

4

Visual Elements and Graphics

Chapter Objectives

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

- Create and format lines and borders on web pages
- Decide when to use graphics and what graphics are appropriate
- Apply the image element to add graphics to web pages
- Optimize an image for web page display
- Configure images as backgrounds on web pages
- Configure images as hyperlinks
- Configure rounded corners, box shadow, text shadow, opacity, and gradients with CSS3
- Configure RGBA color with CSS3
- Configure HSLA color with CSS3
- Use HTML5 elements to caption a figure
- Use the HTML5 meter and progress elements
- Find free and fee-based graphics sources
- Follow recommended web design guidelines when using graphics on web pages

A key component of a compelling website is the use of interesting and appropriate graphics. This chapter introduces you to working with visual elements on web pages.

When you include images on your website, it is important to remember that not all users are able to view them. Some users may have vision problems and need assistive technology such as a screen reader application that reads the web page to them. In addition, search engines send out spiders and robots to walk the web and catalog pages for their indexes and databases; such programs do not access your images. Visitors using a mobile device may have images disabled. As a web developer, strive to create pages that are enhanced by graphical elements, but are usable without them.

4.1 Configuring Lines and Borders

Web designers often use visual elements such as lines and borders to separate or define areas on web pages. In this section, you'll explore two coding techniques to configure a line on a web page: the HTML horizontal rule element and the CSS border and padding properties.

The Horizontal Rule Element

A **horizontal rule element**, `<hr>`, visually separates areas of a page and configures a horizontal line across a web page. Since the horizontal rule element does not contain any text, it is coded as a void tag and not in a pair of opening and closing tags. The horizontal rule element has a new semantic meaning in HTML5, it can be used to indicate a thematic break or change in the content.



Hands-On Practice 4.1

Open the web page found at chapter4/starter1.html in the student files in a text editor. Add an `<hr>` tag after the opening footer element.

Save your file as hr.html, and test it in a browser. The lower portion of your web page should look similar to the partial screenshot shown in Figure 4.1. Compare your work with the solution in the student files (chapter4/4.1/hr.html). While a horizontal rule can be easily created using HTML, a more modern technique for configuring lines on web pages is to use CSS to configure a border.

Meeting Your Business Needs

Our expert designers are creative and eager to work with you.

Copyright © 2016 Your Name Here

Figure 4.1 The `<hr>` tag configures a horizontal line

The border and padding Properties

As you may have noticed when you configured background colors for heading elements in Chapter 3, block display HTML elements form the shape of a rectangular box on a web page. This is an example of the CSS box model, which you will explore in detail in Chapter 6. For now, let's focus on two CSS properties that can be configured for the "box": the `border` and `padding` properties.

The `border` Property

The **border property** configures the border, or boundary, around an element. By default, the border has a width set to 0 and does not display. You can set the `border-width`, `border-color`, and `border-style` with the `border` property. And there's more—you

can even configure individual settings for the top, right, bottom and left borders using the `border-top`, `border-right`, `border-bottom`, and `border-left` properties.

The `border-style` Property

The **`border-style` property** configures the type of line displayed in the border. The formatting options include `inset`, `outset`, `double`, `groove`, `ridge`, `solid`, `dashed`, and `dotted`. Be aware that these property values are not all uniformly applied by browsers. Figure 4.2 shows how a recent version of Firefox renders various border-style values.

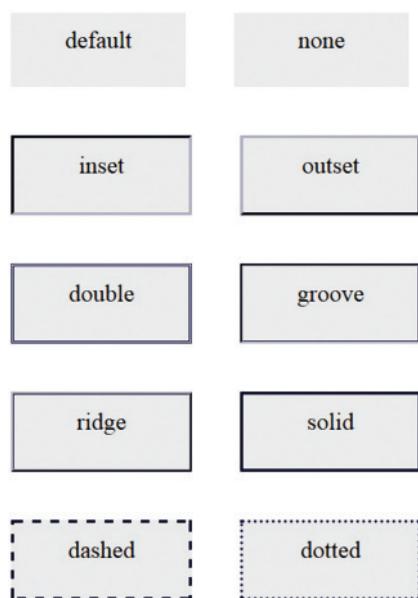


Figure 4.2 Examples of the various border-style values rendered by Firefox

The CSS to configure the borders shown in Figure 4.2 uses a `border-width` of 3 pixels, the value indicated for the `border-style` property, and a `border-color` of `#000000`. For example, the style rule to configure the dashed border follows:

```
.dashedborder { border-width: 3px;  
    border-style: dashed;  
    border-color: #000033; }
```

A shorthand notation allows you to configure all the border properties in one style rule by listing the values of `border-width`, `border-style`, and `border-color`, as in the following example:

```
.dashedborder { border: 3px dashed #000033; }
```

The `padding` Property

The **`padding` property** configures empty space between the content of the HTML element (usually text) and the border. By default, the padding is set to 0. If you configure a background color for an element, the color is applied to both the padding and the content areas. You'll apply the padding and `border` properties in the next Hands-On Practice. You may want to refer to Table 4.1, which presents a description of the CSS properties introduced in this chapter, as you work through the Hands-On Practice exercises.

Table 4.1 New CSS properties introduced in this chapter

Property	Description	Values
background-attachment	Configures whether the background image scrolls with the page or is fixed in place	fixed, scroll (default)
background-clip	Configures the background painting area. This CSS3 property is not supported in all browsers.	padding-box, border-box, or content-box
background-image	Background image on an element	To display an image, use <code>url(imagename.gif)</code> , <code>url(imagename.jpg)</code> , or <code>url(imagename.png)</code> . To disable image display, use <code>none</code> (default).
background-origin	Configures the background positioning area. This CSS3 property is not supported in all browsers.	padding-box, border-box, or content-box
background-position	Position of the background image	Two percentage values or numeric pixel values. The first value configures the horizontal position, and the second configures the vertical position starting from the upper left corner of the container's box. Text values can also be used: <code>left</code> , <code>top</code> , <code>center</code> , <code>bottom</code> , and <code>right</code> .
background-repeat	Controls how the background image will repeat	Text values <code>repeat</code> (default), <code>repeat-y</code> , (vertical repeat), <code>repeat-x</code> (horizontal repeat), and <code>no-repeat</code> (no repeat). The new CSS3 values (<code>space</code> , <code>round</code>) are not supported in all browsers.
background-size	Configures the size of the background image. This CSS3 property is not supported in all browsers.	Two percentages, pixel values, <code>auto</code> , <code>contain</code> , or <code>cover</code> . The first value indicates width. The second value indicates height. If only one value is provided, the second value defaults to <code>auto</code> . The value <code>contain</code> causes the background image to be scaled (with aspect ratio intact) to horizontally fill the container. The value <code>cover</code> causes the background image to be scaled (with aspect ratio intact) to vertically fill the container.
border	Shorthand notation to configure the values for <code>border-width</code> , <code>border-style</code> , and <code>border-color</code> of an element	The values for <code>border-width</code> , <code>border-style</code> , and <code>border-color</code> separated by spaces—for example, <code>border: 1px solid #000000;</code>
border-bottom	Shorthand notation to configure the bottom border of an element	The values for <code>border-width</code> , <code>border-style</code> , and <code>border-color</code> separated by spaces—for example, <code>border-bottom: 1px solid #000000;</code>
border-bottom-left-radius	Configures rounded corners on the bottom left corner of a border. This CSS3 property is not supported in all browsers.	One numeric value (px or em) or percentage that configures the radius of the corner.
border-bottom-right-radius	Configures rounded corners on the bottom right corner of a border. This CSS3 property is not supported in all browsers.	One numeric value (px or em) or percentage that configures the radius of the corner.
border-color	The color of the border around an element	Any valid color value
border-left	Shorthand notation to configure the left border of an element	The values for <code>border-width</code> , <code>border-style</code> , and <code>border-color</code> separated by spaces—for example, <code>border-left: 1px solid #000000;</code>

Table 4.1 (Continued)

Property	Description	Values
border-radius	Configures rounded corners on a element. This CSS3 property is not supported in all browsers.	One to four numeric values (px or em) or percentages that configure the radius of the corners. If a single value is provided, it configures all four corners. The corners are configured in order of top left, top right, bottom right, and bottom left.
border-right	Shorthand notation to configure the right border of an element	The values for border-width, border-style, and border-color separated by spaces—for example, border-right: 1px solid #000000;
border-style	The type of border around an element	Text values double, groove, inset, none (the default), outset, ridge, solid, dashed, dotted, and hidden
border-top	Shorthand notation to configure the top border of an element	The values for border-width, border-style, and border-color separated by spaces—for example, border-top: 1px solid #000000;
border-top-left-radius	Configures a rounded top left corner. This CSS3 property is not supported in all browsers.	One numeric value (px or em) or percentage that configures the radius of the corner.
border-top-right-radius	Configures a rounded top right corner. This CSS3 property is not supported in all browsers.	One numeric value (px or em) or percentage that configures the radius of the corner.
border-width	The width of a border around an element	A numeric pixel value (such as 1px) or the text values thin, medium, and thick
box-shadow	Configures a drop shadow on an element. This CSS3 property is not supported in all browsers.	Two to four numerical values (px or em) to indicate horizontal offset, vertical offset, blur radius (optional), and spread distance (optional), and a valid color value. Use the inset keyword to configure an inner shadow.
height	The height of an element	A numeric pixel value or percentage
linear-gradient	Configures a linear blending of shades from one color to another. This CSS3 property is not supported in all browsers.	Numerous syntax options for starting points of the gradient and color values. For example, the following configures a two-color linear gradient: <code>linear-gradient(#FFFFFF, #8FA5CE);</code>
max-width	Configures a maximum width for an element	A numeric pixel value or percentage
min-width	Configures a minimum width for an element	A numeric pixel value or percentage
opacity	Configures the transparency of an element. This CSS3 property is not supported in all browsers.	Numeric values between 1 (fully opaque) and 0 (completely transparent). This property is inherited by all child elements.
padding	Shorthand notation to configure the amount of padding—the blank space between the element and its border	<ol style="list-style-type: none"> 1. A single numeric value (px or em) or percentage; configure padding on all sides of the element. 2. Two numeric values (px or em) or percentages; the first value configures the top and bottom padding, and the second value configures the left and right padding—for example, <code>padding: 20px 10px;</code> 3. Three numeric values (px or em) or percentages; the first value configures the top padding, the second value configures the left and right padding, the third value configures the bottom padding—for example, <code>padding: 5px 30px 10px;</code> 4. Four numeric values (px or em) or percentages; the values configure the padding in the following order: <code>padding-top, padding-right, padding-bottom, padding-left.</code>

(Continued)

Table 4.1 (Continued)

Property	Description	Values
padding-bottom	Blank space between an element and its bottom border	A numeric value (px or em) or percentage
padding-left	Blank space between an element and its left border	A numeric value (px or em) or percentage
padding-right	Blank space between an element and its right border	A numeric value (px or em) or percentage
padding-top	Blank space between an element and its top border	A numeric value (px or em) or percentage
radial-gradient	Configures a radial (circular) blending of shades from one color to another. This CSS3 property is not supported in all browsers.	Numerous syntax options for starting points of the gradient and color values. For example, the following configures a two-color radial gradient: <code>radial-gradient (#FFFFFF, #8FA5CE);</code>



Hands-On Practice 4.2

In this Hands-On Practice you will work with the `border` and `padding` properties. Launch a text editor, and open the web page found at chapter4/starter1.html in the student files. You will modify the CSS style rules for the `h1` element selector, `h2` element selector, and `footer` element selector. When you are finished, your page should look similar to the one shown in Figure 4.3.

Edit the CSS style rules as follows:

1. Code a style for the `h1` element selector to set the padding to 1em. The code follows:

```
h1 {padding: 1em;}
```

2. Add a style rule to the `h2` element selector to configure a 2-pixel, dashed, bottom border in the color #191970. The code follows:

```
border-bottom: 2px dashed #191970;
```

3. Add styles to the `footer` element selector to configure a thin, solid, top border in the color #aeaed4 along with 10 pixels of top padding. Also configure the footer to have gray text. The new style declarations follow:

```
border-top: thin solid #aeaed4;
padding-top: 10px;
color: #333333;
```

Save your file as `border.html`.

Test your page in multiple browsers. Expect your page to look slightly different in various browsers. See Figure 4.3 for a screenshot of the page using Firefox. Figure 4.4 shows the page displayed in Internet Explorer. The student files contain a sample solution (chapter4/4.2/border.html).

