

Heading Level 3

Heading Level 4

Heading Level 5

Heading Level 6



Hands-On Practice 2.2

To create the web page shown in Figure 2.6, launch Notepad or another text editor. Select File > Open to edit the HTML5 template file located at chapter2/template.html in the student files. Modify the title element and add heading tags to the body section as indicated by the following highlighted code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Heading Example</title>
<meta charset="utf-8">
</head>
<body>
<h1>Heading Level 1</h1>
<h2>Heading Level 2</h2>
<h3>Heading Level 3</h3>
<h4>Heading Level 4</h4>
<h5>Heading Level 5</h5>
<h6>Heading Level 6</h6>
</body>
</html>
```

Save the document as heading.html on your hard drive or flash drive. Launch a browser such as Microsoft Edge or Firefox to test your page. It should look similar to the page shown in Figure 2.6. You can compare your work with the solution found in the student files (chapter2/heading.html).



Hands-On Practice 2.4

To create the web page shown in Figure 2.9, launch a text editor. Select File > Open to edit the file located at chapter2/paragraph.html in the student files. Modify the text contained between the title tags to be “Line Break Example”. Place your cursor after the first sentence in the paragraph (after “This is a sample paragraph.”). Press the Enter key. Save your file. Test your page in a browser, and notice that even though your source code showed the “This is a sample paragraph.” sentence on its own line, the browser did not render it that way. A line break tag is needed to configure the browser to display the second sentence on a new line. Edit the file in a text editor, and add a `
` tag after the first sentence in the paragraph, as shown in the following code snippet:

```
<body>
<h1>Heading Level 1</h1>
<p>This is a sample paragraph. <br> Heading tags can help to make your
pages more accessible and usable. It is good coding practice to use
heading tags to outline the structure of your web page content.
</p>
<h2>Heading Level 2</h2>
<h3>Heading Level 3</h3>
<h4>Heading Level 4</h4>
<h5>Heading Level 5</h5>
<h6>Heading Level 6</h6>
</body>
```

Save your file as `linebreak.html`. Launch a browser to test your page. It should look similar to the page shown in Figure 2.9. You can compare your work with the solution found in the student files (`chapter2/linebreak.html`).



Hands-On Practice 2.5

To create the web page shown in Figure 2.10, launch a text editor. Select File > Open to edit the template file located at chapter2/template.html in the student files. Modify the title element. Add a heading tag, a paragraph tag, and a blockquote tag to the body section as indicated by the following highlighted code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Blockquote Example</title>
<meta charset="utf-8">
</head>
<body>
<h1>The Power of the Web</h1>
<p>According to Tim Berners-Lee, the inventor of the World Wide Web,  
at http://www.w3.org/WAI/:</p>
<blockquote>
The power of the Web is in its universality. Access by everyone  
regardless of disability is an essential aspect.
</blockquote>
</body>
</html>
```

2.11 Phrase Elements

Save the document as `blockquote.html` on your hard drive or flash drive. Launch a browser to test your page. It should look similar to the page shown in Figure 2.10. You can compare your work with the solution (`chapter2/blockquote.html`) found in the student files.



Hands-On Practice 2.6

In this Hands-On Practice, you will use a heading and an ordered list on the same page. To create the web page shown in Figure 2.13, launch a text editor. Select File > Open to edit the template file located at chapter2/template.html in the student files. Modify the title element and add h1, ol, and li elements to the body section as indicated by the following highlighted code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Heading and List</title>
<meta charset="utf-8">
</head>
```

Hands-On Practice 2.7

In this Hands-On Practice, you will use a heading and an unordered list on the same page. To create the web page shown in Figure 2.15, launch a text editor. Select File > Open to edit the template file located at chapter2/template.html in the student files. Modify the title element and add h1, ul, and li tags to the body section as indicated by the following highlighted code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Heading and List</title>
<meta charset="utf-8">
</head>
<body>
<h1>My Favorite Colors</h1>
<ul>
    <li>Blue</li>
    <li>Teal</li>
    <li>Red</li>
</ul>
</body>
</html>
```



Figure 2.15 An unordered list

Save your file as ul.html. Launch a browser and test your page. It should look similar to the page shown in Figure 2.15. You can compare your work with the solution in the student files (chapter2/ul.html).



Hands-On Practice 2.8

In this Hands-On Practice, you will use a heading and a description list on the same page. To create the web page shown in Figure 2.16, launch a text editor. Select File > Open to edit the template file located at chapter2/template.html in the student files. Modify the title element and add h1, dl, dd, and dt tags to the body section as indicated by the following highlighted code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Description List</title>
```

2.14 Description List

```
<meta charset="utf-8">
</head>
<body>
<h1>Sample Description List</h1>
<dl>
    <dt>TCP</dt>
        <dd>Transmission Control Protocol is a method (protocol) used along with the Internet Protocol (IP) to send data in the form of message units, called packets, between computers over the Internet.</dd>
    <dt>IP</dt>
        <dd>Internet Protocol is the method or protocol by which data is sent from one computer to another on the Internet. Each computer on the Internet is uniquely identified by an IP address.</dd>
    <dt>FTP</dt>
        <dd>File Transfer Protocol is a protocol used to exchange files between computers on the Internet.</dd>
    <dt>HTTP</dt>
        <dd>Hypertext Transfer Protocol is the protocol used for exchanging text, graphic images, sound, video, and other multimedia files on the Web.</dd>
</dl>
</body>
</html>
```

Save your file as `description.html`. Launch a browser and test your page. It should look similar to the page shown in Figure 2.16. Don't worry if the word wrap is a little different; the important formatting is that each `<dt>` term should be on its own line and the corresponding `<dd>` description should be indented under it. Try resizing your browser window, and notice how the word wrap on the description text changes. You can compare your work with the solution in the student files (`chapter2/description.html`).



Hands-On Practice 2.9

Figure 2.17 shows the web page you will create in this Hands-On Practice. Launch a text editor. Select File > Open to edit the template file located at chapter2/template.htm in the student files. Save your file as design.html. Modify the title of the web page by changing the text between the `<title>` and `</title>` tags to “Web Design Steps.”

The sample page shown in Figure 2.17 contains a heading, an unordered list, and a right information. Configure the heading “Web Design Steps” as a level 1 heading (`<h1>`) as follows:

