2

HTML Basics

Chapter Objectives

In this chapter, you will learn how to . . .

- Describe HTML, XHTML, and HTML5
- Identify the markup language in a web page document
- Use the html, head, body, title, and meta elements to code a template for a web page
- Configure the body of a web page with headings, paragraphs, line breaks, divs, lists, and blockquotes
- Configure text with phrase elements
- Configure special characters
- Use the new HTML5 header, nav, main, and footer elements
- Use the anchor element to link from page to page
- Create absolute, relative, and e-mail hyperlinks
- Code, save, and display a web page document
- Test a web page document for valid syntax

This chapter gets you started on your very first web page. You'll be introduced to Hypertext Markup Language (HTML), the language used to create web pages; eXtensible Hypertext Markup Language (XHTML), the most recent standardized version of HTML; and HTML5, the newest version of HTML. The chapter begins with an introduction to the syntax of HTML5; continues with sample web pages; and introduces HTML structural, phrase, and hyperlink elements as more example web pages are created. You will learn more if you work along with the sample pages in the text. Coding HTML is a skill, and every skill improves with practice.

2.1 HTML Overview

Markup languages consist of sets of directions that tell the browser software (and other user agents such as mobile phones) how to display and manage a web document. These directions are usually called tags and perform functions such as displaying graphics, formatting text, and referencing hyperlinks.

The World Wide Web is composed of files containing **Hypertext Markup Language (HTML)** and other markup languages that describe web pages. Tim Berners-Lee developed HTML using Standard Generalized Markup Language (SGML). SGML prescribes a standard format for embedding descriptive markup within a document and for describing the structure of a document. SGML is not in itself a document language, but rather a description of how to specify one and create a document type definition (DTD). The W3C (http://www.w3c.org) sets the standards for HTML and its related languages. Like the Web itself, HTML is in a constant state of change.

HTML

HTML is the set of markup symbols or codes placed in a file that is intended for display on a web page. These markup symbols and codes identify structural elements such as paragraphs, headings, and lists. HTML can also be used to place media (such as graphics, video, and audio) on a web page and describe fill-in forms. The browser interprets the markup code and renders the page. HTML permits the platform-independent display of information across a network. No matter what type of computer a web page was created on, any browser running on any operating system can display the page.

Each individual markup code is referred to as an **element** or **tag**. Each tag has a purpose. Tags are enclosed in angle brackets, the < and > symbols. Most tags come in pairs: an opening tag and a closing tag. These tags act as containers and are sometimes referred to as container tags. For example, the text that is between the `<title>` and `</title>` tags on a web page would display in the title bar on the browser window. Some tags are used alone and are not part of a pair. For example, a `<br>` tag that configures a line break on a web page is a stand-alone, or self-contained, tag and does not have a closing tag. Most tags can be modified with **attributes** that further describe their purpose.

XML

XML (eXtensible Markup Language) was developed by the W3C to create common information formats and share the format and the information on the Web. It is a text-based syntax designed to describe, deliver, and exchange structured information, such as RSS feeds. XML is not intended to replace HTML, but to extend the power of HTML by separating data from presentation. Using XML, developers can create any tags they need to describe their information.

XHTML

The most recent standardized version of HTML used today is **eXtensible HyperText Markup Language (XHTML)**. XHTML uses the tags and attributes of HTML4 along with the syntax of XML. XHTML has been used on the Web for over a decade and you'll find many web pages coded with this markup language. At one point the W3C was working on a new version of XHTML, called XHTML 2.0. However, the W3C stopped development of XHTML 2.0 because it was not backward compatible with HTML4. Instead, the W3C decided to move forward with HTML5.

HTML5

HTML5 is intended to be the successor to HTML4 and will replace XHTML. HTML5 incorporates features of both HTML and XHTML, adds new elements of its own, provides new features such as form edits and native video, and is intended to be backward compatible.

The W3C approved HTML5 for Candidate Recommendation status in late 2012. HTML5 reached final Recommendation status in late 2014. The W3C continues its development of HTML and is adding more new elements, attributes, and features in HTML 5.1, which is currently in draft form.

Recent versions of popular browsers, such as Internet Explorer, Microsoft Edge, Firefox, Safari, Google Chrome, and Opera, already support most features of HTML5. When new versions of each browser are released, you can expect increased support of HTML5. You'll learn to use HTML5 syntax as you work through this textbook. W3C HTML5 documentation is available at http://www.w3.org/TR/html5/.

FAQ What software do I need?

No special software is needed to create a web page document; all you need is a text editor. The Notepad text editor is included with Microsoft Windows. TextEdit is distributed with the Mac OS X operating system. (See http://support.apple.com/kb/TA20406 for configuration information.) An alternative to the operating system's basic text editor is one of the many free or shareware editors that are available, such as Notepad++ for Windows (http://notepad-plus-plus.org/download) and TextWrangler for Macs (http://www.barebones.com/products/textwrangler/download.html). Another commonly used alternative is a commercial web-authoring tool, such as Adobe Dreamweaver. Regardless of the software or program you use, having a solid foundation in HTML will be useful.

You will need to test your web pages in the most popular browsers, which are listed as follows, along with the URLs where you can download them for free:

- Internet Explorer (http://windows.microsoft.com/en-US/internet-explorer/download-ie)
- Mozilla Firefox (http://www.mozilla.com/en-US/products/download.html)
- Apple Safari (http://www.apple.com/safari/download/)
- Google Chrome (http://www.google.com/chrome)

You will also find the Web Developer Extension for Firefox (https://addons.mozilla.org/en-us/firefox/addon/web-developer) to be useful.

2.2 Document Type Definition

Because multiple versions and types of HTML and XHTML exist, the W3C recommends identifying the type of markup language used in a web page document with a **Document Type Definition (DTD)**. The DTD identifies the version of HTML contained in your document. Browsers and HTML code validators can use the information in the DTD when processing the web page. The DTD statement, commonly called a **doctype** statement, is the first line of a web page document. The DTD for HTML5 is:

```
<!DOCTYPE html>
```

2.3 Web Page Template

You already know that the HTML markup language tells browsers how to display information on a web page. Let's take a closer look at what's "under the hood" of every web page you create. Every single web page you create will include the DTD and the html, head, title, meta, and body elements. We will follow the coding style to use lowercase letters and place quotes around attribute values. A basic HTML5 web page template (found in the student files at chapter2/template.html) is as follows:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Page Title Goes Here</title>
<meta charset="utf-8">
</head>
<body>
... body text and more HTML tags go here ...
</body>
</html>
```

With the exception of the specific page title, the first seven lines will usually be the same on every web page that you create. Review the code above and notice that the document type definition statement has its own formatting and that the HTML tags all use lowercase letters. Next, let's explore the purpose of the html, head, title, meta, and body elements.

2.4 HTML Element

The purpose of the html element is to indicate that the document is HTML formatted. The html element tells the browser how to interpret the document. The opening **<html>** tag is placed on a line below the DTD. The closing `</html>` tag indicates the end of the web page and is placed after all other HTML elements in the document.

The html element also needs to indicate the spoken language, such as English, of the text in the document. This additional information is added to the `<html>` tag in the form of an attribute, which modifies or further describes the function of an element. The **lang attribute** specifies the spoken language of the document. For example, `lang="en"` indicates the English language. Search engines and screen readers may access this attribute.

2.5 Head, Title, Meta, and Body Elements

There are two sections on a web page: the head and the body. The **head section** contains information that describes the web page document. The **body section** contains the actual tags, text, images, and other objects that are displayed by the browser as a web page.

The Head Section

Elements that are located in the head section include the title of the web page, meta tags that describe the document (such as the character encoding used and information that may be accessed by search engines), and references to scripts and styles. Many of these features do not show directly on the web page.

The **head element** contains the head section, which begins with the **<head>** tag and ends with the **</head>** tag. You will always code at least two other elements in the head section: a title element and a meta element.

The first element in the head section, the **title element**, configures the text that will appear in the title bar of the browser window. The text between the **<title>** and **</title>** tags is called the title of the web page and is accessed when web pages are bookmarked and printed. Popular search engines, such as Google, use the title text to help determine keyword relevance and even display the title text on the results page of a search. A descriptive title that includes the website or organization name is a crucial component for establishing a brand or presence on the Web.

The **meta element** describes a characteristic of a web page, such as the character encoding. **Character encoding** is the internal representation of letters, numbers, and symbols in a file such as a web page or other file that is stored on a computer and may be transmitted over the Internet. There are many different character-encoding sets. However, it is common practice to use a character-encoding set that is widely supported, such as utf-8, which is a form of Unicode (http://www.unicode.org). The meta tag is not used as a pair of opening and closing tags. It is considered to be a stand-alone, or **self-contained**, tag (referred to as a **void element** in HTML5). The meta tag uses the **charset attribute** to indicate the character encoding. An example meta tag is as follows:

```
<meta charset="utf-8">
```

The Body Section

The body section contains text and elements that display directly on the web page in the browser window, also referred to as the browser viewport. The purpose of the body section is to configure the contents of the web page.

The **body element** contains the body section, which begins with the **<body>** tag and ends with the **</body>** tag. You will spend most of your time writing code in the body of a web page. Text and elements typed between the opening and closing body tags will display on the web page in the browser viewport.

2.6 Your First Web Page

Hands-On Practice 2.1

VideoNote
Your First
Web Page

Now that you're familiar with basic elements used on every web page, it's your turn to create your first web page, shown in Figure 2.1.

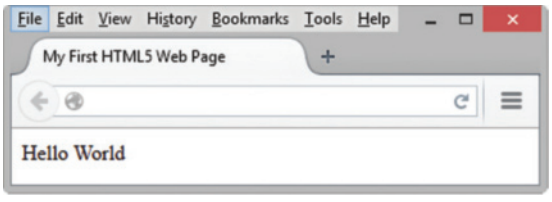


Figure 2.1 Your first webpage

Create a Folder

You'll find it helpful to create folders to organize your files as you develop the web pages in this book and create your own websites. Use your operating system to create a new folder named mychapter2 on your hard drive or a portable flash drive.

To create a new folder on a Mac:

1. Launch Finder, and select the location where you would like to create the new folder.
2. Choose File > New Folder to create an untitled folder.
3. To rename the folder, select the folder and click on the current name. Type a name for the folder, and press the Return key.

To create a new folder with Windows (version 7 and below):

1. Launch Windows Explorer (either press the Windows key or select Start > All Programs > Accessories > Windows Explorer).
2. Navigate to the location where you would like to create the new folder, such as My Documents or your C: drive.
3. Select New Folder.
4. To rename the New Folder, right-click on it, select Rename from the context-sensitive menu, type in the new name, and press the Enter key.

To create a new folder with Windows (version 8 and above):

1. Launch File Explorer (formerly called Windows Explorer):
   a. Display the Desktop.
   b. Right-click on the Start button and select File Explorer.
2. Navigate to the location where you would like to create the new folder, such as Documents, your C: drive, or an external USB drive.
3. Select the Home tab. Select New Folder.
4. To rename the New Folder, right-click on it, select Rename from the context-sensitive menu, type in the new name, and press the Enter key.

FAQ **Why should I create a folder, why not just use the desktop?**

Folders will help you to organize your work. If you just use the desktop, it would quickly become cluttered and disorganized. It's also important to know that web-sites are organized on web servers within folders. By starting to use folders right away to organize related web pages, you are on your way to becoming a suc-cessful web designer.

Your First Web Page

Now you are ready to create your first HTML5 web page. Launch Notepad or another text editor. Type in the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>My First HTML5 Web Page</title>
<meta charset="utf-8">
</head>
<body>
Hello World
</body>
</html>
```

Notice that the first line in the file contains the doctype. The HTML code begins with an opening `<html>` tag and ends with a closing `</html>` tag. The purpose of these tags is to indicate that the content between them makes up a web page. The head section is delimited by `<head>` and `</head>` tags and contains a pair of title tags with the words "My First HTML5 Web Page" in between, along with a **`<meta>`** tag to indicate the character encoding.

The body section is delimited by `<body>` and `</body>` tags. The words "Hello World" are typed on a line between the body tags. See Figure 2.2 for a screenshot of the code as it would appear in Notepad. You have just created the source code for a web page document.