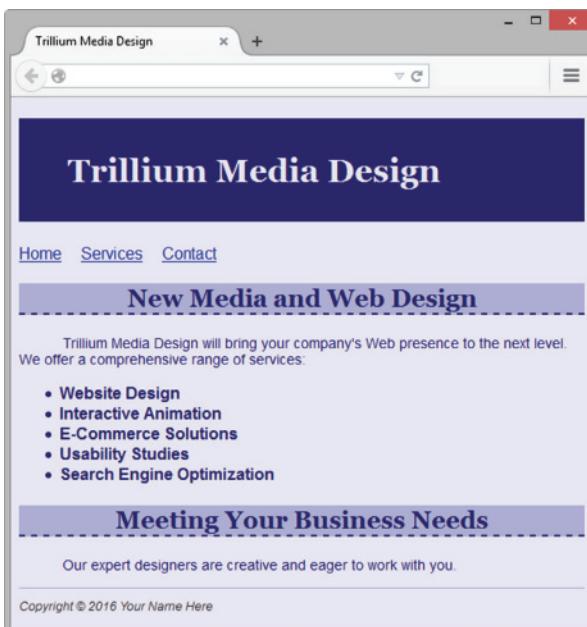


Figure 4.3 CSS border and padding properties add visual interest to the page. Screenshots of Internet Explorer. Copyright by Microsoft Corporation. Used by permission of Microsoft Corporation

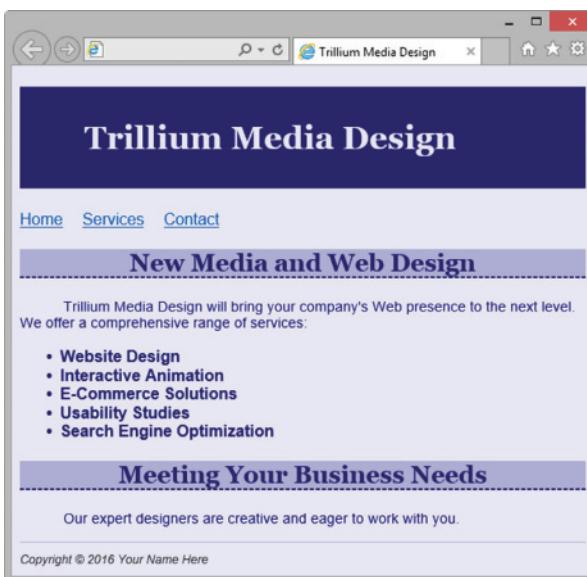


Figure 4.4 Internet Explorer renders the dashed border differently than Firefox. Screenshots of Internet Explorer. Copyright by Microsoft Corporation. Used by permission of Microsoft Corporation



FAQ My web page looks different in some browsers. What can I do?

Do not expect your web pages always to look the same in every browser and every browser version. Web pages that look slightly different in various browsers are a part of life in the world of web developers.



Checkpoint 4.1

1. Is it reasonable to try to code a web page that looks exactly the same on every browser and every platform? Explain your answer.
2. Describe what is incorrect with the following CSS code, which causes a web page containing it not to display a border when rendered in a browser:

```
h2 { background-color: #FF0000  
     border-top: thin solid #000000 }
```

3. True or False: CSS can be used to configure visual elements such as rectangular shapes and lines on web pages.

4.2 Types of Graphics

Graphics can make web pages compelling and engaging. This section discusses features of graphic files commonly used on the Web: GIF, JPEG, and PNG. A new web graphic format, WebP, is also introduced.

Graphic Interchange Format (GIF) Images



Figure 4.5

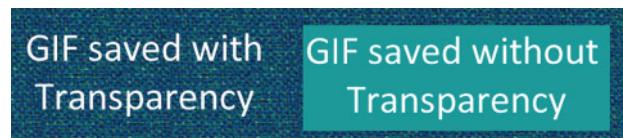
This logo is a GIF

Transparency

The format GIF89A used by GIF images supports image **transparency**. In a graphics application, such as the open-source GIMP, one color (typically the background color) of the image can be set to be transparent. The background color (or background image) of the web page shows through the transparent area in the image. Figure 4.6 displays two GIF images on a blue textured background.

Figure 4.6

Comparison of transparent and nontransparent GIFs



Animation

An **animated GIF** consists of several images or frames, each of which is slightly different. When the frames display on the screen in order, the image appears animated. Animated GIFs can be created in a graphics application such as Adobe Fireworks.

Compression

Lossless compression is used when a GIF is saved. This means that nothing in the original image is lost and that the compressed image, when rendered by a browser, will contain the same pixels as the original.

Optimization

To avoid slow-loading web pages, graphic files should be optimized for the Web. Photos taken with a digital camera are too large—in both their dimensions and their file size—to display well on a web page. **Image optimization** is the process of creating an image with the lowest file size that still renders a good-quality image—balancing image quality and file size. GIF images are typically optimized by using a graphics application to reduce the number of colors in the image. Popular graphics applications used by web developers include Adobe Photoshop, Adobe Fireworks, and GIMP (<http://gimp.org>).

Interlacing

Browsers render, or display, web page documents in order, line by line, starting at the top of the document. They display standard images as the files are read in order from top to bottom. The top of a standard image begins to display after 50% of the image has been read by a browser. When a GIF graphic file is created, it can be configured as interlaced. An **interlaced image** progressively displays and seems to fade in as it downloads. The image first appears fuzzy but gradually becomes clearer and sharper, which can help to reduce the perceived load time of your web page.

Joint Photographic Experts Group (JPEG) Images

JPEG images are best used for photographs. In contrast to a GIF image, a JPEG image can contain 16.7 million colors. However, JPEG images cannot be made transparent, and they cannot be animated. JPEG images have a .jpg or .jpeg file extension.

Compression

JPEG images are saved using **lossy compression**. This means that some pixels in the original image are lost or removed from the compressed file. When a browser renders the compressed image, the display is similar to, but not exactly the same as, the original image.

Optimization

There are trade-offs between the quality of the image and the amount of compression. An image with less compression will have higher quality and result in a larger file size. An image with more compression will have lower quality and result in a smaller file size. Graphics applications such as Adobe Photoshop, Adobe Fireworks, Adobe Lightroom, and GIMP allow you to configure the compression quality and choose the image that best suits your needs.

When you take a photo with a digital camera, the file size is too large for optimal display on a web page. Figure 4.7 shows an optimized version of a digital photo with an original file size of



Figure 4.7 A JPEG saved at 80% quality (55KB file size) displays well on a web page



Figure 4.8 JPEG saved at 20% quality (19KB file size)



Figure 4.9
This small thumbnail image is only 5KB

250KB. The image was optimized using a graphics application set to 80% quality, is now only 55KB, and displays well on a web page.

Figure 4.8 was saved with 20% quality and is only 19KB, but its quality is unacceptable. The quality of the image degrades as the file size decreases. The square blockiness you see in Figure 4.8 is called **pixelation** and should be avoided.

Adobe Photoshop and Adobe Fireworks are often used by web professionals to optimize images for the Web. Pixlr offers a free, easy-to-use, online photo editor at <http://pixlr.com/editor>. GIMP (<http://www.gimp.org>) is a popular open-source image editor that supports multiple platforms.

Another technique used with web graphics is to display a small version of the image, called a **thumbnail image**. Often, the thumbnail is configured as an image hyperlink that displays the larger image when clicked. Figure 4.9 shows a thumbnail image.

Progressive JPEG

When a JPEG file is created, it can be configured as progressive. A **progressive JPEG** is similar to an interlaced GIF in that the image progressively displays and seems to fade in as it downloads.

Portable Network Graphic (PNG) Images

PNG images combine the best of GIF and JPEG images and will be a replacement for GIF images in the future. PNG (pronounced “ping”) graphics can support millions of colors, support variable transparency levels, and use lossless compression. PNG images also support interlacing. Table 4.2 summarizes the characteristics of GIF, JPEG, and PNG image files.

Table 4.2 Overview of common web graphic file types

Image Type	File Extension	Compression	Transparency	Animation	Colors	Progressive Display
GIF	.gif	Lossless	Yes	Yes	256	Interlacing
JPEG	.jpg or .jpeg	Lossy	No	No	Millions	Progressive
PNG	.png	Lossless	Yes	No	Millions	Interlacing

New WebP Image Format

Google’s new **WebP image format** offers improved lossy compression for photographic images, but it’s not yet ready for use in commercial websites. WebP (pronounced “weppy”) graphics are currently supported only by the Google Chrome browser. Visit <https://developers.google.com/speed/webp> for more information on this new image format.

4.3 Image Element

The **image element** configures graphics on a web page. These graphics can be photographs, banners, company logos, navigation buttons, and so on; you are limited only by your creativity and imagination.

The image element is a void element and is not coded as a pair of opening and closing tags. The following code example configures an image named logo.gif, which is located in the same folder as the web page:

```

```

The **src attribute** specifies the file name of the image. The **alt attribute** provides a text replacement, typically a text description, of the image. The browser reserves the correct amount of space for your image if you use the **height** and **width** attributes with values either equal to or approximately the size of the image. Provide accurate values for the height and width of the image to retain the image's **aspect ratio** which is the proportional relationship between the width and height of an image. The image could be skewed or distorted by the browser if you provide inaccurate values for the image height and/or width. Table 4.3 lists **** tag attributes and their values. Commonly used attributes are shown in bold.

Table 4.3 Attributes of the `` tag

Attribute	Value
align	right, left (default), top, middle, bottom; obsolete—use the CSS float or position property instead (see Chapter 6)
alt	Text phrase that describes the image
border	Image border size in pixels; <code>border="0"</code> prevents the border of an image hyperlink from being displayed; obsolete—use the CSS border property instead
height	Height of image in pixels
hspace	Amount of space, in pixels, that is blank to the left and right of the image; obsolete—use the CSS padding property instead
id	Text name—alphanumeric, beginning with a letter, no spaces; the value must be unique and not used for other id values on the same web page document
longdesc	URL of a resource that contains an accessible description of a complex image
name	Text name—alphanumeric, beginning with a letter, no spaces; this attribute names the image so that it can be easily accessed by client-side scripting languages such as JavaScript; obsolete—use the id attribute
src	The URL or file name of the image
srcset	New HTML 5.1 attribute that supports the browser display of responsive images (see Chapter 7)
title	A text phrase containing advisory information about the image; typically more descriptive than the alt text
vspace	Amount of space, in pixels, that is blank above and below the image; obsolete—use the CSS padding property instead
width	Width of image in pixels

Notice that several attributes in Table 4.3 are marked as obsolete. Although obsolete in HTML5, they are still valid in XHTML so you'll see them coded in existing web pages. As you work through this book, you'll use CSS to re-create the functions of these now-obsolete attributes.

Accessibility and Images



Focus on Accessibility

Use the `alt` attribute to provide accessibility. Recall from Chapter 1 that Section 508 of the Rehabilitation Act requires the use of accessibility features for new information technology (including websites) associated with the federal government. The `alt` attribute configures an alternative text description of the image. This alt text is used by the browser in two ways. The browser will display the alt text in the image area before the graphic is downloaded and displayed. Some browsers will also display the alt text as a tool tip whenever a visitor to the web page places the mouse cursor over the image area. Applications such as screen readers will read the text in the `alt` attribute out loud. A mobile browser may display the alt text instead of the image.

Standard browsers such as Internet Explorer and Safari are not the only type of application or user agent that can access your website. Major search engines run programs called spiders or robots; these programs index and categorize websites. They cannot process text within images, but some process the value of the `alt` attributes in image tags.

The W3C recommends that alt text be no longer than 100 characters. Avoid using the file name or words like picture, image, and graphic as the value of the `alt` attribute. Instead, use a brief phrase that describes the image. If the purpose of an image, such as a logo, is to display text, then configure the text as the value of the `alt` attribute.



Hands-On Practice 4.3

In this Hands-On Practice you will place a logo graphic and a photograph on a web page. Create a new folder called kayakch4. The images used in this Hands-On Practice are located in the student files chapter4/starters folder. Copy the kayakdc.gif and hero.jpg files into your kayakch4 folder. A starter version of the KayakDoorCounty.net Home page is ready for you in the student files. Copy the chapter4/starter2.html file into your kayakch4 folder. Rename the file as index.html. When you complete this Hands-On Practice, your page will look similar to the one shown in Figure 4.10—with two images. Launch a text editor and open the index.html file.