```
<h1>Web Design Steps</h1>
```

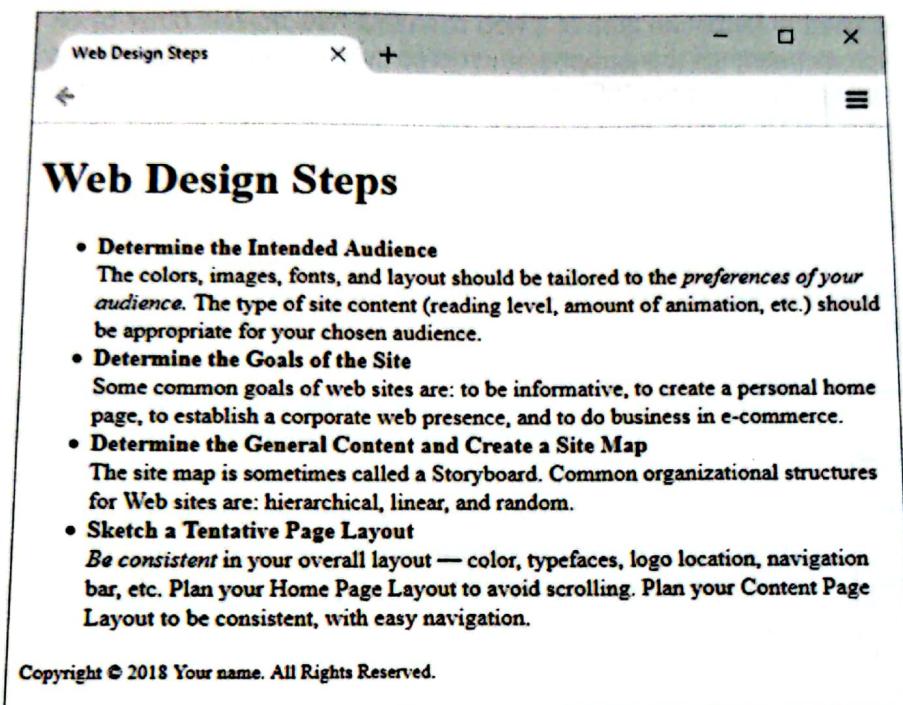


Figure 2.17 The design.html web page

Now create the unordered list. The first line of each bulleted item is the title of the web design step, which should be strong or stand out from the rest of the text. The code for the beginning of the unordered list is as follows:

```
<ul>
  <li><strong>Determine the Intended Audience</strong><br>
    The colors, images, fonts, and layout should be tailored to the
    <em>preferences of your audience.</em> The type of site content
    (reading level, amount of animation, etc.) should be appropriate for
    your chosen audience.</li>
```

Now code the entire unordered list in your design.html file. Remember to code the closing `` tag at the end of the list. Don't worry if your text wraps a little differently; your screen resolution or browser window size may be different from what is displayed in Figure 2.17.

Finally, configure the copyright information with the small element. Use the special character © for the copyright symbol. The code for the copyright line is as follows:

```
<p><small>Copyright &copy; 2018 Your name. All Rights Reserved.
</small></p>
```

How did you do? Compare your work to the sample in the student files (chapter2/design.html).



Hands-On Practice 2.10

In this Hands-On Practice you will use structural elements as you create the Trillium Media Design home page, shown in Figure 2.19. Launch a text editor, and open the template.html file from the chapter2 folder in the student files. Edit the code as follows:

1. Modify the title of the web page by changing the text between the `<title>` and `</title>` tags to Trillium Media Design.
2. Position your cursor in the body section and code the header element with the text, "Trillium Media Design" contained in an `h1` element:

```
<header>
  <h1> Trillium Media Design</h1>
</header>
```

3. Code a nav element to contain text that will indicate the main navigation for the website. Configure bold text (use the `b` element) and use the ` ` special character to add extra blank space:

```
<nav>
  <b>Home &nbsp; Services &nbsp; Contact</b>
</nav>
```

4. Code a main element that contains the `h2` and paragraph elements:

```
<main>
  <h2>New Media and Web Design</h2>
  <p>Trillium Media Design offers a comprehensive range of
  services to take your company's Web presence to the next
  level.</p>
  <h2>Meeting Your Business Needs</h2>
  <p>Our expert designers will listen to you as they create a
  website that helps to promote and grow your business.</p>
</main>
```

5. Configure the footer element to contain a copyright notice displayed in small font size (use the `small` element) and italic font (use the `i` element). Be careful to properly nest the elements as shown here:

```
<footer>
  <small><i>Copyright &copy; 2018 Your Name Here</i></small>
</footer>
```

Save your page as `structure.html`. Test your page in a browser. It should look similar to Figure 2.19. You can compare your work to the sample in the student files (`chapter2/structure.html`).

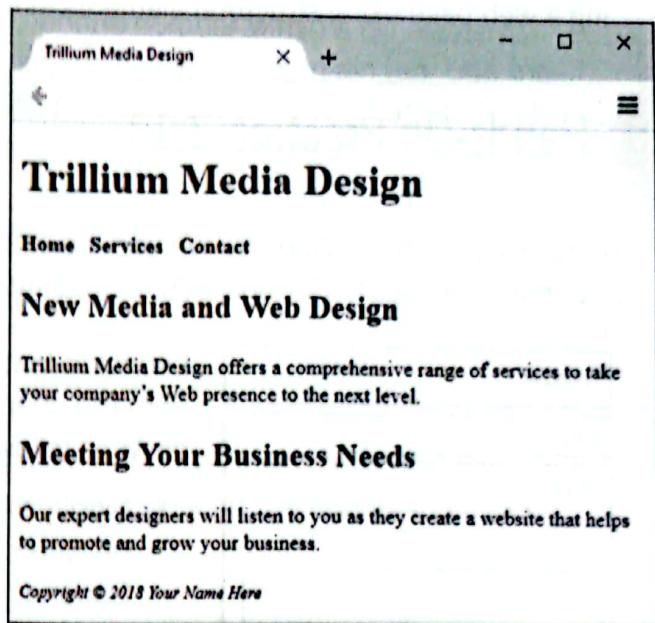


Figure 2.19 Trillium home page

Hands-On Practice 2.11

In this Hands-On Practice you will use the wireframe shown in Figure 2.20 as a guide as you create the Casita Sedona Bed & Breakfast web page, shown in Figure 2.21.

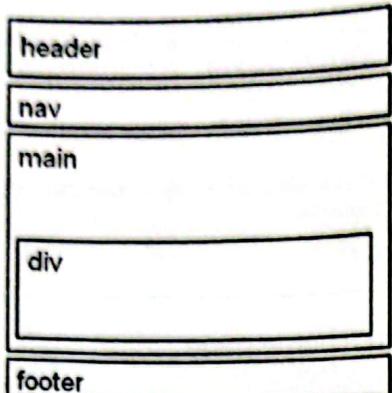


Figure 2.20 Wireframe for Casita Sedona

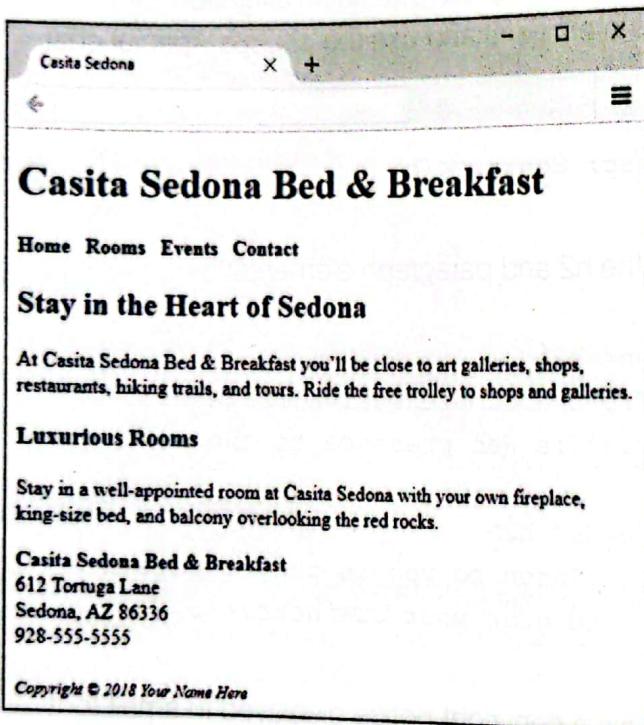


Figure 2.21 Casita Sedona web page

Launch a text editor, and open the template.html file from the chapter2 folder in the student files. Edit the code as follows:

1. Modify the title of the web page by changing the text between the `<title>` and `</title>` tags to Casita Sedona.
2. Position your cursor in the body section and code the header element with the text, "Casita Sedona Bed & Breakfast" contained in an `h1` element. Be sure to use the special character & for the ampersand.

```
<header>
  <h1>
    Casita Sedona Bed & Breakfast
  </h1>
</header>
```

3. Code a nav element to contain text that will indicate the main navigation for the website. Configure bold text (use the `b` element) and use the special character to add extra blank space:

```
<nav>
  <b>
    Home &nbsp;
    Rooms &nbsp;
    Events &nbsp;
    Contact
  </b>
</nav>
```

4. Code the content within a main element. Start with the `h2` and `paragraph` elements:
`<main>`
`<h2>Stay in the Heart of Sedona</h2>`
`<p>At Casita Sedona Bed & Breakfast you'll be close to art galleries, shops, restaurants, hiking trails, and tours. Ride the free trolley to shops and galleries.</p>`
`<h3>Luxurious Rooms</h3>`

```
<p>Stay in a well-appointed room at Casita Sedona with your own  
fireplace, king-size bed, and balcony overlooking the red rocks.</p>  
</main>
```

5. Configure the company name, address, and phone number within a div element. Code the div element *within* the main element before the closing main tag. Use line break tags to display the name, address, and phone information on separate lines and to create extra empty space before the footer.

```
<div>  
  <strong>Casita Sedona Bed & Breakfast</strong><br>  
  612 Tortuga Lane<br>  
  Sedona, AZ 86336<br>  
  928-555-5555<br><br>  
</div>
```

6. Configure the footer element to contain a copyright notice displayed in small font size (use the small element) and italic font (use the i element). Be careful to properly nest the elements as shown here:

```
<footer>  
  <small><i>Copyright © 2018 Your Name Here</i></small>  
</footer>
```

Save your page as casita.html. Test your page in a browser. It should look similar to Figure 2.21. You can compare your work to the sample in the student files (chapter2/casita.html). Older browsers (such as Internet Explorer 8 and earlier) do not support the new HTML5 structural elements. In Chapter 6, we'll explore coding techniques that will force older browsers to correctly display HTML5 structural elements. For now, be sure to use a current version of any popular browser to test your pages.



Hands-On Practice 2.12

To create the web page shown in Figure 2.22, launch a text editor. Select File > Open to edit the template file located at chapter2/template.html in the student files. Modify the title element and add anchor tags to the body section as indicated by the following highlighted code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Anchor Example</title>
<meta charset="utf-8">
</head>
<body>
<a href="http://webdevfoundations.net">Web Development & Design Foundations</a>
</body>
</html>
```

Save the document as anchor.html on your hard drive or flash drive. Launch a browser to test your page. It should look similar to the page shown in Figure 2.22. You can compare your work with the solution found in the student files (chapter2/anchor.html).



Hands-On Practice 2.13

The best way to learn how to code web pages is by actually doing it! Let's practice and create three pages in the website shown in Figure 2.23: home page (index.html) with two content pages: services page (services.html) and contact page (contact.html).

