



Ferramentas de sistema web

Ferramentas para sistemas web

Virgínia Mara Cardoso

© 2017 por Editora e Distribuidora Educacional S.A.
Todos os direitos reservados. Nenhuma parte desta publicação poderá ser reproduzida ou transmitida de qualquer modo ou por qualquer outro meio, eletrônico ou mecânico, incluindo fotocópia, gravação ou qualquer outro tipo de sistema de armazenamento e transmissão de informação, sem prévia autorização, por escrito, da Editora e Distribuidora Educacional S.A.

Dados Internacionais de Catalogação na Publicação (CIP)

Cardoso, Virgínia Mara
C268f Ferramentas para sistemas web / Virgínia Mara Cardoso.
– Londrina : Editora e Distribuidora Educacional S.A., 2017.
 80 p.

ISBN 978-85-8482-621-6

1. Sites da web. I. Título.

CDD 005.3

2017
Editora e Distribuidora Educacional S.A.
Avenida Paris, 675 – Parque Residencial João Piza
CEP: 86041-100 – Londrina – PR
e-mail: editora.educacional@kroton.com.br
Homepage: <http://www.kroton.com.br/>

Sumário

Tema 1 Evolução da internet e ferramentas para projeto e construção de websites: estrutura básica da linguagem HTML	7
Tema 2 Arquivos de páginas web e o (X)HTML	25
Tema 3 Trabalhando com imagens e links	47
Tema 4 Estilos utilizados – CSS e HTML	63

Convite à leitura

A leitura deste tema possibilitará a você conhecer um assunto de muito interesse nos dias atuais, que são as ferramentas para um sistema web. Para iniciar, vamos entender melhor a evolução da Internet, esta grande rede das redes que movimenta as informações do mundo todo e nos dá acesso a toda hora. Temos a Internet, mas como podemos ter tanta informação nesta rede, ou melhor, como esses documentos ficam inseridos? Como devem ser estes documentos? Há uma maneira especial para montarmos as páginas que encontramos quando navegamos? Estas questões são tratadas nesse tema e, além disso, vamos aprender também a colocar um documento na web se desejarmos. Mas, para isso, iniciaremos aqui a discorrer sobre a linguagem HTML (HyperText Markup Language), mostrando sua base e suas versões até o (X)HTML (Extensible HyperText Markup Language), e também sobre o CSS (Cascade Style Sheets), que veio para agregar funcionalidades e permitir o desenvolvimento de páginas mais sofisticadas em termos de design e estruturas de distribuição de elementos na tela. É, ainda, abordado o discurso sobre a estrutura básica do HTML, para que possamos começar a aprender a desenvolver a nossa página. Concluindo o assunto, neste tema é mostrado um pequeno exemplo de como iniciar o desenvolvimento de uma página web.

Evolução da internet e ferramentas para projeto e construção de websites: estrutura básica da linguagem HTML



POR DENTRO DO TEMA

Evolução da internet

A evolução da comunicação permite e facilita o acesso às notícias e acontecimentos do mundo. Nos dias atuais, convivemos com a disseminação da informação com tanta facilidade que nem notamos a tecnologia que está nos auxiliando. Enviamos e-mails, conversamos, trocamos fotos e vídeos com grupos de amigos, às vezes utilizando telefones celulares, outras vezes recorrendo a notebooks, tablets, dentre outros equipamentos. Se desejamos viajar, logo estaremos com o destino conhecido através de imagens e já poderemos fazer a reserva do hotel. Lidamos com as contas mensais e com a conta bancária dentro de nossas casas. Se precisamos de livros, entramos virtualmente em várias bibliotecas de várias partes do planeta. Temos acesso a diversas tarefas, bem como a entretenimentos, tudo isso porque temos uma conexão com o mundo através de um sistema de comunicação denominado Internet. Se buscarmos um conceito, encontraremos que essa é a rede das redes.

Para conceituar melhor a palavra, temos que “nets” são redes e que “inter” quer dizer que há interação; neste caso, trata-se de redes ligadas, conectadas entre si. Sendo assim, a Internet consiste em várias redes interligadas, com serviços comuns.

Essa interconexão, ou seja, a Internet, teve início em 1969. A história nos descreve que na década de 1960 ainda não havia redes e tínhamos apenas computadores sem ligação uns com os outros, que trabalhavam isoladamente. Depois de muitas pesquisas surgiu a ideia de conectar computadores para compartilharem dados. De acordo com Mileto (2014, p. 16), a Internet foi criada por Tim Berners-Lee quando ele realizava pesquisas para interligar e compartilhar documentos entre os pesquisadores. Foi

somente em 1969 que a ARPA (Advanced Research Projects Agency) do departamento de defesa dos Estados Unidos apresentou a ARPANET, que consistia em uma pequena rede de computadores conectados.

A Internet é constituída por redes locais e utiliza serviços de provedores de acesso à Internet. A Internet não é de responsabilidade governamental, mas é atuada por empresas privadas. Seus provedores podem ser internacionais, nacionais, regionais ou locais.

Conforme a definição de Castro (2010), a Internet reúne computadores conectados uns com os outros. Quanto às conexões, algumas pessoas têm internet banda larga em alta velocidade por 24 horas diárias, através de **DSL** (Digital Subscriber Line), cabo ou satélite, e outras utilizam o **modem** para conectar seus computadores pessoais, mas por um determinado período de tempo. O que é relevante é que todos podem estar, de uma maneira ou de outra, conectados à Internet, obtendo informações do mundo todo.

A Web

Neste contexto de Internet, sabemos que temos acesso a diversas informações, mas podemos questionar: visto que são vários tipos de documentos, como todos são inseridos e como conseguimos com tanta facilidade consultá-los? Neste caso, temos a World Wide Web – ou WWW – que se trata de um sistema de documentos em **hipermídia** que são interligados e executados na Internet. Para Castro (2010), a World Wide Web pode transformar milhões de documentos para que possam ser acessados pela Internet, mas esses documentos são escritos, de alguma forma, utilizando o HTML (HyperText Markup Language). Para acompanhar toda a evolução da Internet a Figura 1.1 mostra em síntese este desenvolvimento.

Figura 1.1 A Evolução da Internet.

Resumo histórico do surgimento da Internet	
1991	Surgimento da lista de discussão eletrônica WWW-talk, cujo foco é compartilhar experiências relacionadas com a linguagem HTML.
1992	Desenvolvimento do navegador Mosaic.
1993	Criação do navegador de texto Lynx.
1994	Estabelecimento da especificação HTML 2 (unificação das diferentes formas de HTML). Criação da empresa Netscape Communications Corporation. Surgimento do W3C (<i>World Wide Web Consortium</i>).
1995	Definição de novas marcações para o HTML. Criação do conceito de frames. Lançamento do navegador Internet Explorer 2.0.
1996	Lançamento do HTML versão 4.
1997	O HTML passa a incorporar tabelas, <i>applets</i> , subscritos, sobreescritos e texto combinado com imagens.
1998	Reformulação do HTML em XML.
1999	Lançamento do XHTML 1.0.
2004	Criação do grupo WHATWG (<i>Web Hypertext Application Technology Working Group</i>).
2007	Início do desenvolvimento do HTML 5, que incorpora TAGs para animações.

Fonte: Adaptado de Mileto (2014).

É notável que a Internet mudou os hábitos de muitas pessoas no que diz respeito à comunicação, entretenimentos, comportamentos e também consumos. Neste cenário, as aplicações web tiveram que acompanhar todo este desenvolvimento com a utilização de métodos e ferramentas específicas. As aplicações web podem ser classificadas da seguinte maneira (MILETO, 2014):

1. Site de conteúdo: exibe o conteúdo de uma determinada área de forma linear, com recursos de tabelas, imagens, gráficos, dentre outros. Possui navegação simples.
2. Site de registro de dados ou entrada do usuário: utiliza como recurso principal formulários que são preenchidos pelo usuário, com dados relacionados com o objetivo do site.
3. Portal: é um conjunto de páginas; links que convergem para um determinado contexto.
4. Aplicação orientada à transação: é um modelo em que o usuário envia solicitações que a aplicação processa em um banco de dados, retornando ao usuário a resposta.

Baseado em Mileto (2014), somente depois que é conhecido o propósito da aplicação é que se deve desenvolver os modelos de projeto e, na etapa final, deve ocorrer a construção das páginas com o uso de tecnologias específicas para web, como HTML, CSS, PHP, Javascript e também ferramentas específicas para edição de códigos.

HTML

De acordo com Castro (2010), o HTML é a abreviação de HyperText Markup Language, que quer dizer linguagem de marcação de hipertexto, ou seja, é a linguagem em que são escritas as páginas da web que são utilizadas e, na maioria das vezes, é facilmente interpretada pelo navegador. Esta linguagem apresenta duas características: o hipertexto e a universalidade. O hipertexto são as informações ligadas na forma de páginas com interconexões entre elas - os links, em que a informação na web pode ser acessada em direções diferentes. Os documentos em HTML são salvos como arquivos de texto de modo que qualquer computador possa ler a página, não importando seu sistema operacional, ou seja, a web é aberta para todos, o que causa a universalidade.

Na década de 1990, o HTML ganhou várias versões e, como já dito, foi criado para ser uma linguagem independente de plataformas, browsers e outros meios de acesso. O HTML pode ser entendido universalmente, possibilitando a reutilização da informação de acordo com as limitações de cada meio de acesso. Assim, várias organizações se uniram, devido ao não contentamento com detalhes, e escreveram o que hoje chamamos de HTML5 (FERREIRA, 2013). A W3C, organização World

Wide Web Consortium, tem como objetivos trabalhar com a comunidade web a importância da universalidade, melhorar a aparência das páginas e evitar novos problemas.

As versões mais antigas reuniam conteúdo e instruções de formatação em um único documento. Mas, foi criado um novo sistema de instruções de formatação, o CSS (Cascade Style Sheets), que trouxe novas funcionalidades, em particular, a capacidade de posicionar elementos em uma página web com maior precisão, como também criar layouts de aparência profissional. Como o HTML não era uma plataforma suficientemente firme para que se pudesse construir sobre ela, o W3C decidiu que era necessária uma estrutura maior, surgindo, assim, o XML (Extensible Markup Language). Embora seja muito parecido com o HTML, o XML não é somente uma linguagem para criar páginas web, mas sim uma linguagem para criar outras linguagens. Neste âmbito, houve uma solução ideal, quando o W3C reescreveu o HTML em XML: tendo os recursos do HTML com a sintaxe e flexibilidade do XML, surgiu o nomeado XHTML. Sendo assim, o XHTML e o CSS se tornaram uma combinação poderosa para os web designers (CASTRO, 2010).

Estrutura Básica da Linguagem HTML

Se observarmos as páginas da web atualmente, poderemos dizer que estas têm se tornado mais complicadas ou difíceis. Porém, se olharmos através de outro foco, é perceptível que a estrutura fundamental se mantém simples desde o início. De acordo com Castro (2010), a página web é feita de três componentes principais:

1. Conteúdo de texto: são os cabeçalhos e parágrafos da página.
2. Referências: são os links, imagens e animações em flash.
3. Marcação: regras ou instruções para descrever as referências e o conteúdo a serem exibidos.

Mas, além destes três componentes, as páginas web abrangem informações sobre a linguagem e a codificação, ou seja, o script no qual o texto foi escrito e também o **doctype**.

Marcação

São as instruções de formatação, em que se incluem também detalhes sobre os relacionamentos que existem entre as partes dos documentos. Há três tipos de marcação (CASTRO, 2010):

1. Elementos são etiquetas que identificam e dão estrutura às diferentes partes da página da web. Essas partes podem ser um parágrafo, um cabeçalho ou uma informação que mereça destaque, que seja importante. Os elementos podem conter mais de um atributo, conter texto ou podem ser vazios. São delimitados por

tags, que são comandos específicos da linguagem. Um elemento não vazio inclui uma tag de abertura, o conteúdo e uma tag de fechamento. A tag de abertura contém o nome do elemento e atributos que estão escritos e inclusos dentro dos sinais de maior e menor. A tag de fechamento também fica dentro dos sinais de maior e menor, porém há uma barra diagonal seguida pelo nome do elemento. Na Figura 1.2 podemos verificar como é um elemento não vazio e um vazio.

Figura 1.2 Elementos.



Fonte: Adaptado de Castro (2010).

De acordo com Mileto (2014), os elementos que compõem uma página são posicionados por meio de tags. Para conceituar melhor, uma tag é uma palavra específica que deve ser colocada entre os sinais de maior e menor. De modo geral, aparecem em pares, para uma indicar o início e a outra o fim da marcação, como já mostrado na Figura 1.2. Para criar uma página web devem ser utilizadas tags básicas, como mostra a Figura 1.3.

Figura 1.3 Tags utilizadas.

TAGs HTML	
TAG	Função
<code>html</code>	Marca o início e o fim da página Web, informando ao navegador que o texto contido no documento está escrito em HTML.
<code>head</code>	Marca o início e o fim do cabeçalho, a área onde serão descritos os cabeçalhos e o título da página. Pode ser utilizado, ainda, para declarar scripts em JavaScript ou definir formatações CSS.
<code>title</code>	Marca o início e o fim do título da página, que sempre está posicionado na barra superior do browser.
<code>body</code>	Marca o início e o fim do corpo da página, que contém imagens, textos, títulos, links, etc.

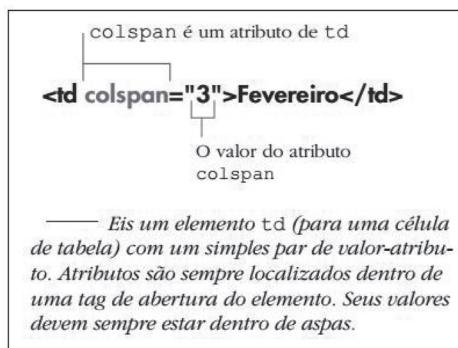
Fonte: Adaptado de Mileto (2014).

2. Atributos contêm informações sobre os dados em um documento. Alguns atributos aceitam qualquer valor, mas outros são mais limitados.

3. Valores devem estar sempre entre aspas. Dentro do conjunto de atributos, alguns aceitam quaisquer valores e outros têm suas limitações. Quando os atributos fazem referência a outros arquivos, estes devem conter valores na forma de **URL** (Uniform Resource Locator).

Para melhor compreensão, na Figura 1.4 é apresentado o exemplo de um atributo simples, em que é possível ver como ele deve ser escrito.