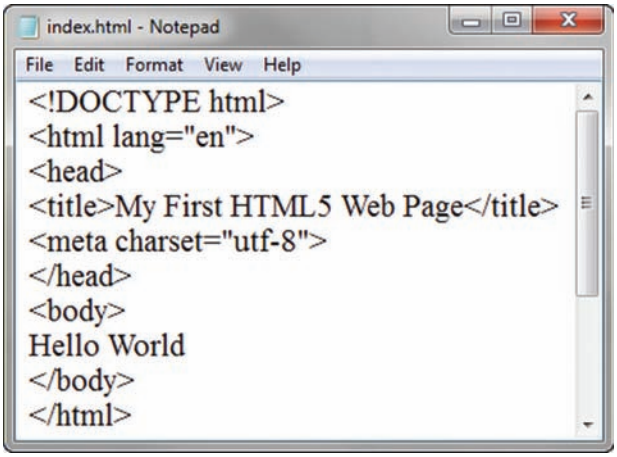


Figure 2.2 Code displayed in Notepad. Screenshot from Microsoft® Notepad®. Used by permission of Microsoft Corporation

FAQ **Do I have to start each tag on its own line?**

No, you are not required to start each tag on a separate line. A browser can display a page even if all the tags follow each other on one line with no spaces. Humans, however, find it easier to write and read web page code if line breaks and indentation are used.

Save Your File

You will save your file with the name of index.html. A common file name for the home page of a website is index.html or index.htm. Web pages use either a .htm or a .html file extension. The web pages in this book use the .html file extension. Display your file in Notepad or another text editor. Select File from the menu bar, and then select Save As. The Save As dialog box will appear. Navigate to your mychapter2 folder. Using Figure 2.3 as an example, type the file name. Click the Save button after you type the file name. Sample solutions for the exercises are available in the student files. If you like, you can compare your work with the solution in the student files at chapter2/index.html before you test your page.

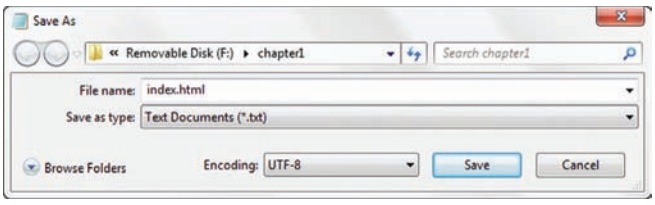


Figure 2.3 The Save As dialog box. Screenshot from Microsoft® Notepad®. Used by permission of Microsoft Corporation

FAQ **Why does my file have a .txt file extension?**

In some older versions of Windows, Notepad will automatically append a .txt file extension. If this happens, type the name of the file within quotation marks ("index.html"), and save your file again.

Test Your Page

There are two ways to test your page:

1. Launch Windows Explorer (Windows 7 or earlier), File Explorer (Windows 8 or later), or Finder (Mac). Navigate to your index.html file. Double-click index.html. The default browser will launch and will display your index.html page. If your default browser is Internet Explorer, your page should look similar to the one shown in Figure 2.4. A display of the page using Firefox is shown in Figure 2.1.

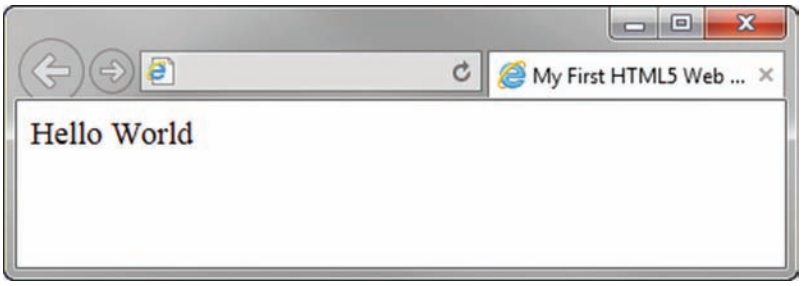


Figure 2.4 Web page displayed by Internet Explorer. Screenshots of Internet Explorer. Copyright by Microsoft Corporation. Used by permission of Microsoft Corporation

2. Launch a browser. (If you are using Internet Explorer 9 or later, right-click in the area at the top of the browser window and select the Menu bar.) Select File > Open > Browse. Navigate to your index.html file. Double-click index.html, and click OK. If you used Internet Explorer, your page should look similar to the one shown in Figure 2.4. A display of the page using Firefox is shown in Figure 2.1.

   Examine your page. Look carefully at the browser window. Notice how the browser title bar or browser tab displays the title text, "My First HTML5 Web Page." Some search engines use the text enclosed within the `<title>` and `</title>` tags to help determine the relevancy of keyword searches, so make certain that your pages contain descriptive titles. The title element is also used when viewers bookmark your page or add it to their Favorites. An engaging and descriptive page title may entice a visitor to revisit your page. If your web page is for a company or an organization, it's a best practice to include the name of the company or organization in the title.

FAQ When I viewed my page in a browser, the file name was index.html.html–why did this happen?

This usually happens when your operating system is configured to hide file extension names. You will correct the file name, using one of the following two methods:

- Use the operating system to rename the file from "index.html.html" to "index.html".

OR

- Open the index.html.html file in your text editor and save it with the name "index.html".

It's a good idea to change the settings in your operating system to show file extension names. Access the system help for your operating system or the resources below for information about how to configure your operating system to show file extension names:

- *Windows:* http://windows.microsoft.com/en-us/windows/show-hide-file-name-extensions#show-hide-file-name-extensions=windows-7, http://www.pcadvisor.co.uk/how-to/software/3341794/how-show-or-hide-file-extensions/
- *Mac:* http://www.fileinfo.com/help/mac_show_extensions

Checkpoint 2.1

1. Describe the origin, purpose, and features of HTML.

2. Describe the software needed to create and test web pages.

3. Describe the purpose of the head and body sections of a web page.

2.7 Heading Element

Heading elements are organized into six levels: h1 through h6. The text contained within a heading element is rendered as a "block" of text by the browser (referred to as **block display**) and displays with empty space (sometimes called "white space") above and below. The size of the text is largest for **<h1>** (called the heading 1 tag) and smallest for **<h6>** (called the heading 6 tag). Depending on the font being used (more on font sizes in Chapter 3), the text contained within **<h4>**, **<h5>**, and **<h6>** tags may be displayed smaller than the default text size. All text contained within heading tags is displayed with bold font weight. Figure 2.5 shows a web page document with six levels of headings.

FAQ Why doesn't the heading tag go in the head section?

It's common for students to try to code the heading tags in the head section of the document, but doing this is not valid and will cause issues with the way the browser displays the web page. Even though "heading tag" and "head section" sound similar, always code heading tags in the body section of the web page document.

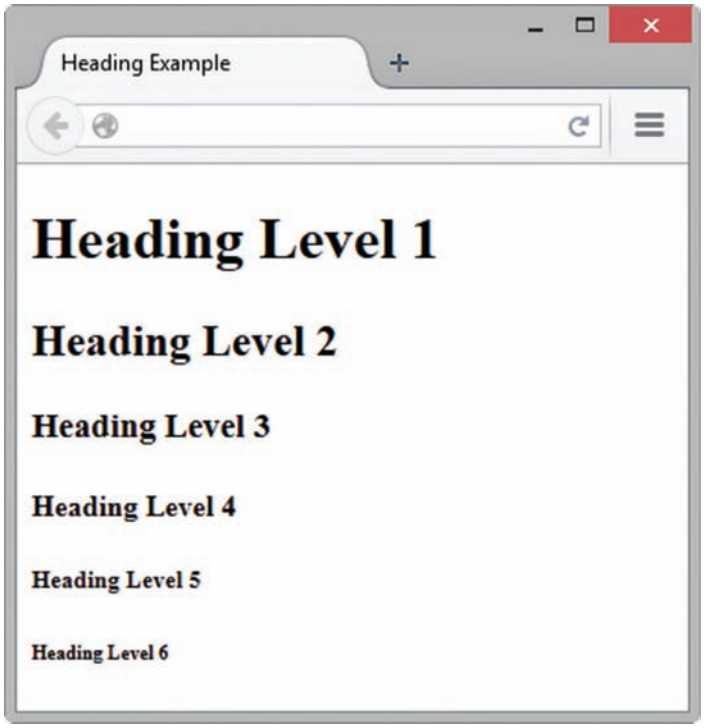


Figure 2.5 Sample headings

Hands-On Practice 2.2

To create the web page shown in Figure 2.5, 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 Internet Explorer or Firefox to test your page. It should look similar to the page shown in Figure 2.5. You can compare your work with the solution found in the student files (chapter2/heading.html).

Accessibility and Headings

Focus on Accessibility

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. To indicate areas within a page hierarchically, code heading tags numerically as appropriate (h1, h2, h3, and so on), and include page content in block display elements such as paragraphs and lists. In Figure 2.6, the `<h1>` tag contains the name of the website in the logo header area at the top of the web page, the `<h2>` tag contains the topic or name of the page in the content area, and other heading elements are coded in the content area as needed to identify major topics and subtopics.

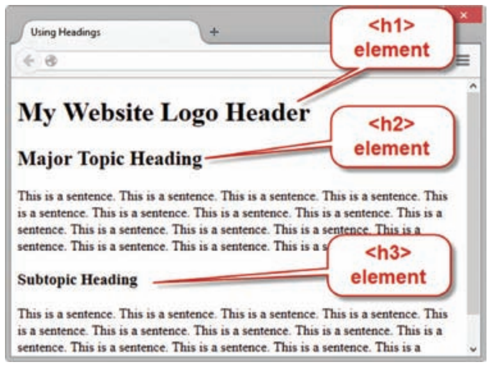


Figure 2.6 Heading tags outline the page

Visually challenged visitors who are using a screen reader can direct the software to display a list of the headings used on a page to focus on the topics that interest them. Your well-organized page will be more usable for every visitor to your site, including those who are visually challenged.

2.8 Paragraph Element

Paragraph elements are used to group sentences and sections of text together. Text that is contained by `<p>` and `</p>` tags display as a "block" (referred to as block display) and will appear with empty space above and below it. Figure 2.7 shows a web page document with a paragraph after the first heading.