- 1. Create a Folder.** If you had printed papers to organize you would probably store them in a paper folder. Web designers store and organize their computer files by creating a folder on a hard drive (or portable storage such as an SD card or Flash drive) for each website. This helps them to be efficient as they work with many different websites. You will organize your own web design work by creating a new folder for each website and storing your files for that website in the new folder. Use your operating system to create a new folder named mypractice for your new website.
- 2. Create the Home Page.** Use the Trillium Media Design web page (Figure 2.19) from Hands-On Practice 2.10 as a starting point for your new home page (shown in Figure 2.24). Copy the sample file for Hands-On Practice 2.10 (chapter2/structure.html) into your mypractice folder. Change the file name of structure.html to index.html. It's common practice to use the file name index.html for the home page of a website.

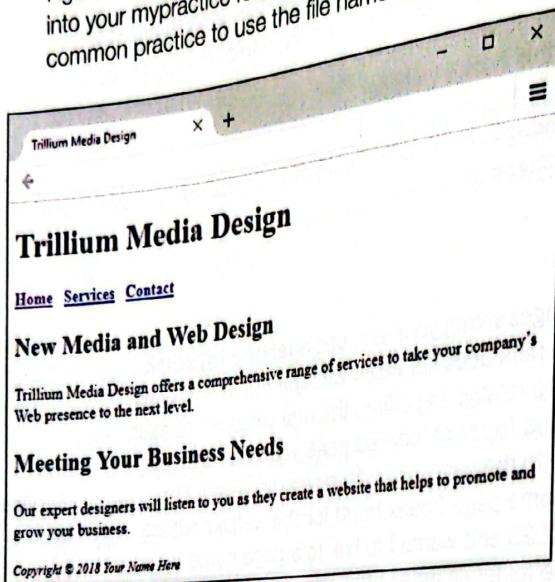


Figure 2.24 New index.html web page

Launch a text editor, and open the index.html file.

- a. The navigation hyperlinks will be located within the nav element. You will edit the code within the nav element to configure three hyperlinks:

- The text "Home" will hyperlink to index.html
- The text "Services" will hyperlink to services.html
- The text "Contact" will hyperlink to contact.html

Modify the code within the nav element as follows:

```
<nav>
  <b><a href="index.html">Home</a> &ampnbsp
    <a href="services.html">Services</a> &ampnbsp
    <a href="contact.html">Contact</a>
  </b>
</nav>
```

- b. Save the index.html file in your mypractice folder. Test your page in a browser. It should look similar to Figure 2.24. You can compare your work to the sample in the student files (chapter2/2.13/index.html).

3. **Create the Services Page.** It is common practice to create a new web page based on an existing page. You will use the index.html file as a starting point for the new services page, shown in Figure 2.25.

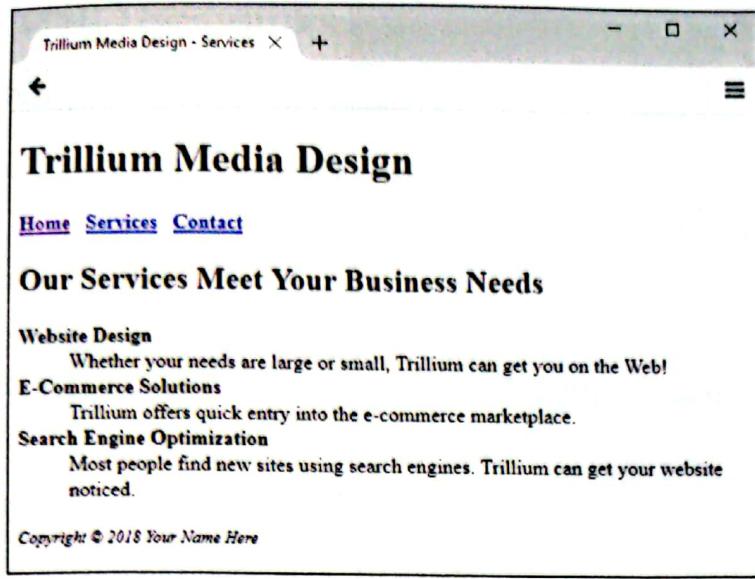


Figure 2.25 The services.html web page

Open your index.html file in a text editor and save the file as services.html. Edit the code as follows:

- a. Modify the title of the web page by changing the text between the `<title>` and `</title>` tags to "Trillium Media Design - Services". In order to create a consistent header, navigation, and footer for the web pages in this website, do not change the code within the header, nav, or footer elements.
- b. Position your cursor in the body section and delete the code and text between the opening and closing main tags. Code the main page content (heading 2 and description list) for the services page between the main tags as follows:

```
<h2>Our Services Meet Your Business Needs</h2>
<dl>
    <dt><strong>Website Design</strong></dt>
        <dd>Whether your needs are large or small, Trillium can
            get you on the Web!</dd>
    <dt><strong>E-Commerce Solutions</strong></dt>
        <dd>Trillium offers quick entry into the e-commerce
            marketplace.</dd>
    <dt><strong>Search Engine Optimization</strong></dt>
        <dd>Most people find new sites using search engines.
            Trillium can get your website noticed.</dd>
</dl>
```

- c. Save the services.html file in your mypractice folder. Test your page in a browser. It should look similar to Figure 2.25. You can compare your work to the sample in the student files (chapter2/2.13/services.html).
4. **Create the Contact Page.** Use the index.html file as a starting point for the new Contact page, shown in Figure 2.26. Open your index.html file in a text editor and save the file as contact.html. Edit the code as follows:

- a. Modify the title of the web page by changing the text between the `<title>` and `</title>` tags to "Trillium Media Design – Contact". In order to create a consistent header, navigation, and footer for the web pages in this website, do not change the code within the header, nav, or footer elements.



