa que este elemento aparecerá à esquerda.
- `width: 600px`: define que o elemento terá 600 pixels de largura.
- `margin: 5px`: define que a margem do elemento terá 5 pixels.
- `background-color: #ffffff`: define a cor de fundo do elemento (branca).
- `border-radius: 15px`: define que as quinas do elemento devem ser arredondadas em 15 pixels, causando um efeito mais suave nas quinas do `<div>` que fará uso deste estilo.

Após formatar visualmente o elemento `<div id="cabecalho">`, outros estilos devem ser criados para formatar algumas propriedades das imagens utilizadas no topo da página. O estilo que formata o logotipo da empresa pode ser visualizado a seguir:

```
.imagemMesmaLinhaLogotipo
{
  float: left;
  margin: 10px;
  margin-right: 30px;
}
```

O estilo `.imagemMesmaLinhaLogotipo` informa que o alinhamento da imagem em relação aos outros elementos da página deve ser feito para a esquerda (`left`), por meio da propriedade `float`. Esta instrução evita também que ocorra uma quebra de linha entre as marcações que fizerem uso deste estilo. Este estilo define ainda que as margens dos quatro lados da imagem devem ser de `10px`, por meio da propriedade `margin`, e, por último, informa que a margem direita (`margin-right`) deve ser de `30px`, para que exista um pouco mais de espaço entre a imagem do logotipo e as imagens do menu de opções que irão aparecer do lado direito. Você pode testar os valores das margens trocando de `10` para outros valores e de `30` para outros valores, para visualizar exatamente como cada uma dessas propriedades impacta no resultado final.

Já para as imagens do menu de opções um novo estilo foi definido. Isso ocorreu devido ao fato de que o espaçamento entre as imagens do menu apresentarão um valor diferente para a propriedade de margem. Para as imagens do menu, o seguinte estilo deve ser definido:

```
.imagemMesmaLinhaMenu
{
  float: left;
  margin: 10px;
  margin-top: 50px;
}
```

As imagens do menu de opções continuam sendo alinhadas à esquerda em relação aos demais elementos, por meio da propriedade `float: left`, evitando que ocorra quebra de linha entre as marcações `<figure>`. Ainda, a margem de espaçamento novamente foi definida com `10px` por meio da propriedade `margin`, mas observe que não há mais a definição da propriedade `margin-right` que existia no estilo utilizado no logotipo. Ou seja, as imagens do menu ficarão bem mais próximas umas das outras. Ainda, como as imagens do menu têm altura menor que a do logotipo, elas foram afastadas do topo da página em `50px` por meio da propriedade `margin-top: 50px`. Novamente, você pode alterar esses valores para visualizar outros resultados.

Com isso, temos o topo da página e o menu de opções prontos para serem utilizados! Mas ainda há muito a ser feito. Vamos então para a próxima parte: corpo da página.

14.3. Conteúdo principal

O conteúdo principal da página será construído novamente entre uma marcação `<div>` aberta na linha 1 e fechada na linha 19 do código-fonte apresentado a seguir. Para este elemento também será criado um estilo próprio para formatação, mas por enquanto vamos focar no arquivo HTML. Observe a seguir o código-fonte utilizado para criar o conteúdo principal desta página:


```

01 <div id="conteudo">
02   <section>
03     <article>
04       <header>
05         <h1>Cursos online</h1>
06       </header>
07       <p>
08         Quer aprender com qualidade? Venha fazer parte dos
09         melhores, venha aprender programação na Softblue!
10         Aprenda <strong>HTML</strong> por meio do nosso
11         <strong>e-book de HTML</strong> agora mesmo!
12       </p>
13       <figure>
14         
15         <figcaption>E-book de HTML</figcaption>
16       </figure>
17     </article>
18   </section>
19 </div>

```

Dentro do elemento `<div id="conteudo">` optou-se em utilizar a marcação `<section>`, apresentada entre as linhas 2 e 18, para delimitar a seção de conteúdo principal da página. Para popular essa seção, o conteúdo principal será apresentado dentro de uma marcação `<article>`, criada entre as linhas 3 e 17. Assumindo que é a página que apresenta os cursos da Softblue, foi dado o título `Cursos online` para o conteúdo da página, por meio da marcação `<h1>` existente na linha 5, apresentada entre a abertura e fechamento da marcação `<header>` nas linhas 4 e 6, que informam o título do artigo do qual fazem parte.