1. Delete the text contained between the `h1` opening and closing tags. Code an `` tag for kayakdc.gif in this area. Remember to include the `src`, `alt`, `height`, and `width` attributes. Sample code follows:

```

```

2. Code an image tag to display the hero.jpg image below the `h2` element. The image is 500 pixels wide and 350 pixels high. Configure appropriate alt text for the image.
3. Save your page in the kayakch4 folder. Launch a browser and test your page. It should look similar to the one shown in Figure 4.10.



Figure 4.10 A web page with images.

Note: If the images did not display on your web page, verify that you have saved the files inside the kayakch4 folder and that you have spelled the file names correctly in the `` tags. The student files contain a sample solution in the chapter4/4.3 folder. Isn't it interesting how images can add visual interest to a web page?



FAQ What if I don't know the height and width of an image?

Most graphics applications can display the height and width of an image. If you have a graphics application such as Adobe Photoshop, Adobe Fireworks, or GIMP handy, launch the application and open the image. These applications include options that will display the properties of the image, such as height and width.

If you don't have a graphics application available, you can determine the dimensions of an image by using a browser. Display the image on a web page. Right-click on the image to display the context-sensitive menu. Select a menu option such as Properties or "View Image Info" to view the dimensions (height and width) of the image. (*Warning:* If the height and width are specified on the web page, those values will be displayed even if the image's actual height and width are different.)

Image Hyperlinks

Writing the code to make an image function as a hyperlink is very easy. To create an **image link** all you need to do is surround your `` tag with anchor tags. For example, to place a link around an image called `home.gif`, use the following code:

```
<a href="index.html"></a>
```

A thumbnail image link is a small image configured as an image link with an href attribute value that points to another image file instead of to a web page. For example,

```
<a href="sunset.jpg"></a>
```

To see this in action, launch a browser and view chapter4/thumb.html in the student files.



Hands-On Practice 4.4

You will add image links to the KayakDoorCounty.net Home page in this Hands-On Practice. You should already have the index.html, kayakdc.gif, and hero.jpg files in your kayakch4 folder. The new graphics used in this Hands-On Practice are located in the student files in the chapter4/starters folder. Copy the home.gif, tours.gif, reservations.gif, and contact.gif files into your kayakch4 folder. View Figure 4.11 to see how your page should look after you are done with this Hands-On Practice.

Let's get started. Launch a text editor and open index.html. Notice that the anchor tags are already coded—you'll just need to convert the text links to image links!

1. Whenever the main navigation consists of media, such as an image, some individuals may not be able to see the images (or may have images turned off in their browser). To provide navigation that is accessible to all, configure a set of plain text navigation links in the page footer area. Copy the <nav> element containing the navigation area to the lower portion of the page and paste it within the footer element, above the copyright line.
2. Locate the style tags in the head section and code the following style rules:
 - a. Configure a green background color for an id named bar:


```
#bar { background-color: #152420; }
```
 - b. Some browsers display a border around image links by default. Prevent the border display by setting the border property to none for the img element selector:


```
img { border: none; }
```
3. Now, focus on the top navigation area. Code id="bar" on the opening nav tag. Next, replace the text contained between each pair of anchor tags with an image

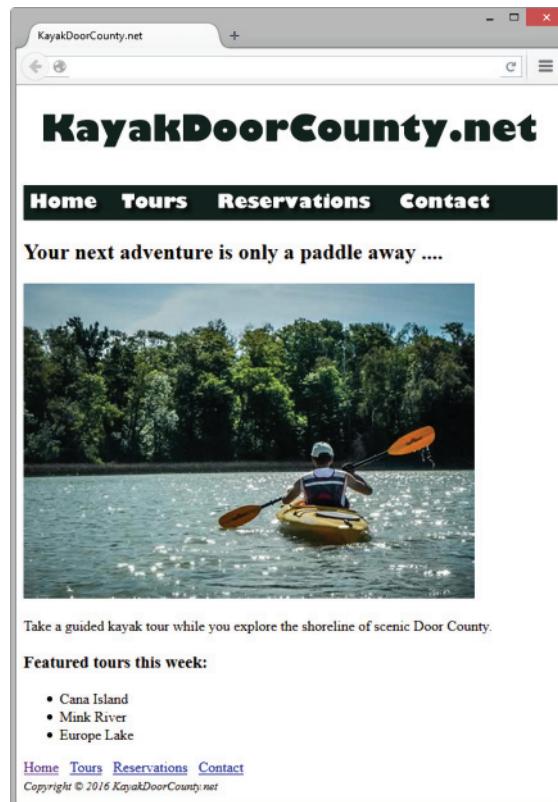


Figure 4.11 The new Home page navigation with image links.

navigation that is accessible to all, configure a set of plain text navigation links in the page footer area. Copy the <nav> element containing the navigation area to the lower portion of the page and paste it within the footer element, above the copyright line.

2. Locate the style tags in the head section and code the following style rules:
 - a. Configure a green background color for an id named bar:


```
#bar { background-color: #152420; }
```
 - b. Some browsers display a border around image links by default. Prevent the border display by setting the border property to none for the img element selector:


```
img { border: none; }
```
3. Now, focus on the top navigation area. Code id="bar" on the opening nav tag. Next, replace the text contained between each pair of anchor tags with an image

element. Use home.gif for the link to index.html, tours.gif for the link to tours.html, reservations.gif for the link to reservations.html, and contact.gif for the link to contact.html. Be careful not to leave any extra spaces between the img tag and the opening and closing anchor tags. A sample follows:

```
<a href="index.html"></a>
```

As you code the img tags be mindful of the width of each image: home.jpg (90 pixels), tours.jpg (90 pixels), reservations.jpg (190 pixels), and contact.jpg (130 pixels).

4. Save your page as index.html. Launch a browser and test your page. It should look similar to the one shown in Figure 4.11.

The student files contain a sample solution in the chapter4/4.4 folder.

Accessibility and Image Hyperlinks

When using an image for main navigation, there are two methods to provide for accessibility:

1. Add a row of plain text navigation hyperlinks in the page footer. These won't be noticed by most people but could be helpful to a person using a screen reader to visit your web page.
2. Configure the alt attribute for each image to contain the exact text that displays in the image. For example, code alt="Home" in the tag for the Home button.



**Focus on
Accessibility**



FAQ What if my images don't display?

The following are common reasons for an image to not display on a web page:

- Is your image *really* in the website folder? Use Windows Explorer or the Mac Finder to double check.
- Did you code the HTML and CSS correctly? Perform W3C CSS and HTML validation testing to find syntax errors that could prevent the image from displaying.
- Does your image have the exact file name that you have used in the CSS or HTML code? Attention to detail and consistency will be very helpful here.



FAQ How should I name my image files?

Guidelines for naming image files:

- Use all lowercase letters.
- Do not use punctuation symbols and spaces.
- Do not change the file extensions (should be .gif, .jpg, .jpeg, or .png).
- Keep your file names short, but descriptive. Here are some examples:
i1.gif is probably too short.
myimagewithmydogonmybirthday.gif is too long.
dogbdy.gif may be just about right.

4.4 HTML5 Visual Elements

You'll explore configuring images with captions in this section. In the next Hands-On Practice you'll configure an image and a caption using a div element as a container. Next, you'll explore an approach to configure a image with a caption that implements new HTML5 figure and figcaption elements.



Hands-On Practice 4.5

In this Hands-On Practice you will configure an image with a caption on a web page. The photo used in this Hands-On Practice is located in the student files chapter4/starters folder. Save the myisland.jpg file in a folder named mycaption.

Step 1: 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 an image tag to the body section to display the myisland.jpg image as follows:

```

```

Save the file as index.html in the mycaption folder. Launch a browser to test your page. It should look similar to the page shown in Figure 4.12.



Figure 4.12 The image is displayed on the web page.
Screenshots of Mozilla Firefox.
Courtesy of Mozilla Foundation.

Step 2: Configure a figure caption and border for the image. To do so, launch a text editor and open the web page file. Add embedded CSS to the head section that configures an id named figure that is 640 pixels wide, has a border, has padding set to 5px, and has centered text using the Papyrus font typeface (or the default fantasy family font). The code follows:

```
<style>
#figure { width: 640px;
           border: 1px solid #000000;
           padding: 5px;
           text-align: center;
           font-family: Papyrus, fantasy; }
</style>
```

Edit the body section to add a div to contain the image. Add the text “Tropical Island Getaway” below the image but within the div element. Assign the div to the id named `figure`. Save the file as index.html in the mycaption folder. Launch a browser to test your page. It should look similar to the page shown in Figure 4.13. The student files contains a sample solution in the chapter4/4.5 folder.

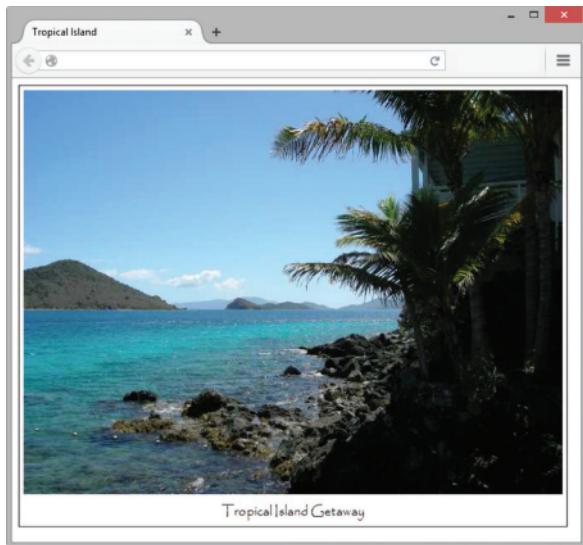


Figure 4.13 CSS configures the placement of the border and figure caption. Screenshots from Microsoft® Excel®. Used by permission of Microsoft® Corporation

HTML5 Figure and Figcaption Elements

The block display **figure element** comprises a unit of content that is self-contained, such as an image, along with one optional **figcaption element**.

The block display **figcaption element** provides a caption for a figure. The figure and figcaption elements require a modern browser that supports HTML5 such as Safari, Firefox, Chrome, Opera, Microsoft Edge, or Internet Explorer (version 9 or later).

You might be wondering why these new HTML5 elements were created when the same design can be configured using a div element as a container. The reason is semantics. The div element is useful but very generic in nature. When the figure and figcaption elements are used, the structure of the content is well defined.



Hands-On Practice 4.6

In this Hands-On Practice you will configure an area on a web page that contains an image with a caption by using the HTML5 figure and figcaption elements. The graphic used in this Hands-On Practice is located in the student files chapter4/starters folder. Save the `myisland.jpg` file in a folder named `mycaption2`.

Step 1: 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 an image tag to the body section to display the `myisland.jpg` image as follows:

```

```

Save the file as index.html in the mycaption2 folder. Launch a browser to test your page. It should look similar to the page shown in Figure 4.12.

Step 2: Configure a figure caption and border for the image. Launch a text editor and open the web page file. Add embedded CSS to the head section that configures the figure element selector to be 640 pixels wide, with a border, and with padding set to 5px. Configure the figcaption element selector to have centered text using the Papyrus font typeface (or the default fantasy family font). The code follows:

```
<style>
figure { width: 640px; padding: 5px;
          border: 1px solid #000000; }
figcaption { text-align: center;
            font-family: Papyrus, fantasy; }
</style>
```

Edit the body section. Below the image, add a `figcaption` element that contains the following text: “Tropical Island Getaway.” Configure a `figure` element that contains both the image and the `figcaption`. The code follows:

```
<figure>
  
  <figcaption> Tropical Island Getaway</figcaption>
</figure>
```

Save the file as index.html in the mycaption2 folder. Launch a browser to test your page. It should look similar to the page shown in Figure 4.14. The student files contains a sample solution in the chapter4/4.6 folder.