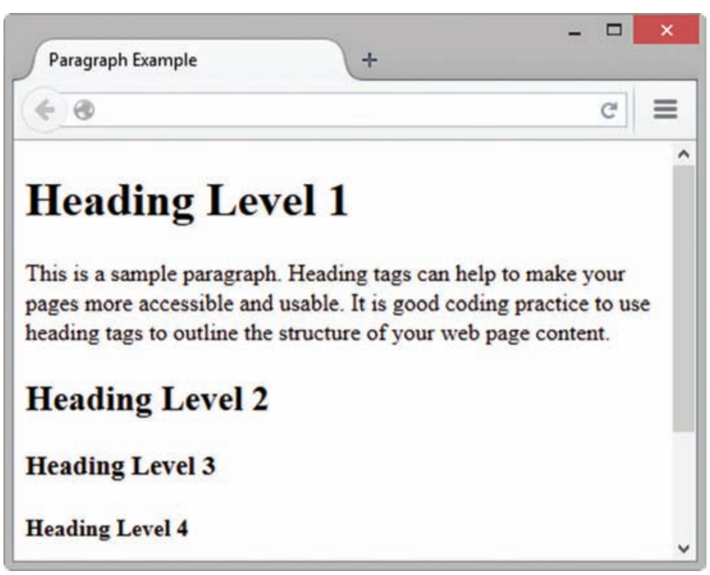


Figure 2.7 Web page using headings and a paragraph

Hands-On Practice 2.3

To create the web page shown in Figure 2.7, launch a text editor. Select File > Open to edit the file located at chapter2/heading.html in the student files. Modify the page title, and add a paragraph of text to your page below the line with the `<h1>` tags and above the line with the `<h2>` tags. Use the following code as an example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Paragraph Example</title>
<meta charset="utf-8">
</head>
<body>
<h1>Heading Level 1</h1>
<p>This is a sample paragraph. 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>
</html>
```

Save the document as paragraph.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 2.7. You can compare your work with the solution (chapter2/paragraph.html) found in the student files. Notice how the text in the paragraph wraps automatically as you resize your browser window.

Alignment

As you tested your web pages, you may have noticed that the headings and text begin near the left margin. This placement is called **left alignment** and is the default alignment for web pages. There are times, however, when you want a paragraph or heading to be centered or right aligned. The align attribute can be used for this purpose. The purpose of an attribute is to modify the properties of an HTML element. In this case, the **align** attribute modifies the element's horizontal alignment (left, center, or right) on a web page. To center an element on a web page, use the attribute `align="center"`. To right-align the text within an element, use the `align="right"` attribute. In XHTML syntax, the align attribute can be used with a number of block display elements, including the paragraph (`<p>`) and heading (`<h1>` through `<h6>`) tags. The align attribute is **obsolete** in HTML5, which means that while it may be used in XHTML, the attribute has been removed from the W3C HTML5 specification and is invalid. In Chapter 6, you will learn how to configure alignment using a more modern approach with Cascading Style Sheets (CSS).

2.9 Line Break Element

The **line break element** causes the browser to advance to the next line before displaying the next element or portion of text on a web page. The line break tag is not coded as a pair of opening and closing tags. It is a stand-alone, or void element, and is coded as **
**. Figure 2.8 shows a web page document with a line break after the first sentence in the paragraph.

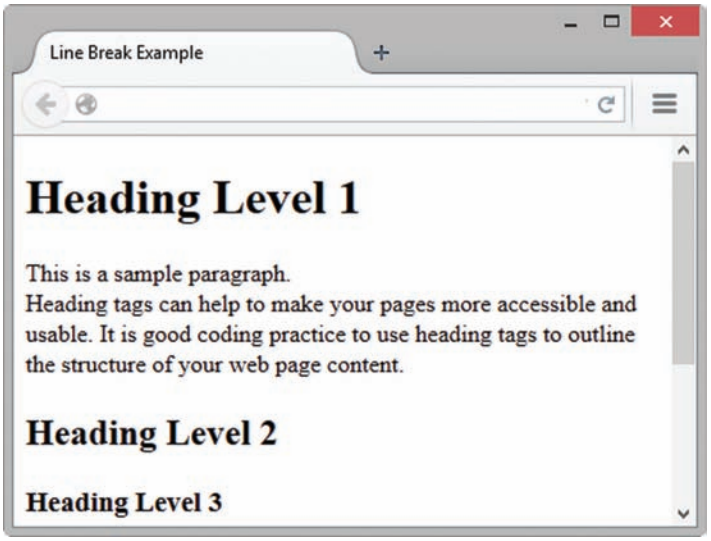


Figure 2.8
Notice the line break after the first sentence

Hands-On Practice 2.4

To create the web page shown in Figure 2.8, 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 `<br>` 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 2.8. You can compare your work with the solution found in the student files (chapter2/linebreak.html).

FAQ Why does my web page still look the same?

Often, students make changes to a web page document, but get frustrated because their browser shows an older version of the page. The following troubleshooting tips are helpful when you know you modified your web page, but the changes do not show up in the browser:

1. Make sure you save your page after you make the changes.
2. Verify the location that you are saving your page to—a specific folder on the hard drive or removable storage.
3. Verify the location that your browser is requesting the page from—a specific folder on the hard drive or removable storage.
4. Be sure to click the Refresh or Reload button in your browser.

Figure 2.9 The text within the blockquote element is indented

2.10 Blockquote Element

In addition to organizing text in paragraphs and headings, sometimes you need to add a quotation to a web page. The **blockquote element** is used to display a block of quoted text in a special way—indented from both the left and right margins. A block of indented text begins with a **<blockquote>** tag and ends with a **</blockquote>** tag. Figure 2.9 shows a web page document with a heading, a paragraph, and a blockquote.

Hands-On Practice 2.5

To create the web page shown in Figure 2.9, 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>
```

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 2.9. You can compare your work with the solution (chapter2/blockquote.html) found in the student files.

You have probably noticed how convenient the `<blockquote>` tag could be if you need to indent an area of text on a web page. You may have wondered whether it would be OK to use the blockquote element anytime you would like to indent text or whether the blockquote element is reserved only for long quotations. The semantically correct use of the blockquote element is only for displaying large blocks of quoted text within a web page. Why should you be concerned about semantics? Consider the future of the Semantic Web, described in *Scientific American* (http://www.scientificamerican.com/article.cfm?id=the-semantic-web) as "A new form of Web content that is meaningful to computers [that] will unleash a revolution of new possibilities." Using HTML in a semantic, structural manner is one step toward the Semantic Web. So, avoid using a blockquote element just to indent text. You will learn modern techniques to configure margins and padding on elements later in this book.

2.11 Phrase Elements

Phrase elements indicate the context and meaning of the text between the container tags. It is up to each browser to interpret that style. Phrase elements are displayed right in line with the text (referred to as **inline display**) and can apply to a section of text or even just a single character of text. For example, the **`<strong>`** element indicates that the text associated with it has strong importance and should be displayed in a "strong" manner in relation to normal text on the page. Table 2.1 lists common phrase elements and examples of their use. Notice that some tags, such as `<cite>` and `<dfn>`, result in the same type of display (italics) as the `<em>` tag in popular browsers. These tags semantically describe the text as a citation or definition, but the physical display is usually italics in both cases.

Table 2.1 Phrase elements

Element	Example	Usage
`<abbr>`	WIPO	Identifies text as an abbreviation; configure the title attribute with the full name
`<b>`	**bold** text	Text that has no extra importance, but is styled in bold font by usage and convention
`<cite>`	*cite* text	Identifies a citation or reference; usually displayed in italics
`<code>`	`code` text	Identifies program code samples; usually a fixed-space font
`<dfn>`	*dfn* text	Identifies a definition of a word or term; usually displayed in italics
`<em>`	*emphasized* text	Causes text to be emphasized in relation to other text; usually displayed in italics
`<i>`	*italicized* text	Text that has no extra importance, but is styled in italics by usage and convention
`<kbd>`	`kbd` text	Identifies user text to be typed; usually a fixed-space font
`<mark>`	mark text	Text that is highlighted in order to be easily referenced
`<samp>`	`samp` text	Shows program sample output; usually a fixed-space font
`<small>`	small text	Legal disclaimers and notices ("fine print") displayed in small font size
`<strong>`	**strong** text	Strong importance; causes text to stand out from surrounding text; usually displayed in bold
`<sub>`	sub text	Displays a subscript as small text below the baseline
`<sup>`	sup text	Displays a superscript as small text above the baseline
`<var>`	*var* text	Identifies and displays a variable or program output; usually displayed in italics

Each phrase element is a container element, so an opening and a closing tag must be used. As shown in Table 2.1, the element indicates that the text associated with it has "strong" importance. Usually, the browser (or other user agent) will display text in bold font type. A screen reader, such as JAWS or Window-Eyes, might interpret text to indicate that the text should be more strongly spoken. In the following line, the phone number is displayed with strong importance:

```
Call for a free quote for your web development needs:  888.555.5555
```

The corresponding code is

```
<p>Call for a free quote for your web development needs:
<strong>888.555.5555</strong></p>
```

Notice that the opening and closing tags are contained within the paragraph tags (and). This code is properly nested and is considered to be **well formed**. When improperly nested, the and tag pairs overlap each other instead of being nested within each other. Improperly nested code will not pass validation testing (see Section 2.18, "HTML Validation") and may cause display issues.

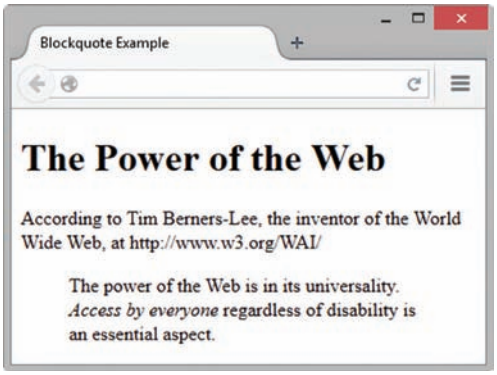


Figure 2.10 The tag in action

Figure 2.10 shows a web page document (also found in the student files at chapter2/em.html) that uses the tag to display the emphasized phrase "Access by everyone" in italics.

The code snippet is

```
<blockquote>
The power of the Web is in its universality.
<em>Access by everyone</em>
regardless of disability is an essential aspect.
</blockquote>
```

2.12 Ordered List

Lists are used on web pages to organize information. When writing for the Web, headings, short paragraphs, and lists can make your page more clear and easy to read. HTML can be used to create three types of lists—description lists, ordered lists, and unordered lists. All lists are rendered as block display with an empty space above and below. This section focuses on the **ordered list**, which displays a numbering or lettering system to itemize the information contained in the list. Ordered lists can be organized by the use of numerals (the default), uppercase letters, lowercase letters, uppercase Roman numerals, and lowercase Roman numerals. See Figure 2.11 for a sample ordered list.

Figure 2.11 Sample ordered list

Ordered lists begin with an **
** tag and end with an **
** tag. Each list item begins with an **
** tag and ends with an **
** tag. The code to configure the heading and ordered list shown in Figure 2.11 follows:

```
<h1>My Favorite Colors</h1>
<ol>
      <li>Blue</li>
      <li>Teal</li>
      <li>Red</li>
</ol>
```

The Type, Start, and Reversed Attributes

The **type attribute** configures the symbol used for ordering the list. For example, to create an ordered list organized by uppercase letters, use `<ol type="A">`. Table 2.2 documents the type attribute and its values for ordered lists.

Table 2.2 The type attribute for ordered lists

Value	Symbol
1	Numerals (the default)
A	Uppercase letters
a	Lowercase letters
I	Roman numerals
i	Lowercase Roman numerals

The **start attribute** is useful when you need a list to begin with an integer value other than 1 (for example, `start="10"`). Use the new HTML5 **reversed attribute** (set `reversed="reversed"`) to configure the list markers to display in descending order.

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 2.12, 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>
```

```
<body>
<h1>My Favorite Colors</h1>
<ol>
    <li>Blue</li>
    <li>Teal</li>
    <li>Red</li>
</ol>
</body>
</html>
```

Save your file as ol.html. Launch a browser and test your page. It should look similar to the page shown in Figure 2.12. You can compare your work with the solution in the student files (chapter2/ol.html).

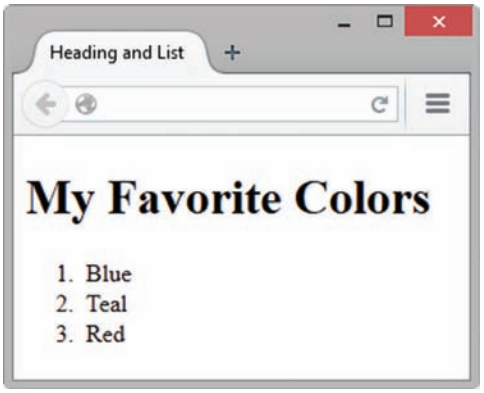


Figure 2.12 An ordered list

Take a few minutes to experiment with the type attribute. Configure the ordered list to use uppercase letters instead of numerals. Save your file as ola.html. Test your page in a browser. You can compare your work with the solution in the student files (chapter2/ola.html).

2.13 Unordered List

An **unordered list** displays a bullet, or list marker, before each entry in the list. This bullet can be one of several types: disc (the default), square, and circle. See Figure 2.13 for a sample unordered list.

Unordered lists begin with a **
** tag and end with a **
** tag. Each list item begins with an **
** tag and ends with an **
** tag. The code to configure the heading and unordered list shown in Figure 2.13 is

Figure 2.13 Sample unordered list

```
<h1>My Favorite Colors</h1>
<ul>
    <li>Blue</li>
    <li>Teal</li>
    <li>Red</li>
</ul>
```

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 2.14, 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:

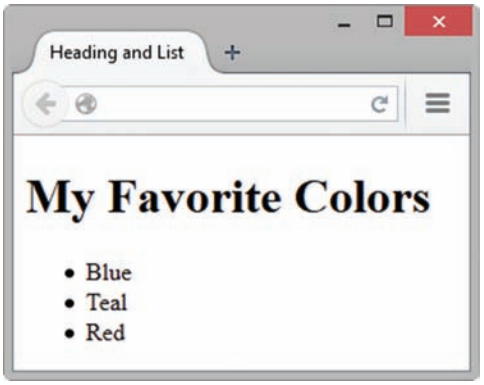


Figure 2.14 An unordered list