Após o título do conteúdo principal ter sido apresentado, alguns parágrafos com o conteúdo da página podem ser informados. Neste caso, há apenas um parágrafo, entre as linhas 7 e 12. Observe que, como o conteúdo principal desta página é sobre os materiais da Softblue, a marcação `` foi utilizada para dar maior relevância a algumas palavras-chave do conteúdo, entre elas `HTML` e `e-book de HTML`, respectivamente nas linhas 10 e 11.

Há também no conteúdo principal da página uma imagem representando o e-book de HTML da Softblue, informado por meio de uma marcação `<figure>` entre as linhas 13 e 16. Observe que, neste caso, optou-se por utilizar a marcação `<figcaption>` para atrelar uma legenda à imagem, na linha 15.

Apesar de poucas marcações fazerem relação com algum estilo CSS por meio da propriedade `id` ou `class`, várias marcações HTML sofreram alterações no arquivo CSS neste exemplo, a fim de apresentar uma formatação mais agradável.

O primeiro estilo adicionado no arquivo CSS é o `div#conteudo`, formatando o elemento `<div id="conteudo">`, da seguinte forma:

```

div#conteudo
{

```

```
float: left;
width: 405px;
height: 350px;
margin: 5px;
background-color: #ffffff;
border-radius: 15px 50px 15px 50px;
}
```

Nesta altura do e-book você já deve estar familiarizado com todas as propriedades CSS utilizadas no estilo `div#conteudo`, mas não custa revisar. São elas:

- `float: left`: define que o elemento deve ser alinhado à esquerda em relação aos outros elementos.
- `width: 405px`: define que a largura do elemento será de 405 pixels.
- `height: 350px`: define que a altura do elemento será de 350 pixels.
- `margin: 5px`: define que a margem do elemento será de 5 pixels.
- `background-color: #ffffff`: define que a cor de fundo do elemento será `#ffffff` (branca).
- `border-radius: 15px 50px 15px 50px`: Define que as quinas do elemento serão arredondadas e, neste caso, com valores diferentes de acordo com cada quina. A primeira é a quina superior esquerda, em seguida a quina superior direita, inferior direita e inferior esquerda. Neste caso, duas quinas serão do mesmo tamanho que as quinas já apresentadas no topo da página, e duas serão maiores.

Alguns outros estilos foram adicionados também no arquivo CSS para formatar algumas marcações HTML em particular. São eles:

```
h1
{
  color: #26c4f3;
  margin-left: 20px;
  font-size: 18pt;
}
```

O estilo `h1` irá formatar todas as marcações `<h1>` no arquivo HTML, colorindo seus textos com a cor azul `#26c4f3` por meio da propriedade `color`, e aplicará uma margem na lateral esquerda de 20 pixels, afastando o elemento da lateral esquerda do site, por meio da propriedade `margin-left`. Ainda neste estilo, foi definido que o texto terá o tamanho de 18 pontos, por meio da propriedade `font-size` com o valor `18pt`.

Já para formatar os parágrafos que apresentam o conteúdo da página, foi criado o seguinte estilo:

```
p
{
  margin-left: 20px;
}
```

A única formatação realizada para as marcações `<p>` é o espaçamento com a lateral esquerda do site, criando uma margem de 20 pixels por meio da propriedade `margin-left: 20px`.

Para as figuras apresentadas na página, optou-se em centralizá-las horizontalmente com os demais conteúdos. Para isso, o estilo `figure` foi criado:

```
figure
{
  text-align: center;
}
```

O estilo `figure` centraliza as imagens das marcações `<figure>` por meio da propriedade `text-align` definida como `center`. As legendas apresentadas nas marcações `<figcaption>` dentro das marcações `<figure>` são alinhadas no centro por consequência. E para alterar o tamanho da fonte das legendas das figuras, o estilo `figcaption` foi criado, definindo a propriedade `font-size` com 10 pontos:

```
figcaption
{
  font-size: 10pt;
}
```

Com estas formatações, o conteúdo principal da página está pronto. No próximo tópico vamos adicionar um conteúdo lateral na nossa página.