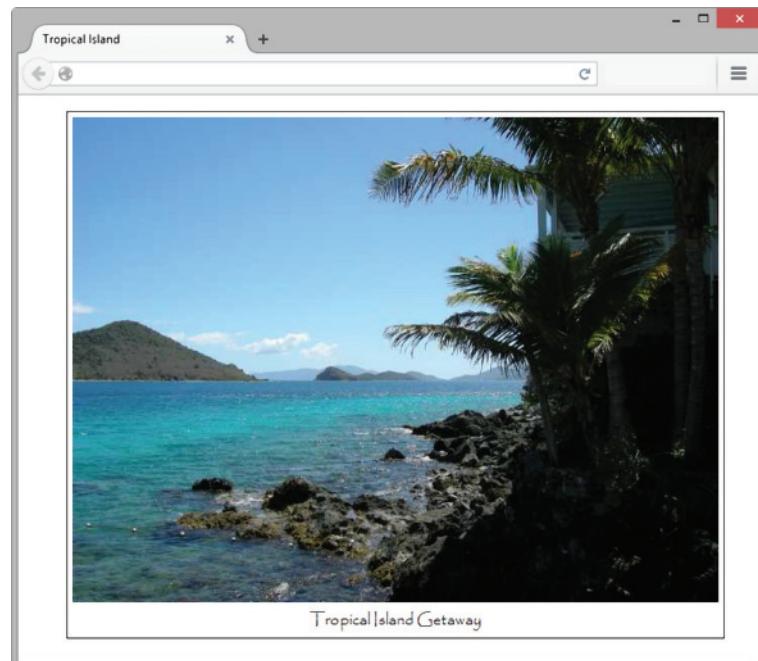


Figure 4.14 The HTML5 figure and figcaption elements were used in this web page

HTML5 Meter Element

The **meter** element displays a visual gauge of a numeric value within a known range, typically as part of a bar chart. At the time this was written, this new HTML5 element was not supported in Internet Explorer. The meter element is configured with several attributes, including `value` (the value displayed), `min` (the lowest value in the range), and `max` (the highest possible value in the range). The following code snippet (student files chapter4/meter.html) configures the display of a report that shows total visits and the number of visits by users for each browser:

```
<h1>Monthly Browser Report</h1>
<meter value="14417" min="0" max="14417">14417</meter>14,417 Total
Visits<br>
<meter value="7000" min="0" max="14417">7000</meter> 7,000 Firefox<br>
<meter value="3800" min="0" max="14417">3800</meter> 3,800 Internet
Explorer<br>
<meter value="2062" min="0" max="14417">2062</meter> 2,062 Chrome<br>
<meter value="1043" min="0" max="14417">1043</meter> 1,043 Safari<br>
<meter value="312" min="0" max="14417">312</meter> 312 Opera<br>
<meter value="200" min="0" max="14417">200</meter> 200 other<br>
```

As shown in Figure 4.15, the meter element provides a handy way to display a bar chart on a web page. Visit <http://caniuse.com> to get current information on browser support of this element.

HTML5 Progress Element

The **progress** element displays a bar that depicts a numeric value within a specified range. At the time was written, this new HTML5 element was supported by modern browsers, including Internet Explorer 10. The progress element is configured with the `value` (the value displayed) and `max` (highest possible value) attributes. Place information for nonsupporting browsers to display between the opening and closing progress tags. The following code snippet (student files chapter4/progress.html) shows 50% completion of a task:

```
<h1>Progress Report</h1>
<progress value="5" max="10">50%</progress>
Progress Toward Our Goal
```

Figure 4.16 displays the page in the Firefox browser. Visit <http://caniuse.com> to get current information on browser support of HTML5 elements.

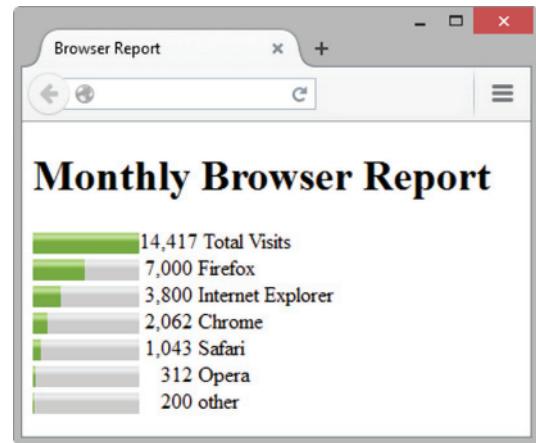


Figure 4.15 The `meter` element

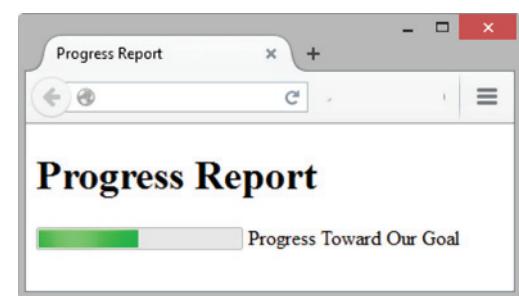


Figure 4.16 The `progress` element

4.5 Background Images

Back in Chapter 3, you learned how to configure background color with the CSS `background-color` property. For example, the following CSS code configures the background of a web page to be a soft yellow:

```
body { background-color: #ffff99; }
```

The `background-image` Property



CSS Background Images

Use the CSS **`background-image` property** to configure a background image. The following CSS code configures the HTML body selector with a background of the graphic `texture1.png` located in the same folder as the web page file:

```
body { background-image: url(texture1.png); }
```

Using Both Background Color and a Background Image

You can configure both a background color and a background image. The background color (specified by the `background-color` property) will display first. Next, the image specified as the background will be displayed as it is loaded by the browser.

By coding both a background color and a background image, you provide your visitor with a more pleasing visual experience. If the background image does not load for some reason, the background color will still have the expected contrast with your text color. If the background image is smaller than the web browser window and the web page is configured with CSS not to automatically tile (repeat the image), the background color of the page will display in areas not covered by the background image. The CSS for a page with both a background color and a background image follows:

```
body { background-color: #99cccc;
       background-image: url(background.jpg); }
```

Browser Display of a Background Image

You may think that a graphic created to be the background of a web page would always be about the size of the browser window viewport. However, the dimensions of the background image are often much smaller than the typical viewport. The shape of a background image is often either a thin rectangle or a small rectangular block. Unless otherwise specified in a style rule, browsers repeat, or tile, these images to cover the page's background, as shown in Figures 4.17 and 4.18. The images have small file sizes so that they download as quickly as possible.

Background Image

Web Page with Background Image



Figure 4.17 A long, thin background image tiles down the page. Screenshots of Mozilla Firefox. Courtesy of Mozilla Foundation

Background Image

Web Page with Background Image

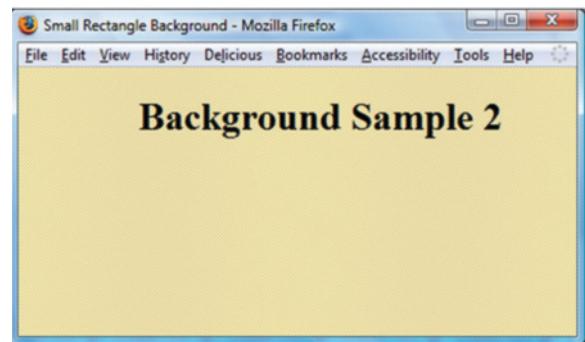


Figure 4.18 A small rectangular background is repeated to fill the web page window. Screenshots of Mozilla Firefox. Courtesy of Mozilla Foundation

The `background-repeat` Property

As just discussed, the default behavior of a browser is to repeat, or tile, background images to cover the entire element's background. This behavior also applies to other elements, such as backgrounds for headings, paragraphs, and so on. You can modify this tiling behavior with the CSS **background-repeat** property. The values for the `background-repeat` property include `repeat` (default), `repeat-y` (vertical repeat), `repeat-x` (horizontal repeat), and `no-repeat` (no repeat). Figure 4.19 provides examples of the actual background image and the result of applying various `background-repeat` property values.

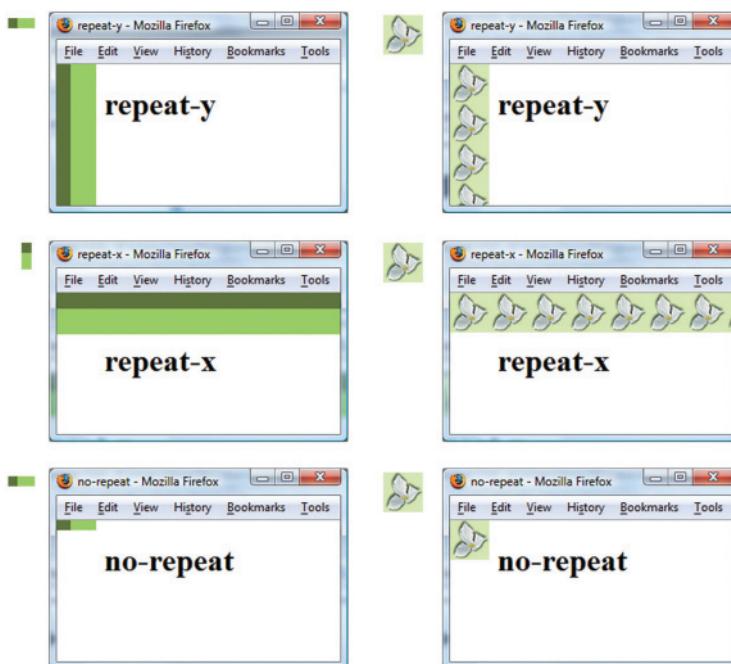


Figure 4.19 Examples of the CSS `background-repeat` property. Screenshots of Mozilla Firefox. Courtesy of Mozilla Foundation

CSS3 provides for additional values for the `background-repeat` property that are not yet well supported by browsers:

- `background-repeat: space;`
- Repeats the image in the background without clipping (or cutting off) parts of the image by adjusting empty space around the repeated images.
- `background-repeat: round;`

Repeats the image in the background and scales (adjusts) the dimensions of the image to avoid clipping.



Hands-On Practice 4.7

Let's practice using a background image. In this exercise, you will use your files in the kayakch4 folder from Hands-On Practice 4.4 (also see the student files chapter4/4.4 folder) as a starting point. In this Hands-On Practice you will configure the main element selector with a background image that does not repeat. Obtain the heroback.jpg image from the student files chapter4/starters folder. Copy the image into your kayakch4 folder. When you have completed this exercise, your page should look similar to the one shown in Figure 4.20. Launch a text editor and open index.html.

1. Locate the style tags in the head section. Code a new style rule for the main element selector to configure the `background-image` and `background-repeat` properties. Set the background image to be heroback.jpg. Set the background not to repeat. The main element selector style rules follow:

```
main { background-image: url(heroback.jpg);
background-repeat: no-repeat; }
```

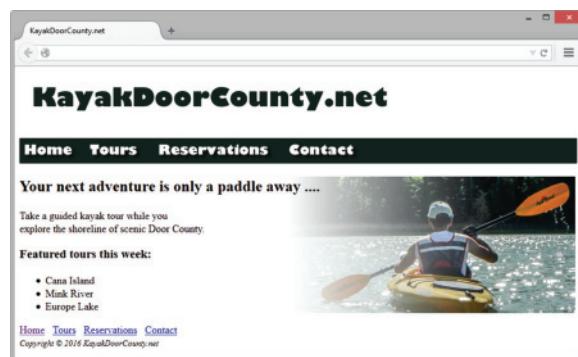


Figure 4.20 The background image in the `<main>` area is configured with `background-repeat: no-repeat`.

2. Remove the `img` tag that displays the hero.jpg image from the body of the web page.
3. Save your page as index.html. Launch a Firefox or Chrome browser, and test your page. You may notice that the text within the main element is displayed over the background image. In this case, the page would look more appealing if the paragraph did not extend across the background image. Open index.html in a text editor and code a line break tag before the word "explore".

4. Save and test your page again in Firefox or Chrome. It should look similar to the page shown in Figure 4.20. The student files contain a sample solution in the chapter4/4.7 folder. At the time this was written, Internet Explorer did not support default styles for the HTML5 main element. You may need to nudge this browser to comply by adding the `display: block;` declaration (see Chapter 6) to the styles for the main element selector. An example solution is in the student files (chapter4/4.7/iefix.html).



FAQ What if my images are in their own folder?

It's a good idea to organize your website by placing all your images in a folder. Notice that the CircleSoft website whose file structure is shown in Figure 4.21 contains a folder called images, which contains GIF and JPEG files. To refer to these files in code, you also need to refer to the images folder. The following are some examples:

- The CSS code to configure the background.gif file from the images folder as the page background follows:

```
body { background-image:  
       url(images/background.gif); }
```

- The HTML to display the logo.jpg file from the images folder follows:

```

```

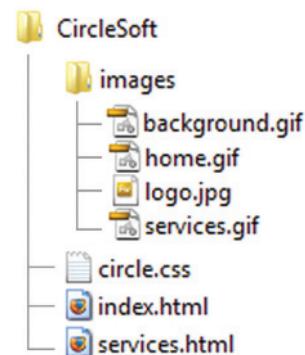


Figure 4.21 A folder named “images” contains the graphic files

The `background-position` Property

You can specify other locations for the background image besides the default top left location by using the **background-position** property. Valid values for the background-position property include percentages; pixel values; or left, top, center, bottom, and right. The first value indicates horizontal position. The second value indicates vertical position. If only one value is provided, the second value defaults to center. In Figure 4.22, the small flower image has been placed in the background on the right side of the element by using the following style rule:

```
h2 { background-image: url(flower.gif);  
     background-position: right;  
     background-repeat: no-repeat; }
```

New Media and Web Design



Figure 4.22 The flower background image was configured to display on the right side with CSS

The background-attachment Property

Use the **background-attachment** property to configure whether the background image remains fixed in place or scrolls along with the page in the browser viewport. Valid values for the background-attachment property include `fixed` and `scroll` (the default).