```
<!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>
```

Save your file as ul.html. Launch a browser and test your page. It should look similar to the page shown in Figure 2.14. You can compare your work with the solution in the student files (chapter2/ul.html).

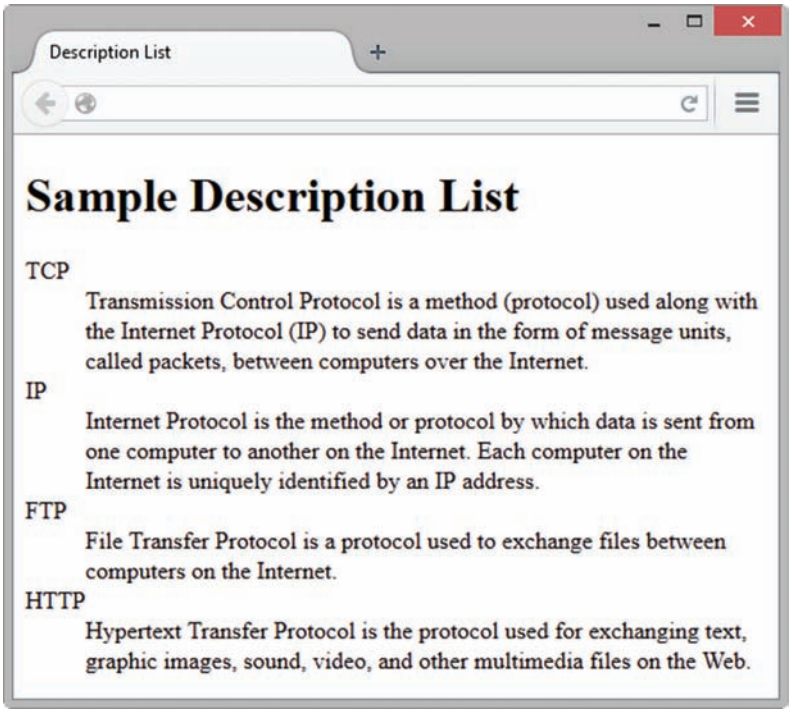
FAQ Can I change the "bullet" in an unordered list?

Back in the day before HTML5, the type attribute could be included with a <ul> tag to change the default list marker to a square (`type="square"`) or open circle (`type="circle"`). However, be aware that using the type attribute on an unordered list is considered obsolete in HTML5 because it is decorative and does not convey meaning. No worries, though—you'll learn techniques to configure list markers (bullets) in Chapter 6 to display images and shapes.

2.14 Description List

HTML5 introduces a new element name, **description list**, to replace the definition list element (used in XHTML and earlier versions of HTML). A description list is useful for organizing terms and their descriptions. The terms stand out, and their descriptions can be as long as needed to convey your message. Each term begins on its own line at the margin. Each description begins on its own line and is indented. Description lists are also handy for organizing Frequently Asked Questions (FAQs) and their answers. The questions and answers are offset with indentation. Any type of information that consists of a number of corresponding terms and longer descriptions is well suited to being organized in a description list. See Figure 2.15 for an example of a web page that uses a description list.

Figure 2.15
A description list



Description lists begin with the **<dl>** tag and end with the **</dl>** tag. Each term or name in the list begins with the **<dt>** tag and ends with the **</dt>** tag. Each description begins with the **<dd>** tag and ends with the **</dd>** tag.

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 2.15, 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, 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>
```

```
<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 2.15. 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).

FAQ Why is the HTML code in the Hands-On Practice examples indented?

Actually, it doesn't matter to the browser if web page code is indented, but humans find it easier to read and maintain code when it is logically indented. Review the description list created in Hands-On Practice 2.8. Notice how the `<dt>` and `<dd>` tags were indented. This makes it easier for you or another web developer to understand the source code in the future. There is no rule as to how many spaces to indent, although your instructor or the organization you work for may have a standard. Consistent indentation helps to create more easily maintainable web pages.

Checkpoint 2.2

1. Describe the features of a heading element and how it configures the text.

2. Describe the difference between ordered lists and unordered lists.

3. Describe the purpose of the blockquote element.

2.15 Special Characters

In order to use special symbols such as quotation marks, the greater-than sign (>), the less-than sign (<), and the copyright symbol (©) in your web page document, you need to use **special characters**, sometimes called **entity characters**. For example, if you wanted to include a copyright line on your page as follows:

```
© Copyright 2015 My Company. All rights reserved.
```

Use the special character code **`©`** to display the copyright symbol, as shown in the following code:

```
&copy; Copyright 2015 My Company. All rights reserved.
```

Another useful special character code is **` `**, which stands for nonbreaking space. You may have noticed that web browsers treat multiple spaces as a single space. If you need a small number of spaces in your text, you may use ` ` multiple times to indicate multiple blank spaces. This practice is acceptable if you simply need to tweak the position of an element a little. However, if you find that your web pages contain many ` ` special characters in a row, you should use a different method, such as configuring the padding or margin with Cascading Style Sheets (see Chapters 4 and 6). Table 2.3 and Appendix C, "Special Entity Characters," provide descriptions of special characters and their corresponding code.

Table 2.3 Common special characters

Character	Entity Name	Code
"	Quotation mark	"
'	Apostrophe	'
©	Copyright symbol	©
&	Ampersand	&
Empty space	Nonbreaking space	
—	Long dash	—
|	Vertical Bar		

Hands-On Practice 2.9

Figure 2.16 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.html 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 2.16 contains a heading, an unordered list, and copyright information. Configure the heading "Web Design Steps" as a level 1 heading (`<h1>`) as follows:

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

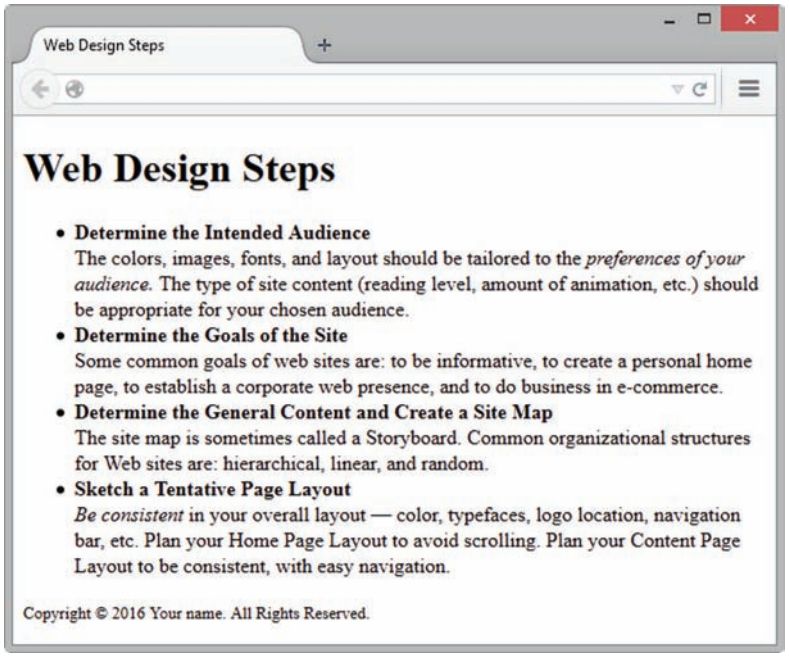


Figure 2.16 The design.html web page

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 `</ul>` 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 2.16.

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; 2016 Your name. All Rights Reserved.
</small></p>
```

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

2.16 Structural Elements

The Div Element

The **div element** has been used for many years to configure a generic structural area or "division" on a web page as a block display with empty space above and below. A div element begins with a ** ** tag and ends with a ** ** tag. Use a div element

when you need to format an area of a web page that may contain other block display elements such as headings, paragraphs, unordered lists, and even other div elements. You'll use Cascading Style Sheets (CSS) later in this book to style and configure the color, font, and layout of HTML elements.

HTML5 Structural Elements

In addition to the generic div element, HTML5 introduces a number of semantic structural elements that can be used to configure specific areas on a web page. These new HTML5 elements are intended to be used in conjunction with div and other elements to structure web page documents in a more meaningful manner that indicates the purpose of each structural area. You'll explore three of these new elements in this section: the header element, the nav element, and the footer element. Figure 2.17 shows a diagram of a page (called a wireframe) that indicates how the structure of a web page could be configured with the header, nav, main, div, and footer elements.

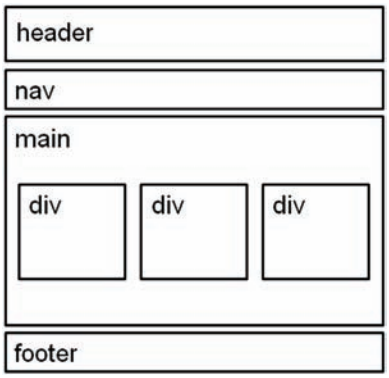


Figure 2.17 Structural elements

The Header Element

The purpose of the HTML5 **header element** is to contain the headings of either a web page document or an area within the document such as a section or article (more on the section element and article element in Chapter 6). The header element begins with the **<header>** tag and ends with the **</header>** tag. The header element is block display and typically contains one or more heading level elements (h1 through h6).

The Nav Element

The purpose of the HTML5 **nav element** is to contain a section of navigation links. The block display nav element begins with the **<nav>** tag and ends with the **</nav>** tag.

The Main Element

The purpose of the HTML5 **main element** is to contain the main content of a web page document. There should be only one main element per web page. The block display main element begins with the **<main>** tag and ends with the **</main>** tag.

The Footer Element

The purpose of the HTML5 **footer element** is to contain the footer content of a web page or section of a web page. The block display footer element begins with the **<footer>** tag and ends with the **</footer>** tag.

Hands-On Practice 2.10

In this Hands-On Practice you will practice using structural elements as you create the Trillium Media Design home page, shown in Figure 2.18. Launch a text editor, and open the template.html file from the chapter2 folder in the student files. Edit the code as follows:

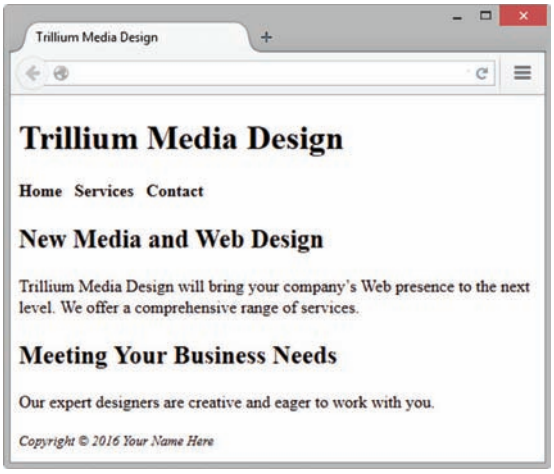


Figure 2.18 Trillium home page

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   Services   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 will bring your company&#39;s Web
presence to the next level. We offer a comprehensive range of
services.</p>
  <h2>Meeting Your Business Needs</h2>
  <p>Our expert designers are creative and eager to work with
you.</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; 2016 Your Name Here</i></small>
</footer>
```

Save your page as structure.html. Test your page in a browser. It should look similar to Figure 2.18. You can compare your work to the sample in the student files (chapter2/structure.html).

Practice with Structural Elements

Coding HTML is a skill and skills are best learned by practice. You'll get more practice coding a web page using structural elements in this section.

Hands-On Practice 2.11

In this Hands-On Practice you will use the wireframe shown in Figure 2.19 as a guide as you create the Casita Sedona Bed & Breakfast web page, shown in Figure 2.20.

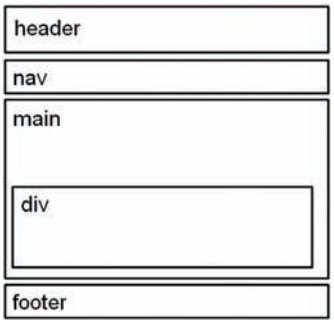


Figure 2.19 Wireframe for Casita Sedona.

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.

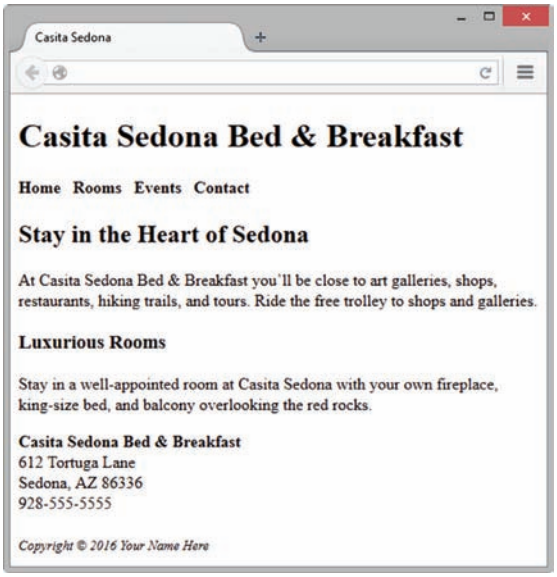


Figure 2.20 Casita Sedona web page.