14.4. Conteúdo lateral

Boa parte das páginas na internet fazem o uso de conteúdo lateral para apresentar alguma informação secundária. Para ilustrar essa estrutura, o seguinte código foi adicionado ao arquivo HTML:

```
01 <div id="lateral">
02   <aside>
03     <a href="paginaFunciona.htm">
04       <p id="lateral">Como funciona?</p>
05     </a>
06     <a href="paginaCertificado.htm">
07       <p id="lateral">Certificado</p>
08     </a>
09     <a href="paginaInstrutores.htm">
10       <p id="lateral">Nossos instrutores</p>
11     </a>
12     <a href="paginaPagamento.htm">
13       <p id="lateral">Formas de pagamento</p>
14     </a>
15   </aside>
16 </div>
```

O conteúdo lateral de uma página deve ser apresentado por meio da marcação `<aside>`, existente entre as linhas 2 e 15, que organiza semanticamente essa estrutura. Para apresentá-la, optou-se em criar um `<div>` próprio, com a propriedade `id` definida como `lateral`, aberta e fechada respectivamente nas linhas 1 e 16, para formatar o conteúdo lateral posteriormente por meio do arquivo CSS. Neste caso, o seguinte estilo foi criado:

```
div#lateral
{
  float: left;
  width: 185px;
  margin: 5px;
  height: 350px;
  background-color: #ffffff;
  border-radius: 15px 50px 15px 50px;
}
```

Observe que o estilo `div#lateral` é muito parecido com o estilo `div#conteudo`, alterando apenas a largura do elemento, sem novidades a serem apresentadas no momento. A única observação é que o conteúdo lateral aparece ao lado direito do conteúdo principal, pois foi adicionado no HTML depois do bloco em questão. Caso você queira apresentar o conteúdo lateral do lado esquerdo, basta posicioná-lo antes do bloco do conteúdo principal em seu arquivo HTML.

Talvez o fator mais importante a ser considerado entre os estilos deste exemplo diz respeito às larguras. Cuidado para não errar na matemática! Se o elemento `<div>` raiz do site foi definido para ter 610 pixels de largura, a soma das larguras do elemento `<div>` do conteúdo principal e do conteúdo lateral, acrescido de suas margens, não pode ser superior a este valor. Caso seja, o elemento `<div>` raiz utilizará quebra de linha para que o elemento apresentado por último possa ser apresentado com seu tamanho maior em uma nova linha, ou seja, o conteúdo lateral poderá aparecer abaixo do conteúdo principal.

As contas sobre a largura dos elementos funcionam da seguinte forma: largura do primeiro elemento + margem esquerda do primeiro elemento + margem direita do primeiro elemento + largura do segundo elemento + margem esquerda do segundo elemento + margem direita do segundo elemento = largura do elemento raiz. Neste caso o elemento principal tem 405 de largura + 5 de margem esquerda + 5 de margem direita, que deve ser somado com a largura do conteúdo lateral que tem 185 de largura + 5 de margem esquerda + 5 de margem direita, o que totaliza 610, exatamente o tamanho comportado pelo elemento `<div>` raiz do site.

A intenção neste exemplo foi a de apresentar no conteúdo lateral algumas opções de links com dúvidas gerais que alguns usuários poderiam ter ao visualizar o conteúdo principal da página, como por exemplo, o funcionamento dos cursos da Softblue, certificado, instrutores e formas de pagamento. Para isso, foram criados quatro parágrafos, cada um deles com um link para acessar a respectiva página. O ponto de maior destaque no arquivo CSS é o fato das marcações `<p>` terem a propriedade `id` atribuída com o valor `lateral`. Isso foi feito porque os parágrafos do conteúdo lateral terão uma formatação visual um pouco diferente dos demais parágrafos da página. Para isso, o seguinte estilo foi adicionado no arquivo CSS:

```
p#lateral
{
  margin-left: 10px;
}
```

Para o conteúdo lateral, ficou estipulado que os parágrafos terão uma margem menor em relação ao lado esquerdo do conteúdo, neste caso definido pela propriedade `margin-left` apresentada com o valor `10px`.