Checkpoint 4.2

1. Describe the CSS to configure a graphic named circle.jpg to display once in the background of all `<h1>` elements. Code the CSS.
2. Describe the CSS that configures a file named bg.gif to repeat vertically down the background of a web page. Code the CSS.
3. Explain how the browser will render the web page if you use CSS to configure both a background image and a background color.

4.6 More About Images

This section introduces several additional techniques used with images on web pages. Topics discussed include image maps, the favorites icon, image slicing, and CSS Sprites.

Image Maps

An **image map** is an image that can be used as one or more hyperlinks. An image map will typically have multiple clickable or selectable areas that link to another web page or website. The selectable areas are called **hotspots**. Image maps can configure selectable areas in three shapes: rectangles, circles, and polygons. An image map requires the use of the `image` element, `map` element, and one or more `area` elements.

Map Element

The **map element** is a container tag that indicates the beginning and ending of the image map description. The `name` attribute is coded to associate the `<map>` tag with its corresponding image. The `id` attribute must have the same value as the `name` attribute. To associate a `map` element with an image, configure the `image` tag with the **usemap attribute** to indicate which `<map>` to use.

Area Element

The **area element** defines the coordinates or edges of the clickable area. It is a void tag that uses the `href`, `alt`, `title`, `shape`, and `coords` attributes. The `href` attribute identifies the web page to display when the area is clicked. The `alt` attribute provides a text description for screen readers. Use the `title` attribute to specify text that some browsers may display as a tooltip when the mouse is placed over the area. The `coords` attribute indicates the coordinate position of the clickable area. Table 4.4 describes the type of coordinates needed for each `shape` attribute value.

Table 4.4 Shape coordinates

Shape	Coordinates	Meaning
rect	"x1,y1, x2,y2"	The coordinates at point (x1,y1) represent the upper-left corner of the rectangle. The coordinates at point (x2,y2) represent the lower-right corner of the rectangle.
circle	"x,y,r"	The coordinates at point (x,y) indicate the center of the circle. The value of r is the radius of the circle, in pixels.
polygon	"x1,y1, x2,y2, x3,y3", etc.	The values of each (x,y) pair represent the coordinates of a corner point of the polygon.

Exploring a Rectangular Image Map

We'll focus on a rectangular image map. For a rectangular image map, the value of the shape attribute is `rect`, and the coordinates indicate the pixel positions as follows:

- distance of the upper-left corner from the left edge of the image
- distance of the upper-left corner from the top of the image
- distance of the lower-right corner from the left edge of the image
- distance of the lower-right corner from the top of the image.

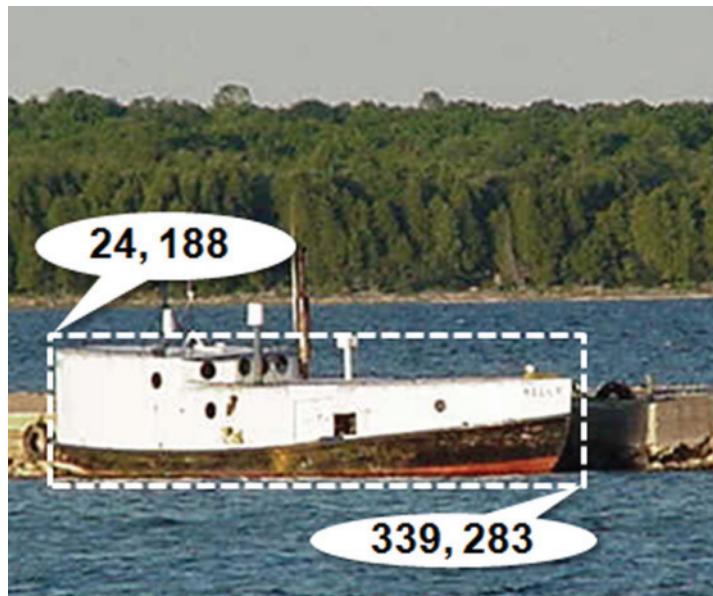
**Figure 4.23** Sample image map

Figure 4.23 shows an image of a fishing boat. The dotted rectangle around the fishing boat indicates the location of the hotspot. The coordinates shown (24, 188) indicate that the top-left corner is 24 pixels from the left edge of the image and 188 pixels from the top of the image. The pair of coordinates in the lower-right corner (339, 283) indicates that this corner is 339 pixels from the left edge of the image and 283 pixels from the top of

the image. This example is in the student files at chapter4/map.html. The HTML code to create this image map follows:



**Focus on
Accessibility**

```
<map name="boat" id="boat">
  <area href="http://www.fishingdoorcounty.com" shape="rect"
        coords="24, 188, 339, 283" alt="Door County Fishing Boat">
</map>

```

Note the use of the `alt` attribute on the `area` element in the previous code sample. Configure a descriptive `alt` attribute for each `area` element associated with an image map to provide for accessibility.

Most web developers do not hand-code image maps. Web authoring tools, such as Adobe Dreamweaver, have features that help you to generate image maps. There are also free online image map generators available at:

- <http://www.maschek.hu/imagemap/imgmap>
- <http://image-maps.com>
- http://mobilefish.com/services/image_map/image_map.php

The Favorites Icon

Ever wonder about the small icon you sometimes see in the address bar or tab of a browser? That's a favorites icon, usually referred to as a **favicon**, which is a square image (either 16×16 pixels or 32×32 pixels) associated with a web page. The favicon, shown in Figure 4.24, may display in the browser's address bar, tab, or the favorites and bookmarks lists.



Figure 4.24 The favorites icon displays in the browser tab and address bar

Configuring a Favorites Icon

While some versions of Internet Explorer (version 6 and earlier) expected the file to be named `favicon.ico` and to reside in the root directory of the web server, a more modern approach is to associate the `favicon.ico` file with a web page by using the `link` element. Recall that in Chapter 3, you coded the `<link>` tag in the head section of a web page to associate an external style sheet file with a web page file. You can also use the `<link>` tag to associate a favorites icon with a web page. Three attributes are used to associate a web page with a favorites icon: `rel`, `href`, and `type`. The value of the `rel` attribute is `icon`. The value of the `href` attribute is the name of the image file. The value of the `type` attribute describes the MIME type of the image—which defaults to `image/x-icon`.

for .ico files. The HTML code to associate a favorites icon named favicon.ico to a web page follows:

```
<link rel="icon" href="favicon.ico" type="image/x-icon">
```

Be aware that Internet Explorer's support of the favorites icon is somewhat buggy. You may need to publish your files to the Web (see the FTP tutorial in the Appendix) in order for the favicon to display even in current versions of Internet Explorer. Other browsers, such as Firefox, Safari, Google Chrome, and Opera, display favicons more reliably and also support GIF and PNG image formats.



Hands-On Practice 4.8

Let's practice using a favorites icon. Obtain the favicon.ico file from the student files in the chapter4/starters folder. In this exercise, you will use your files in the kayakch4 folder from Hands-On Practice 4.7 (also see the student files chapter4/4.7 folder) as a starting point.

1. Launch a text editor, and open index.html. Add the following link tag to the head section of the web page:

```
<link rel="icon" href="favicon.ico" type="image/x-icon">
```

2. Save your file. Launch the Firefox browser, and test your page. You should notice the small kayaker image in the Firefox browser tab as shown in Figure 4.25. The student files contain a sample solution in the chapter4/4.8 folder.



Figure 4.25 The favorites icon displays in the Firefox browser tab



FAQ How can I create my own favorites icon?

You can create your own favicon with a graphics application, such as GIMP, Adobe Fireworks, or with one of the following online tools:

- <http://favicon.cc>
- <http://www.favicongenerator.com>
- <http://www.freelfav.com>
- <http://www.xiconeditor.com>

Image Slicing

Graphic artists and designers can create complex web page images. Sometimes, parts of these images are better optimized as GIFs than as JPEGs, while other parts of the images may be better optimized as JPEGs than as GIFs. By **image slicing** the single, complex images into multiple, smaller images, you can optimize all portions for the most efficient display. In addition, there may be times when you plan special mouse rollover effects for parts of a large, complex image. In this case, parts of the image need to be individually accessible to scripting languages, and so the image needs to be sliced. When an image is sliced, it is broken into multiple graphic files. Most graphics applications, such as Adobe Fireworks and Adobe Photoshop, have features for image slicing that automatically create the HTML for you.

CSS Sprites

A modern technique to optimize the use of images on web pages is called CSS Sprites. A **sprite** is an image file that contains multiple small graphics that are configured as background images for various web page elements. The CSS `background-image`, `background-repeat`, and `background-position` properties are used to manipulate the placement of the image. Having just a single image saves download time, because the browser needs to make only one http request for the combined image instead of many requests for the individual smaller images. You'll work with CSS Sprites in Chapter 7.

4.7 Sources and Guidelines for Graphics

Sources of Graphics

There are many ways to obtain graphics: You can create them using a graphics application, download them from a website providing them for free, purchase and download them from a graphics website, purchase a graphics collection on a DVD, take digital photographs, scan photographs, scan drawings, or hire a graphic designer to create graphics for you. Popular graphic applications include Adobe Photoshop and Adobe Fireworks. Popular free graphics applications include GIMP (<http://gimp.org>), Google's Picasa (<http://picasa.google.com>), and the Pixlr web application (<http://pixlr.com/editor>). These applications usually include tutorials and sample images to help you get started. Visit the textbook's website at <http://webdevfoundations.net/8e/chapter4.html> for tutorials on using Adobe Fireworks and Adobe Photoshop to create a logo banner image.

Sometimes you might be tempted to right-click on an image on a web page and download it for use on your own website. Be aware that materials on a website are copyrighted (even if a copyright symbol or notice does not appear) and are not free to use unless the owner of the site permits it. So, contact the owner of an image and request permission for use rather than just taking it. If you're using Flickr (<http://flickr.com>) to search for images, select the advanced-search page and check "*Only search within Creative Commons-licensed content.*" Be sure to follow the instructions for attribution when indicated.



Focus on
Ethics

There are many web sites that offer free and low-cost graphics. Choose a search engine and search for “free graphics”—you’ll get more results than you have time to view. The following are a few sites that you may find helpful when looking for images:

- Free Images: <http://www.freeimages.com>
- Free Stock Photo Search Engine: <http://www.everystockphoto.com>
- Free Digital Photos: <http://www.freedigitalphotos.net>
- The Stock Solution: <http://www.tssphoto.com>
- Pixabay: <http://pixabay.com>
- iStockphoto: <http://www.istockphoto.com>

It is also possible to create a banner or button image online. There are a number of sites that offer this feature. Some include advertising with your free image, some offer paid memberships, and others are simply free. Search for “create free online banner” to find sites offering this service. The following are a few useful sites for creating banners and button images:

- LogoCreator: <http://creatr.cc/creatr>
- Cool Text <http://www.cooltext.com>
- Da Button Factory: <http://dabuttonfactory.com>

Guidelines for Using Images

Images enhance your web page by creating an engaging, interesting user experience. Images can also hurt your web page by slowing down its performance to a crawl and discouraging visitors. This section explores some guidelines for using images on web pages.

Reuse Images

Once an image from your site is requested for a web page, it is stored in the cache on your visitor’s hard drive. Subsequent requests for the image will use the file from the hard drive instead of another download. This approach results in faster page loads for all pages that use the image. It is recommended that you reuse common graphics such as logos and navigation buttons on multiple pages instead of creating different versions of these common graphics.

Consider the Size vs. Quality Issue

You can choose among varying levels of image quality when using a graphics application to create or optimize an image. There is a correlation between the quality of the image and the size of the image file: The higher the quality, the larger the file size will be. Choose the smallest file that gives you appropriate quality. You may need to experiment until you get the right match.

Consider Image Load Time

Be careful when using images on web pages—it takes time for them to load. Optimize the file size and the dimensions of images for efficient web page display.