```
<header>
  <h1>
  Casita Sedona Bed &amp;
    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  
    Rooms  
    Events  
    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 &amp; Breakfast you&#39;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 &amp; 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 &copy; 2016 Your Name Here</i></small>
</footer>
```

Save your page as casita.html. Test your page in a browser. It should look similar to Figure 2.20. 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 elements.

In Chapter 6, we'll explore coding techniques that will force older browsers to correctly display HTML5 structural tags. For now, be sure to use a current version of any popular browser to test your pages.

FAQ Are there other new structural elements in HTML5 that configure areas on web pages?

Yes, one of the characteristics of HTML5 is an emphasis on semantics. While the div element is useful, it is also quite generic. HTML5 offers a variety of special-purpose structural elements, including section, article, header, nav, main, aside, and footer. You'll explore the section, article, and aside elements in Chapter 7.

2.17 Anchor Element

Use the **anchor element** to specify a **hyperlink**, often referred to as a *link*, to another web page or file that you want to display. Each anchor element begins with an **<a>** tag and ends with a **** tag. The opening and closing anchor tags surround the text to click to perform the hyperlink. Use the **href attribute** to configure the hyperlink reference, which identifies the name and location of the file to access. Figure 2.21 shows a web page document with an anchor tag that configures a hyperlink to this book's website, http://webdevfoundations.net.

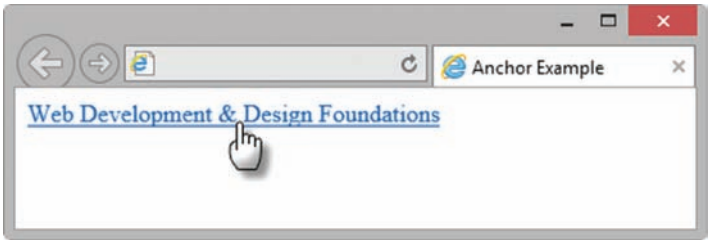


Figure 2.21 Sample hyperlink. Screenshots of Internet Explorer. Copyright by Microsoft Corporation. Used by permission of Microsoft Corporation

The code for the anchor tag in Figure 2.21 is as follows:

```
<a href="http://webdevfoundations.net">Web Development &amp; Design
  Foundations</a>
```

Notice that the href value is the URL for the website. The text that is typed between the two anchor tags displays on the web page as a hyperlink and is underlined by most browsers. When you move the mouse cursor over a hyperlink, the cursor changes to a pointing hand, as shown in Figure 2.21.

Hands-On Practice 2.12

To create the web page shown in Figure 2.21, 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 &amp; 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 2.21. You can compare your work with the solution found in the student files (chapter2/anchor.html).

FAQ Can images be hyperlinks?

Yes. Although we'll concentrate on text hyperlinks in this chapter, it's also possible to configure an image as a hyperlink. You'll get practice with image links in Chapter 4.

Absolute Hyperlinks

An **absolute hyperlink** indicates the absolute location of a resource on the Web. Use absolute hyperlinks when you need to link to resources on other websites. The href value for an absolute hyperlink to the home page of a website includes the `http://` protocol and the domain name. The following hyperlink is an absolute hyperlink to the home page of this book's website:

```
<a href="http://webdevfoundations.net">Web Development &amp; Design
Foundations</a>
```

Note that if we want to access a web page other than the home page on the book's website, we could also include a specific folder name and file name. For example, the following anchor tag configures an absolute hyperlink for a file named chapter1.html located in a folder named 8e on this book's website:

```
<a href="http://webdevfoundations.net/8e/chapter1.html">Web Development
&amp; Design Foundations Chapter 1</a>
```

Relative Hyperlinks

When you need to link to web pages within your site, use a **relative hyperlink**. The href value for a relative hyperlink does not begin with the http:// and does not include a domain name. For a relative hyperlink, the href value will contain only the file name or file name and folder of the web page you want to display. The hyperlink location is relative to the page currently being displayed. For example, if you were coding a home page (index.html) for the website whose site map is illustrated in Figure 2.22 and wanted to link to a page named contact.html located in the same folder as index.html, you would use the following code sample:

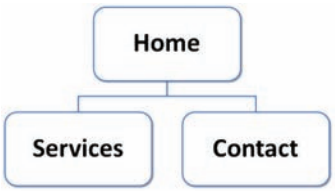


Figure 2.22 Site map