Figure 2.26 The contact.html web page

- b.** Position your cursor in the body section and delete the code and text contained between the opening main tag and the closing main tag. Code the main page content for the contact page between the main tags:

```
<h2>Contact Trillium Media Design Today</h2>
<ul>
    <li>E-mail: contact@trilliummediadesign.com</li>
    <li>Phone: 555-555-5555</li>
</ul>
```

- c.** Save the contact.html file in your mypractice folder. Test your page in a browser. It should look similar to Figure 2.26. Test your page by clicking each link. When you click the "Home" hyperlink, the index.html page should display. When you click the "Services" hyperlink, the services.html page should display. When you click the "Contact" hyperlink, the contact.html page will display. You can compare your work to the sample in the student files (chapter2/2.13/contact.html).



FAQ

What if my relative hyperlink doesn't work?

Check the following:

- Did you save files in the correct folder?

For example, to create an e-mail hyperlink to the e-mail address help@terrymorris.net, code the following:

```
<a href="mailto:help@terrymorris.net">help@terrymorris.net</a>
```

It is good practice to place the e-mail address both on the web page and within the anchor tag. Not everyone has an e-mail program configured with his or her browser. By placing the e-mail address in both places, you increase usability for all of your visitors.



Hands-On Practice 2.14

In this Hands-On Practice you will modify the contact page (contact.html) of the website you created in Hands-On Practice 2.13 and configure an e-mail link in the page content area. Launch a text editor, and open the contact.html file from your mypractice folder. This example uses the contact.html file found in the student files in the chapter2/2.13 folder.

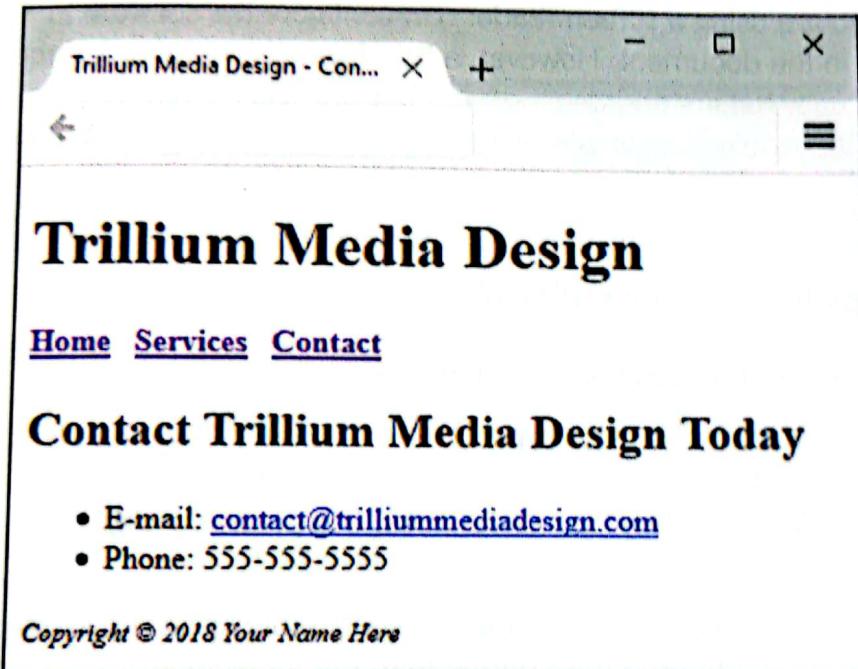


Figure 2.27 An e-mail hyperlink has been configured on the contact page

Configure the e-mail address in the main content area as an e-mail hyperlink as follows:

```
<li>E-mail:  
<a href="mailto:contact@trilliummediadesign.com">contact@  
trilliummediadesign.com</a>  
</li>
```

Save and test the page in a browser. The browser display should look similar to the page shown in Figure 2.27. You can compare your work with the sample in the student files (chapter2/2.14/contact.html).



Hands-On Practice 2.15

In this Hands-On Practice, you will use the W3C Markup Validation Service to validate a web page file. This example uses the page completed in Hands-On Practice 2.9 (located in the student files at chapter2/design.html). Open design.html in a text editor. Add an error to the design.html page by deleting the first closing `` tag. This modification should generate several error messages.

Next, attempt to validate the design.html file. Launch a browser and visit the W3C Markup Validation Service file upload page at http://validator.w3.org/#validate_by_upload. Click the Browse button, and select the chapter2/design.html file from your computer. Click the Check button to upload the file to the W3C site (Figure 2.28).

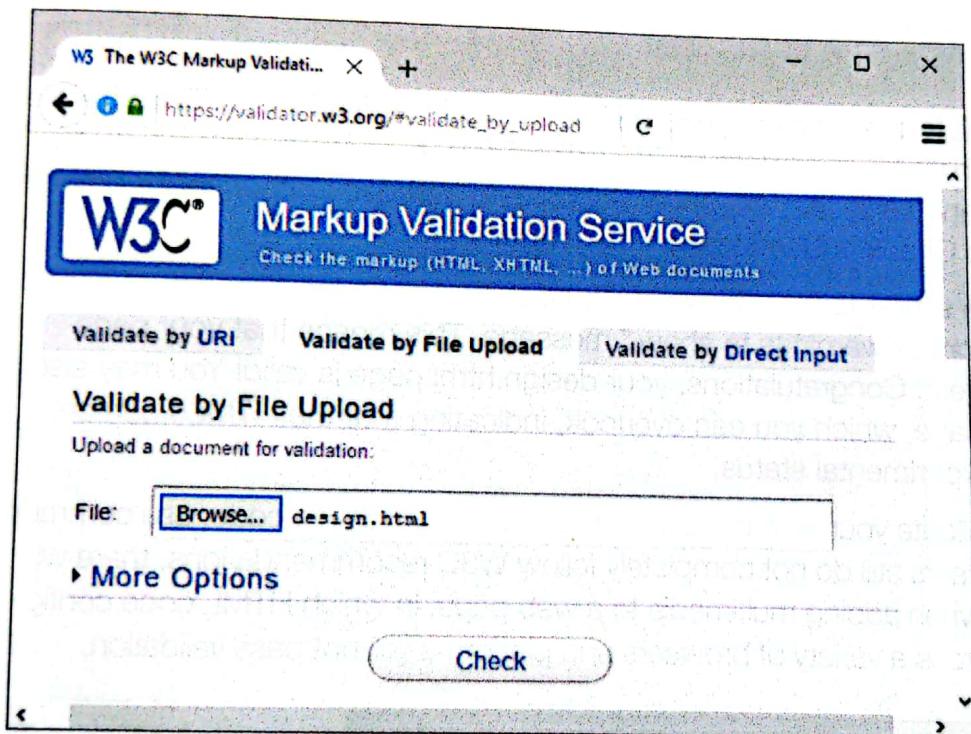


Figure 2.28 Validating a page with the W3C Markup Validation Service. Screenshots of W3C. Courtesy of W3C (World Wide Web Consortium).