Outra alteração interessante que pode ser feita com arquivos CSS é a alteração da cor dos links HTML. Observe os seguintes estilos:

```
a:link
{
  color: #26c4f3;
  text-decoration: none;
}

a:visited
{
  color: #26c4f3;
  text-decoration: none;
}

a:hover
{
  text-decoration: underline;
}
```

O estilo `a:link` define que os links ainda não visitados no site terão a cor azul por meio da propriedade `color` definida com o valor `#26c4f3`. Ainda, a propriedade `text-decoration` definida com o valor `none` informa que os links do site não devem aparecer com o tradicional efeito de sublinhado.

A mesma formatação foi aplicada aos links já visitados, por meio do estilo `a:visited`, atribuindo as mesmas características. E para adicionar o efeito de sublinhado ao link somente quando o usuário passar o mouse sobre o mesmo, foi definido o estilo `a:hover`, com o efeito `text-decoration: underline`.

Estamos quase chegando ao fim da construção da página. Para isso, o próximo tópico vai abordar a parte da estrutura do rodapé.

14.5. Rodapé da página

Para finalizar a página deste exemplo, o rodapé será construído. Seguindo os mesmos padrões já apresentados, uma marcação `<div>` específica será utilizada para formatar visualmente algumas propriedades do rodapé, que aparece criado pela marcação `<footer>` entre as linhas 2 e 4 do código-fonte apresentado a seguir:

```
01 <div id="rodape">
02   <footer>
```

```

03   <p><b>Softblue</b> - Todos os direitos reservados</p>
04   </footer>
05 </div>

```

O rodapé é bastante simples, apresentando apenas informações gerais da empresa, colocando o nome da mesma em negrito, mas sem relevância semântica. Sabe-se que o conteúdo apresentado nesta página é da Softblue. Contudo, o foco do conteúdo da página neste caso é o e-book de HTML. Por este motivo, optou-se em não dar maior relevância na palavra Softblue presente no rodapé.

Para formatar o rodapé, o seguinte estilo foi adicionado no arquivo CSS:

```

div#rodape
{
    float: left;
    width: 600px;
    margin: 5px;
    background-color: #ffffff;
    text-align: center;
    border-radius: 15px;
}

```

A essa altura do campeonato, estou certo de que você já compreende completamente as informações contidas neste estilo, que são basicamente as mesmas já apresentadas no decorrer deste e-book.

14.6. Código na íntegra

O código-fonte HTML completo do exemplo construído nos últimos tópicos pode ser baixado no site da Softblue, na área deste e-book, e também pode ser visualizado a seguir:

```

01 <!DOCTYPE html>
02 <html lang="pt-br">
03   <head>
04     <meta charset="utf-8">
05     <meta name="description" content="Site de cursos online">
06     <meta name="keywords" content="Java, PHP, HTML, Android, iOS">
07     <meta name="author" content="André Milani (Softblue)">
08     <title>Softblue</title>
09     <link rel="stylesheet" type="text/css" href="css/site.css">
10   </head>
11   <body>
12     <div id="raiz">
13
14       <div id="cabecalho">
15         <header>
16           <figure class="imagemMesmaLinhaLogotipo">
17             
19           </figure>
20
21           <nav>
22             <a href="paginaContato.htm">
23               <figure class="imagemMesmaLinhaMenu">