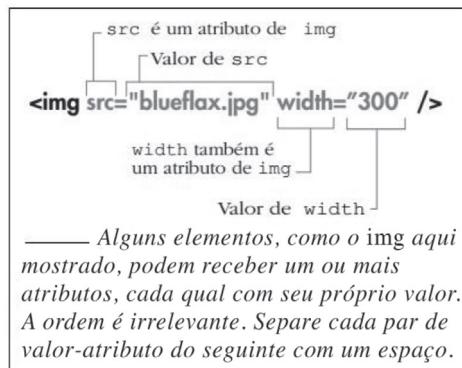
Figura 1.4 Atributo simples.



Fonte: Adaptado de Castro (2010).

Na Figura 1.5 são apresentados atributos, mas neste caso cada um tem seu próprio valor. Estes valores se apresentam entre aspas, mas, em HTML, às vezes as aspas podem ser omitidas.

Figura 1.5 Atributos com valores.



Fonte: Adaptado de Castro (2010).

Na sequência, a Figura 1.6 apresenta um exemplo de atributos que não aceitam quaisquer valores.

Figura 1.6 Atributos com valores específicos.

```
<link rel="folhadeestilo" type="text/css"
      media="screen" href="blueflax.css" /
```

↓
Valor pré-definido

Alguns atributos apenas aceitam valores específicos. Por exemplo, o atributo media no elemento link pode ser ajustado para screen, handheld ou print, entre outros, mas você não pode criar um valor para ele.

Fonte: Adaptado de Castro (2010).

Abordando ainda os elementos, estes podem ser de nível parágrafo ou linha. Elementos de nível parágrafo são as partes estruturais que podem ser consideradas as maiores da página web. Estes elementos podem conter outros elementos de nível parágrafo, linha e texto. Sobre a estrutura de parágrafos, a tag responsável pela quebra de parágrafos é a <P>, que finaliza o parágrafo e insere automaticamente uma linha em branco entre parágrafos. O elemento <div> é utilizado para fornecer uma identidade, uma referência (PRADO, 2011). Um exemplo de elemento de nível parágrafo e de nível linha pode ser visto na Figura 1.7.

Figura 1.7 Elementos de nível parágrafo.

```
<div>
<p>Sempre fico surpresa com as belas e delicadas Blue Flax que, de alguma forma, aparecem em meu jardim.
Elas são abundantes de cor todas as manhãs, mas nem uma única flor resiste à tarde.
São a própria definição de <em>efêmero</em>.</p>
<p>&copy; 2002 by Blue Flax Society.</p>
</div>
```

Os elementos nível de parágrafo, aqui exibidos destacados em negrito, são div e p. Os elementos de linha, destacados mas não em negrito, são img e em.

Fonte: Adaptado de Castro (2010).

Dentre os elementos, pode-se dizer que há uma pequena hierarquia, ou seja, são designados aqueles que são pais e, consequentemente, os filhos. Para esta classificação é estabelecido que quando um elemento contém o outro ele é o pai, e o elemento contido é o filho. Os filhos são os descendentes, ou seja, é criada uma família em uma página web (CASTRO, 2010). Na Figura 1.8 temos um exemplo dessa hierarquia dos elementos.

Na página web temos que usar caracteres e, às vezes, símbolos especiais. Para isso, há a opção de digitar os caracteres como são e codificar em **Unicode**, sendo que os navegadores que não compreendem Unicode interpretam a porção ASCII do documento. Há um detalhe importante: o símbolo & não deve ser digitado diretamente, visto que possui um significado especial em (X)HTML.

Figura 1.8 Exemplo de elementos pais e filhos.

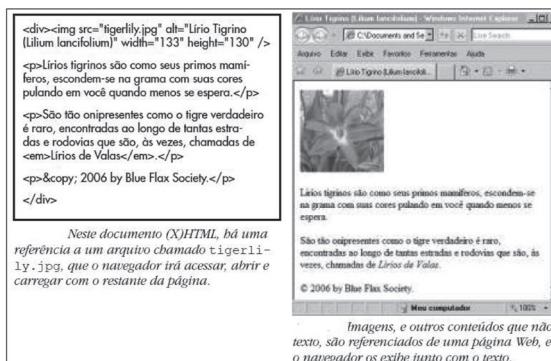
```
<div>
  
  <p>... de
  <em>efêmero</em>
  </p>
  <p>...pela Blue Flax Society</p>
</div>

O elemento div é pai dos elementos
img e todos os p. Por outro lado, os elementos
img e todos os p são filhos (e descendentes) de
div. O primeiro elemento p é pai da tag em. O
em é filho do primeiro p e, também, descendente
(mas não filho) de div.
```

Fonte: Adaptado de Castro (2010).

Um destaque importante e que fascina os navegadores são os links, imagens, animações em flash, filmes em QuickTime, músicas em MP3, dentre outros. Estes arquivos são salvos separadamente e dentro da página, sendo referenciados no texto. Na Figura 1.9 pode-se ver a referência a uma imagem.

Figura 1.9 Exemplo de imagens na página.



Fonte: Adaptado de Castro (2010).

Como todo documento, a página web precisa ser identificada e, assim, é necessário dar-lhe o nome de um arquivo. Esse nome deve ter letras minúsculas e sua extensão deverá ser htm ou html.

URL

Não podemos deixar de abordar a URL (Uniform Resource Locator), que contém informações sobre onde o arquivo está e a sua funcionalidade, ou seja, o que o navegador pode fazer com este arquivo. Cada arquivo na Internet tem uma única URL. A URL é dividida em partes, sendo a primeira o esquema, que informa como lidar com o arquivo que será aberto. Este, na maioria das vezes, é seguido de dois pontos e duas barras diagonais. Um exemplo de esquema comum é o HTTP (Hypertext Transfer Protocol), o protocolo de transferência de hipertexto, utilizado para acessar páginas web. A próxima parte é o nome do servidor onde o arquivo está localizado, e também o caminho que leva ao arquivo, como pode ser visto na Figura 1.10.

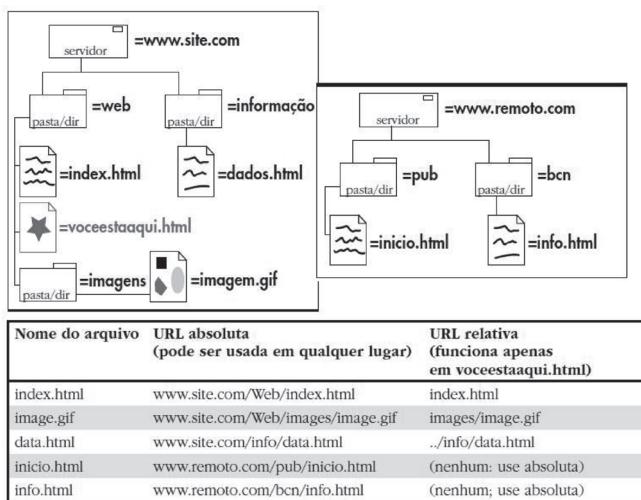
Figura 1.10 Exemplo de URL básica.



Fonte: Adaptado de Castro (2010).

As URLs podem ser absolutas ou relativas. Uma URL é absoluta quando mostra todo o caminho para o arquivo: esquema, nome do servidor, caminho completo e nome do arquivo. Ela é dita completa, o que significa que não tem influência sobre a localização do arquivo referenciado, ou seja, não importa se é uma página web neste ou naquele servidor, ela parecerá exatamente a mesma. A URL absoluta é necessária quando referencia um arquivo do servidor de alguém, para sites **FTP** ou URL que não usem protocolo HTTP. Quando a URL é relativa, a localização do arquivo no servidor é relativa, como mostra a Figura 1.11, junto com o quadro que mostra as URLs e onde podem ser usadas.

Figura 1.11 URLs Absolutas e Relativas.



Fonte: Adaptado de Castro (2010).

Exemplo

Como já foi explanada toda a evolução da linguagem e seus principais elementos, é importante conhecer os requisitos básicos para o desenvolvimento de uma página. De acordo com Prado (2011), temos como princípios básicos: conhecer a linguagem HTML e, depois, abrir um editor de texto para gerar o código fonte. O editor pode ser um bloco de notas. Para podermos visualizar as páginas, é necessário um navegador de Internet, o browser. Sobre os comandos, já conhecemos as Tags, que fazem indicações a um browser e devem aparecer entre sinais de maior e menor, utilizadas aos pares e, além disso, a tag de finalização de um comando deve ser precedida de uma barra. O documento HTML fica contido entre as tags:

`<HTML>` e `</HTML>`, que indicam o início e o fim do documento.

`<HEAD>` e `</HEAD>`, que indicam parâmetros de configuração do documento.

`<TITLE>` e `</TITLE>`, que indicam o título do documento para o browser e devem estar dentro das tags `<HEAD>` e `</HEAD>`.

`<BODY>` e `</BODY>`, onde fica o corpo do documento.

Agora, com alguns conhecimentos, será montado um pequeno exemplo: uma página com uma frase e uma cor de fundo. Primeiramente, devemos ter uma pasta específica para o arquivo e abrir o editor de texto, nomeando o arquivo como `index.html`. Caso você tente abri-lo, vai aparecer uma página em branco, pois o

arquivo está vazio. Para digitar o código vá ao seu arquivo e, com o botão direito do mouse, clique em “editar”. Assim, você inclui o seu código, que pode ser:

```
<html>
<head>
<title>
    Exemplo de HTML
</title>
</head>
<body bgcolor="blue" >
    Primeira página HTML com cor de fundo.
</body>
</html>
```

Salve o seu código, feche o seu arquivo e, depois, abra o arquivo HTML, e você poderá ver a sua página. Na linha sete foi incluída a instrução “bgcolor” para indicar a cor do fundo. Como você já iniciou a construção de uma página, tente mudar a sua cor do fundo. Agora você já sabe como iniciar a montagem de uma página.



ACOMPANHE NA WEB

Evolução da Internet: do início secreto à explosão mundial (TAIT, 2007)

Para o melhor entendimento sobre a evolução da Internet, é indicada a leitura de um artigo que discorre sobre este assunto com uma linguagem bem objetiva e com ricos detalhes.

Disponível em: <<http://homepages.dcc.ufmg.br/~mlbc/cursos/internet/historia/Brasil.html>>. Acesso em: 03 nov. 2014.

História da Internet

Sobre o assunto abordado neste tema e também para uma assimilação melhor do que foi a evolução da Internet, é indicado um vídeo bem explicativo.

Disponível em: <<https://www.youtube.com/watch?v=sqoxRynXxyg>>. Acesso em:

03 nov. 2014.

Tempo: 7:44.

A História do HTML (WILLIAM, 2012)

Sobre a Linguagem HTML, item relevante neste tema, é feita a indicação de leitura de um artigo, que trata da história dessa linguagem de uma forma bem clara. No artigo, o autor abrange todos os tópicos pertinentes a este assunto.

Disponível em: <<https://fit.faccat.br/~sorgetz/ArtigoHTML.pdf>>. Acesso em: 04 nov. 2014.

Apostila de HTML e CSS (PIVETTA, 2010)

É indicada a você uma apostila que abrange a linguagem HTML e o CSS. A apostila apresenta no Capítulo 1 uma introdução ao HTML e, no Capítulo 10, discorre sobre o que é o CSS.

Disponível em: <<https://www.telecom.uff.br/pet/petws/downloads/tutoriais/css/css2k80912.pdf>>. Acesso em: 03 nov. 2014.



AGORA É A SUA VEZ

Instruções:

Agora, chegou a sua vez de exercitar seu aprendizado. A seguir, você encontrará algumas questões de múltipla escolha e dissertativas. Leia cuidadosamente os enunciados e atente-se para o que está sendo pedido.

Questão 1

A Internet, a rede das redes, é uma rede mundial largamente utilizada. A maioria dos usuários não sabe como ela funciona e não conhece a sua estrutura, mas se aproveita dos seus recursos. Como você conceituaría a Internet?

Questão 2

Sobre a classificação das aplicações web, confira as afirmativas e responda:

- I. Site de conteúdo: exibe o conteúdo de uma determinada área de forma linear, com navegação simples.
- II. Site de registro de dados: utiliza como recurso principal sites já existentes.

Portal: é um conjunto de páginas; links que convergem para um determinado contexto.

Aplicação orientada à transação: é um modelo em que o usuário envia solicitações que a aplicação processa em um site, retornando ao usuário a resposta.

- a) Todas as afirmativas estão corretas.
- b) Somente as afirmativas I, II e III estão corretas.
- c) Somente as afirmativas II, III e IV estão corretas.
- d) Somente as alternativas I e III estão corretas.
- e) Somente as alternativas I e IV estão corretas.

Questão 3

Relacione:

1. HTML	() Sistema novo de instruções de formatação, com novas funcionalidades, como a capacidade de posicionar elementos em uma página web com maior precisão e criar layouts de aparência profissional.
2. CSS	() Não é somente uma linguagem para criar páginas web e sim uma linguagem para criar outras linguagens.
3. W3C	() Linguagem de marcação de hipertexto, ou seja, é a linguagem em que são escritas as páginas da web.
4. XML	() Organização que tem como objetivo trabalhar com a comunidade web a importância da universalidade e melhorar a aparência das páginas.

Questão 4

As páginas da web atualmente têm se tornado mais complicadas, mas a estrutura fundamental se mantém simples desde o início. Cite os três componentes principais com que as páginas são feitas.

Questão 5

Recorrendo a algumas informações mencionadas neste caderno sobre a linguagem HTML, monte um pequeno exemplo de uma página com uma frase e uma cor de fundo.



FINALIZANDO

Este tema possibilitou a você o conhecimento sobre a evolução da Internet. Agora, você pode entender melhor o processo de colocar documentos nesta rede e poder acessá-los, mas com detalhes sobre as linguagens utilizadas. Sobre este assunto, foi narrado a respeito da linguagem HTML e suas versões, bem como o CSS. Você pôde começar a conhecer e entender os fundamentos básicos da linguagem HTML, quais são os comandos básicos, como iniciar a construção de uma página e quais são os recursos que podem ser utilizados. Para completar este assunto, você pôde ver e fazer um exemplo básico de uma página com uma frase e uma cor de fundo.



REFERÊNCIAS

CASTRO, Elizabeth. HTML5 e CSS3: guia prático e visual. São Paulo: Alta Books, 2010.

FERREIRA, Elcio; EIS, Diego. HTML 5. Rio Grande do Sul: Oficina da Net, 2011. Disponível em: <<http://www.oficinadanet.com.br/images/apostilas/703/html5-web.pdf>>. Acesso em: 04 nov. 2014.

MEYER, Érica A. Smashing CSS. Técnicas profissionais para um layout moderno. Dados Eletrônicos. Porto Alegre: Bookman, 2011.