```
<a href="contact.html">Contact Us</a>
```

Site Map

A **site map** represents the structure, or organization, of pages in a website in a visual manner. Each page in the website is represented by a box on the site map. Figure 2.22 displays the site map for a website that contains a Home page and two content pages: a Services page and a Contact page. Review Figure 2.22 and notice that the Home page is at the top of the site map. The second level in a site map shows the other main pages of the website. In this very small three-page website, the other two pages (Services and Contact) are included on the second level. The main navigation of a website usually includes hyperlinks to the pages shown on the first two levels of the site map.

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 2.22: 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 2.18) from Hands-On Practice 2.10 as a starting point for your new home page (shown in Figure 2.23). 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.

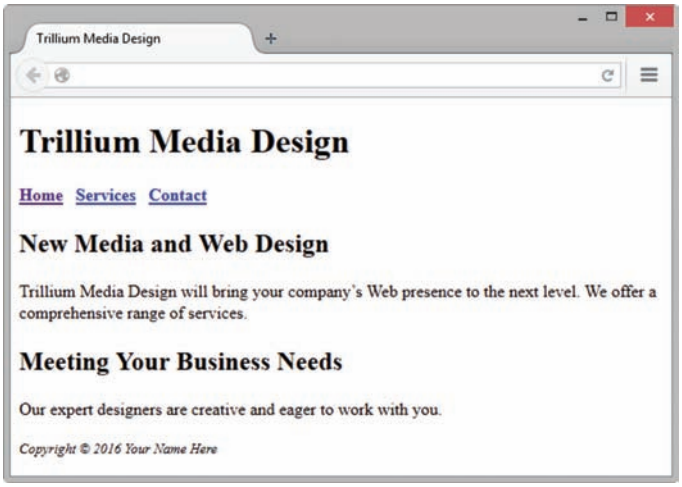


Figure 2.23 New index.html web page

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>   
         <a href="services.html">Services</a>   
         <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 2.23. 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 2.24.

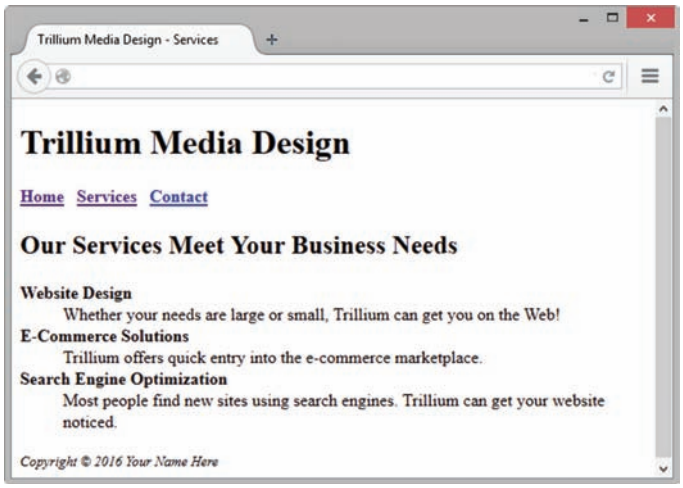


Figure 2.24 The services.html web page

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 2.24. 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 2.25. 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.

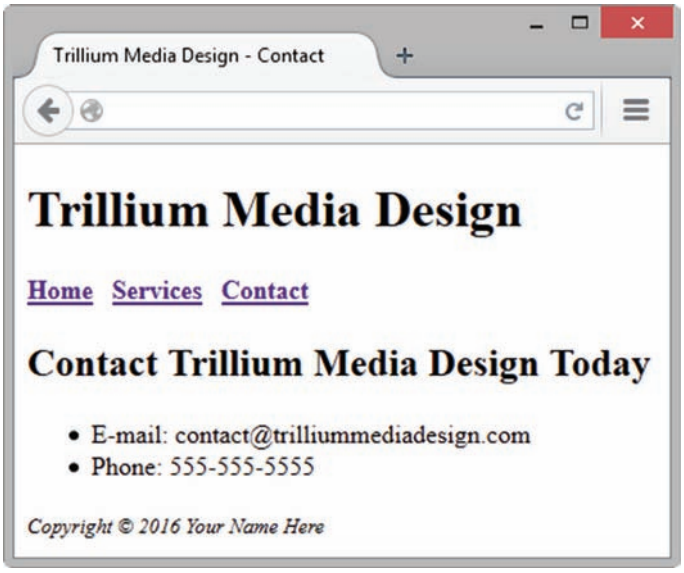


Figure 2.25 The contact.html web page.

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 2.25. 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 specified folder?
- Did you save the files with the names as requested? Use Windows Explorer, My Computer, or Finder (Mac users) to verify the actual names of the files you saved.
- Did you type the file names correctly in the anchor tag's href attribute? Check for typographical errors.
- When you place your mouse over a link, the file name of a relative link will display in the status bar in the lower edge of the browser window. Verify that this is the correct file name. On many operating systems, such as UNIX or Linux, the use of uppercase and lowercase letters in file names matters—make sure that the file name and the reference to it are in the same case. It's a good practice to always use lowercase for file names used on the Web.

E-Mail Hyperlinks

The anchor tag can also be used to create **e-mail hyperlinks**. An e-mail hyperlink will automatically launch the default mail program configured for the browser. It is similar to an external hyperlink with the following two exceptions:

- It uses `mailto:` instead of `http://`.
- It launches the default e-mail application for the visitor's browser with your e-mail address as the recipient.

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.

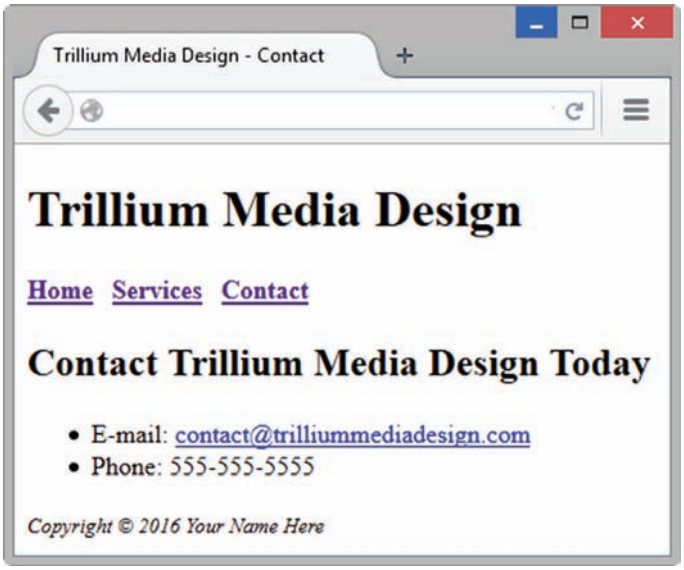


Figure 2.26 An e-mail hyperlink has been configured on the contact page

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 2.26. You can compare your work with the sample in the student files (chapter2/2.14/contact.html).

FAQ Won't displaying my actual e-mail address on a web page increase spam?

Yes and no. While it's possible that some unethical spammers may harvest web pages for e-mail addresses, the chances are that your e-mail application's built-in spam filter will prevent your inbox from being flooded with messages. When you configure an easily readable e-mail hyperlink you increase the usability of your website for your visitors in the following situations:

- The visitor may be at a public computer with no e-mail application configured. In this case, when the e-mail hyperlink is clicked, an error message may display, and the visitor will have difficulty contacting you using the e-mail link.
- The visitor may be at a private computer but may prefer not to use the e-mail application (and address) that is configured by default to work with the browser. Perhaps he or she shares the computer with others, or perhaps he or she wishes to preserve the privacy of the default e-mail address

If you prominently displayed your actual e-mail address, in both of these situations the visitor can still access your e-mail address and use it to contact you (in either their e-mail application or via a web-based e-mail system such as Google's Gmail). The result is a more usable website for your visitors.

Accessibility and Hyperlinks

Focus on Accessibility

Visually challenged visitors who are using a screen reader can configure the software to display a list of the hyperlinks in the document. However, a list of links is useful only if the text describing each link is actually helpful and descriptive. For example, on your college website, a "Search the course schedule" link would be more useful than a link that simply says, "More information."

FAQ What are some tips for using hyperlinks?

- Make your link names descriptive and brief to minimize possible confusion.
- Avoid using the phrase "Click here" in your hyperlinks. In the early days of the Web, this phrase was needed because clicking links was a new experience for web users. Now that the Web is a daily part of our lives, this phrase is slightly redundant, and even archaic.
- Try not to bury hyperlinks within large blocks of text; use lists of hyperlinks instead. Be aware that it is more difficult to read web pages than printed pages.
- Be careful when linking to external websites. The Web is dynamic, and it's possible that the external site may change the name of the page, or even delete the page. If this happens, your link will be broken.

Checkpoint 2.3

1. Describe the purpose of special characters.

2. Describe when to use an absolute link. Is the http protocol used in the href value?

3. Describe when to use a relative link. Is the http protocol used in the href value?

2.18 HTML Validation

The W3C's free Markup Validation Service, available at http://validator.w3.org, will validate your HTML code and check it for syntax errors. HTML **validation** provides students with quick self-assessment—you can prove that your code uses correct syntax. In the working world, HTML validation serves as a quality assurance tool. Invalid code may cause browsers to render the pages slower than otherwise.

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 `</strong>` tag. This modification should generate several error messages.

VideoNote
HTML Validation

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 2.27).

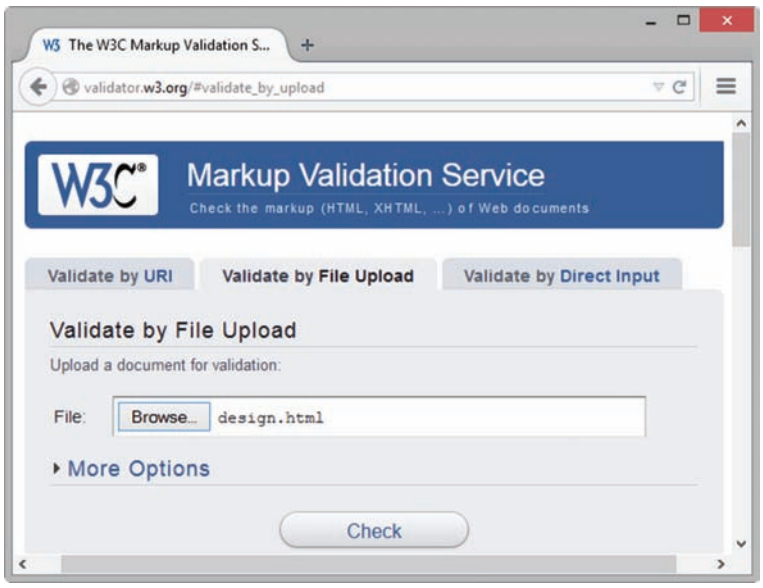


Figure 2.27 Validating a page with the W3C Markup Validation Service. Screenshots of W3C. Courtesy of W3C (World Wide Web Consortium)

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 2.28.

Notice that the message indicates line 10, which is after the missing closing `</strong>` 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.

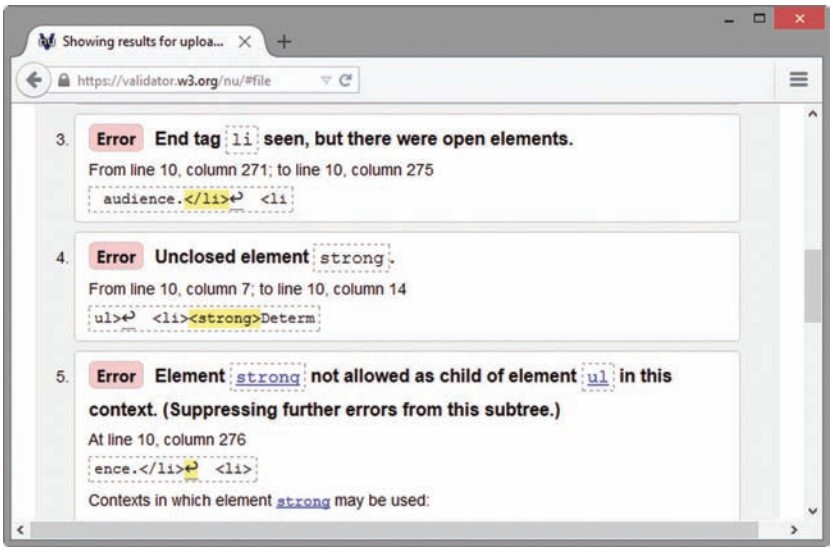


Figure 2.28 The service indicates errors. Screenshots of W3C. Courtesy of W3C (World Wide Web Consortium)

Edit the design.html file in a text editor, and add the missing `</strong>` 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 2.29. 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.

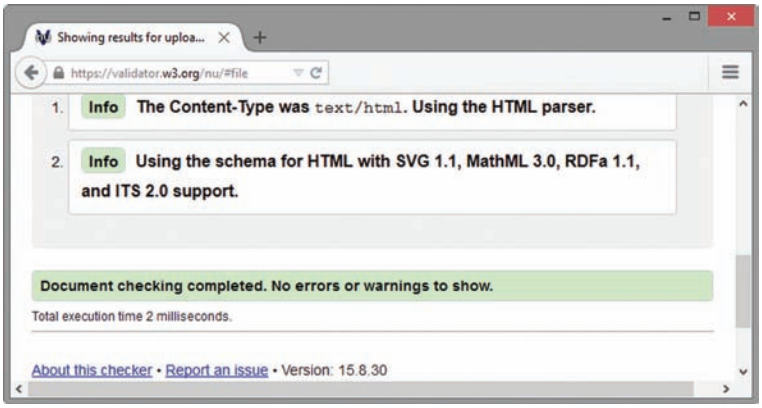


Figure 2.29 The page has passed the validation test. Screenshots of W3C. Courtesy of W3C (World Wide Web Consortium)

FAQ **Are there other ways to validate my HTML?**

In addition to the W3C validation service, there are other tools that you can use to check the syntax of your code. Explore the HTML5 validator at http://html5.validator.nu and the HTML & CSS Validation Service at http://www.onlinewebcheck.com.

Chapter Summary

This chapter has provided an introduction to HTML, XHTML, and HTML5. The basic elements that are part of every web page were demonstrated. HTML elements including div, paragraph, blockquote, header, nav, main, and footer were presented. Additional topics included configuring lists and using special characters, phrase elements, and hyperlinks. You have practiced testing your HTML5 code for valid syntax. If you worked along with the samples in the chapter, you should be ready to create some web pages on your own. The Hands-On Exercises and Web Case Studies that follow will provide some additional practice.

Visit this textbook's website at http://www.webdevfoundations.net for links to the URLs listed in this chapter and to view updated information.

Key Terms

`©`
` `
`<a>`
`<abbr>`
`<b>`
`<blockquote>`
`<body>`
`<br>`
`<cite>`
`<code>`
`<dd>`
`<dfn>`
`<div>`
`<dl>`
`<dt>`
`<em>`
`<footer>`
`<h1>`
`<h6>`
`<head>`
`<header>`
`<html>`
`<i>`
`<kbd>`
`<li>`
`<main>`
`<mark>`
`<meta>`
`<nav>`
`<ol>`
`<p>`
`<samp>`
`<small>`
`<strong>`
`<sub>`
`<sup>`
`<title>`
`<ul>`
`<var>`
`align`
absolute hyperlink
anchor element
attributes
block display
blockquote element
body element
body section
character encoding
description list
div element
doctype
Document Type Definition (DTD)
element
e-mail hyperlinks
entity element
eXtensible HyperText Markup Language (XHTML)
footer element
head element
head section
header element
heading element
href attribute
HTML5
hyperlink
Hypertext Markup Language (HTML)
inline display
lang attribute
left alignment
line break element
main element
markup languages
meta element
nav element
obsolete
ordered list
paragraph elements
phrase elements
relative hyperlink
reversed attribute
self-contained
sitemap
special characters
stand-alone
start attribute
tag
title element
type element
unordered list
validation
void element
well-formed
XML (eXtensible Markup Language)

Review Questions

Multiple Choice

1. Which tag is used to link web pages to each other?
   a. `<link>` tag
   b. `<hyperlink>` tag
   c. `<a>` tag
   d. `<body>` tag

2. Which tag configures the next element or portion of text to display on a new line?
   a. `<line>`
   b. `<nl>`
   c. `<br>`
   d. `<new>`

3. Which tag pair configures a structural area on a web page?
   a. `<area> </area>`
   b. `<div> </div>`
   c. `<cite> </cite>`
   d. `<strong> </strong>`

4. Which tag pair is used to create the largest heading?
   a. `<h1> </h1>`
   b. `<h9> </h9>`
   c. `<h type="largest"> </h>`
   d. `<h6> </h6>`

5. What is the default alignment for elements on a web page?
   a. center
   b. left
   c. right
   d. wherever you type them in the source code

6. When do you need to use a fully qualified URL in a hyperlink?
   a. always
   b. when linking to a web page file on the same site
   c. when linking to a web page file on an external site
   d. never

7. Which of the following is a reason that the text contained by the title tag should be descriptive and include the name of the business or organization?
   a. The title is saved by default when a visitor bookmarks a web page.
   b. The title may be printed when a visitor prints a web page.
   c. The title may be listed in search engine results.
   d. All of the above are reasons that the text contained by the title tag should be descriptive and include the name of the business or organization.

8. Which type of HTML list will automatically number the items for you?
   a. numbered list
   b. ordered list
   c. unordered list
   d. definition list

9. Which of the following is an HTML5 element used to indicate navigational content?
   a. main
   b. nav
   c. header
   d. a

10. What does an e-mail link do?
    a. automatically sends you an e-mail message with the visitor's e-mail address as the reply-to field
    b. launches the default e-mail application for the visitor's browser, with your e-mail address as the recipient
    c. displays your e-mail address so that the visitor can send you a message later
    d. links to your mail server

Fill in the Blank

11. Use the _____ element to configure text to have strong importance and display in a bold font weight.

12. _____ can be used to display characters such as the copyright symbol.

13. The `<meta>` tag can be used to _____.

14. The _____ is used to place a non-breaking space on a web page.

15. Use the _____ element to configure text to be emphasized and displayed in an italic font style.

Short Answer

16. Explain why it is good practice to place the e-mail address on the web page and within the anchor tag when creating an e-mail link.

Apply Your Knowledge

1. Predict the Result. Sketch out and briefly describe the web page that will be created with the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Predict the Result</title>
  <meta charset="utf-8">
</head>
<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; 2016 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.</dd>
   <_>
   </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>
```

Hands-On Exercises

1. Write the HTML to display your name with the largest heading element.
2. Write the HTML to create an absolute link to your school's website.
3. Write the HTML for an unordered list to display the days of the week.
4. Write the HTML for an ordered list that uses uppercase letters to order the items. This ordered list should display the following terms: HTML, XML, and XHTML.
5. Think of a favorite quotation by someone you admire. Write the HTML code to display the person's name in a heading and the quotation in a blockquote element.
6. Modify the following code snippet to indicate that the term "site map" should have strong importance:

```
<p>A diagram of the organization of a website is called a site map.
A site map represents the structure, or organization, of pages in
a website in a visual manner. Creating the site map is one of the
initial steps in developing a website.</p>
```

7. Modify the blockquote.html web page you created in Hands-On Practice 2.5. Configure the URL http://www.w3.org/WAI/ as a hyperlink. Save the file as blockquote2.html.
8. Create a web page that uses a description list to display three network protocols (see Chapter 1) and their descriptions. Include a hyperlink to a website that provides information about the protocols. Add an appropriate heading to the page. Save the page as network.html.
9. Create a web page about your favorite musical group. Include the name of the group, the individuals in the group, a hyperlink to the group's website, your favorite three (or fewer if the group is new) CD releases, and a brief review of each CD.
   - Use an unordered list to organize the names of the individuals.
   - Use a description list for the names of the CDs and your reviews.

   Save the page as band.html.
10. Create a web page about your favorite recipe. Use an unordered list for the ingredients and an ordered list to describe the steps needed to prepare the food. Include a hyperlink to a website that offers free recipes. Save the page as recipe.html.

Web Research

There are many HTML5 tutorials on the Web. Use your favorite search engine to discover them. Choose two that are helpful. For each, print out the home page or other pertinent page and create a web page that contains the answers to the following questions:

a. What is the URL of the website?

b. Is the tutorial geared toward the beginner level, intermediate level, or both levels?

c. Would you recommend this site to others? Why or why not?

d. List one or two concepts that you learned from this tutorial.

Focus on Web Design

You are learning the syntax of HTML5. However, coding alone does not make a web page; design is also very important. Surf the Web and find two web pages, one that is appealing to you and one that is unappealing to you. Print each page. Create a web page that answers the following questions for each of your examples:

a. What is the URL of the website?

b. Is the page appealing or unappealing? List three reasons for your answer.

c. If the page is unappealing, what would you do to improve it?

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 2.30. The site map describes the architecture of the website, which consists of a "Home" page with three main content pages: "Menu," "Music," and "Jobs."

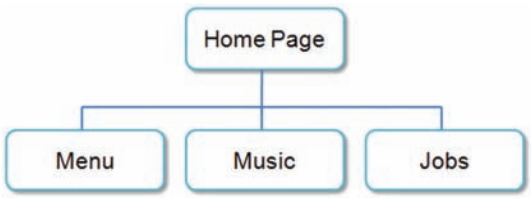


Figure 2.30 JavaJam site map

Figure 2.31 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.