```

```

24         
25     </figure>
26 </a>
27 
29 <a href="paginaCursos.htm">
30     <figure class="imagemMesmaLinhaMenu">
31         
32     </figure>
33 </a>
34 
36 <a href="paginaContato.htm">
37     <figure class="imagemMesmaLinhaMenu">
38         
39     </figure>
40 </a>
41 </nav>
42 </header>
43 </div>
44
45 <div id="conteudo">
46     <section>
47         <article>
48             <header>
49                 <h1>Cursos online</h1>
50             </header>
51             <p>
52                 Quer aprender com qualidade? Venha fazer parte dos
53                 melhores, venha aprender programação na Softblue!
54                 Aprenda <strong>HTML</strong> por meio do nosso
55                 <strong>e-book de HTML</strong> agora mesmo!
56             </p>
57             <figure>
58                 
59                 <figcaption>E-book de HTML</figcaption>
60             </figure>
61         </article>
62     </section>
63 </div>
64
65 <div id="lateral">
66     <aside>
67         <a href="paginaFunciona.htm">
68             <p id="lateral">Como funciona?</p>
69         </a>
70         <a href="paginaCertificado.htm">
71             <p id="lateral">Certificado</p>
72         </a>
73         <a href="paginaInstrutores.htm">
74             <p id="lateral">Nossos instrutores</p>
75         </a>
76         <a href="paginaPagamento.htm">
77             <p id="lateral">Formas de pagamento</p>
78         </a>
79     </aside>
80 </div>
81
82 <div id="rodape">
83     <footer>
84         <p><b>Softblue</b> - Todos os direitos reservados</p>
85     </footer>

```

```

86     </div>
87
88     </div>
89 </body>
90 </html>

```

Código-Fonte 14.1. paginaEstruturada.htm

A seguir o código-fonte do arquivo CSS, que também pode ser baixado no site da Softblue, na área deste e-book:

```

01 body
02 {
03     background-color: #26c4f3;
04     font-family: Century Gothic, Tahoma, Arial;
05 }
06
07 @font-face
08 {
09     font-family: Century Gothic;
10     src: url('/fonts/GOTHIC.ttf');
11 }
12
13 @font-face
14 {
15     font-family: Century Gothic;
16     src: url('/fonts/GOTHICB.ttf');
17     font-weight: bold;
18 }
19
20 div#raiz
21 {
22     width: 610px;
23     margin: 0 auto;
24 }
25
26 div#cabecalho
27 {
28     float: left;
29     width: 600px;
30     margin: 5px;
31     background-color: #ffffff;
32     border-radius: 15px;
33 }
34
35 .imagemMesmaLinhaLogotipo
36 {
37     float: left;
38     margin: 10px;
39     margin-right: 30px;
40 }
41
42 .imagemMesmaLinhaMenu
43 {
44     float: left;
45     margin: 10px;
46     margin-top: 50px;
47 }
48
49 div#conteudo
50 {

```



```
51 float: left;
52 width: 405px;
53 height: 350px;
54 margin: 5px;
55 background-color: #ffffff;
56 border-radius: 15px 50px 15px 50px;
57 }
58
59 h1
60 {
61 color: #26c4f3;
62 margin-left: 20px;
63 font-size: 18pt;
64 }
65
66 p
67 {
68 margin-left: 20px;
69 }
70
71 figure
72 {
73 text-align: center;
74 }
75
76 figcaption
77 {
78 font-size: 10pt;
79 }
80
81 div#lateral
82 {
83 float: left;
84 width: 185px;
85 margin: 5px;
86 height: 350px;
87 background-color: #ffffff;
88 border-radius: 15px 50px 15px 50px;
89 }
90
91 p#lateral
92 {
93 margin-left: 10px;
94 }
95
96 a:link
97 {
98 color: #26c4f3;
99 text-decoration: none;
100 }
101
102 a:visited
103 {
104 color: #26c4f3;
105 text-decoration: none;
106 }
107
108 a:hover
109 {
110 text-decoration: underline;
111 }
112
```

```
113 div#rodape
114 {
115     float: left;
116     width: 600px;
117     margin: 5px;
118     background-color: #ffffff;
119     text-align: center;
120     border-radius: 15px;
121 }
```

Código-Fonte 14:2. site.css

15. Ufa, chegamos ao final de uma jornada!

Não foi fácil, mas certamente foi bastante produtivo. As principais marcações do HTML5 e ainda várias e várias propriedades do CSS 3 foram abordadas, capacitando você a trabalhar com essas tecnologias em seus projetos.

Obviamente, existem dezenas de outras marcações HTML e propriedades CSS que, com o tempo e necessidade, você pesquisará para incluir em seus projetos. O que importa mesmo neste momento é que você conquistou uma base sólida de conhecimento sobre esses assuntos e já pode utilizá-los em seu dia a dia.

Deixo aqui um grande abraço e meus mais sinceros parabéns por você ter concluído esta etapa! Encontro você em breve nos eventos ou materiais da Softblue. Até mais!