MILETO, Evandro Manara; BERTAGNOLLI, Sílvia de Castro. Desenvolvimento de Software II: introdução ao desenvolvimento web com HTML, CSS, JavaScript e PHP. Dados Eletrônicos. Porto Alegre: Bookman, 2014.

PIVETTA, Elisa Maria. Apostila de HTML e CSS. Colégio Agrícola Frederico Westphalen – UFSM. 2010. Disponível em: <<https://www.telecom.uff.br/pet/petws/downloads/tutoriais/css/css2k80912.pdf>>. Acesso em: 03 nov. 2014.

PRADO, Márcio Feliciano do. Apostila de HTML e CSS. Minicursos Virtuais. Campinas: Unicamp, 2002. Disponível em: <<http://www.marcoprado.eti.br/ensino.html?download=6%3Aapostilahtml>>. Acesso em: 05 nov. 2014.

TAIT, Tania Fatima Calvi. Evolução da Internet: do início secreto à explosão mundial. PET Informática. 2007. Disponível em: <<http://homepages.dcc.ufmg.br/~mlbc/cursos/internet/historia/Brasil.html>>. Acesso em: 03 nov. 2014.

VÍDEO. História da Internet. YouTube. 2013. Disponível em: <<https://www.youtube.com/watch?v=sqoxRynXxyg>>. Acesso em: 03 nov. 2014.

WILLIAM, David. A História do HTML. Front End Brasil. 2012. Disponível em: <<https://fit.faccat.br/~sorgetz/ArtigoHTML.pdf>>. Acesso em: 04 nov. 2014.



GLOSSÁRIO

Doctype: tipo de marcação que descreve o texto escrito na página web.

DSL: Digital Subscriber Line, tecnologia de transmissão digital de dados via rede de telefonia.

FTP: File Transfer Protocol, ou seja, protocolo de transferência de arquivo.

Hipermídia: como é hiper, então, é possível dizer que são vários meios, um conjunto que é possível acessar simultaneamente, incluindo textos, imagens e sons. Resumindo, é a reunião de várias mídias em um sistema computacional de comunicação.

Modem: dispositivo eletrônico de entrada e saída utilizado para transmissão e processamento de dados entre computadores.

Unicode: é um superconjunto de ASCII que fornece um número único para cada caractere.

URL: Uniform Resource Locator, ou seja, localizador de recursos uniforme, o endereço único de um arquivo na web.

Arquivos de páginas web e o (X)HTML



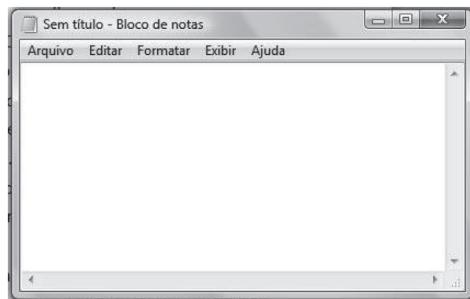
POR DENTRO DO TEMA

Arquivos de Página na Web

Como já tivemos contato com a linguagem HTML, vamos agora entender com detalhes tudo o que é necessário para criar uma página na web. Todo o texto deste tema está baseado na leitura de Castro (2013), Ferreira e Eis (2011), Manzano e Toledo (2010), Meyer (2011), Mileto e Bertagnolli (2014) e Prado (2002). São vários os passos para criar uma página e, se verificarmos, hoje temos muitas opções e páginas com estilos bem ousados, além de outras mais simples. Vamos iniciar com o básico e aos poucos vamos aprofundando e melhorando nossa página.

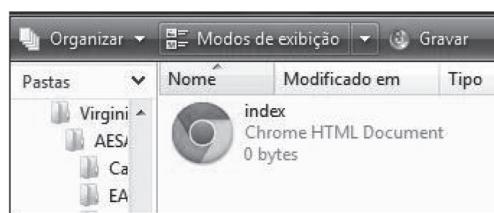
Baseados em Castro (2013), vamos seguir as regras iniciais para criar uma página na web. A primeira questão que surge é sobre qual editor de texto devemos usar. Podemos usar qualquer editor de texto como, por exemplo, o bloco de notas (Figura 2.1). Há editores de textos específicos para o HTML e alguns até gratuitos. Um detalhe importante é não utilizar **processadores de texto** como o Microsoft Word, pois ele pode incluir códigos inválidos e atrapalhar o desenvolvimento de sua página. A página é criada com um editor de texto, mas pode ser visualizada por múltiplos navegadores de diversas plataformas. Além desta opção, tem-se utilizado bastante no mercado de trabalho o software Dreamweaver (Adobe).

Figura 2.1 Editor de texto Bloco de Notas.



Para que a página seja reconhecida, ou seja, interpretada com suas devidas marcações, os arquivos devem ter extensão .html ou .htm, o que irá distingui-la. A extensão .html é mais indicada que o .htm. No bloco de notas, é comum termos a extensão .txt, mas a extensão .html.txt não será vista, sendo assim, deixe somente .html. A Figura 2.2 mostra como ficará seu arquivo na pasta.

Figura 2.2 Arquivo index.html na pasta.



É comum escolher o formato do texto, que sempre é salvo com caracteres padrão do sistema. Se optar por outra codificação que tenha símbolos especiais deve ser usado um editor de texto que permita a escolha da codificação. O **UTF-8** é a melhor escolha.

O arquivo deve ser salvo como index.html, que é o nome para designar a página como página padrão a ser aberta na pasta em que se encontra. Este nome é reconhecido pela maioria dos servidores web como página padrão em cada pasta. A página padrão – index.html – que é criada no topo do diretório web, ou raiz, é a homepage. É possível criar uma página padrão para todos os diretórios de seu site, mantendo os arquivos na pasta e nas subpastas.

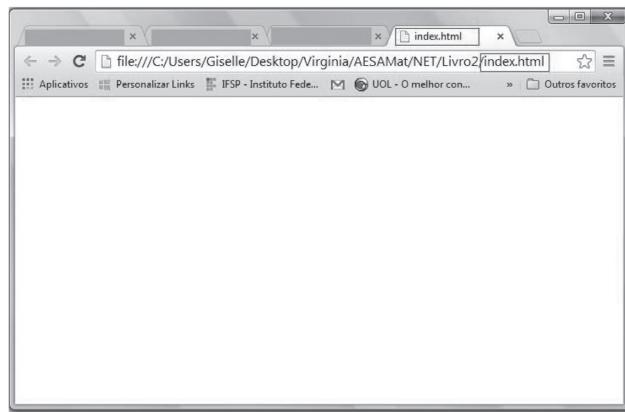
Caso não haja uma página padrão em cada diretório, alguns servidores podem mostrar uma lista com esses conteúdos. O ideal é criar uma página padrão para todos os diretórios do site que contenham páginas HTML. Neste caso, há a opção de modificar a configuração do **servidor** para que a lista de arquivos fique oculta, o que é recomendável quando há pastas com **assets**, como as imagens, arquivos de mídias, folhas de estilo e arquivos de **Javascript**.

Na maioria das vezes, temos vários arquivos e devemos organizá-los para depois não correr o risco de perdê-los ou não localizá-los. A principal dica é criar uma pasta central ou diretório onde deve ficar todo o material que estará disponível no site. Além da pasta central, você pode criar subpastas incluídas nela, mas estas devem refletir a organização de seu site. É comum criar uma pasta acima de todas para as imagens e também criar a pasta principal com nome Assets, colocando nela a pasta de imagens com as outras que contêm vídeos, folhas de estilo, dentre outros.

Depois de criada a página, é necessário visualizá-la, assim, abra o navegador, vá até seu arquivo e escolha a opção “abrir página” ou “abrir arquivo”. Há também a opção

de clicar duas vezes sobre o ícone do arquivo e assim será vista a página, como ilustra a Figura 2.3.

Figura 2.3 Página no navegador.



Para um bom desenvolvedor é importante pesquisar e conhecer outras páginas, mas como ver o código HTML de outros? Há a opção de abrir a página desejada em qualquer navegador e depois escolher “visualizar fonte”, assim, o código HTML será exibido.

(X)HTML e sua Estrutura

Até este ponto, foi vista a página no editor de texto sem codificação, somente com a estrutura para criar, salvar e mostrar. Agora é importante conhecer os componentes básicos e necessários para iniciar de fato a página. Na Figura 2.4, é apresentado um código que será explicado em detalhes.

Figura 2.4 Codificação da Página.

A screenshot of a code editor window displaying the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <title></title>
</head>
<body>

</body>
</html>
```

The code editor has a dark theme with syntax highlighting for HTML tags. The scroll bar on the right indicates there is more code than visible in the current view.

Fonte: Adaptada de Castro (2013).

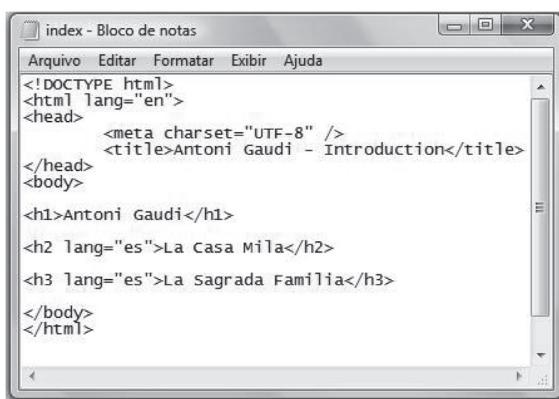
As páginas HTML basicamente são divididas em duas partes: cabeçalho, head, e corpo, body. É fundamental que elas se iniciem com o **DOCTYPE**. O código da Figura 2.4 é o alicerce de toda página HTML5. A primeira linha apresenta `<!DOCTYPE html>` que declara a página como um documento HTML5. Na segunda linha, temos `<html lang="língua-do-código">` para iniciar o HTML do seu documento, em que a língua do código determina a língua padrão, que no código da Figura 2.4 está com "en" para o inglês, mas que poderia ser outra, como "fr" para o francês.

O `<head>` indica o início do cabeçalho do documento da página. Nesta parte, define-se o título da página, que pode incluir informação sobre ela para sites de busca, como, por exemplo, o Google, que carrega folhas de estilo e também pode carregar arquivos JavaScript, o que não é recomendável. Seguindo, temos `<meta charset="UTF-8" />`, que declara a codificação dos caracteres como UTF-8.

Na sequência, há a codificação `<title>` e `</title>`, na qual deve ser inserido o título da página. É necessário que a página tenha título e este deve ser curto, descritivo e único para cada página. Ele não pode conter qualquer formatação, HTML, imagens ou links para outras páginas. O título é geralmente usado como o texto vinculado que aponta o resultado da busca para sua página e também determina a relevância dela em uma busca.

Na linha seguinte há o `</head>`, que encerra o documento de cabeçalho da página. O HTML fornece seis níveis de cabeçalho para que, assim, possa haver uma hierarquia das informações das páginas. Dessa maneira, podem ser marcados os elementos, como mostra a Figura 2.5, em que o cabeçalho `h1` é o mais alto e o `h2` é um subcabeçalho. Mas, para cada nível de cabeçalho, este deve ser encerrado com a mesma indicação, conforme destacado na Figura 2.5. Eles estão entre os elementos mais importantes do HTML de qualquer página.

Figura 2.5 Codificação da Página – cabeçalho.



```

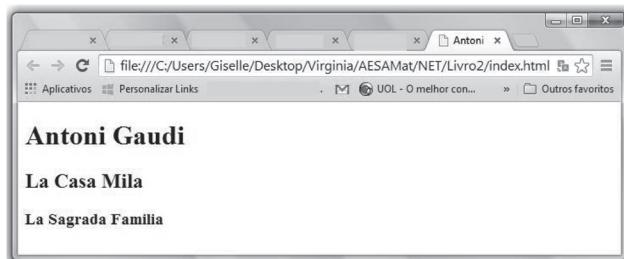
index - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <title>Antoni Gaudi - Introduction</title>
</head>
<body>
    <h1>Antoni Gaudi</h1>
    <h2 lang="es">La Casa Mila</h2>
    <h3 lang="es">La Sagrada Familia</h3>
</body>
</html>

```

Fonte: Adaptada de Castro (2013).

Com o código da Figura 2.5, pode-se ver a página com os cabeçalhos e perceber que, para cada nível, há uma fonte diferente, sendo o maior com uma fonte maior e assim sucessivamente, como ilustra a Figura 2.6.

Figura 2.6 Página com cabeçalho em três níveis.



Fonte: Adaptada de Castro (2013).

Os documentos HTML possuem um outline básico, que é como um índice, definido pelos elementos de cabeçalho, mas este não é exibido explicitamente na página. Há ferramentas que permitem conferir o outline do documento.

É possível agrupar cabeçalhos, pois eles podem ter diversos níveis consecutivos, para isso há um elemento, o hgroup, que indica a relação entre eles. Cada hgroup pode conter dois ou mais cabeçalhos h1-h6, mas nenhum outro elemento é permitido, somente o nível mais alto em um hgroup, que aparece no outline do documento. Para agrupar dois ou mais cabeçalhos é somente seguir a sequência:

Digitar <hgroup>.

Digitar na linha sequente <hn>, em que n pode variar de 1 a 6.

Na frente desta instrução indicada em 2, digite o conteúdo do cabeçalho.

Para encerrar, na frente de cada cabeçalho, como já mostrado anteriormente, deve-se digitar </hn>.

Deve-se repetir os passos de 2 a 4 para o número de cabeçalhos que devem conter o hgroup.

Para finalizar este bloco de instruções, digite </hgroup>.

O exemplo do código e da página utilizando o elemento hgroup pode ser visto na Figura 2.7.

Figura 2.7 Página com exemplo do uso do elemento <hgroup>.

```

<!DOCTYPE html>
<html>
  <head>
    <hgroup>
      <h1>As estrelas</h1>
      <h2>As estrelas que brilham forte !</h2>
    </hgroup>
  </head>
  <body bgcolor="#FEEDEA">
  </body>
</html>

```

O <body> depois do encerramento do cabeçalho é usado para dar início ao corpo da página. Neste espaço, deve ser incluído o conteúdo, por isso há algumas linhas em branco. O exemplo está na Figura 2.8. Este conteúdo pode incluir texto, imagens, formulários, áudio, vídeo, dentre outros. Para encerrar, primeiro é necessária a instrução </body>, que encerra o corpo da página, e na próxima linha </html> para finalizar a página.

Figura 2.8 Página com exemplo do uso do elemento <body>.