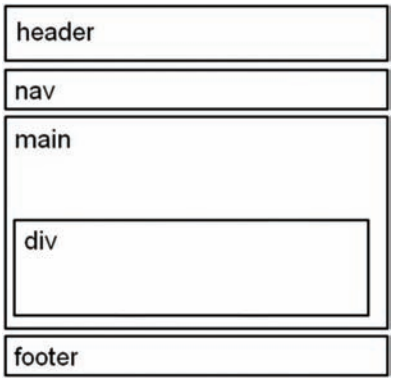


Figure 2.31 JavaJam wireframe

Hands-On Practice Case Study

Task 1: 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 2.32.

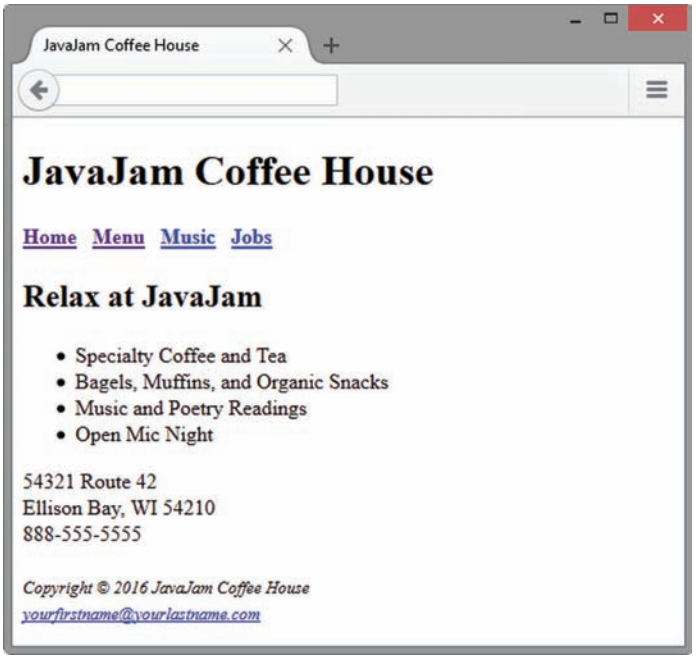


Figure 2.32 JavaJam index.html

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 a business website.
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 <b> 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.

   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 Organic Snacks

   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.

   54321 Route 42

   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 © 2016 JavaJam Coffee House

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

The page in Figure 2.32 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 2.33. 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 2.33 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 `<strong>` 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 $2.00

   Cafe au Lait

   House blended coffee infused into a smooth, steamed milk.

   Single $2.00 Double $3.00

   Iced Cappuccino

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

   Single $4.75 Double $5.75

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.

Fish Creek Animal Hospital

Magda Patel is a veterinarian and owner of the Fish Creek Animal Hospital. Her customers are local pet owners who range from children to senior citizens. Magda would like a website to provide information to her current and potential customers. She has requested a home page, a services page, a page for advice from a veterinarian, and a contact page.

A site map for the Fish Creek Animal Hospital website is shown in Figure 2.34. The site map describes the architecture of the website, which consists of a "Home" page with three main content pages: "Services," "Ask the Vet," and "Contact."

Figure 2.34 Fish Creek site map

Figure 2.35 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.

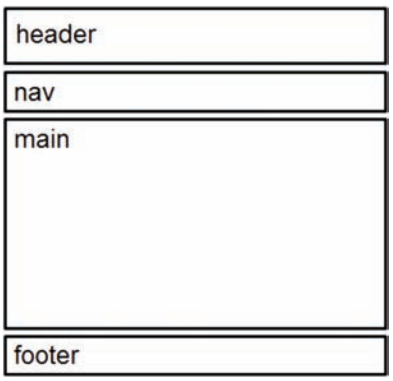


Figure 2.35 Fish Creek wireframe

You have three tasks in this case study:

1. Create a folder for the Fish Creek website.
2. Create the Home page: index.html.
3. Create the Services page: services.html.

Hands-On Practice Case Study

Task 1: Create a folder on your hard drive or portable storage device (thumb drive or SD card) called "fishcreek" to contain your Fish Creek website files.

Task 2: The Home Page. You will use a text editor application to create the Home page for the Fish Creek Animal Hospital website. The Home page is shown in Figure 2.36.

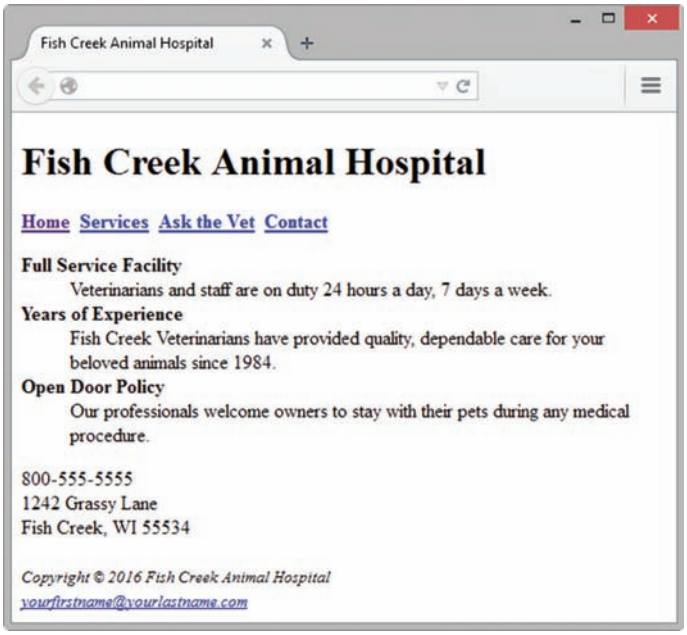


Figure 2.36 Fish Creek index.html

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 a business website.
2. **Wireframe Header.** Code the header element with the text, "Fish Creek Animal Hospital" contained within a heading 1 element.
3. **Wireframe Navigation.** Place the following text within a nav element with bold text (use the `<b>` element)

   Home Services Ask the Vet Contact

   Code anchor tags so that "Home" links to index.html, "Services" links to services.html, "Ask the Vet" links to askvet.html, and "Contact" links to contact.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.

   a. Code the following content in a description list. Configure the text in each dt element to have strong importance and display in bold font weight:

   Full Service Facility

   Veterinarians and staff are on duty 24 hours a day, 7 days a week.

   Years of Experience

   Fish Creek Veterinarians have provided quality, dependable care for your beloved animals since 1984.

Open Door Policy

Our professionals welcome owners to stay with their pets during any medical procedure.

b. Configure the following address and phone number contact information within a div element below the description list. Use line break tags to help you format this area.

800-555-5555

1242 Grassy Lane

Fish Creek, WI 55534

5. **Wireframe Footer.** Code 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 © 2016 Fish Creek Animal Hospital

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

The page in Figure 2.36 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 `<br>` 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 fishcreek folder, and name it index.html.

Task 3: The Services Page. Create the Services page shown in Figure 2.37. 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 Services page will use the index.html page as a starting point.

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

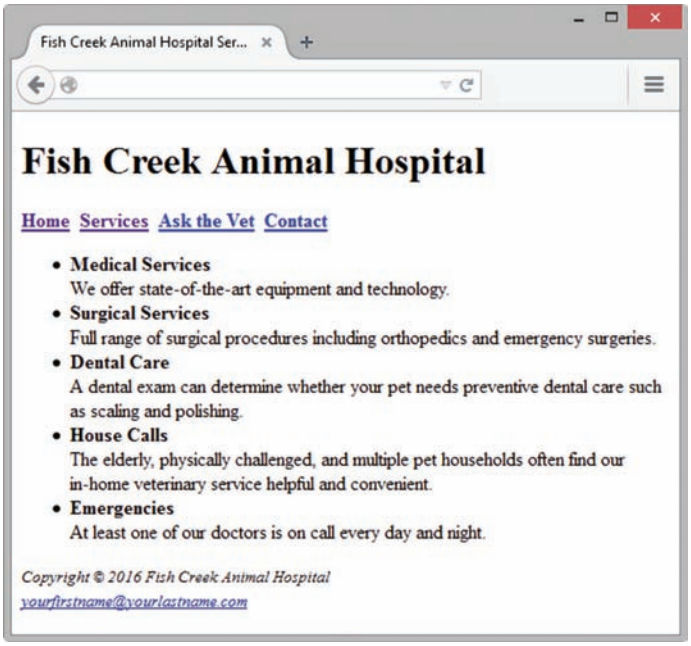


Figure 2.37 Fish Creek services.html

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

   Fish Creek Animal Hospital Services

2. **Wireframe Main Content.**

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

   b. Use an unordered list to add the services content to the page. Configure the name of each services category to use bold font weight and have strong emphasis. (Use the tag.) Use line break tags to help you configure this area. The service categories and descriptions are as follows:

   Medical Services

   We offer state-of-the-art equipment and technology.

   Surgical Services

   Full range of surgical procedures including orthopedics and emergency surgeries.

   Dental Care

   A dental exam can determine whether your pet needs preventive dental care such as scaling and polishing.

   House Calls

   The elderly, physically challenged, and multiple pet households often find our in-home veterinary service helpful and convenient.

   Emergencies

   At least one of our doctors is on call every day and night.

Save your page, and test it in a browser. Test the hyperlink from the services.html page to index.html. Test the hyperlink from the index.html page to services.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.

Pacific Trails Resort

Melanie Bowie is the owner of Pacific Trails Resort, located on the California North Coast. The resort offers a quiet getaway, with luxury camping in yurts along with an upscale lodge for dining and visiting with fellow guests. The target audience for Pacific Trails Resort is couples who enjoy nature and hiking. Melanie would like a website that emphasizes the uniqueness of the location and accommodations. She would like the website to include a home page, a page about the special yurt accommodations, a reservations page with a contact form, and a page to describe the activities available at the resort.

A site map for the Pacific Trails Resort website is shown in Figure 2.38. The site map describes the architecture of the website, which consists of a "Home" page with three main content pages: "Yurts," "Activities," and "Reservations."

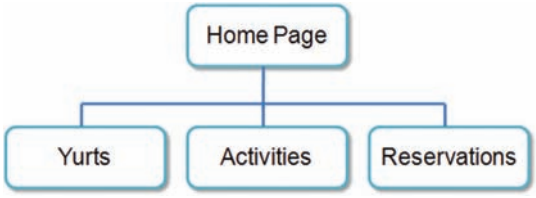


Figure 2.38 Pacific Trails Resort site map

Figure 2.39 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.

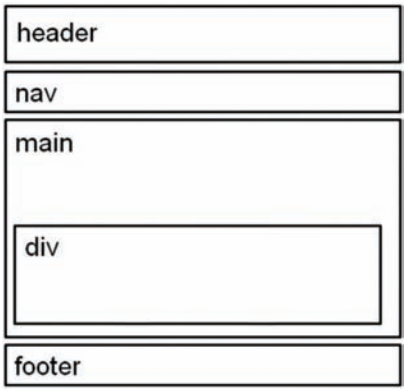


Figure 2.39 Pacific Trails Resort wireframe

You have three tasks in this case study:

1. Create a folder for the Pacific Trails website.
2. Create the Home page: index.html.
3. Create the Yurts page: yurts.html.

Hands-On Practice Case Study

Task 1: Create a folder on your hard drive or portable storage device (thumb drive or SD card) called "pacific" to contain your Pacific Trails Resort website files.

Task 2: The Home Page. You will use a text editor to create the Home page for the Pacific Trails Resort website. The Home page is shown in Figure 2.40.