Lista de Figuras

Figura 3-1. Exemplo de página HTML	10
Figura 5-1. Título da página apresentado no topo da janela do navegador	15
Figura 5-2. Caracteres corretamente interpretados pelo navegador	16
Figura 5-3. Página simples com conteúdo	18
Figura 5-4. Formatando margens com CSS	19
Figura 6-1. Textos com quebras de linhas e espaços em branco sem marcações	25
Figura 6-2. Textos com quebras de linhas e espaços em branco com marcações	26
Figura 6-3. Diferentes tipos de alinhamentos de parágrafos via CSS	27
Figura 6-4. Trecho de parágrafo formatado isoladamente.....	28
Figura 6-5. Textos com formatações clássicas via marcações HTML	30
Figura 6-6. Textos com formatações clássicas via marcações HTML com relevância semântica	32
Figura 6-7. Texto marcado com relevância sem formatação.....	33
Figura 6-8. Texto marcado com relevância em diferentes situações	34
Figura 6-9. Textos com alinhamentos acima e abaixo do padrão	36
Figura 6-10. Texto com título principal e subtítulo.....	37
Figura 7-1. Artigos organizados por seção	41
Figura 7-2. Página com links (pontos de acesso).....	42
Figura 8-1. Página com imagem	45
Figura 8-2. Página com imagem marcada de conteúdo relevante e legenda	46
Figura 8-3. Página com vídeo	47
Figura 9-1. Página com lista sem numeração	49
Figura 9-2. Página com lista numerada	50
Figura 10-1. Tabela construída em HTML.....	52
Figura 11-1. Formulário construído em HTML	56
Figura 11-2. Preenchendo campos e selecionando valores via HTML.....	59
Figura 12-1. Página com painel	60
Figura 14-1. Exemplo fictício de página da Softblue para ser construída.....	65

Lista de Códigos-Fonte

Código-Fonte 3-1. exemploHtml.htm	9
Código-Fonte 5-1. marcacaoHtml.htm	12
Código-Fonte 5-2. marcacaoHtmlAtributoLang.htm	13
Código-Fonte 5-3. marcacaoHead.htm	14
Código-Fonte 5-4. marcacaoTitle.htm	15
Código-Fonte 5-5. marcacaoMeta.htm	16
Código-Fonte 5-6. marcacaoMetaOutrasConfiguracoes.htm	16
Código-Fonte 5-7. marcacaoBody.htm	17
Código-Fonte 5-8. estiloCssFormatoAtributo.htm	18
Código-Fonte 5-9. estiloCssFormatoMarcacao.htm	20
Código-Fonte 5-10. estiloCssNomeMarcacao.htm	20
Código-Fonte 5-11. arquivoDeEstilos.css	22
Código-Fonte 5-12. estiloCssFormatoArquivo.htm	22
Código-Fonte 6-1. textosSemMarcacao.htm	24
Código-Fonte 6-2. textosComMarcacao.htm	26
Código-Fonte 6-3. textosAlinhamentoParagrafos.htm	27
Código-Fonte 6-4. textosTrechosMarcados.htm	28
Código-Fonte 6-5. textosFormatacaoClassicaMarcacaoHtml.htm	29
Código-Fonte 6-6. textosFormatacaoClassicaEstiloCss.htm	31
Código-Fonte 6-7. textosFormatacaoClassicaMarcacaoHtmlRelevanciaSemantica.htm	32
Código-Fonte 6-8. textosMarcacaoStrongSemFormatacao.htm	32
Código-Fonte 6-9. textosMarcacoesMaiorRelevancia.htm	34
Código-Fonte 6-10. textosAlinhamentoAcimaAbaixo.htm	36
Código-Fonte 6-11. textosTitulos.htm	37
Código-Fonte 7-1. conteudoOrganizadoComoArtigo.htm	38
Código-Fonte 7-2. artigosOrganizadosPorSecao.htm	40
Código-Fonte 7-3. marcacaoA.htm	42
Código-Fonte 8-1. marcacaoImg.htm	44
Código-Fonte 8-2. marcacaoFigure.htm	45
Código-Fonte 8-3. marcacaoVideo.htm	47
Código-Fonte 9-1. marcacaoUl.htm	49
Código-Fonte 9-2. marcacaoOl.htm	50
Código-Fonte 10-1. marcacaoTableCaptionTrThTd.htm	52
Código-Fonte 11-1. marcacaoFormLabelInputSelect.htm	56
Código-Fonte 11-2. atributosCheckedSelectedDisabled.htm	59
Código-Fonte 12-1. marcacaoDiv.htm	60
Código-Fonte 13-1. paginaEstruturadaCrua.htm	63
Código-Fonte 14-1. paginaEstruturada.htm	80
Código-Fonte 14-2. site.css	82