```

<!DOCTYPE html>
<html>
  <head>
    <title>Teste com HTML</title>
  </head>
  <BODY>
    Bem-vindo a esta página HTML. Aqui estaremos trabalhando com alguns elementos mais utilizados nesta linguagem.
  </BODY>
  <body bgcolor="#FEEDEA">
  </body>
</html>

```

Nos sites, há quatro componentes principais:

Uma cabeça com a navegação.

Um artigo na área do conteúdo principal.

Uma barra lateral com informação complementar.

Um rodapé.

A semântica aplicada à construção das páginas comuns é muito semelhante, sem importar o layout.

Para melhor compreensão, é importante trabalhar com os elementos e entendê-los separadamente para ir inserindo-os no código. Começando com o elemento header, se uma seção da página tiver um grupo de conteúdo introdutório ou navegação, este deve ser marcado com o header. Na página pode haver qualquer número de elementos header, com vários significados, dependendo do contexto. Um header geralmente inclui a seção de cabeçalho, mas não é obrigatório. Para se criar um cabeçalho com a tag header, deve-se seguir os seguintes passos:

1. No código, posicionar o cursor dentro do elemento em que vai criar o cabeçalho.
2. Digite a instrução <header>.
3. Em sequência, digite o conteúdo do header. Este conteúdo pode incluir diversos tipos de conteúdos marcados com os elementos HTML. Um header pode conter cabeçalhos, série de logos, navegação, campo de busca, dentre outros.
4. Para finalizar esta tag, digite </header>.

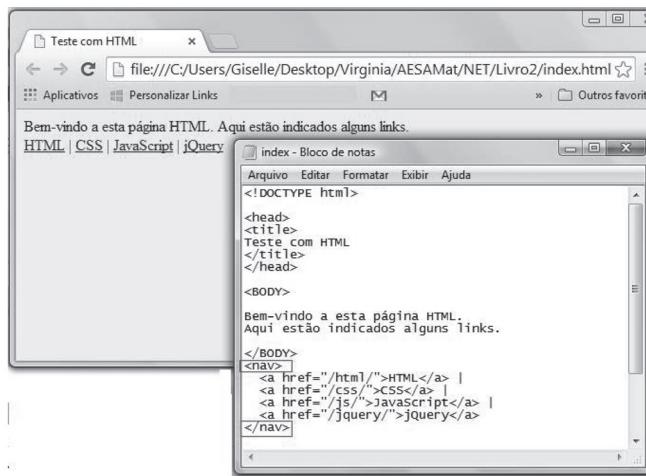
Atenção: um header não pode ser substituído por um cabeçalho.

No HTML5, há um elemento que pode representar explicitamente uma seção dos links de navegação principal, o elemento nav. Os links em um nav podem apontar para um conteúdo dentro da página, para outras páginas, recursos ou ambos. Mas o nav deve ser usado somente para os grupos de links mais importantes. Para usar o nav deve-se realizar a sequência:

1. Digite a instrução <nav>.
2. Em sequência, digite a lista de links estruturados.
3. Finalize digitando </nav>.

Uma indicação do HTML5 é não envolver links no rodapé de páginas auxiliares em um elemento nav. A figura 2.9 mostra o exemplo deste elemento.

Figura 2.9 Página com exemplo do uso do elemento <nav>.



O HTML5 criou o elemento article, que é para receber conteúdos como um artigo de uma revista, uma postagem em um blog, um comentário de um usuário ou qualquer outro item independente de conteúdo, que pode ser visto na Figura 2.10. Uma página pode conter diversos elementos article. Para criar um artigo, basta seguir a sequência:

1. Primeiro digite <article>.
2. Na sequência, digite o conteúdo. Este pode ser qualquer número de elementos, como parágrafos, listas, áudio, vídeo, imagens, figuras dentre outros.
3. Para finalizar, digite </article>.

A linguagem HTML permite a definição de uma seção com o elemento section, considerado um primo semântico menos específico, que também é exclusivo do HTML5. Este elemento representa uma seção genérica de um documento ou aplicativo, é um grupo temático de conteúdo normalmente com um cabeçalho. Para este elemento, são exemplos os capítulos, as várias abas de páginas em uma caixa de diálogo tabulada, seções numeradas de uma tese e também a divisão da homepage de um site. Pode-se dizer que a section é semelhante ao article, e, para diferenciá-los, é necessário verificar a barra lateral, que é mostrada na Figura 2.10. Para definir uma seção é simples:

1. O primeiro passo é indicá-la digitando <section>.
2. Como estamos na área da seção, agora deve ser digitado o conteúdo. Este pode ser parágrafos, listas, áudio, vídeo, imagens, figuras dentre outros.

- Para finalizar, é só digitar </section>.

Figura 2.10 Página com exemplo do uso dos elementos <article> e <section>.

```

<article>
  <hgroup>
    <h1>Título do artigo que deseja incluir</h1>
    <h2>subtítulo do artigo</h2>
  </hgroup>
  <p>introdução ou parágrafo...</p>
  <section>
    <h3>título da primeira seção</h3>
    <p>Conteúdo da primeira seção</p>
  </section>
  <section>
    <h3>título da segunda seção</h3>
    <p>Conteúdo da segunda seção</p>
  </section>
</article>

```

Outro elemento próprio do HTML5 é o aside, que é usado para indicar, de forma semântica, uma seção de conteúdo que é relacionada ao conteúdo principal que permite ser sustentada por conta própria. Como exemplo, podemos citar: uma citação em destaque, uma barra lateral, uma caixa de links de artigos relacionados em um site e uma lista de produtos relacionados em um site de vendas, como ilustrado na Figura 2.11. Para especificá-lo:

- O primeiro passo é digitar <aside>.
- Depois de indicá-lo, é necessário digitar o conteúdo, que pode conter qualquer número de elementos, como parágrafos, listas, áudio, vídeo, imagens, figuras e outros.
- Para encerrar, deve-se digitar </aside>.

Figura 2.11 Página com exemplo do uso do elemento <aside>.

```

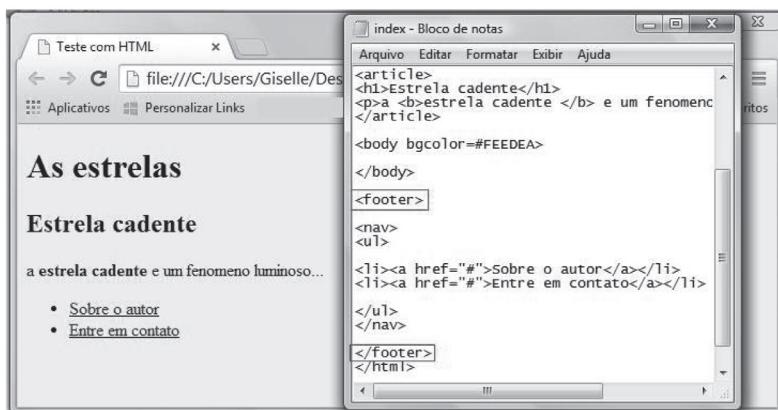
<header>
  <h1>As estrelas</h1>
</header>
<article>
  <h2>Estrela cadente</h2>
  <p>a estrela cadente é um fenômeno luminoso...</p>
  <h2>Conheca as estrelas</h2>
  <ul>
    <li><a href="#">Estrela</a></li>
    <li><a href="#">Constelação</a></li>
  </ul>
</article>
<aside>
  <h3>Conheca as estrelas</h3>
  <ul>
    <li><a href="#">Estrela</a></li>
    <li><a href="#">Constelação</a></li>
  </ul>
</aside>

```

Como é criado o cabeçalho, em uma página bem estruturada, cria-se também o rodapé, assim, há o elemento footer do HTML5. Este elemento, como o header, pode ser usado em outro lugar. Este é o rodapé da página toda cujo ancestral mais próximo é o body. O rodapé sempre possui informações sobre a seção e você pode aninhar um header ou outro footer dentro de um footer. A Figura 2.12 ilustra um exemplo de footer. Para ter o rodapé em sua página realize a sequência:

1. O primeiro passo é posicionar o cursor dentro do elemento que vai criar o rodapé.
2. Na sequência, digite <footer>.
3. Depois de indicado o rodapé, basta incluir o conteúdo.
4. Para finalizar o rodapé, digite </footer>.

Figura 2.12 Página com exemplo do uso do elemento <footer>.



Na página, pode-se adicionar comentários, como ilustra o código da Figura 2.13, que é um recurso de que precisamos para documentá-la, e o comentário só vai aparecer se o documento for aberto com um editor de texto ou na opção de visualizar o código. Para adicioná-lo:

1. Escolha o local dentro de seu documento onde deseja inserir o comentário, deve-se digitar <!--.
2. Na sequência, digite o comentário.
3. Para completar e encerrar o comentário, digite --> .

Figura 2.13 Código com comentário no código.

```

<header>
<h1>As estrelas</h1>
</header>
<article>
<h1>Estrela cadente</h1>
<p>a <b>estrela cadente </b> é um fenômeno</p>
</article>

<body bgcolor="#FEEDEA">
<!--
    Isto é um comentário onde você pode
    documentar sua página
-->

</body>
<footer>
<nav>
<ul>
    <li><a href="#">Sobre o autor</a></li>

```

Textos no (X)HTML

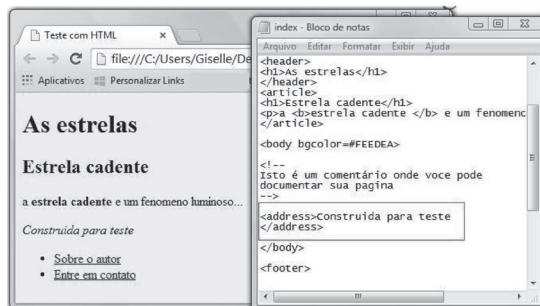
Em qualquer página, é fácil notar que a maior parte do seu conteúdo é o texto. Para iniciar o texto, devemos iniciar o parágrafo. Segundo Castro (2013), o HTML não reconhece os retornos ou espaços em branco digitados no editor de texto. O elemento usado para o parágrafo é o p, usado nos códigos dos exemplos das figuras anteriores. Este deve realizar a sequência:

1. Para iniciar o parágrafo, digite <p>.
2. Na sequência, digite o conteúdo do parágrafo.
3. Para finalizar o parágrafo, digite </p>.

No texto, há muitos tipos de informações que devem ser colocadas. Por exemplo, em um artigo devemos incluir informações de contato do autor, mas como adicioná-la? O HTML possui o elemento address, que define a informação de contato para pessoas em uma página, como mostra a Figura 2.14. Para este elemento, devemos realizar a sequência:

1. Caso a informação de contato seja para um article, o cursor deverá ser colocado dentro deste elemento. O cursor também pode ser colocado dentro de body ou no footer, se a informação for geral.
2. Para iniciar, digite <address>.
3. Na sequência, digite o endereço do e-mail do autor, um link para uma página com informação para contato.
4. Para finalizar, digite </address>.

Figura 2.14 Página com exemplo do elemento <address>.



```

<header>
<h1>As estrelas</h1>
</header>
<h2>Estrela cadente</h2>
<p>a estrela cadente é um fenômeno luminoso...</p>
<article>
<h3>Construída para teste</h3>