Use Appropriate Resolution

Most desktop and laptop web browsers display images at relatively low **resolution**—72ppi (pixels per inch) or 96ppi. Many digital cameras and scanners can create images with much higher resolution. Of course, higher resolution means larger file size. Even though the browser does not display the depth of resolution, more bandwidth is still used for the large file size. Be careful when taking digital photographs or scanning images. Use a resolution setting appropriate for web pages. A one-inch image saved at 150ppi could appear close to two inches wide on a 72ppi monitor. Be aware that some devices (such as tablets and smartphones) have high pixel density displays which can affect the rendering of an image. In Chapter 7, you'll be introduced to configuring flexible, responsive images for multiple devices.

Specify Dimensions

Use accurate height and width attributes on image tags. This will allow the browser to allocate the appropriate space on the web page for the image and load the page faster. Do not try to resize the appearance of an image by modifying the settings of the height and width attributes. While this approach will work, your page will load more slowly, and your image quality may suffer. Instead, use a graphics application to create a smaller or larger version of the graphic when needed.

Be Aware of Brightness and Contrast

Gamma refers to the brightness and contrast of the monitor display. Monitors used with Macintosh and Windows operating systems use a different default gamma setting (Macintosh uses 1.8; Windows uses 2.2). Images that have good contrast on a computer running Windows may look slightly washed out on a Macintosh. Images created on a Macintosh may look darker, with less contrast, when displayed on a computer with a Windows operating system. Be aware that even monitors on the same operating system may have slightly different gamma values than the default for the platform. A web developer cannot control gamma, but should be aware that images will look different on various platforms because of this issue.

Accessibility and Visual Elements



Focus on
Accessibility

Even though images help to create a compelling, interesting website, remember that not all your visitors will be able to view your images. The Web Accessibility Initiative's WCAG 2.0 includes a number of guidelines for web developers in the use of color and images:

- Don't rely on color alone. Some visitors may have color perception deficiencies. Use high contrast between background and text color.
- Provide a text equivalent for every nontext element. Use the `alt` attribute on your image tags.
 - If an image displays text, configure that text as the value of the `alt` attribute.
 - Use `alt=""` for an image that is purely decorative.
- If your site navigation uses image hyperlinks, provide simple text links at the bottom of the page.

Vinton Cerf, the co-inventor of TCP/IP and the former chairman of the Internet Society, said, "The Internet is for everyone." Follow web accessibility guidelines to ensure that this is true.



Checkpoint 4.3

1. Search for a site that uses image hyperlinks to provide navigation. List the URL of the page. What colors are used on the image links? If the image links contain text, is there good contrast between the background color and the letters on the image links? Would the page be accessible to a visitor who is sight challenged? How have accessibility issues been addressed? Is the `alt` attribute used to describe the image link? Is there a row of text links in the footer section of the page? Answer these questions and discuss your findings.
2. When configuring an image map, describe the relationship between the image, map, and area tags.
3. True or False: You should save your images using the smallest file size possible.

4.8 CSS3 Visual Effects

This section introduces new CSS3 properties that provide visual effects on web pages, including background clipping and scaling, multiple background images, rounded corners, box shadows, text shadows, opacity effects, transparent color with RGBA, transparent color with HSLA, and gradients.

The CSS3 `background-clip` Property

The new CSS3 **`background-clip` property** confines the display of the background image with the following values:

- `content-box` (clips the display to the area behind the content)
- `padding-box` (clips the display to the area behind the content and padding)
- `border-box` (default; clips the display to the area behind the content, padding, and border; similar to the `padding-box` property except that the image will display behind a border configured to be transparent)

The `background-clip` property is supported by modern browsers, including Internet Explorer (version 9 and later). Figure 4.26 shows div elements configured with different values of the `background-clip` property. Note that the dashed border is intentionally large in these examples. The student files (chapter4/clip folder) contains an example page.

The CSS is shown as follows:

```
.test { background-image: url(myislandback.jpg);  
        background-clip: content-box;  
        width: 400px;  
        padding: 20px;  
        margin-bottom: 10px;  
        border: 1px dashed #000; }
```

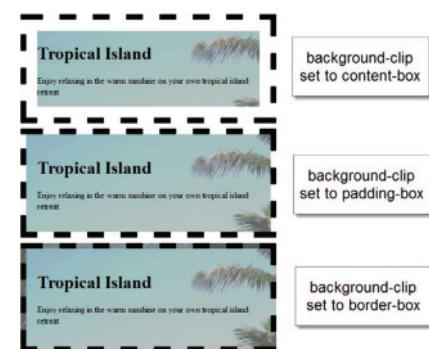


Figure 4.26 The CSS3 `background-clip` property

The CSS3 background-origin Property

The new CSS3 **background-origin property** positions the background image, using the following values:

- **content-box** (positions relative to the content area)
- **padding-box** (default; positions relative to the padding area)
- **border-box** (positions relative to the border area)



Figure 4.27 The CSS3 background-origin property

The **background-origin** property is supported by current versions of modern browsers. Figure 4.27 shows div elements configured with different values of the **background-origin** property. The sample page is located in the student files (chapter4/origin folder).

The CSS for the first div follows:

```
.test { background-image: url(trilliumsolo.jpg);
    background-origin: content-box;
    background-repeat: no-repeat;
    background-position: right top;
    width: 200px; margin-bottom: 10px;
    padding: 20px; border: 1px solid #000; }
```

You may have noticed that it's common to use several CSS properties when configuring background images. The properties typically work together. However, be aware that the **background-origin** property has no effect if the **background-attachment** property is set to the value "fixed".

The CSS3 background-size Property

The CSS3 **background-size property** can be used to resize or scale the background image. The **background-size** property is supported by current versions of modern browsers, including Internet Explorer (version 9 and later). Valid values for the **background-size** property can be:

- a pair of percentage values (width, height)

If only one percentage value is provided, the second value defaults to `auto` and is determined by the browser.

- a pair of pixel values (width, height)

If only one numeric value is provided, the second value defaults to `auto` and is determined by the browser.

- `cover`

The value `cover` will preserve the aspect ratio of the image as it scales the background image to the *smallest size* for which both the height and width of the image can completely cover the area.

- `contain`

The value `contain` will preserve the aspect ratio of the image as it scales the background image to the *largest size* for which both the height and width of the image will fit within the area.

Figure 4.28 shows two div elements that are each configured with the same background image to display without repeating.

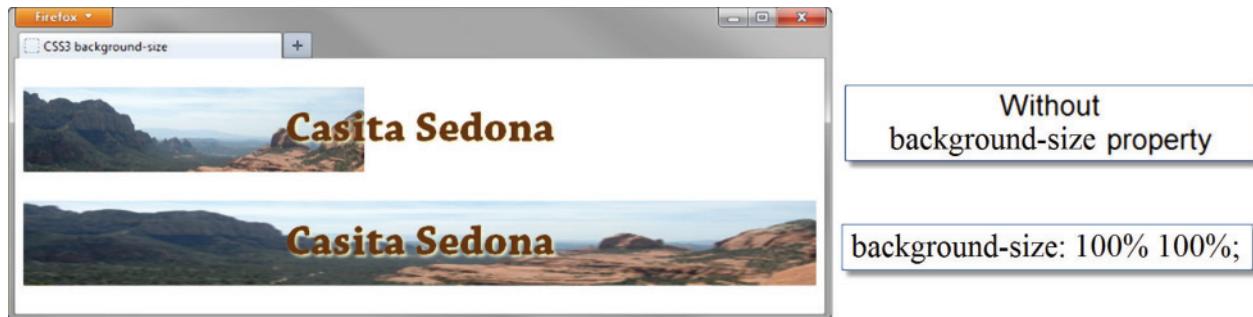


Figure 4.28 Configuring a background image. Screenshots of Mozilla Firefox. Courtesy of Mozilla Foundation

The `background-size` property is not configured for the first div element's background image which only partially fills the space. The CSS for the second div configures the `background-size` to be `100% 100%` so the browser scales and resizes the background image to fill the space. The sample page is located in the student files (chapter4/size/sedona.html). The CSS for the second div follows:

```
#test1 { background-image: url(sedonabackground.jpg);  
         background-repeat: no-repeat;  
         background-size: 100% 100%; }
```

Figure 4.29 demonstrates use of the `cover` and `contain` values to configure the display of a 500×500 background image within a 200 pixel wide area on a web page. The web page on the left uses `background-size: cover;` to scale and resize the image to completely cover the area while keeping the aspect ratio of the image intact. The web page on the

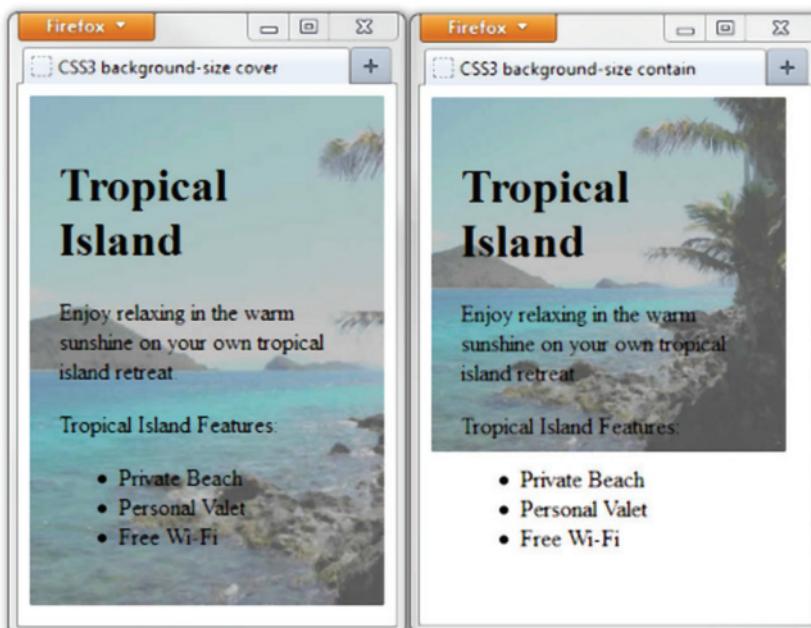


Figure 4.29 Examples of `background-size: cover;` and `background-size: contain`. Screenshots of Mozilla Firefox. Courtesy of Mozilla Foundation

right uses `background-size: contain;` to scale and resize the image so that both the height and width of the image will fit within the area. Review the sample pages in the student files (`chapter4/size/cover.html` and `chapter4/size/contain.html`).

CSS3 Multiple Background Images

Let's explore applying multiple background images to a web page. Although the CSS3 Backgrounds and Borders module is still in candidate recommendation status, current versions of most popular web browsers support the use of multiple background images.

Figure 4.30 shows a web page with two background images configured on the body selector: a green gradient image that repeats across the entire browser viewport, and a flower image that displays once in the right footer area. Use the CSS3 `background` property to configure multiple background images. Each image declaration is separated by a comma. You can optionally add property values to indicate the image's position and whether the image repeats. The `background` property uses a shorthand notation: Just list the values that are needed for relevant properties such as `background-position` and `background-repeat`.

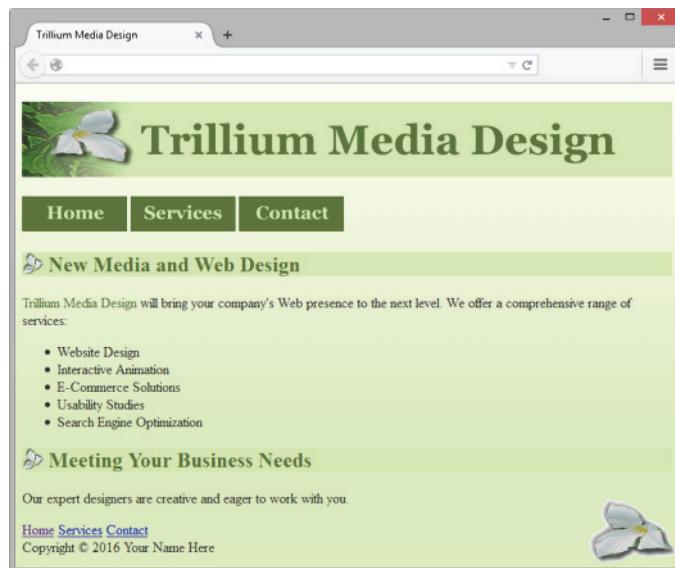


Figure 4.30 The Firefox browser displays multiple background images

Progressive Enhancement

Multiple background images are currently supported by recent versions of Firefox, Chrome, Safari, Opera, and Internet Explorer (version 9 and later). Be aware that multiple background images are not supported by earlier versions of Internet Explorer. You'll use the technique of **progressive enhancement**, which is defined by web developer and HTML5 evangelist Christian Heilmann as “starting with a baseline of usable functionality, then increasing the richness of the user experience step by step by testing for support for enhancements before applying them.” In other words, start with a web page that displays well in most browsers and then add new design techniques, such as multiple background images, in a way that enhances the display for visitors who are using browsers that support the new technique.