Índice Remissivo

@font-face, 67, 68, 80
 <!--, 12, 13, 14, 63, 66
 <!DOCTYPE html>, 12
 <a>, 42, 70
 <article>, 38, 39, 40, 62, 63, 64, 73, 79
 <aside>, 63, 64, 75, 79
 , 29
 <body>, 9, 17, 18, 20, 24, 25, 27, 28, 29,
 31, 32, 33, 36, 37, 38, 40, 42, 44, 45, 47,
 51, 55, 58, 60, 62, 63, 64, 65, 66, 78

, 9, 25, 26, 35, 42, 56, 58
 <caption>, 51, 52
 <code>, 33, 35
 <div>, 60, 61, 65, 66, 68, 69, 71, 72, 75,
 76, 77
 , 33, 35
 <figcaption>, 45, 46, 70, 73, 75
 <figure>, 45, 46, 70, 71, 72, 73, 75, 79
 <footer>, 63, 64, 77, 79
 <form>, 54, 55, 56
 <h1>, 36, 41, 73, 74
 <h2>, 36, 41
 <h3>, 36
 <h4>, 36
 <h5>, 36
 <h6>, 36
 <head>, 9, 14, 15, 16, 17, 18, 19, 20, 22,
 24, 25, 27, 28, 29, 31, 32, 33, 36, 37, 38,
 39, 42, 44, 45, 47, 49, 50, 51, 55, 58, 60,
 63, 65, 66, 78
 <header>, 38, 39, 40, 41, 62, 63, 64, 69,
 73, 78, 79
 <html>, 12, 13, 14, 15, 17, 18, 65, 66
 <i>, 29
 , 44, 45, 46, 70, 71
 <input>, 55, 56, 57, 58, 59
 <kbd>, 33
 <label>, 55, 56, 57
 , 49, 50
 <link>, 22, 57, 66
 <nav>, 62, 63, 64, 69, 70, 78
 , 50
 <option>, 55, 58
 <p>, 21, 25, 28, 33, 34, 36, 37, 38, 73, 74,
 76, 78, 79
 <s>, 29
 <samp>, 33, 35
 <section>, 39, 40, 62, 63, 64, 73, 79
 <select>, 55, 58, 59
 <source>, 47
 , 27, 28, 30
 , 31, 32, 34, 35, 73
 <sub>, 35, 36
 <sup>, 35, 36
 <table>, 9, 51, 52
 <td>, 9, 51, 53
 <th>, 51, 53
 <title>, 14, 18, 65, 66
 <tr>, 9, 51, 52, 53
 <u>, 29
 , 49, 50
 <var>, 33, 35
 <video>, 47, 48
 -->, 12, 13, 14, 63, 66
 a: hover, 77
 a: link, 77
 a: visited, 77
 action, 54, 55, 58
 alt, 44, 45, 46, 69, 73, 78, 79
 author, 16, 66, 78
 border, 51, 52, 71, 74, 76, 78, 80, 81, 82
 charset, 9, 15, 16, 17, 18, 19, 20, 22, 24,
 25, 27, 28, 29, 31, 32, 33, 36, 37, 38, 39,
 42, 44, 45, 47, 49, 50, 51, 55, 58, 60, 63,
 66, 78
 checkbox, 55, 56, 57, 58, 59
 checked, 58, 59
 class, 19, 20, 21, 22, 27, 28, 31, 32, 60, 61,
 69, 71, 73, 78, 79
 color, 28, 61, 67, 71, 74, 76, 77, 78, 80, 81,
 82
 colspan, 51, 52, 53
 content, 16, 66, 78
 controls, 47