An error page will display. Notice the “Errors found while checking this document” message. You can view the errors by scrolling down the page, as shown in Figure 2.29.

Notice that the first message indicates line 12, which is after the missing closing `` tag. HTML error messages often point to a line that follows the error. The text of the message, “End tag li seen, but there were open elements,” lets you know that something is wrong. It is up to you to figure out what it is. A good place to start is to check your container tags and make sure they are in pairs. In this case, that is the problem. You can scroll down to view the other errors. However, since multiple error messages are often displayed after a single error occurs, it is a good idea to fix one item at a time and then revalidate.

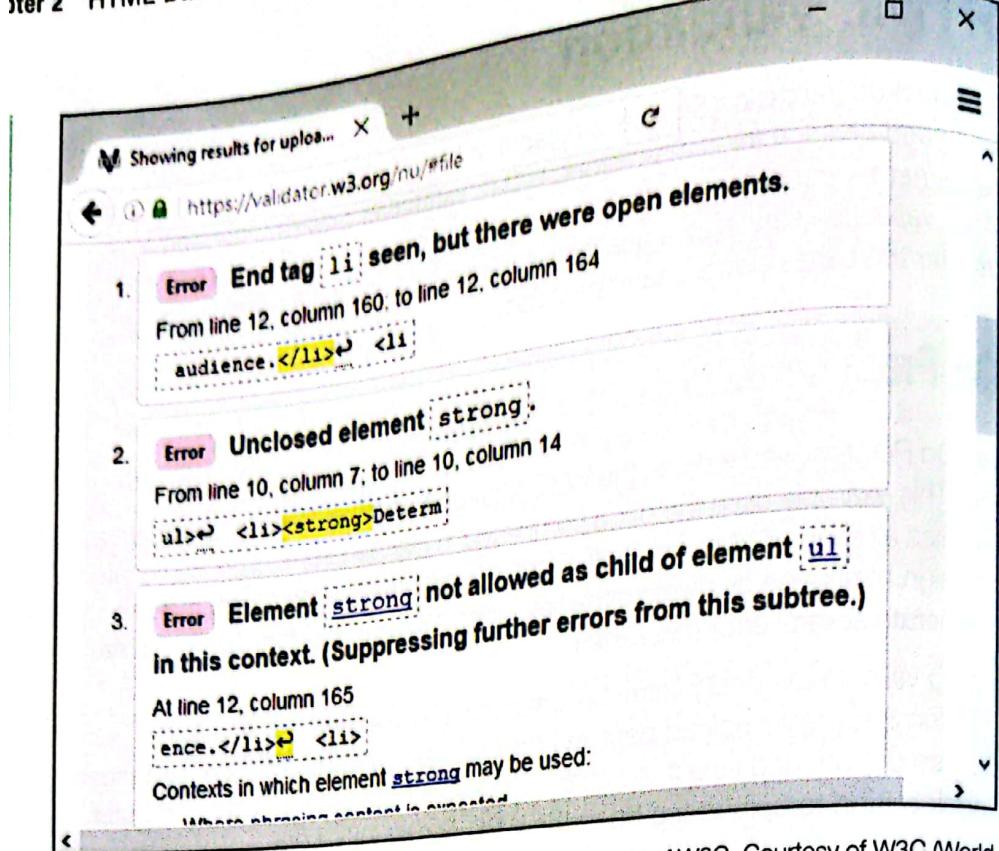


Figure 2.29 The service indicates errors. Screenshots of W3C. Courtesy of W3C (World Wide Web Consortium).

Edit the design.html file in a text editor, and add the missing `` tag. Save the file. Launch a browser, and visit http://validator.w3.org/#validate_by_upload. Select your file, select More Options, and verify that the Show Source and Verbose Output check boxes are checked. Click the Check button to begin the validation.

Your display should be similar to that shown in Figure 2.30. Notice the “Document checking completed. No errors or warnings to show.” message. This means that your page passed the validation test. Congratulations, your design.html page is valid! You may also notice a warning message, which you can overlook, indicating that the HTML5 conformance checker is in experimental status.

It is good practice to validate your web pages. However, when validating code, use common sense. Since web browsers still do not completely follow W3C recommendations, there will be situations, such as when adding multimedia to a web page, in which HTML code configured to work reliably across a variety of browsers and platforms will not pass validation.

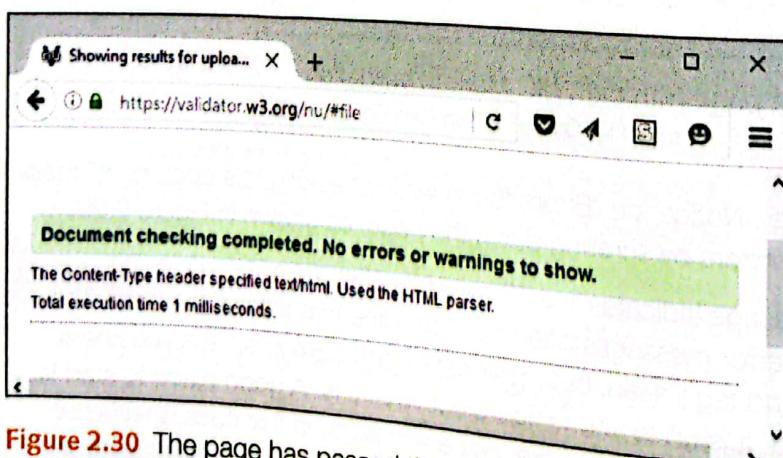


Figure 2.30 The page has passed the validation test. Screenshots of W3C. Courtesy of W3C (World Wide Web Consortium).