To provide for progressive enhancement when using multiple background images, first configure a separate `background-image` property with a single image (rendered by most browsers) before the `background` property with multiple images (rendered by supporting browsers and ignored by nonsupporting browsers).



Hands-On Practice 4.9

Let's practice configuring multiple background images. In this Hands-On Practice you will configure the body element selector to display multiple background images on the web page. Create a new folder named trilliumch4. Copy all the files from the student files chapter4/trillstarters folder into your trilliumch4 folder. You'll update the index.html file. Launch a text editor and open index.html.

1. Modify the style rule for the body element selector. Use the `background-image` property to display `trilliumgradient.png`. This style rule will be applied by browsers that do not support multiple background images. Configure a `background` property to display both the `trilliumfoot.gif` image and the `trilliumgradient.png` image. The `trilliumfoot.gif` image should not repeat and should be displayed in the lower right corner. The body selector style rules are as follows:

```
body { background-color: #f4ffe4; color: #333333;  
      background-image: url(trilliumgradient.png);  
      background: url(trilliumfoot.gif) no-repeat right bottom,  
                 url(trilliumgradient.png); }
```

2. Save your page as `index.html`. Launch a browser and test your page in Firefox, Chrome, Microsoft Edge, or Internet Explorer version 9 or later. Your display should be similar to Figure 4.30.
3. There is usually more than one way to design a web page. Let's consider the placement of the flower image in the footer area of the web page. Why not configure the gradient image as the body element selector background and the flower image as the footer element selector background? This will provide for a similar display on all currently popular browsers. Let's try this out. Edit the `index.html` file. Remove the `background` property from the body element selector. A code sample is

```
body { background-color: #f4ffe4; color: #333333;  
      background-image: url(trilliumgradient.png); }
```

Next, configure the `trilliumfoot.gif` image as the background for the footer element selector. Configure a height value that will be large enough to display the image. The code is

```
footer { background-image: url(trilliumfoot.gif);  
        background-repeat: no-repeat;  
        background-position: right top;  
        height: 75px; }
```

4. Save your page as `index2.html`. Launch a browser and test your page. It should look similar to Figure 4.30 on all popular modern browsers. See the chapter4/4.9 folder in the student files for solutions to this Hands-On Practice.



CSS3 Rounded Corners

As you've worked with borders and the box model, you may have begun to notice a lot of rectangles on your web pages! CSS3 introduces the **border-radius** property, which can be used to create rounded corners and soften up those rectangles. The border-radius property is supported by current versions of major browsers, including Internet Explorer (version 9 and later).

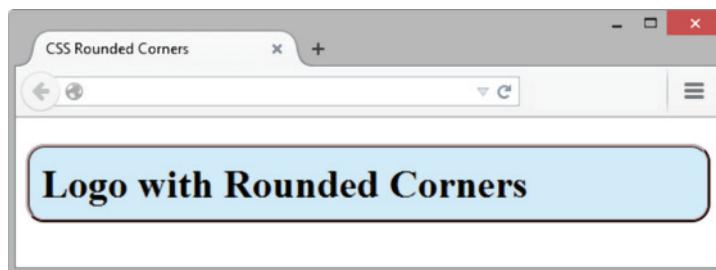
Valid values for the border-radius property include one to four numeric values (using pixel or em units) or percentages that configure the radius of the corner. If a single value is provided, it configures all four corners. If four values are provided the corners are configured in order of top left, top right, bottom right, and bottom left. You can configure corners individually with the border-bottom-left-radius, border-bottom-right-radius, border-top-left-radius, and border-top-right-radius properties.

CSS declarations to set a border with rounded corners are shown in the next segment of code. If you would like a visible border to display, configure the border property. Then set the value of the border-radius property to a value below 20px for best results. For example:

```
border: 3px ridge #330000;  
border-radius: 15px;
```

See Figure 4.31 (chapter4/box.html in the student files) for an example of this code in action. With progressive enhancement in mind, note that your visitors using older versions of Internet Explorer will see right-angle rather than rounded corners. However, the functionality and usability of the web page will not be affected. Also, keep in mind that another approach to getting a rounded look is to create a rounded rectangle background image with a graphics application.

Figure 4.31 Rounded corners were configured with CSS



Hands-On Practice 4.10

You'll configure a logo header area that uses a background image and rounded borders in this Hands-On Practice. When complete, your web page will look similar to the one shown in Figure 4.32.

1. Create a new folder called borderch4. Copy the lighthouselogo.jpg and the background.jpg files in the chapter4/starters folder to your borderch4 folder. A starter file is ready for you in the student files. Save the chapter4/starter3.html file to your borderch4 folder. Launch a browser to display the starter3.html web page shown in Figure 4.33.

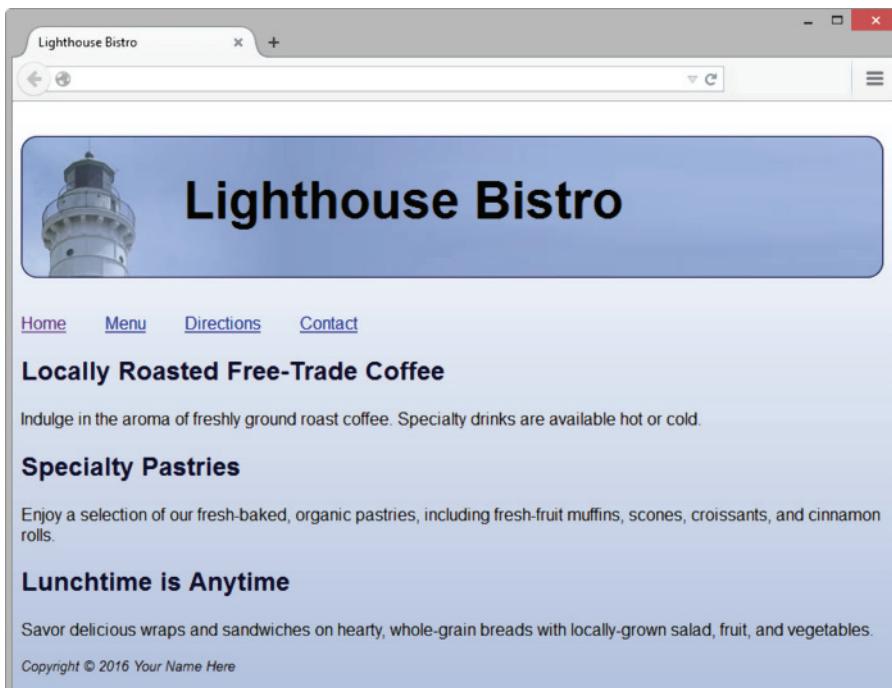


Figure 4.32 The web page with the logo area configured

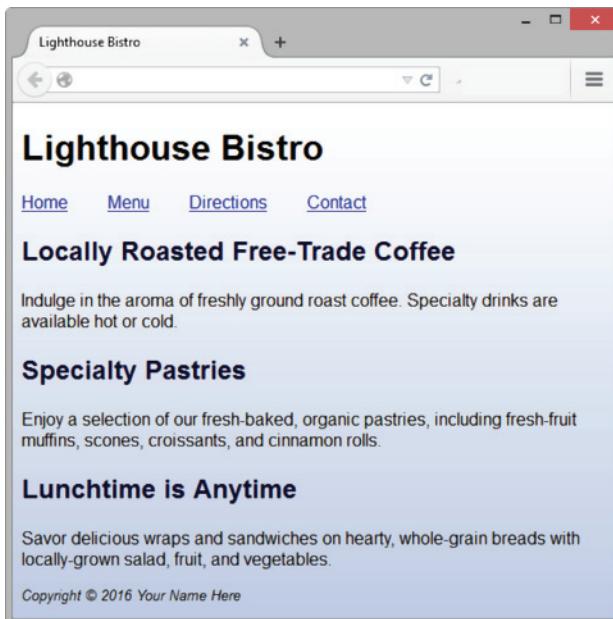


Figure 4.33 The starter3.html file

2. Launch a text editor, and open the starter3.html file. Save the file as index.html. Edit the embedded CSS, and add the following style declarations to the h1 element selector that will configure the lighthouselogo.jpg image as a background image that does not repeat: height set to 100px, width set to 650px, font size set to 3em,

150px of left padding, 30px of top padding, and a solid dark-blue border (#000033) with a border radius of 15px. The style declarations are as follows:

```
h1 { background-image: url(lighthouselogo.jpg);  
background-repeat: no-repeat;  
height: 100px; width: 650px; font-size: 3em;  
padding-left: 150px; padding-top: 30px;  
border: 1px solid #000033;  
border-radius: 15px; }
```

3. Save the file. When you test your index.html file in a browser, it should look similar to the one shown in Figure 4.32 if you are using a browser that supports rounded corners. Otherwise the logo will have right-angle corners, but the web page will still be usable. Compare your work with the solution in the student files (chapter4/4.10/index.html).

The CSS3 box-shadow Property

The CSS3 **box-shadow property** can be used to create a shadow effect on block-display elements such as div and paragraph elements. The box-shadow property is supported by current versions of major browsers, including Internet Explorer (version 9 and later). Configure a box shadow by coding values for the shadow's horizontal offset, vertical offset, blur radius (optional), spread distance (optional), and color:

- **Horizontal offset.** Use a numeric pixel value. Positive value configures a shadow on the right. Negative value configures a shadow on the left.
- **Vertical offset.** Use a numeric pixel value. Positive value configures a shadow below. Negative value configures a shadow above.
- **Blur radius (optional).** Configure a numeric pixel value. If omitted, defaults to the value 0 which configures a sharp shadow. Higher values configure more blur.
- **Spread distance (optional).** Configure a numeric pixel value. If omitted, defaults to the value 0. Positive values configure the shadow to expand. Negative values configure the shadow to contract.
- **Color value.** Configure a valid color value for the shadow.

Here's an example that configures a dark-gray drop shadow with 5px horizontal offset, 5px vertical offset, and a 5px blur radius:

```
box-shadow: 5px 5px 5px #828282;
```

Inner Shadow Effect. To configure an inner shadow, include the optional `inset` value. For example:

```
box-shadow: inset 5px 5px 5px #828282;
```



Hands-On Practice 4.11

You'll configure a centered content area and apply the `box-shadow` and `text-shadow` properties in this Hands-On Practice. When complete, your web page will look similar to the one shown in Figure 4.34.

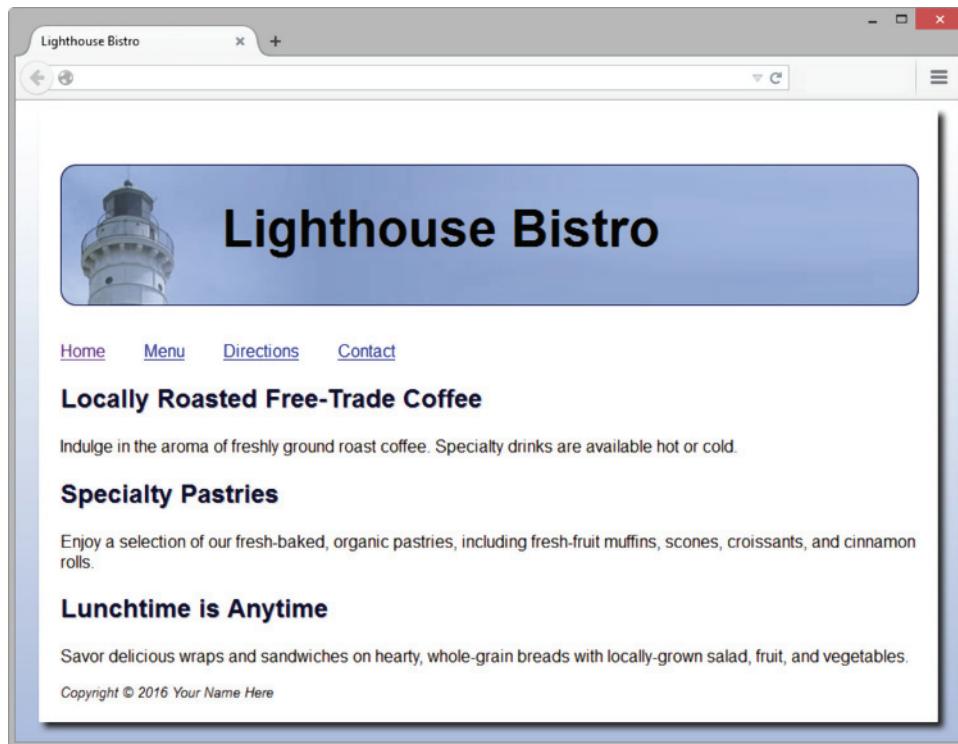


Figure 4.34 The centered neutral background along with shadow properties add dimension

Create a new folder called shadowch4. Copy the lighthouselogo.jpg and the background.jpg files from the chapter4/starters folder to your shadowch4 folder. Launch a text editor, and open the chapter4/4.10/index.html file (shown in Figure 4.32). Save the file in your shadowch4 folder.