CSS, 10, 11, 14, 18, 19, 20, 21, 22, 23, 24,
 26, 27, 28, 30, 31, 32, 35, 53, 60, 61, 64,
 66, 67, 68, 69, 70, 71, 73, 74, 75, 76, 77,
 78, 80, 83
 description, 16, 66, 78
 disabled, 55, 58, 59
 float, 71, 72, 74, 76, 78, 80, 81, 82
 font-family, 28, 29, 67, 68, 80
 font-size, 21, 28, 29, 74, 75, 81
 font-weight, 30, 32, 68, 80
 get, 54
 height, 47, 61, 74, 76, 81
 href, 22, 24, 25, 27, 28, 29, 31, 32, 33, 36,
 37, 38, 39, 42, 44, 45, 47, 49, 50, 51, 55,
 58, 60, 66, 69, 75, 78, 79
 HTML, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16,
 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,
 28, 29, 30, 31, 32, 33, 35, 36, 37, 38, 39,
 42, 43, 44, 45, 47, 48, 49, 50, 51, 52, 53,
 54, 55, 56, 57, 58, 59, 60, 62, 63, 65, 66,
 68, 71, 72, 73, 74, 75, 76, 77, 78, 79, 83
 HTML5, 10, 12, 48, 62, 83
 Hyper Text Markup Language, 7
 id, 21, 66, 68, 69, 71, 73, 75, 76, 77, 78, 79
 keywords, 16, 66, 78
 LATIN1, 17
 left, 26, 71, 72, 74, 76, 78, 80, 81, 82
 marcações, 9
 margin, 18, 19, 20, 22, 68, 71, 72, 74, 76,
 77, 78, 80, 81, 82
 margin-left, 18, 19, 20, 22, 74, 77, 81
 margin-right, 72
 margin-top, 18, 19, 20, 22, 72, 80
 meta, 9, 14, 15, 16, 17, 18, 19, 20, 22, 24,
 25, 27, 28, 29, 31, 32, 33, 36, 37, 38, 39,
 42, 44, 45, 47, 49, 50, 51, 55, 58, 60, 63,
 65, 66, 78
 method, 54, 55, 58
 name, 16, 55, 56, 57, 58, 66, 78
 number, 55, 56, 57, 58
 post, 54, 55, 58
 radio, 55, 56, 57, 58
 rel, 22, 24, 25, 27, 28, 29, 31, 32, 33, 36,
 37, 38, 39, 42, 44, 45, 47, 49, 50, 51, 55,
 58, 60, 66, 78
 reset, 55, 56, 58
 rowspan, 51, 52, 53
 selected, 58, 59
 semântica, 10, 12, 16, 31, 32, 46, 62, 69,
 78
 sites de buscas, 10
 src, 9, 44, 45, 47, 67, 68, 69, 73, 78, 79, 80
 style, 18, 19, 20, 22, 30
 stylesheet, 22, 24, 25, 27, 28, 29, 31, 32,
 33, 36, 37, 38, 39, 42, 44, 45, 47, 49, 50,
 51, 55, 58, 60, 66, 78
 submit, 55, 56, 58
 text, 22, 24, 25, 26, 27, 28, 29, 30, 31, 32,
 33, 36, 37, 38, 39, 42, 44, 45, 47, 49, 50,
 51, 55, 56, 58, 60, 66, 75, 77, 78, 81, 82
 text/css, 22, 24, 25, 27, 28, 29, 31, 32, 33,
 36, 37, 38, 39, 42, 44, 45, 47, 49, 50, 51,
 55, 58, 60, 66, 78
 text-decoration, 30, 77, 81
 type, 22, 24, 25, 27, 28, 29, 31, 32, 33, 36,
 37, 38, 39, 42, 44, 45, 47, 48, 49, 50, 51,
 55, 56, 58, 60, 66, 78
 underline, 30, 77, 81
 UTF-8, 15, 17
 value, 55, 56, 57, 58, 59
 web 2.0, 8
 width, 47, 61, 68, 71, 74, 76, 78, 80, 81, 82