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Predict the Result</title>
<meta charset="utf-8">
<body>
<header><h1><i>favorite sites</i></h1></header>
<main>
<ol>
<li><a href="http://facebook.com">Facebook</a></li>
<li><a href="http://google.com">Google</a></li>
</ol>
</main>
<footer>
<small>Copyright &copy; 2018 Your name here</small>
</footer>
</body>
</html>
```

2. **Fill in the Missing Code.** The web page defined by the given code should display a heading and a description list, but some HTML tags, indicated by <_>, are missing. Fill in the missing code.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Door County Wildflowers</title>
<meta charset="utf-8">
</head>
<body>
<header><_>Door County Wild Flowers<_></header>
<main>
<dl>
<dt>Trillium<_>
<_>This white flower blooms from April through June in
wooded areas.<_>
<_>Lady Slipper<_>
<_>This yellow orchid blooms in June in wooded areas.</d
<_>
</main>
</body>
</html>
```

- 3. Find the Error.** All the text on the web page defined by the given code displays in large and bold font typeface. Explain why this is happening.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Find the Error</title>
    <meta charset="utf-8">
</head>
<body>
    <h1>My Web Page</h1>
    <p>This is a sentence on my web page.</p>
</body>
</html>
```

WEBSITE CASE STUDY

Each of the case studies in this section continues throughout most of the text. This chapter introduces each website scenario, presents the site map, and directs you to create two pages for the site.

JavaJam Coffee House

Julio Perez is the owner of the JavaJam Coffee House, a gourmet coffee shop that serves snacks, coffee, tea, and soft drinks. Local folk music performances and poetry readings are held a few nights during the week. The customers of JavaJam are mainly college students and young professionals. Julio would like a web presence for his shop that will display his services and provide a calendar for the performances. He would like a home page, menu page, music performance schedule page, and job opportunities page.

A site map for the JavaJam Coffee House website is shown in Figure 2.31. The site map describes the architecture of the website, which consists of a "Home" page with three main content pages: "Menu," "Music," and "Jobs."

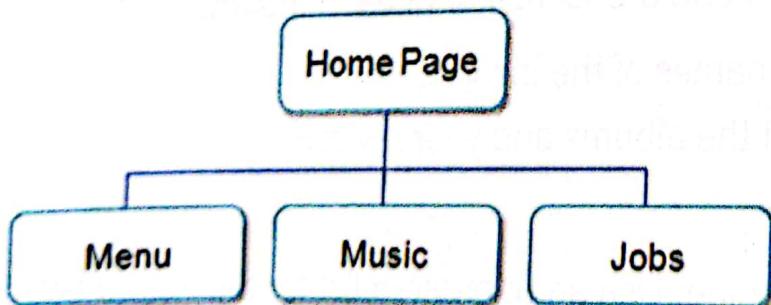


Figure 2.31 JavaJam site map

Figure 2.32 displays a wireframe sketch of the page layout for the website. It contains a header area, a navigation area, a main content area, and a footer area for copyright information.

You have three tasks in this case study:

1. Create a folder for the JavaJam website.
2. Create the Home page: index.html.
3. Create the Menu page: menu.html.

Hands-On Practice Case Study

Task 1: The Website Folder. Create a folder on your hard drive or portable storage device (thumb drive or SD card) called “javajam” to contain your JavaJam website files.

Task 2: The Home Page. You will use a text editor to create the Home page for the JavaJam Coffee House website. The Home page is shown in Figure 2.33.

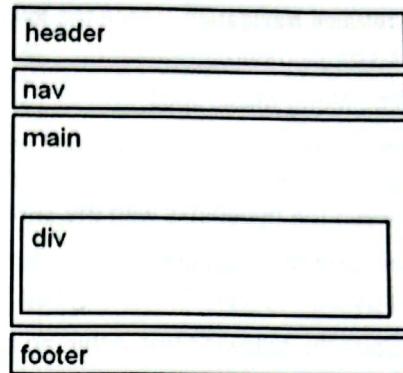


Figure 2.32 JavaJam wireframe

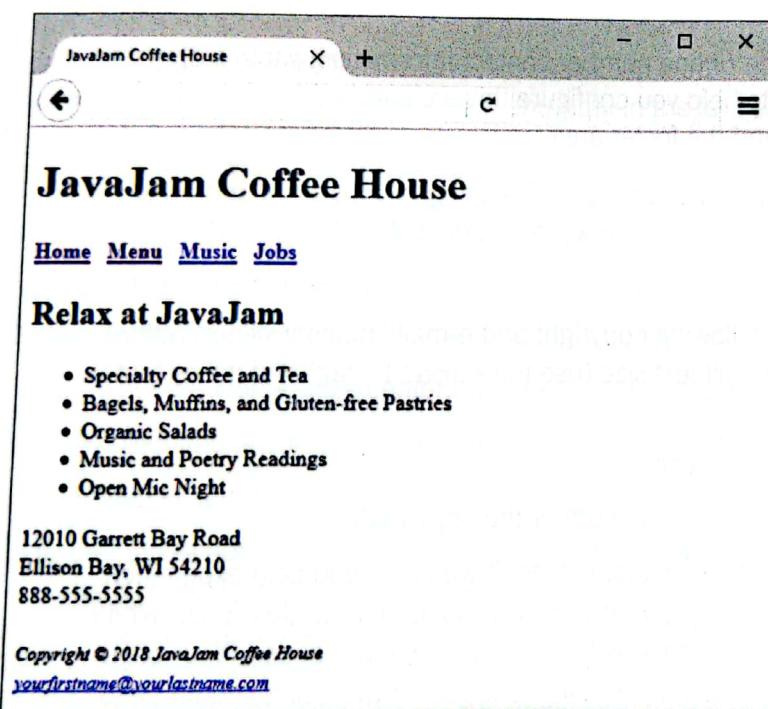


Figure 2.33 JavaJam index.html

Launch a text editor, and create a web page with the following specifications:

1. **Web Page Title.** Use a descriptive page title. The company name is a good choice for the home page of a business website. On pages other than the home page, the title typically contains both the company name and a word or a phrase that describes the purpose of the page.
2. **Wireframe Header.** Code the header element with the text, “JavaJam Coffee House” contained within a heading 1 element.

3. Wireframe Navigation. Place the following text within a nav element with bold text (use the **** element):