- Sobre o autor
- Entre em contato


</article>
<body bgcolor="#FEEDEA">
<!--
Isto é um comentário onde você pode
documentar sua página
-->
<address>Construída para teste</address>
</body>
<footer>

```

Em quase todas as páginas encontram-se figuras, mas há figuras de tipos variados, como mostra a Figura 2.15. No HTML5, há o elemento figure, que é uma parte do conteúdo autocontida, associada ao conteúdo principal do documento. Há o elemento figcaption, que é opcional, é a legenda da figura e pode aparecer no início ou no final do conteúdo da figura. Para incluir uma figura com legenda, deve-se seguir os seguintes passos:

1. Inicie digitando <figure>.
2. Se desejar incluir legenda digite <figcaption>.
3. Se for digitado o <figcaption>, na sequência, digite o texto da legenda.
4. Para finalizar a legenda, digite </figcaption>.
5. Crie a figura adicionando o código para imagens, vídeos, tabelas, etc.
6. Se desejar colocar legenda neste espaço, siga os passos de 2 a 4.
7. Para finalizar digite </figure>.

Figura 2.15 Exemplo do elemento <figure>.

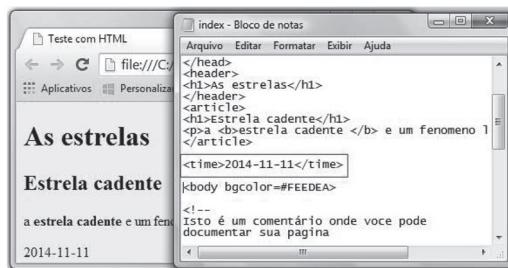


Fonte: Adaptada de: <<http://html5doctor.com/the-figure-figcaption-elements/>>. Acesso em: 23 jan. 2015.

São vários itens que incluímos em um texto, como a data, às vezes é necessário especificar o tempo, e este se faz por meio do elemento time e do atributo datetime. Se o atributo datetime for omitido, é fornecida a data e a hora no formato válido, ilustrado na Figura 2.16. Quando é incluído o datetime, você pode definir a representação da data. Para uma hora exata e uma data do calendário, basta seguir estes passos:

1. Para iniciar, digite <time>, se não utilizar o datetime.
2. Se desejar especificar o formato pode digitar o datetime sendo então: <time datetime="formato">.
3. Caso a hora represente a data de publicação, neste caso, deve-se digitar pubdate="pubdate" ou pubdate.
4. Para finalizar, digite </time>.

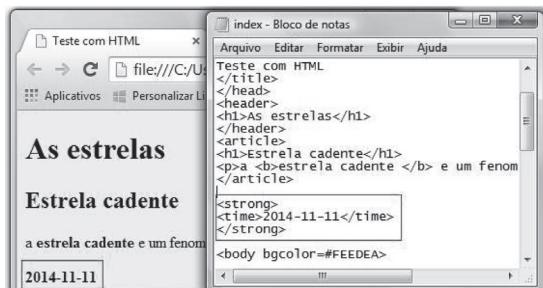
Figura 2.16 Exemplo do elemento <time>.



O texto pode e às vezes deve ser marcado como importante, como mostra a Figura 2.17, e, para este recurso, há o elemento strong. Assim:

1. Para iniciar, digite .
2. Na sequência, deve ser digitado o texto ou parte do texto que precisa ser marcado como importante.
3. Para finalizar, digite .

Figura 2.17 Exemplo do elemento .



Se a necessidade for de enfatizar o texto, as instruções são as seguintes:

1. Para inicializar este procedimento, digite .
2. Nesta parte, deve ser digitado o texto a ser enfatizado.
3. Para finalizar, dite .

É possível marcar e enfatizar o texto ao mesmo tempo de acordo com o que for preciso. Neste contexto, é possível também indicar uma citação ou referência, o que é feito através do elemento cite. Neste caso, pode-se ter títulos de uma peça, roteiro ou livro, nome de música ou filme, etc. Deve ser dentro da regra:

1. Para iniciar, digite <cite>.
2. Na sequência, deve-se introduzir a referência.
3. Para finalizar, digite </cite>.

Para citar fontes é diferente, há o elemento blockquote que é usado quando a citação é longa. Há também o elemento q para citações curtas, geralmente dentro de uma sentença.

Para longas citações, deve ser usado como segue:

1. Inicia-se a partir da digitação de <blockquote>.
2. Se precisar, digite 'cite="url"', em que "url" é o endereço da fonte da citação.
3. Não se esqueça de digitar > para completar a tag inicial.
4. Na sequência, digite o texto.
5. Para finalizar, digite </blockquote>.

Em casos de citações ou frases curtas, deve-se seguir:

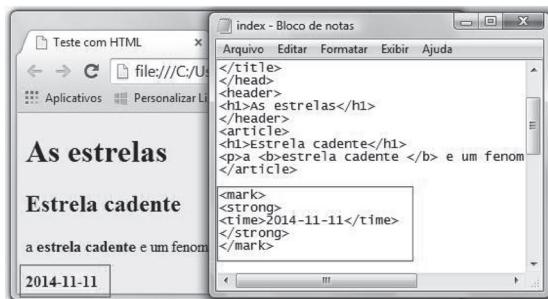
1. Para iniciar digite <q>.
2. Se precisar, digite 'cite="url"', em que "url" é o endereço da fonte da citação.
3. Não se esqueça de digitar > para completar a tag inicial.
4. Na sequência, digite o texto.
5. Para finalizar, digite </q>.

Para destacar ou marcar o texto, tem o elemento mark, como uma caneta marca-texto, que pode ser observado na Figura 2.18. Esta marcação só deve ser utilizada quando necessário, pois é somente para chamar a atenção do leitor. Deve-se seguir

os passos:

1. Inicie digitando <mark>.
2. Na sequência, digite as palavras a serem destacadas.
3. Para encerrar a marcação, digite </mark>.

Figura 2.18 Exemplo do elemento <mark>.



O HTML fornece inúmeros recursos, com vários elementos, para detalhes, às vezes pequenos e bem específicos, para textos.



ACOMPANHE NA WEB

Apostila de HTML e CSS

Leia a Apostila de HTML e CSS que abrange a linguagem HTML. Consulte as páginas de 4 a 17 para ver detalhes dos elementos a serem utilizados nas páginas dos sites.

Disponível em: <<https://www.telecom.uff.br/pet/petws/downloads/tutoriais/css/css2k80912.pdf>>. Acesso em: 10 nov. 2014.

Estrutura do HTML

Para complementar este tema, acesse essa página que trata da estrutura do HTML bem como dos passos iniciais e principais para o desenvolvimento de códigos em HTML.

Disponível em: <<http://www.educacaopublica.rj.gov.br/oficinas/informatica/html/estrutura.htm>>. Acesso em: 11 nov. 2014.



AGORA É A SUA VEZ

Instruções:

Agora, chegou a sua vez de exercitar seu aprendizado. A seguir, você encontrará algumas questões de múltipla escolha e dissertativas. Leia cuidadosamente os enunciados e atente-se para o que está sendo pedido.

Questão 1

Para criar uma página na web, é importante saber qual editor de texto usar. Podemos usar vários editores, como o bloco de notas ou programas específicos para o HTML, sendo alguns até gratuitos, mas nem todos esses editores são recomendáveis. Podemos desenvolver uma página em qualquer processador de texto? Explique.

Questão 2

Para criar uma página, devemos ter um código básico correto. Dentre os códigos mencionados a seguir, assinale o correto:

a)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
<title>
</head>

<body></body>
</html>
```

b)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
<title> </title>
</head>

<body>
</body>
</html>
```

c)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
<title> </title>
</head>
<body>
</body>
</html>
```

d)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
</head>
<title> </title>
</head>

<body>
</html>
```

e)

```
<!DOCTYPE html>
<html lang="en">
  <meta charset="UTF-8" />
<title> </title>
</head>

<body>
</body>
</html>
```

Questão 3

Analise as sentenças a seguir e julgue cada uma delas como verdadeira ou falsa.

Às vezes, é necessário especificar o tempo, e este se faz através do elemento time. Mas, neste contexto, há o atributo datetime. Sobre este atributo, podemos afirmar que:

- () Não pode ser omitido quando utilizado o elemento time, pois o tempo fica sem especificação correta.
- () Se o atributo datetime for omitido, não é fornecida a data e a hora no formato válido.
- () Sobre os elementos time e datetime, utiliza-se somente um, não sendo aceito os dois.
- () Quando é incluído o datetime, a representação da data pode ser definida.

Questão 4

O HTML fornece seis níveis de cabeçalho para que, assim, possa ter uma hierarquia das informações das páginas. Dessa forma, os elementos podem ser marcados, sendo o cabeçalho h1 o mais alto. A partir do código da Figura 2.4, introduza um cabeçalho com três níveis.

Questão 5

Para destacar ou marcar o texto, há o elemento mark, que funciona como uma caneta marca-texto. Aproveitando o código da questão anterior, marque o cabeçalho do nível 2.



FINALIZANDO

Este tema possibilitou a você conhecer a estrutura básica para montar uma página em HTML. Você pode agora começar a desenvolver páginas utilizando os recursos aprendidos, como o código básico, introdução de títulos e cabeçalhos, etc. Na página, a parte principal é o texto inserido, e, para este, foi dedicada um grande espaço, apresentando detalhes específicos e explicando elementos básicos para que se possa inserir um texto com qualidade. Há detalhes nas páginas que todos devem colocar, como a data e as marcações específicas para chamar a atenção do usuário. Para que você pudesse entender bem a linguagem, foram apresentados trechos de códigos com os elementos determinados e também o resultado na página, para que esse acompanhamento não gere dúvidas, mas traga mais conhecimentos.



REFERÊNCIAS

CASTRO, Elizabeth. HTML5 e CSS3: guia prático e visual. Jacaré: Alta Books, 2013.

FERREIRA, Elcio; EIS, Diego. HTML 5: curso W3C Escritório Brasil. Apostila. 2011. Disponível em <<http://www.oficinadanet.com.br/imagens/apostilas/703/html5-web.pdf>>. Acesso em: 4 nov. 2014.

MANZANO, José A. N. G.;TOLEDO, Suely Alves. Guia de orientação e desenvolvimento de sites: HTML, XHTML, CSS e JavaScript/JScript. São Paulo: Érica, 2010.

MEYER, Érica A. Smashing CSS. Técnicas profissionais para um layout moderno. Dados Eletrônicos. Porto Alegre: Bookman, 2011.

MILETO, Evandro Manara; BERTAGNOLLI, Sílvia de Castro. Desenvolvimento de Software II: introdução ao desenvolvimento web com HTML, CSS, JavaScript e PHP. Dados Eletrônicos. Porto Alegre: Bookman, 2014.

PIVETTA, Elisa Maria. Apostila de HTML e CSS. 2010. Disponível em: <<https://www.telecom.uff.br/pet/petws/downloads/tutoriais/css/css2k80912.pdf>>. Acesso em: 3 nov. 2014.

PRADO, Márcio Feliciano do. Apostila de HTML e CSS. UNICAMP. Minicursos virtuais. Campinas: Unicamp, 2002. Disponível em: <<http://www.marcoprado.eti.br/ensino.html?download=6%3Aapostilahtml>>. Acesso em: 5 nov. 2014.



GLOSSÁRIO

Assets: termo utilizado para designar o que complementa o conteúdo das páginas web.

Doctype: tipo de marcação que descreve o texto escrito na página web.

Javascript: é uma linguagem interpretada que pode ser utilizada junto com o HTML.

Processadores de texto: programas específicos para escrever, digitar textos no computador. Com eles é possível criar documentos simples e até arquivos complexos.

Servidor: é um computador em um sistema de computação centralizada, específico para transmitir informações e fornecer serviços a computadores que estiverem conectados a ele em uma rede de computadores.

UTF-8: é uma tabela de caracteres unicode (conjunto de caracteres para documentos HTML). Um tipo de codificação de caracteres.

Trabalhando com imagens e links



POR DENTRO DO TEMA

Imagens para a Web

Todo o texto deste tema está baseado em Castro (2013), incluindo também a leitura de Ferreira e Eis (2011), Manzano e Toledo (2010), Meyer (2011), Mileto e Bertagnolli (2014) e Prado (2002). São vários os elementos que já foram mostrados para que se possa construir uma página, mas esta deve ser ilustrada de maneira que influencie a estética e o visual. Isso é possível com a inserção de imagens, a harmonização das cores e o design a ser considerado para o site. Esse recurso deve ser utilizado com cuidado, pois o excesso de imagens em um site pode torná-lo pesado, o que acarretará dificuldade de visualização em computadores mais simples. Como devemos inserir imagens e exibi-las com precisão? Para saber criar as imagens na web e fazer com que elas fiquem bem dispostas nas páginas, há seis fatores principais para os quais é preciso atentar-se a fim de que elas possam ser diferenciadas. São eles: formato, cor, tamanho/resolução, velocidade, transparência e animação.

Primeiramente, é importante saber sobre o formato, pois a imagem vai ser salva e deve ser acessada sem problemas, não importando os **sistemas operacionais**. Atualmente, os formatos mais utilizados na web são: GIF, PNG e JPEG. Para saber escolher o formato ideal, é preciso pensar naquele que fornece uma melhor qualidade de imagem com menor tamanho de arquivo. Dentre os citados, é possível dividi-los da seguinte forma:

1. JPEG: é indicado para fotos coloridas, já que com ele se manipula uma grande quantidade de cores. Além disso, o arquivo é comprimido, oferecendo a vantagem de a página ser rapidamente carregada. Mas, nesse processo, há a desvantagem de ocorrerem perdas das informações originais da imagem.

2. GIF: é utilizado para arquivos como logos, com grande quantidade de cores sólidas e padrões, ou quando há a necessidade de transparência. Consegue comprimir áreas de cores contínuas ou padrões repetitivos melhor que o formato JPEG.

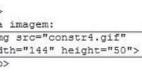
3. PNG: é a melhor escolha, sendo usado também para arquivos como logos com grande quantidade de cores sólidas e padrões ou quando há a necessidade de transparência. Possui um algoritmo de compressão melhor para arquivos menores e tem um suporte para transparência superior.

Na Figura 3.1 apresentam-se o código e duas imagens com formato GIF; a segunda delas, inclusive, é animada, como você pode observar:

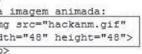
Figura 3.1 Código e imagem com formato GIF

```

<html>
<body>

<p>
Uma imagem:


</p>

<p>
Uma imagem animada:


</p>

<p>
Observe que a sintaxe para inserir uma
imagem animada não é diferente da de
uma imagem não animada.
</p>

```

Fonte: adaptado de <<http://www.clem.ufba.br/tuts/html/c12.htm>>.

Outro aspecto importante são as cores, já que atualmente é possível contar com monitores excelentes, capazes de exibir milhões delas. Alguns formatos têm restrições: os formatos GIF e PNG possuem imagens com uma paleta de 256 cores, o que é satisfatório para logos e ícones. Entretanto, há as fotografias, que são exemplos de imagens que devem ser exibidas com muita precisão, as quais precisam ser salvas no formato JPEG ou PNG-24, em razão de terem mais cores.

Sobre o tamanho e a resolução, é importante saber que as imagens digitais são medidas em **pixels**. O tamanho da imagem depende da resolução do monitor do navegador, devendo-se respeitar a média das páginas na web, que gira em torno de 960 pixels de largura, o que permite aos usuários ver todo o conteúdo da página sem a barra de rolagem na horizontal. Sobre a resolução, pode-se dizer que quanto maior for esta maior será o número de pixels.

Como neste assunto deve-se pensar nos usuários, é necessário abordar a velocidade em que as imagens são carregadas, pois todos desejam vê-las o mais rápido possível, sem ter de ficar esperando seu carregamento. Como o tempo de carregamento deve ser minimizado, é preciso lembrar que quanto maior for a imagem mais tempo será utilizado para que a imagem apareça na tela. Nesse sentido, as imagens pequenas têm mais vantagens. Uma forma utilizada para solucionar a questão é comprimir a imagem, ação para a qual se indica o formato JPEG, pois dessa maneira se reduz drasticamente o tamanho do arquivo. Essa

poderia ser a solução, mas a informação de compressão ocupa muito espaço, o que não é bom para imagens pequenas. Além disso, trata-se de uma compressão com perdas, que elimina detalhes permanentes para economizar espaço. Entretanto, há os formatos PNG e GIF que comprimem a imagem sem perdas.

Outro aspecto a considerar é a transparência, que, apesar de aparentemente não relevante, é usada para criar layouts complexos, permitindo que uma imagem se mova atrás da outra. A transparência pode dar à imagem um contorno não retangular. Os formatos PNG e GIF permitem a transparência, diferentemente do formato JPEG, que não a permite.

As imagens podem ter movimento, ou seja, podem ser animadas e salvas como GIF. Os formatos JPEG ou PNG, por sua vez, não suportam animação. A animação pode ser criada com **Flash**, **CSS Animation** e **JavaScript**.

Há várias formas de arrumar imagens para colocá-las na página. Elas podem ser compradas, podem ser baixadas prontas e podem advir da digitalização de desenhos feitos à mão ou também do escaneamento de fotografias. É necessária muita atenção neste ponto. Essas imagens devem ter barras laterais com licenças Creative Commons para saber sobre os direitos autorais delas.

Há editores de imagens que você pode usar para criar e salvar imagens para a web, os quais possuem ferramentas especiais que auxiliam muito. Dentre esses editores, hoje são muito utilizados o Photoshop e o Adobe Fireworks.

Depois de escolhidas ou feitas as imagens, estas devem ser salvas. Para isso você pode utilizar as versões de teste do Photoshop ou do Fireworks se não tiver um desses softwares instalado no computador. Para utilizar o comando salvar para web no primeiro deles, é preciso cumprir os seguintes passos:

1. Abra o Photoshop e crie ou abra sua imagem.
2. Escolha as opções: Arquivo > Salvar para Web & Dispositivos.
3. Clique na aba 2 ou Mais para uma versão otimizada ou 4 ou mais para três.
4. Escolha o formato desejado.
5. Ajuste as configurações para uma boa qualidade.
6. Clique em salvar.

Agora que já está salva, a imagem deve ser inserida na página de web. Depois de inserida, é hora de ver a imagem, que, por sua vez, deve aparecer automaticamente quando o usuário acessar a página. Para isso, é preciso cumprir as seguintes etapas:

1. Abra o arquivo HTML e posicione o cursor no código.

2. Digite ``. A indicação "image.url" é para localizar o arquivo de imagem no servidor.

3. Para finalizar, digite um espaço em branco e também `/>`.

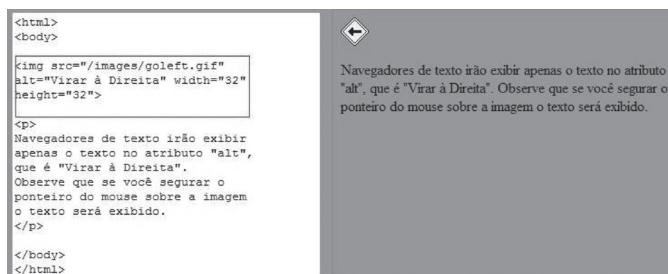
Pode ocorrer de a imagem não aparecer, caso para o qual há a opção de adicionar um texto descritivo, que será lido pelo leitor de telas, como mostra a Figura 3.2. No entanto, você pode acessar a página indicada na Figura 3.2 e visualizar melhor como fica o texto. Para esse recurso, é preciso inserir instruções no seu código, como segue:

1. Na tag img, após o atributo src e seu valor, digite alt="".

2. Digite o texto que deve aparecer se a imagem não for exibida.

3. Digite "".

Figura 3.2 Texto descritivo na imagem



Fonte: adaptado de <<http://www.clem.ufba.br/tuts/html/c12.htm>>.

Um fator muito importante é especificar o tamanho da imagem, pois, se esse detalhe não for considerado, o texto é visto primeiro e, depois que as imagens são carregadas, ele pula ao redor delas para que possa se adaptar. O processo da imagem inicia quando o navegador acessa o código HTML e vai carregar a imagem verificando o tamanho e o espaço que devem ser reservados para ela. Há a opção de deixar as dimensões da imagem já especificadas. Nesse caso, o navegador reservará o espaço da imagem, podendo, enquanto esta é carregada, preencher o texto em volta dela. Dessa forma, o layout fica estável enquanto a imagem é carregada. Para especificar as dimensões da imagem, o que é uma boa solução, usa-se o navegador ou o editor de imagens. Quando a imagem for utilizada em diferentes contextos, o navegador pode aumentá-la ou diminuí-la para que ela se encaixe no tamanho especificado no HTML ou CSS. Para encontrar o tamanho da imagem no navegador, clique com o botão direito na imagem e acesse o menu dela. Nesse menu, dentre as opções, deve-se escolher "Propriedades" ou "Ver", onde há

uma caixa que contém as dimensões da imagem em pixels. Dessa forma, você pode especificar a altura e a largura exatas da imagem em pixels. Seguindo esses passos, o navegador não precisará calcular o tamanho da imagem e esta vai ser exibida com maior rapidez. No Photoshop, deve-se procurar a barra de informação no canto inferior esquerdo. Clique na barra de informação do documento para que uma caixa pequena com informações sobre a imagem apareça incluindo o tamanho. Outra op para especificar o tamanho da imagem é no arquivo HTML. Em primeiro lugar, determine o tamanho da imagem com o navegador, como descrito anteriormente, e, em seguida, dentro da tag ``, após o atributo `src`, digite: `width="x" height="y"`, com os mesmos valores que foram especificados em pixels.

Muitas vezes há imagem com dimensões não propícias para o lugar em que se quer inseri-la, mas é possível modificar o tamanho da exibição especificando uma nova altura e uma nova largura em pixels, utilizando as instruções dadas anteriormente. A Figura 3.3 mostra uma imagem dimensionada em três tamanhos. Para dimensionar uma imagem com o navegador, você deve cumprir os seguintes passos:

1. No código do arquivo HTML, digite: ``.

Figura 3.3 Código e imagem com dimensões diferentes.



Fonte: adaptado de <<http://www.clem.ufba.br/tuts/html/c12.htm>>.

Para o dimensionamento das imagens com utilização de um editor de imagens, há alguns detalhes diferentes a seguir, como no Photoshop:

1. Para iniciar, vá à parte inferior à direita da janela “Salvar para a Web & Dispositivos” e clique na caixa L(largura) ou A(altura) na seção tamanho da imagem.
2. Neste passo, digite a nova altura ou largura em pixels, ou uma porcentagem, e pressione a tecla Tab para a imagem ser redimensionada.
3. Salve a imagem, pois esta só é redefinida após ser salva.

Uma opção existente, que sempre é utilizada, é a de adicionar ícones ou **favicon** no site. Para esse recurso, deve-se cumprir os seguintes passos:

1. A imagem criada ou a ser inserida deve ser de 16x16 pixels e deve ser salva no formato ICO – é também possível salvá-la nos formatos PNG e GIF.
2. Crie uma imagem de 114x114 pixels para aparelhos com touchscreen e salve-a no formato PNG.
3. No arquivo HTML, na seção head, digite `<link rel="shortcut icon" href="favicon.ico" />`, em que “favicon.icon” é o nome e a localização do ícone no servidor. Se a imagem estiver no formato PNG, digite: `<link rel="icon" type="image/png" href="favicon.png" />`. Para imagens no formato GIF, digite: `<link rel="icon" type="image/gif" href="favicon.gif" />`.
4. Ainda na seção head do arquivo HTML, digite: `<link rel="apple-touch-icon" href="/apple-touch-icon.png" />`; neste caso, apple-touch-icon.png é o nome e a localização do ícone no servidor.

Links para a Web

Além de interessante, uma prática satisfatória, que dá vida a web, é a utilização dos links, que permitem conectar uma página a outra, sem ser necessário usar somente uma página. Para entender sobre os links, primeiramente é preciso saber como estes são divididos, já que possuem duas partes principais: um destino e um rótulo.

O destino é a parte mais importante, que especifica o que ocorre quando o link é acionado. Ou seja, indica se o link pode levar para outras páginas ou mesmo saltar para a mesma, a fim de mostrar imagens, baixar um arquivo, mandar um e-mail, entre outras execuções. Os links comuns conectam a outras páginas na web, mas também podem, às vezes, levar para uma localização específica da página, que é determinada de âncora, cujo destino é definido pela adição da URL, sendo visíveis somente na barra de status do navegador.

O rótulo, a segunda parte do link, é a parte que o visitante vê no navegador ou ouve em um leitor de telas e que é utilizada para ativar o alcance do destino, que pode ser um texto, uma imagem ou ambos. O padrão dos navegadores é exibir o texto do rótulo sublinhado em azul, o que, entretanto, pode ser modificado.

O importante é saber criar o link. Para iniciar o processo, pode-se criar um link que conecte outra página; no entanto, há também a opção de criar links para páginas em outros sites. Para criar um link para outra página, como ilustra a Figura 3.4, deve-se cumprir os seguintes passos:

1. Primeiro, digite ``, em que page.html é a URL da página de destino.
2. Neste passo, digite o texto do rótulo, que é destacado por padrão e, quando ativado, leva o usuário à página que foi especificada no item 1.
3. Para finalizar digite ``, completando, assim, a definição do link.

Figura 3.4 Código e link em uma página

```

<html>
<body>

<p>
<a href="lastpage.htm">
Este texto</a> é um vínculo (link) para uma página neste sitio da Web.
</p>

<p>
<a href="http://www.microsoft.com/">
Este texto</a> é um vínculo para uma página na World Wide Web.
</p>

</body>
</html>

```

The screenshot shows a code editor with the above HTML code. Two parts of the code are highlighted with yellow boxes: the first part highlights the entire `` tag and its content "Este texto é um vínculo (link) para uma página neste sitio da Web.", and the second part highlights the entire `` tag and its content "Este texto é um vínculo para uma página na World Wide Web.". To the right of each highlighted area, there is a tooltip-like box containing the text "Este texto é um vínculo (link) para uma página neste sitio da Web." and "Este texto é um vínculo para uma página na World Wide Web." respectively.

Fonte: adaptado de <<http://www.clem.ufba.br/tuts/html/ex20h.htm>>.

Uma grande vantagem é que o HTML5 permite colocar um link em torno de quase todos os tipos de elementos ou grupos de elementos, que são denominados links em nível de bloco. Quando se trabalha com links em nível de bloco, é aconselhável colocar o conteúdo mais pertinente no começo de um link – esse conteúdo no link não deve ser abundante.

Neste contexto, é possível criar âncoras. Mas quando e por que criar âncoras? Com a ativação de um link, este leva o usuário ao topo da página à qual está conectado; entretanto, se desejar que o usuário pule para uma seção específica da página, neste ponto é necessário criar uma âncora e fazer uma referência dela no link. Para criar a âncora, é necessário:

1. No arquivo HTML, posicionar o cursor na tag de início do elemento para o qual você quer que o usuário pule.

2. Deve-se digitar `id="anchor-name"`, em que anchor-name refere-se ao texto que será utilizado internamente para identificar aquela seção da página. É importante certificar-se de que há espaço entre o nome do elemento e o id.

Após a âncora ser criada, é possível definir um link que, assim que o usuário ação-lo, faz com que a página vá diretamente à seção do documento que tem a âncora e não ao topo da página. Deve-se seguir passos determinados para criar um link para a âncora. São eles:

1. Primeiramente, digite ``, em que anchor-name é o valor do destino do atributo id.

2. Na sequência, digite o texto do rótulo, ou seja, o texto que é destacado e que quando ativado possibilita ao usuário ir à seção especificada no passo 1.

3. Para finalizar, digite ``, completando, assim, a definição do link.

Quando criados, os links não são destinados somente para conectar outras páginas, mas sim para qualquer URL. Neste caso, para criar outros tipos de links, é preciso seguir as instruções:

1. Para iniciar, digite `<a href=`.

2. Neste espaço digite a URL.

3. Se o link vai acessar qualquer arquivo na web, desde imagens, planilhas, arquivos compactados etc., é preciso digitar: `http://www.site.com/path/file.ext`, em que:

- a) `www.site.com` é o nome do servidor.

- b) `path/file.ext` é o caminho para o arquivo desejado, com sua extensão.

4. Na sequência digite `>`.

5. Neste passo, digite o rótulo do link, ou seja, o texto destacado que, quando ativado, vai levar o usuário à URL determinada na passo 2.

6. Para finalizar digite ``.

Seguindo os passos e inserindo as instruções determinadas na ordem certa, é possível colocar links e deixar a página com uma boa estrutura e, ao mesmo tempo, com um acesso ideal para o usuário.



ACOMPANHE NA WEB

Linguagem HTML

Acesse uma apostila que abrange a linguagem HTML. É interessante que você faça uma consulta às páginas de 17 a 22 para ver detalhes dos elementos a serem utilizados para lidar com imagens e links em páginas web.

Link para acesso: <<https://www.telecom.uff.br/pet/petws/downloads/tutoriais/css/css2k80912.pdf>>. Acesso em: 16 nov. 2014.

Como trabalhar com imagens

Para complementar este tema, há uma página que apresenta dicas bem interessantes de como trabalhar com imagens, apresentando inovações. Vale a pena você consultá-la.

Link para acesso: <<http://www.kenjiyamamoto.com/blog/2010/08/09/dicas-sobre-html5-e-css3>>. Acesso em: 16 nov. 2014.



AGORA É A SUA VEZ

Instruções:

Agora, chegou a sua vez de exercitar seu aprendizado. A seguir, você encontrará algumas questões de múltipla escolha e dissertativas. Leia cuidadosamente os enunciados e atente-se para o que está sendo pedido.

Questão 1

Quando se deseja inserir imagens em uma página web, é importante exibi-las com precisão. Para criar as imagens na web e deixá-las bem dispostas nas páginas, há seis fatores principais aos quais temos de estar atentos para que elas possam ser diferenciadas.

Quais são esses seis fatores que influenciam na inserção de imagens em uma página web?

Questão 2

Para inserir uma imagem em uma página da web, o formato é muito importante, pois a imagem vai ser salva e deve ser acessada sem problemas, não importando os sistemas operacionais. Atualmente, os formatos mais utilizados na web são: GIF, PNG e JPEG. Sobre estes, é possível afirmar que:

- a) O formato JPEG é indicado para fotos coloridas em razão de manipular uma grande quantidade de cores, havendo, entretanto, desvantagens, dentre as quais se pode destacar o fato de não permitir que a página seja carregada rapidamente.
- b) O formato GIF é utilizado para arquivos como logos, com pequena quantidade de cores sólidas e padrões.
- c) O formato GIF consegue comprimir áreas de cores contínuas ou padrões repetitivos melhor que o formato JPEG.
- d) O formato PNG não é uma boa opção, sendo, às vezes, usado para arquivos como logos, com grande quantidade de cores sólidas.
- e) O formato JPEG, quando comprimido, traz uma vantagem, pois nesse processo não ocorre perdas das informações originais da imagem.

Questão 3

Um fator muito importante é especificar o tamanho da imagem, pois, se esse detalhe não for considerado, o texto é visto primeiro e, depois que as imagens são carregadas, ele pula ao redor destas para que possa se adaptar. No arquivo HTML, para especificar o tamanho da imagem, é indicado:

- () Somente determinar o tamanho da imagem com o navegador.
- () Primeiramente determinar o tamanho da imagem com o navegador e no código HTML, dentro da tag , após o atributo scr, digitar a altura e a largura com os mesmos valores que foram especificados em pixels.
- () Primeiramente determinar o tamanho da imagem com o navegador, e, no código HTML, dentro da tag , antes do atributo scr, devem ser digitadas a altura e a largura com os valores em pixels.
- () No código HTML abaixo da tag , após o atributo scr, digitar: width="x" height="y", com os mesmos valores que foram especificados em pixels.

Questão 4

Os links permitem conectar uma página a outra, sem que se utilize somente uma página. Eles são divididos em duas partes principais: um destino e um rótulo. O destino é a parte mais importante.

Qual é a função da parte “destino” que a faz ser classificada como a mais importante?

Questão 5

É muito interessante e importante haver links em uma página; no entanto, é preciso saber criá-los. É possível criar links que conectem outra página, mas há também a opção de criá-los para páginas em outros sites. Quais são os passos para criar um link para outra página?



FINALIZANDO

A partir do estudo, da leitura, da pesquisa e dos exercícios propostos neste tema, você pôde aprender mais sobre imagens e links. Você agora está apto a ilustrar suas páginas ao inserir imagens com detalhes e precisão necessários. Agora, para criar e inserir uma imagem, você sabe que deve considerar seis fatores que as diferenciam, sendo eles: o formato, a cor, o tamanho/resolução, a velocidade, a transparência e a animação. A ilustração da página é um diferencial que você pôde aprender e um recurso do qual, a partir de agora, você pode fazer uso. Na segunda parte do tema você adquiriu conhecimento sobre os links em uma página em HTML. Você pode agora começar a desenvolver páginas utilizando os recursos aprendidos, como os links. Com eles, você pode conectar outras páginas à sua própria ou a outras de seu interesse. Concluindo, você agora já conhece muitos recursos para que sua página fique com uma boa estrutura e também mostre um resultado satisfatório.



REFERÊNCIAS

CASTRO, Elizabeth. HTML5 e CSS3: Guia Prático e Visual. São Paulo: Alta Books, 2013.

FERREIRA, Elcio; EIS, Diego. HTML 5. Curso W3C Escritório Brasil. Rio Grande do Sul: Oficina da Net, 2011. Disponível em: <<http://www.oficinadanet.com.br/imagens/apostilas/703/html5-web.pdf>>. Acesso em: 4 nov. 2014.

MANZANO, José A. N. G.; TOLEDO, Suely Alves. Guia de Orientação e Desenvolvimento de Sites: HTML, XHTML, CSS e JavaScript/JScript. São Paulo: Érica, 2010.

MEYER, Érica A. Smashing CSS. Técnicas Profissionais para um Layout Moderno. Dados Eletrônicos. Porto Alegre: Bookman, 2011.

MILETO, Evandro Manara; BERTAGNOLLI, Sílvia de Castro. Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP. Dados Eletrônicos. Porto Alegre: Bookman, 2014.

PIVETTA, Elisa Maria. Apostila de HTML e CSS. Colégio Agrícola Frederico Westphalen – UFSM. 2010. Disponível em: <<https://www.telecom.uff.br/pet/petws/downloads/tutoriais/css/css2k80912.pdf>>. Acesso em: 3 nov. 2014.

PRADO, Márcio Feliciano do. Apostila de HTML e CSS. UNICAMP. Minicursos Virtuais. Campinas: Unicamp, 2002. Disponível em: <<http://www.marcoprado.eti.br/ensino.html?download=6%3Apostilahtm>>. 2011. Acesso em: 5 nov. 2014.

YAMAMOTO, Kenji. Dicas sobre HTML5 e CSS3. 2010. Disponível em: <<http://www.kenjiyamamoto.com/blog/2010/08/09/dicas-sobre-html5-e-css3>>. Acesso em: 20 nov. 2014.



GLOSSÁRIO

CSS Animation: parte específica do CSS para trabalhar com animações de imagens.

Favicon: pequeno ícone, que está associado a um site que é visto nas barras de endereço. O favicon é relacionado a abas e favoritos, ou seja, a ícones de favoritos.

Flash: aplicativo para trabalhar com imagens animadas.

JavaScript: é uma linguagem de programação interpretada, baseada em scripts, a qual, incorporada às tags da HTML, permite incrementar as páginas da web.

Pixels: unidade de medida para imagem, sendo o menor elemento em um monitor, em que é possível atribuir uma cor. É considerado o menor ponto que forma uma imagem digital.

Sistemas operacionais: são sistemas que possibilitam a interação entre o usuário e o computador. Possuem softwares que têm a função de administrar várias partes da máquina a fim de facilitar a utilização do usuário; além disso, também gerenciam memória e dispositivos.

Estilos utilizados – CSS e HTML



POR DENTRO DO TEMA

A Estrutura Básica do CSS3

Todo o texto deste tema está baseado em Castro (2013), mas também inclui detalhes da leitura de Manzano e Toledo (2010), Meyer (2011), Mileto e Bertagnolli (2014) e Prado (2002).

Até este ponto, trabalhou-se somente com a linguagem HTML, que, através de suas instruções, não só permite definir o significado do conteúdo da página web, mas também estrutura a página de uma forma básica. Agora você vai conhecer e trabalhar com o CSS (Cascading Style Sheets), ou folhas de estilo de cascata, ferramenta que permite definir a aparência das páginas web.

A folha de estilo é um arquivo texto que contém regras para determinar como os elementos em sua página devem ser exibidos. Essas regras podem ser propriedades ou valores que controlam a formatação básica, tais como tamanho da fonte, cor, propriedades de layout, posicionamento, flutuação, controle de impressão, quebras de página, surgimento e desaparecimento de componentes interativos. Um recurso muito bom é que o CSS pode ser criado fora de uma página, na web, para ser aplicado em todas as páginas do site de uma só vez. Dentre suas vantagens, estão a flexibilidade e a eficiência, que permitem economia de tempo e de banda de conexão.

Para iniciar, é preciso conhecer e construir uma regra de estilo. Cada regra de estilo em uma folha de estilo possui duas partes:

1. Seletor, que determina, ou seja, indica os elementos que serão formatados.
2. Bloco de declaração, que descreve a formatação que deve ser executada.

Para a construção de uma regra de estilo, é preciso cumprir os seguintes passos:

1. Digite selector, que identifica o elemento a ser formatado.

2. Na sequência, digite uma chave para abertura, dando início ao bloco de declaração.
3. Seguindo no bloco de declaração, digite property: value, em que property é o nome da propriedade do CSS que dita a formatação a ser aplicada e value é uma de uma lista de opções permitidas para essa propriedade. A declaração dentro do bloco é composta de propriedade e valor separados por dois pontos e finalizada com ponto e vírgula.
4. O passo 3 pode ser repetido quantas vezes forem necessárias.
5. Para finalizar, digite a chave para fechar o bloco de declaração e a regra de estilo.

Para entender melhor esses passos da construção da regra de estilo, veja o seguinte exemplo:

```
h1 {
    color: red;
}
```

Em que: **h1** é o seletor.
color é a propriedade.
red é o valor.
color: red; é a declaração.

As chaves { e } delimitam o bloco de declaração.

Um recurso utilizado para melhor documentar é o que permite adicionar comentários, curtos ou longos, à regra de estilo. Para isso, basta cumprir os seguintes passos:

1. Na sua folha de estilo, digite /* para começar seu comentário.
2. Em seguida, digite o comentário.
3. Para encerrar o comentário, digite */.

As propriedades do CSS possuem diferentes regras sobre quais valores aceitam, mas aqui serão focalizados os sistemas básicos. A maioria delas possui alguns valores predefinidos que podem ser utilizados. Muitas levam um comprimento em seu valor, que precisa conter uma quantidade e uma unidade, sem espaços entre estas, como exemplo: 24px. É raro haver propriedades que aceitam um valor em um formato de um número sem unidade. A propriedade **float** pode ser configurada para left, right ou none. O valor **inherit** pode ser usado para qualquer propriedade quando precisa especificar que o valor para aquela propriedade é o mesmo que o do pai elemento.

Há propriedades que permitem especificar a URL de outro arquivo, especialmente

imagens. Nesse caso, deve-se utilizar url(file.ext), em que file.ext é o caminho e o nome do arquivo do **asset** desejado. Uma parte interessante é que é possível especificar cores para as propriedades do CSS de várias maneiras. O valor pode ser uma das cores predefinidas. Na prática, é mais comum utilizar as cores que você deseja do CSS com o formato **hexadecimal** ou **RGB**. A cor no CSS pode ser expressa em valor numérico RGB de 0 a 255, como segue no exemplo:

```
color: rgb(89, 0, 127);
```

Sendo: 89 a quantia de vermelho na cor.

0 a quantia de verde na cor.

127 a quantia de azul na cor.

Com números hexadecimais, que é a forma mais comum de definir cor no CSS, ficaria assim:

```
color: #59007f;
```

Sendo 59 a quantia de vermelho na cor

00 a quantia de verde na cor.

7f a quantia de azul na cor.

Na Figura 4.1 é possível observar um exemplo do código e o resultado de cores com a utilização do CSS.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<title>Minha primeira página CSS</title>
<style type="text/css">
body {
    color: purple;
    background-color: #d8da3d
}
</style>
</head>
<body>
[etc.]
```

Figura 4.1 Código e resultado da página com cores



Fonte: adaptado de BOS (2002).

Há várias outras opções para trabalhar com cores no CSS, por meio do RGBA, HSL e HSLA, que são novidades do CSS3.

Folhas de Estilo

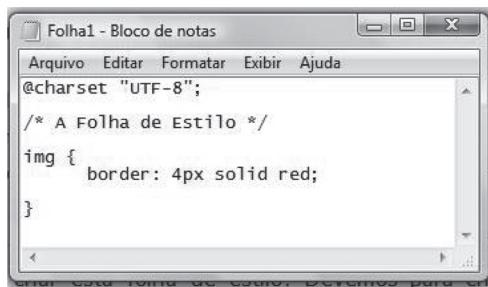
Para trabalhar com folhas de estilo, primeiramente é preciso saber usar os arquivos que as contém, como criá-los e depois usá-los. O CSS permite aplicar as folhas de estilo de um arquivo a várias páginas; pode ser a todo o site, a uma só página ou a algum elemento HTML individual. Para esse recurso há três métodos que se pode utilizar: folhas de estilo externas (a melhor escolha), folhas de estilo incorporadas e estilos **inline**, sendo esta última opção a menos desejável.

Como mencionado, as folhas de estilo externas são as mais indicadas, representando a melhor prática, já que são ideais para dar um visual consistente às páginas. Todos os estilos podem ser definidos como folha de estilo externa. Mas como criar as folhas de estilo externas? Para criá-las, é preciso cumprir os seguintes passos:

1. Inicialmente, crie um arquivo em seu editor de texto.
2. Na sequência, defina, conforme descrição anterior, as regras de estilo para as suas páginas da web e lembre-se de que os comentários podem ser inseridos.
3. Para finalizar, salve o documento em formato de texto na pasta desejada. O nome do arquivo pode ser de sua escolha, mas a extensão deve ser CSS para determinar que é uma folha de estilo em cascata.

A Figura 4.2 mostra o código de uma folha de estilo simples. O importante, vale ressaltar mais uma vez, é lembrar que o arquivo precisa ter extensão CSS.

Figura 4.2 A folha de estilo



```

Folha1 - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
@charset "UTF-8";
/* A Folha de Estilo */
img {
    border: 4px solid red;
}

```

Fonte: adaptado de Castro (2013).

Após criada a folha de estilo, é necessário carregá-la nas páginas HTML, como ilustra o código da Figura 4.3, a fim de que as regras de estilo sejam aplicadas ao conteúdo. Mas para isso é preciso vincular a folha de estilo na seguinte sequência:

1. Para iniciar, digite <link rel="stylesheet" na seção head de cada página HTML em que queira usar a folha de estilo.
2. Em seguida, digite um espaço e href="urlcss", em que url.css é o nome da folha de estilo.
3. Para encerrar digite um espaço e />.

Figura 4.3 Código de uma folha de estilo externa vinculada



```

<head>
<link rel="stylesheet" type="text/css"
      href="meuestilo.css">
</head>

```

Fonte: adaptado de <<http://www.clem.ufba.br/tuts/html/c17.htm>>.

Uma segunda forma de aplicar o CSS a uma página web pode ser criando uma folha de estilo incorporada. Esta permite definir estilos diretamente no arquivo HTML, geralmente na seção head. Em virtude de os estilos estarem no arquivo HTML, o CSS não será aplicado a outras páginas. Para criação deste estilo, é necessário cumprir as seguintes etapas:

1. Como é no documento HTML, dentro deste digite <style> na seção head.
2. Na sequência, podem ser digitadas quantas regras de estilo você escolher.
3. Para finalizar, digite </style>, que encerra a folha de estilo incorporada.

Há também o estilo inline, que corresponde à opção menos desejável, em razão

de entrelaçar o conteúdo HTML e a apresentação CSS. Isso ocorre porque modifica apenas um elemento, fazendo, dessa forma, com que você perca um dos principais benefícios que a folha de estilo fornece de escrever somente uma vez mas de visualizar por toda parte. Para aplicar este estilo, cumpra os passos seguintes:

1. Para iniciar, digite `style="` dentro do elemento HTML que quiser formatar.
2. Em seguida, crie uma regra de estilo sem colchetes ou seletores.
3. Para criar definições de estilo adicionais, digite `";"` (ponto e vírgula) e repita o passo 2.
4. Para finalizar, digite `".`.

Assim são as folhas de estilos que podem ser utilizadas no CSS.

Os Seletores

Os seletores determinam em quais elementos uma regra de estilo, ou seja, a formatação será aplicada, diferenciando-se das declarações, que definem apenas o formato a ser aplicado. Um seletor pode definir até cinco critérios diferentes para escolher os elementos que devem ser formatados, sendo eles:

1. Tipo ou nome do elemento, como mostra o quadro a seguir:

```
h1 {
    color: red;
}
```

Em que: **h1** é o nome do
elemento desejado.

2. O contexto em que o elemento é encontrado, sendo o estilo aplicado somente aos elementos em dentro dos elementos h1, sendo:

```
h1 em {
    color: red;
}
```

Em que: **h1** é o contexto.
em é o nome do elemento
desejado.

3. A class ou id de um elemento. Para isso, o primeiro seletor escolhe todos os elementos que pertencem à classe very, e o segundo seletor escolhe o elemento com o id. É importante saber que um id pode aparecer uma vez em uma página, ao passo que a class pode aparecer várias vezes. Neste caso, tem-se:

```
.very {
    color: red;
}
#nome-id {
    color: red;
}
```

Em que: **.very** é a classe.
#nome-id é o id.

4. A pseudoclasse de um elemento ou pseudoelemento, em que o seletor escolhe elementos a que pertencem à pseudoclasse link.

```
a:link {
    color: red;
}
```

Em que: **a** é o nome.
link é a pseudoclasse.

5. Se um elemento tem ou não certos atributos e valores. Neste caso, é possível usar colchetes para adicionar ao seletor a informação desejada sobre os atributos dos elementos, dos valores ou de ambos.

```
a[name]{
    color: red;
}
```

Em que: **a** é o nome.
[name] é o atributo.

Os seletores podem ser combinados; no entanto, essa combinação deve fazer sentido. Deve-se começar de uma forma simples e tornar mais específico somente quando necessário. Na Figura 4.4 a seguir, evidenciam-se um código e o resultado com o uso de seletores.

Exemplo de CSS para seletores por atributo

```
div[id="cabecalho"]{
    font-size:xx-large;
}

div[id^="parte"]){
    height:50%;
}

div[id$="inferior"]{
    background:#87CEEB;
}

div[id*="roda"]{
    background:red;
}
```

Figura 4.4 Código e resultado de uma página com seletores



Fonte: Rodrigues (2014).