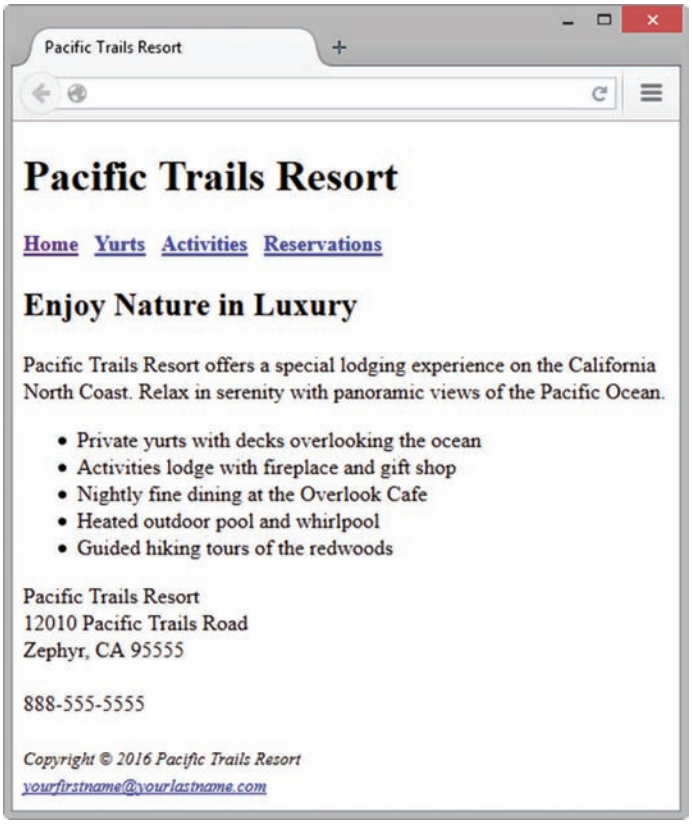


Figure 2.40 Pacific Trails Resort index.html

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 a business website.
2. **Wireframe Header.** Code the header element with the text, "Pacific Trails Resort" contained within a heading 1 element.
3. **Wireframe Navigation.** Place the following text within a nav element with bold text (use the `<b>` element):

   Home Yurts Activities Reservations

   Code anchor tags so that "Home" links to index.html, "Yurts" links to yurts.html, "Activities" links to activities.html, and "Reservations" links to reservations.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.
   a. Code the following text within an h2 element:

      Enjoy Nature in Luxury

   b. Configure the following sentences in a paragraph:

      Pacific Trails Resort offers a special lodging experience on the California North Coast. Relax in serenity with panoramic views of the Pacific Ocean.

   c. Code the following content in an unordered list:

      Private yurts with decks overlooking the ocean

      Activities lodge with fireplace and gift shop

      Nightly fine dining at the Overlook Cafe

      Heated outdoor pool and whirlpool

      Guided hiking tours of the redwoods

   d. Configure the following address and phone number contact information within a div element below the unordered list. Use line break tags to help you format this area.

      Pacific Trails Resort

      12010 Pacific Trails Road

      Zephyr, CA 95555

      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 © 2016 Pacific Trails Resort

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

The page in Figure 2.40 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 `<br>` 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 pacific folder, and name it index.html.

Task 3: The Yurts Page. Create the Yurts page shown in Figure 2.41. 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 Yurts page will use the index.html page as a starting point.

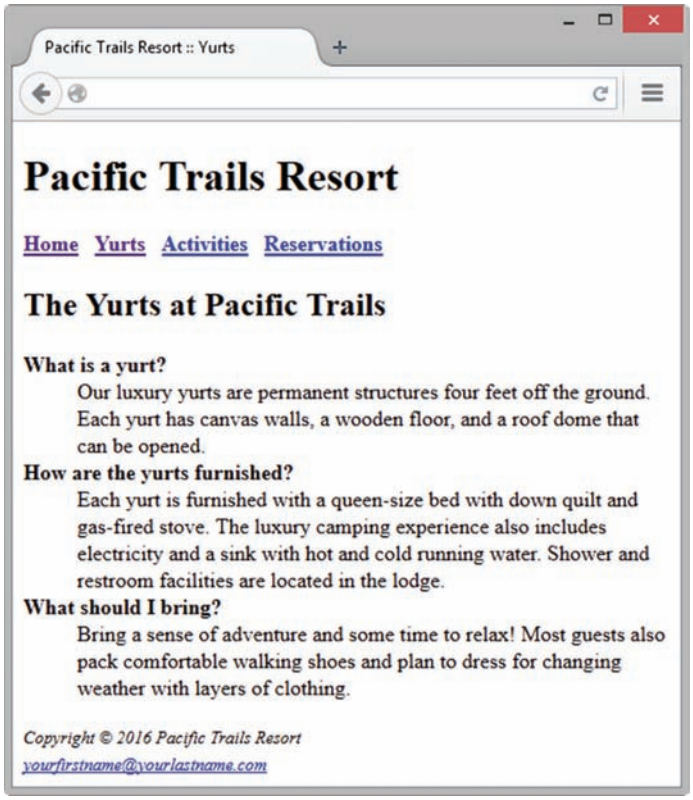


Figure 2.41 Pacific Trails Resort yurts.html

Open the index.html page for the Pacific Trails Resort website in a text editor. Select File > Save As, and save the file with the new name of yurts.html in the pacific 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:

   Pacific Trails Resort :: Yurts

2. **Wireframe Main Content.**

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

   The Yurts at Pacific Trails

   b. Delete the Home page content paragraph, unordered list, and contact information.

   c. Add the yurts content to the page as a FAQs (frequently asked questions) list by using a description list. Configure each question to have strong importance and bold font weight (use the `<strong>` phrase element) within a dt element. Configure each answer within a dd element. The text is shown as follows:

   What is a yurt?

   Our luxury yurts are permanent structures four feet off the ground. Each yurt has canvas walls, a wooden floor, and a roof dome that can be opened.

How are the yurts furnished?

Each yurt is furnished with a queen-size bed with down quilt and gas-fired stove. The luxury camping experience also includes electricity and a sink with hot and cold running water. Shower and restroom facilities are located in the lodge.

What should I bring?

Bring a sense of adventure and some time to relax! Most guests also pack comfortable walking shoes and plan to dress for changing weather with layers of clothing.

Save your page, and test it in a browser. Test the hyperlink from the yurts.html page to index.html. Test the hyperlink from the index.html page to yurts.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 the changes.

Path of Light Yoga Studio

Path of Light Yoga Studio is a small, recently opened yoga studio. The owner, Ariana Starrweaver, would like a website to showcase her yoga studio and provide information for both new and current students. Ariana would like a home page, a classes page that contains information about the types of yoga classes offered, a schedule page, and a contact page.

A site map for the Path of Light Yoga Studio website is shown in Figure 2.42. The site map describes the architecture of the website, which consists of "Home" page with three main content pages: "Classes," "Schedule," and "Contact."

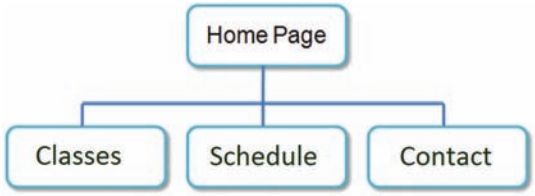


Figure 2.42 Path of Light Yoga Studio site map

Figure 2.43 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.

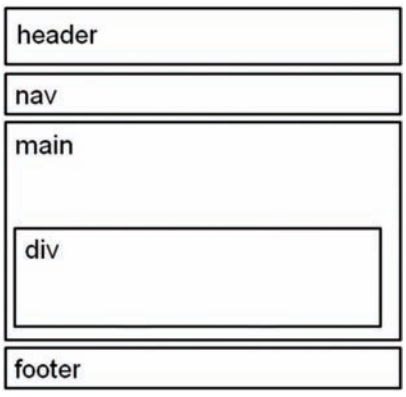


Figure 2.43 Path of Light Yoga Studio wireframe

You have three tasks in this case study:

1. Create a folder for the Path of Light Yoga Studio website.
2. Create the Home page: index.html.
3. Create the Classes page: classes.html.

Hands-On Practice Case Study

Task 1: Create a folder on your hard drive or portable storage device (thumb drive or SD card) called "yoga" to contain your Path of Light Yoga Studio web page files.

Task 2: The Home Page. You will use a text editor to create the Home page for the Path of Light Yoga Studio website. The Home page is shown in Figure 2.44.

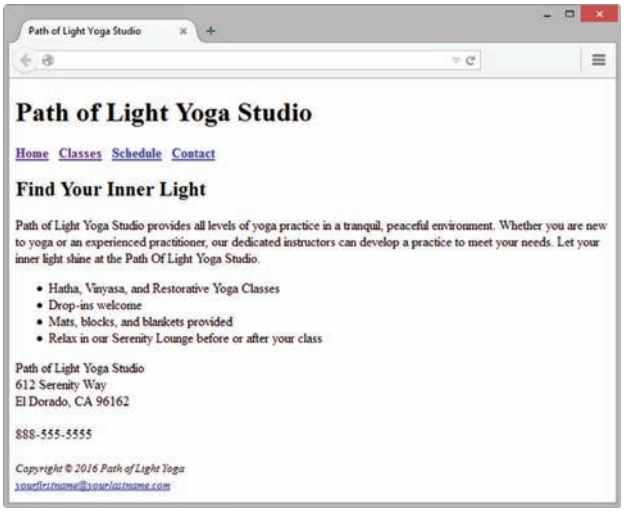


Figure 2.44 Path of Light Yoga Studio index.html

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 a business website.
2. **Wireframe Header.** Code the header element with the text, "Path of Light Yoga Studio" contained within a heading 1 element.
3. **Wireframe Navigation.** Place the following text within a nav element with bold text (use the `<b>` element):

   Home Classes Schedule Contact

   Code anchor tags so that "Home" links to index.html, "Classes" links to classes.html, "Schedule" links to schedule.html, and "Contact" links to contact.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.

   a. Code the following text within an h2 element:

   Find Your Inner Light

b. Configure the following sentences in a paragraph:

Path of Light Yoga Studio provides all levels of yoga practice in a tranquil, peaceful environment. Whether you are new to yoga or an experienced practitioner, our dedicated instructors can develop a practice to meet your needs. Let your inner light shine at the Path of Light Yoga Studio.

c. Configure the following content in an unordered list:

Hatha, Vinyasa, and Restorative Yoga classes

Drop-ins welcome

Mats, blocks, and blankets provided

Relax in our Serenity Lounge before or after your class

d. 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.

Path of Light Yoga Studio

612 Serenity Way

El Dorado, CA 96162

888-555-5555

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

Copyright © 2016 Path of Light Yoga

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

The page in Figure 2.44 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 `<br>` 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 yoga folder, and name it index.html.

Task 3: The Classes Page. Create the Classes page shown in Figure 2.45. 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 Classes page will use the index.html page as a starting point.
Open the index.html page for the Path of Light Yoga Studio website in a text editor. Select File > Save As, and save the file with the new name of classes.html in the yoga 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:

Path of Light Yoga Studio :: Classes

2. **Wireframe Main Content.**

a. Delete the Home Page content paragraphs, unordered list, and contact information.

b. Configure the following text in the heading 2 element:

Yoga Classes

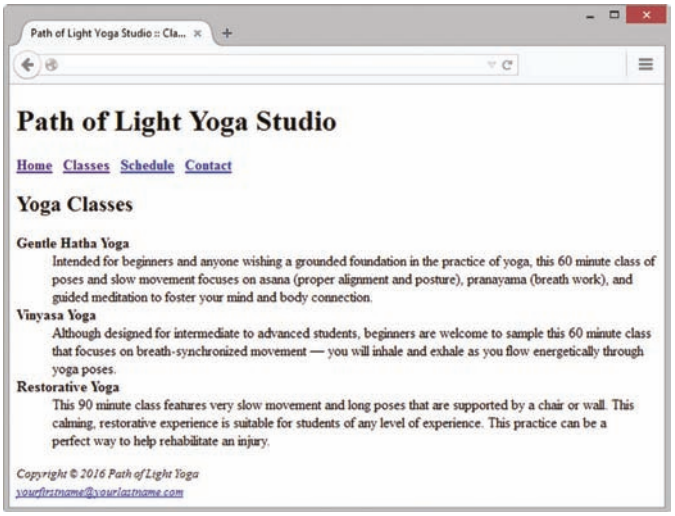


Figure 2.45 Path of Light Yoga Studio classes.html

c. Use a description list to configure information about the yoga classes. Configure the name of each class to have strong importance and bold font weight (use the `<strong>` phrase element) within a `dt` element. Configure `<dd>` elements for the class descriptions. The information follows:

Gentle Hatha Yoga

Intended for beginners and anyone wishing a grounded foundation in the practice of yoga, this 60 minute class of poses and slow movement focuses on asana (proper alignment and posture), pranayama (breath work), and guided meditation to foster your mind and body connection.

Vinyasa Yoga

Although designed for intermediate to advanced students, beginners are welcome to sample this 60 minute class that focuses on breath-synchronized movement—you will inhale and exhale as you flow energetically through yoga poses.

Restorative Yoga

This 90 minute class features very slow movement and long poses that are supported by a chair or wall. This calming, restorative experience is suitable for students of any level of experience. This practice can be a perfect way to help rehabilitate an injury.

Save your page, and test it in a browser. Test the hyperlink from the classes.html page to index.html. Test the hyperlink from the index.html page to classes.html. If your links do not work, review your work with 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 the changes.