Home Menu Music Jobs

Code anchor tags so that "Home" links to index.html, "Menu" links to menu.html, "Music" links to music.html, and "Jobs" links to jobs.html. Add extra blank spaces between the hyperlinks with the special character as needed.

4. Wireframe Main Content. Code the main page content within a main element. Use Hands-On Practice 2.10 as a guide.

Hands-On Practice 2.10 as a guide:

a. Code the following text within an h2 element:

Relax at JavaJam

b. Configure the following content in an unordered list:

Specialty Coffee and Tea

Bagels, Muffins, and Gluten-free Pastries

Organic Salads

Music and Poetry Readings

Open Mic Night

c. Code the following address and phone number contact information within a div element. Use line break tags to help you configure this area and add extra space between the phone number and the footer area.

12010 Garrett Bay Road

Ellison Bay, WI 54210

888-555-5555

5. Wireframe Footer. Configure the following copyright and e-mail link information within a footer element. Format it with small text size (use the <small> tag) and italics font style (use the *<i>* tag).

Copyright © 2018 JavaJam Coffee House

Place your name in an e-mail link on the line under the copyright.

The page in Figure 2.33 may seem a little sparse, but don't worry; as you gain experience and learn to use more advanced techniques, your pages will look more professional. White space (blank space) on the page can be added with **
** tags where needed. Your page does not need to look exactly the same as the sample. Your goal at this point should be to practice and get comfortable using HTML. Save your page in the javajam folder, and name it index.html.

Task 3: The Menu Page. Create the Menu page shown in Figure 2.34. A technique that improves productivity is to create new pages based on existing pages so that you can benefit from your previous work. Your new Menu page will use the index.html page as a starting point.

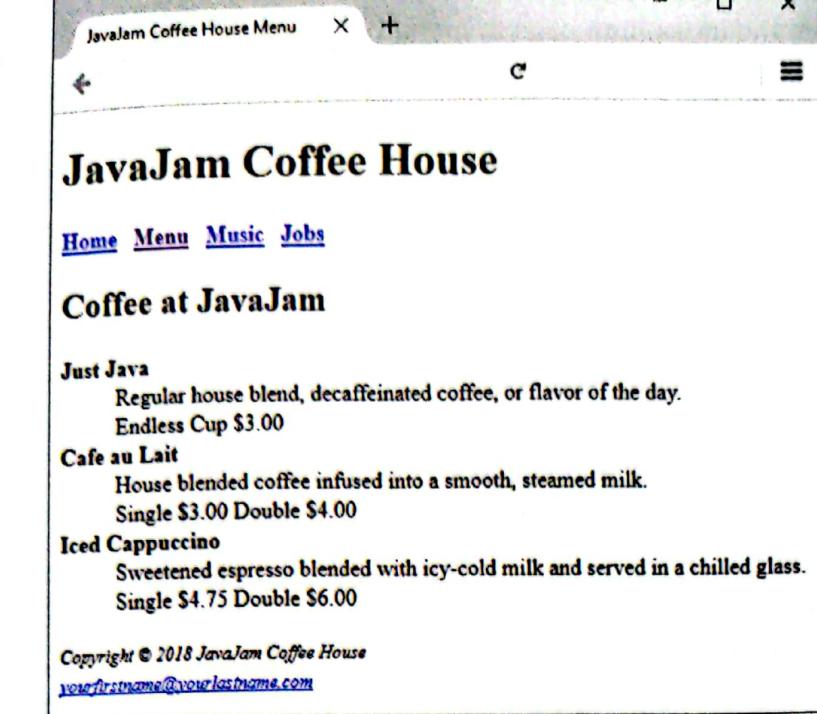


Figure 2.34 JavaJam menu.html

Open the index.html page for the JavaJam website in a text editor. Select File > Save As, and save the file with the new name of menu.html in the javajam folder. Now you are ready to edit the page.

1. **Web Page Title.** Modify the page title. Change the text contained between the `<title>` and `</title>` tags to the following:

JavaJam Coffee House Menu

2. **Wireframe Main Content.**

a. Delete the Home page content unordered list and contact information.

b. Replace the text in the h2 element with the following:

Coffee at JavaJam

c. Use a description list to add the menu content to the page. Use the `<dt>` tag to contain each menu item name. Configure the menu item name to have strong importance and display in bold font weight with the `` tag. Use the `<dd>` tag to contain the menu item description. Configure line break tags as needed to display two lines of information within each dd element. The menu item names and descriptions are as follows:

Just Java

Regular house blend, decaffeinated coffee, or flavor of the day.

Endless Cup \$3.00

Cafe au Lait

House blended coffee infused into a smooth, steamed milk.

Single \$2.00 Double \$4.00

Iced Cappuccino

Sweetened espresso blended with icy-cold milk and served in a chilled glass.

Single \$4.75 Double \$6.00

Chapter 2 HTML Basics

Save your page, and test it in a browser. Test the hyperlink from the menu.html page to index.html. Test the hyperlink from the index.html page to menu.html. If your links do not work, review your work, paying close attention to these details:

- Verify that you have saved the pages with the correct names in the correct folder.
- Verify your spelling of the page names in the anchor elements.

Test again after you make changes.