Texto com Estilos

Com o CSS é possível formatar o texto com estilos, ou seja, pode-se modificar sua fonte, seu tamanho, sua inclinação, sua altura da linha e suas cores tanto de primeiro como do segundo plano, além do espaçamento e do alinhamento. Estas alterações podem ser aplicadas a um documento ou site somente com algumas linhas de código.

Um dos recursos mais práticos utilizados corresponde à família de fontes. A partir desta é possível escolher uma fonte a ser utilizada em títulos e no corpo do texto, mas apenas por uma única família. Para definir essa família de fontes, após ter o seletor desejado na folha de estilo, deve-se digitar `font-family: name;`, em que `name` é a sua primeira escolha da fonte. Quando se quer especificar fontes alternativas, é preciso cumprir os seguintes passos:

1. Primeiro, digite `font-family: name;`, em que `name` é a sua primeira escolha da fonte.
2. Em seguida, digite `nome2,` em que `nome2` é a segunda escolha da fonte. As escolhas precisam ser separadas com uma vírgula e um espaço.
3. O passo 2 pode ser repetido quantas vezes forem necessárias, mas a lista deve ser concluída com um nome de fonte genérico, ou seja, o mais parecido com a fonte favorita.

No código da Figura 4.5 mostram-se uma família de fontes com fontes alternativas e também o resultado desse código.

Ainda sobre fontes, o CSS fornece o recurso para criar itálicos, uma vez que

estes são muito utilizados para indicar citações, dar ênfase em alguma palavra ou trecho do texto, para grafar palavras estrangeiras, nomes científicos, dentre outros. A propriedade do CSS font-style permite que isso seja feito em qualquer elemento. Há a opção de definir no texto a fonte font-style: italic, ou oblique para oblíquo. Para remover um itálico, basta digitar font-style: normal.

Da mesma forma que se coloca o itálico, às vezes é necessário utilizar o negrito, fonte que segue o mesmo padrão, para a qual se deve digitar "font-weight:". Na sequência, é possível digitar bold para dar um peso em negrito ao texto, mas pode-se digitar bolder ou lighter para usar um valor relativo ao peso atual. Se desejar remover o negrito, digite font-weight: normal.

```

Código
<head>
<style type="text/css">
h3 {font-family: times}
p {font-family: courier}
p.sansserif {font-family: sans-serif}
</style>
</head>
<body>
<h3>Isto é título 3</h3>
<p>Este é um parágrafo</p>
<p class="sansserif">Este é um parágrafo</p>
</body>

```

Figura 4.5 Código e resultado em uma página

Resultado
Isto é título 3
Este é um parágrafo
Este é um parágrafo

Fonte: <<http://www.clem.ufba.br/tuts/css/ex13h.htm>>.

Em um texto é importante definir o tamanho da fonte. Para isso, há duas formas básicas: mandando que um tamanho específico seja usado ou mandando que esse tamanho seja relativo ao tamanho da fonte do elemento pai. Para definir um tamanho de fonte específico, é importante cumprir as seguintes etapas:

1. Primeiramente, digite font-size:.
2. Na sequência, digite o tamanho específico, como exemplo 13px. No entanto, é também possível usar uma palavra-chave para especificar o tamanho, como exemplo: xx-small, x-small, médium, x-large.

Por outro lado, para definir um tamanho que dependa do tamanho do elemento pai, deve-se:

1. Primeiro digitar font-size:.
2. Em seguida, digitar o valor relativo após os dois pontos (:), por exemplo, 1.5em ou 150%.
3. Neste caso, pode-se usar uma palavra-chave relacionada, o que não é comum. Como exemplo: larger ou small.

Em um texto, quando um parágrafo é definido, é preciso determinar a altura da linha, pois, a depender dessa altura, é possível facilitar a leitura de seu texto e também definir um estilo seu. Para determinar a altura da linha:

1. Primeiramente, digite line-height: .
2. Na sequência, digite n, que corresponde ao número que será multiplicado pelo tamanho da fonte do elemento para obter a altura da linha.
3. Outra opção é digitar a, onde a é o valor em pixels ou pontos.
4. Pode-se também digitar p%, em que p representa uma porcentagem do tamanho da fonte.

No seu texto você pode ter a opção de mudar a cor dos elementos da sua página cumprindo os seguintes passos:

1. Primeiramente, digite color:.
2. Em seguida, digite o nome da cor, uma das cores predefinidas do CSS.

Na Figura 4.6, são mostrados o exemplo de um código com cores e o resultado na página.

Figura 4.6 Código e resultado em uma página com cores

```

<html>
<head>
<style type="text/css">
<!--
h1 {color: #FF0000;}
h2 {color: #00FF00;}
p {color: rgb(0,0,255);}
-->
</style>
</head>
<body>
<h1>Cabeçalho com letras vermelhas</h1>
<h2>Cabeçalho com letras verdes</h2>
<p>Parágrafo com letras azuis</p>
</body>
</html>

```



```

<html>
<head>
<style type="text/css">
<!--
h1 {text-align: left;}
h2 {text-align: center;}
h3 {text-align: right;}
p {text-align: justify;}
-->
</style>
</head>
<body>
<h1>Cabeçalho com letras verdes</h1>
<h2>Cabeçalho com letras verdes</h2>
<p>Parágrafo com letras azuis</p>
</body>
</html>

```

Fonte: Silva (2005).

Um grande recurso do CSS é o alinhamento do texto, que pode ser à esquerda, à direita, ao centro ou justificado, como mostram o código e o resultado na Figura 4.7.

Figura 4.7 Código e resultado em uma página com texto alinhado

```

<html>
<head>
<style type="text/css">
<!--
h1 {text-align: left;}
h2 {text-align: center;}
h3 {text-align: right;}
p {text-align: justify;}
-->
</style>
</head>
<body>
<h1>Este é o cabeçalho 1</h1>
<h2>Este é o cabeçalho 2</h2>
<h3>Este é o cabeçalho 3</h3>
<p>Este é o parágrafo cujo texto ...</p>
</body>
</html>

```

Este é o cabeçalho 1
Este é o cabeçalho 2
Este é o cabeçalho 3

Fonte: Silva (2003).

Quando se trabalha com folhas de estilo, é possível decorar o texto com sublinhados e linhas. Para esse efeito, ou seja, para decorar o texto, é necessário:

1. Primeiramente, digitar text-decoration:.
2. Em seguida, digitar underline após os dois pontos (:) para sublinhar o texto.
3. Há a opção de digitar overline para adicionar uma linha sobre o texto.
4. Pode-se também digitar line-through para riscá-lo.

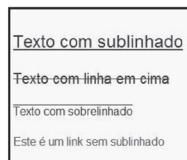
É comum decorar o texto para indicar mudanças. Na Figura 4.8 a seguir pode-se observar o código e o resultado de uma decoração no texto.

Figura 1.1 Pirâmide dos Alimentos

```

<html>
<head>
<style type="text/css">
<!--
h1 {text-decoration: underline;}
h2 {text-decoration: line-through;}
h3 {text-decoration: overline;}
a {text-decoration: none;}
-->
</style>
</head>
<body>
<h1>Texto com sublinhado</h1>
<h2>Texto com linha em cima</h2>
<h3>Texto com sobrelinhado</h3>
<p>
<a href="http://www.maujor.com">Este é um link sem sublinhado</a>
</p>
</body>
</html>

```



Fonte: Silva (2013).

Concluindo, você viu que o CSS fornece inúmeros recursos, com os quais é possível trabalhar de formas bem variadas, que podem compor uma página bem estruturada na web.



ACOMPANHE NA WEB

Apostila HTML e CSS

Acesse uma apostila que abrange a linguagem HTML e o CSS. É interessante que você faça uma consulta a partir da página 48 para ver detalhes dos elementos a serem utilizados nas páginas, aprofundando um pouco mais o modo como lidar com o CSS.

Disponível em: <<https://www.telecom.uff.br/pet/petws/downloads/tutoriais/css/css2k80912.pdf>>. Acesso em: 28 nov. 2014.

Dicas sobre HTML5 e CSS3

Para complementar este tema, há uma página que apresenta o assunto com dicas bem interessantes de como trabalhar com o CSS. Vale a pena consultá-la.

Disponível em: <<http://www.kenjiyamamoto.com/blog/2010/08/09/dicas-sobre-html5-e-css3>>. Acesso em: 1 dez. 2014.



AGORA É A SUA VEZ

Instruções:

Agora, chegou a sua vez de exercitar seu aprendizado. A seguir, você encontrará algumas questões de múltipla escolha e dissertativas. Leia cuidadosamente os enunciados e atente-se para o que está sendo pedido.

Questão 1

O CSS (cascading style sheets, ou folhas de estilo de cascata) é a ferramenta que permite definir a aparência das páginas web. A folha de estilo é um arquivo texto que contém regras para determinar como os elementos em sua página devem ser exibidos. Quais são estas regras?

Questão 2

Um recurso que existe no CSS3 para melhor documentar é o de adicionar comentários, curtos ou longos, à regra de estilo. Como são inseridos comentários no arquivo CSS3?

Assinale a alternativa correta:

- a) "Este é um comentário".
- b) 'Este é um comentário'.
- c) /* Este é um comentário*/.
- d) < Este é um comentário/>.
- e) / Este é um comentário/.

Questão 3

A folha de estilo é um arquivo texto que contém regras para determinar como os elementos em sua página devem ser exibidos. As regras podem controlar a formatação da cor. Assinale a alternativa que mostra a sintaxe correta de uma regra de estilo:

- a) [body; color: red];
- b) {body color: red}
- c) body {color; red}
- d) body {color: red};
- e) body {color: red}

Questão 4

Com base nos códigos mostrados (HTML e CSS), faça uma Folha de Estilo Externa com as seguintes características:

- Cor de fundo laranja.

- Cabeçalho de nível 1 com 36 pt.
- Cabeçalho de nível 2 de cor azul.
- Margem esquerda com 50 pixels.

Questão 5

É possível formatar o texto com estilos, ou seja, com o CSS você pode modificar a fonte, o tamanho, a inclinação, a altura da linha e as cores tanto de primeiro como de segundo plano, além do espaçamento e do alinhamento de texto. Como essas alterações podem ser aplicadas?



FINALIZANDO

Neste tema você pôde conhecer o CSS (Cascading Style Sheets), ferramenta que permite definir a aparência das páginas web. A partir do estudo, da leitura, da pesquisa e dos exercícios propostos neste tema, você pôde aprender a estrutura básica do CSS3. Você agora está apto a trabalhar com as folhas de estilo, os seletores e o texto com estilo. Suas páginas agora terão uma aparência mais bem trabalhada com o uso dos melhores recursos oferecidos pelo CSS3. A princípio você aprendeu a construir uma regra de estilo, em uma folha de estilo com suas duas partes: os seletores e o bloco de declaração. Na sequência, você viu como trabalhar com folhas de estilo, saber usar os arquivos que as contém, como criá-los e depois usá-los. Você foi apresentado, ainda, aos seletores, que determinam em quais elementos uma regra de estilo a formatação será aplicada, diferentemente das declarações, que definem apenas o formato a ser aplicado. Para concluir, você aprendeu, com ajuda do CSS, a formatar o texto com estilos, ou seja, a modificar a fonte, o tamanho, a inclinação, a altura da linha e as cores tanto de primeiro como de segundo plano, o espaçamento e o alinhamento de texto, compreendendo também como decorar o texto.



REFERÊNCIAS

BOS, B. Tutorial CSS. Primeiros Passos com HTML + CSS. Trad. Mauricio Samy Silva. W3C, 2002. Disponível em: <<http://www.maujor.com/w3ctuto/firstcss.html>>. Acesso em: 28 jan. 2015.

CASTRO, Elizabeth. HTML5 e CSS3 guia prático e visual. São Paulo: Alta Books, 2013.

MANZANO, José A. N. G.; TOLEDO, Suely Alves. Guia de Orientação e Desenvolvimento de Sites: HTML, XHTML, CSS e JavaScript/JScript. São Paulo: Érica, 2010.

MEYER, Érica A. Smashing CSS. Técnicas profissionais para um layout moderno. Dados Eletrônicos. Porto Alegre: Bookman, 2011.

MILETO, Evandro Manara; BERTAGNOLLI, Sílvia de Castro. Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP. Dados Eletrônicos. Porto Alegre: Bookman, 2014.

PIVETTA, Elisa Maria. Apostila de HTML e CSS. Colégio Agrícola Frederico Westphalen – UFSM. 2010. Disponível em: <<https://www.telecom.uff.br/pet/petws/downloads/tutoriais/css/css2k80912.pdf>>. 2010. Acesso em: 3 nov. 2014.

PRADO, Márcio Feliciano do. Apostila de HTML e CSS. UNICAMP. Minicursos virtuais. Campinas: Unicamp, 2002. Disponível em <<http://www.marcoprado.eti.br/ensino.html?download=6%3Aapostilahtml>>. 2011. Acesso em: 5 nov. 2014.

RODRIGUES, J. CSS3 Selectors: Conhecendo os Seletores das CSS3. Devmedia, 2014. Disponível em: <<http://www.devmedia.com.br/css3-selectors-conhecendo-os-seletores-das-css3/24782>>. Acesso em: 27 jan. 2015.

SILVA, M. S. Tutoriais CSS, Web Standards, Acessibilidade, HTML, XHTML, Padrões Web – A Propriedade CSS Text. Maujor, 2003. Disponível em: <<http://www.maujor.com/tutorial/fonttut.php>>. Acesso em: 28 jan. 2015.

_____. Tutoriais CSS, Web Standards, Acessibilidade, HTML, XHTML, Padrões Web – A Propriedade CSS Font. Maujor, 2005. Disponível em: <<http://www.maujor.com/tutorial/fonttut.php>>. Acesso em: 28 jan. 2015.

YAMAMOTO, Kenji. Dicas sobre HTML5 e CSS3. 2010. Disponível em: <<http://www.kenjiyamamoto.com/blog/2010/08/09/dicas-sobre-html5-e-css3>>. Acesso em: 20 nov. 2014.



GLOSSÁRIO

Asset: termo utilizado para designar o que complementa o conteúdo das páginas web.

Float: tipos de dados para que se possa trabalhar com números reais.

Hexadecimal: é um sistema de numeração que representa os números na base 16.

Inherit: propriedade do CSS destinada ao uso dos elementos filhos. O valor inherit serve para que o filho sempre herde um valor de uma propriedade do pai.

Inline: é utilizado para designar estilos inline, ou seja, quando as regras de estilos são declaradas dentro do arquivo HTML.

RGB: é um sistema de cores com a abreviatura de vermelho (Red), verde (Green) e azul (Blue). É um modelo aditivo com base no qual as cores vermelho, verde e azul são combinadas para reproduzir outras cores.

ISBN 978-85-8482-621-6



9 788584 826216 >