1. Configure the page content to be centered, with an 800-pixel width, a white background, and some padding.
 - a. Edit the HTML. Configure a div element assigned to the id named `container` that wraps the code within the body section. Code the opening `<div>` tag on a new line after the opening body tag. Code a closing div tag on a new line before the closing body tag.
 - b. Edit the embedded CSS to configure a new selector, an id named `container`. Configure a white background color and 20 pixels of padding. Recall from Chapter 3 the style declarations that will center the page content. Use the `width`, `min-width`, `max-width`, `margin-left`, and `margin-right` properties as follows:

```
#container { background-color: #ffffff;  
            padding: 20px;  
            width: 80%; min-width: 800px; max-width: 960px;  
            margin-right: auto;  
            margin-left: auto; }
```

2. Edit the embedded CSS to add the following style declarations to the `#container` selector to configure a box shadow:

```
box-shadow: 5px 5px 5px #1e1e1e;
```

3. Add the following style declaration to the `h1` element selector to configure a dark-gray text shadow:

```
text-shadow: 3px 3px 3px #666;
```

4. Add the following style declaration to the `h2` element selector to configure a light-gray text shadow with no blur:

```
text-shadow: 1px 1px 0 #ccc;
```

5. Save the file. When you test your `index.html` file in a browser, it should look similar to the one shown in Figure 4.34 if you are using a browser that supports the `box-shadow` and `text-shadow` properties. Otherwise the shadows will not display, but the web page will still be usable. See the student files for a solution (chapter4/4.11/index.html).

Browser support of CSS3 properties may change with each new browser version. There is no substitute for thoroughly testing your web pages. However, several resources provide support lists. The following websites provide this information:

- <http://www.findmebyip.com/litmus>
- <http://www.quirksmode.org/css/contents.html>
- <http://www.impressivewebs.com/css3-click-chart>
- <http://caniuse.com>



Hands-On Practice 4.12

In this Hands-On Practice you will practice your new skills as you configure a web page with centered content and apply CSS properties. When complete, your web page will look similar to the one shown in Figure 4.35.

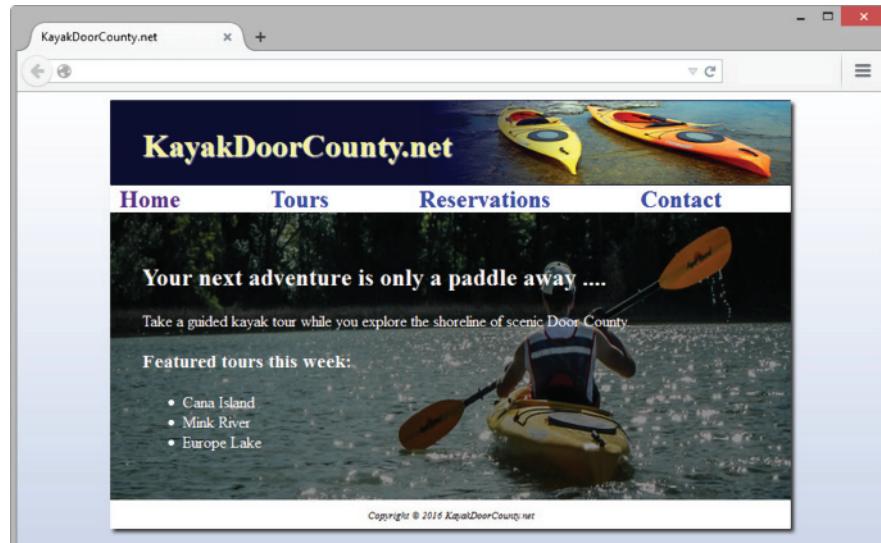


Figure 4.35 CSS drastically changes the look of the web page.

Create a new folder called kayakch4a. Copy the background.jpg, heroback2.jpg, and headerbackblue.jpg files from the chapter4/starters folder to your kayakch4 folder. Launch a text editor and open the chapter4/starter2.html file. Save the file in your kayakch4a folder with the name index.html. Modify the file as follows:

1. Center the page content.

- a. Configure embedded CSS between the style tags and code a new selector, an id named `container` with style declarations for the `width`, `margin-left`, and `margin-right` properties as follows:

```
#container { margin-left: auto;  
            margin-right: auto;  
            width: 80%; }
```

- b. Edit the HTML. Configure a `div` element assigned to the id `container` that “wraps” or contains the code within the body section. Code an opening `div` tag on a new line after the opening `body` tag. Assign the `div` to the id named `container`.

2. Configure embedded CSS.

- a. **The body element selector.** Configure a declaration to display `background.jpg` as the background image.

```
body { background-image: url(background.jpg); }
```

- b. **The container id selector.** Add declarations to configure a white background color, 650px minimum width, 1280px maximum width, and a box shadow with a 3px offset in the color #333.

```
#container { margin-left: auto;  
            margin-right: auto;  
            width: 80%;  
            background-color: #FFFFFF;  
            min-width: 650px; max-width: 1280px;  
            box-shadow: 3px 3px 3px #333; }
```

- c. **The header element selector.** Configure declarations to configure #000033 background color, #FF9 text color, a display of the `headerbackblue.jpg` image on the right without repeating, 80px height, 5px top padding, 2em left padding, and a text shadow in the color #FFF with a 1px offset.

```
header { background-color: #000033; color:#FF9;  
         background-image: url(headerbackblue.jpg);  
         background-position: right;  
         background-repeat: no-repeat;  
         height: 80px;  
         padding-top: 5px;  
         padding-left: 2em;  
         text-shadow: 1px 1px 1px #FFF; }
```

- d. **The nav element selector.** Code declarations to configure bold, 1.5em size, and centered text with 1em word-spacing.

```
nav { word-spacing: 1em;  
      font-weight: bold;  
      font-size: 1.5em;  
      text-align: center; }
```

- e. **The nav a descendant selector.** Code a declaration to eliminate the underline from hyperlinks.

```
nav a { text-decoration: none; }
```

- f. **The main element selector.** Code declarations to configure heroback2.jpg as the background image and set background-size: 100% 100%; Also configure white text (use #FFF) and 2em of padding.

```
main { background-image: url(heroback2.jpg);  
       background-size: 100% 100%;  
       color: #FFF;  
       padding: 2em; }
```

- g. **The footer element selector.** Configure declarations for italic, .80em size centered text with 0.5em of padding.

```
footer { font-style: italic; font-size: .80em;  
         text-align: center; padding: 0.5em; }
```

3. Save the file. When you test your index.html file in a modern browser such as Firefox or Chrome, it should look similar to the one shown in Figure 4.35. Compare your work with the solution in the student files (chapter4/4.12/index.html). Note that if you display the page in a browser (such as Internet Explorer 11) that does not support the new HTML5 main element, the display will not look as you expect. At the time this was written, Internet Explorer did not support default styles the HTML5 main element. You may need to nudge this browser to comply by adding the display: block; declaration (see Chapter 6) to the styles for the main element selector. An example solution is in the student files (chapter4/4.12/iefix.html).

The CSS3 opacity Property

The CSS3 **opacity property** configures the transparency of an element. The **opacity** property is supported by current versions of major browsers, including Internet Explorer (version 9 and later). Opacity values range from 0 (which is completely transparent) to 1 (which is completely opaque and has no transparency). An important consideration when using the **opacity** property is that this property applies to both the text and the background. If you configure a semi-transparent **opacity** value for an element with the **opacity** property, both the background and the text displayed will be semi-transparent. See Figure 4.36 for an example of using the **opacity** property to configure a white background that is only 60% opaque. If you look very closely at Figure 4.36 or view the actual web page (student files chapter4/4.13/index.html), you'll see that both the white background and the black text in the h1 element are semi-transparent. The **opacity** property was applied to both the background color and to the text color.

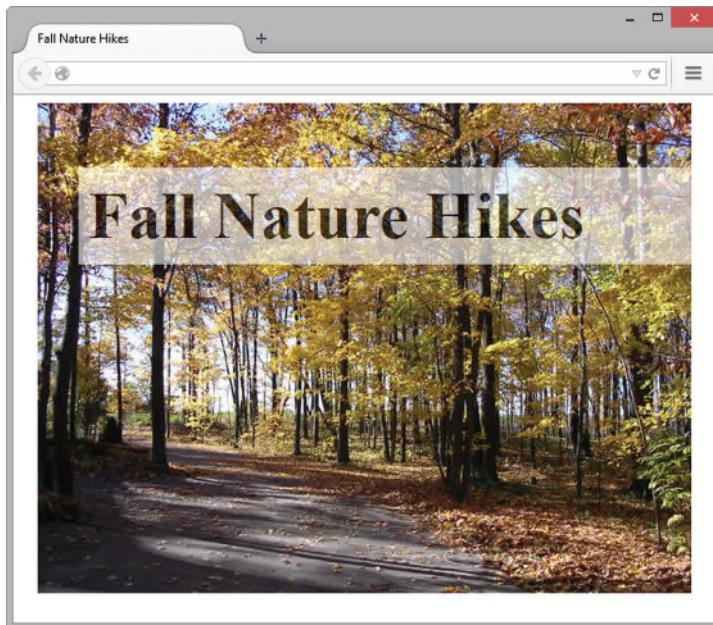


Figure 4.36 The background of the h1 area is transparent



Hands-On Practice 4.13

In this Hands-On Practice you'll work with the opacity property as you configure the web page shown in Figure 4.36.

1. Create a new folder called opacitych4. Copy fall.jpg file from the chapter4/starters folder to your opacitych4 folder. Launch a text editor and open the chapter2/template.html file. Save it in your opacitych4 folder with the name index.html. Change the page title to "Fall Nature Hikes".
2. Let's create the structure of the web page with a div that contains an h1 element. Add the following code to your web page in the body section:

```
<div id="content">
  <h1>Fall Nature Hikes</h1>
</div>
```

3. Now, add style tags to the head section, and configure the embedded CSS. You'll create an id named `content` to display the fall.jpg as a background image that does not repeat. The `content` id also has a width of 640 pixels, a height of 480 pixels, left and right auto margins (which will center the object in the browser viewport), and 20 pixels of top padding. The code is

```
#content { background-image: url(fall.jpg);
            background-repeat: no-repeat;
            margin-left: auto;
            margin-right: auto;
            width: 640px;
            height: 480px;
            padding-top: 20px; }
```

4. Now configure the h1 element selector to have a white background color with opacity set to 0.6, font size set to 4em, and 10 pixels of padding. Sample code is

```
h1 { background-color: #FFFFFF;  
     opacity: 0.6;  
     font-size: 4em;  
     padding: 10px; }
```

5. Save the file. When you test your index.html file in a browser that supports opacity, it should look similar to the page shown in Figure 4.36. See the student files for a solution (chapter4/4.13/index.html).

Figure 4.37 shows the web page displayed in Internet Explorer 8, which does not support the `opacity` property. Notice that the visual aesthetic is not exactly the same, but the page is still usable. While Internet Explorer versions 9 and later support opacity, earlier versions support the proprietary **filter property** with an opacity level configured between 1 (transparent) and 100 (opaque). A sample is found in the student files (chapter4/4.13/opacityie.html). The CSS for the filter property is

```
filter: alpha(opacity=60);
```



Figure 4.37 Internet Explorer 8 does not support the `opacity` property and displays an opaque background color. Screenshots of Internet Explorer. Copyright by Microsoft Corporation. Used by permission of Microsoft Corporation

CSS3 RGBA Color

CSS3 supports new syntax for the color property that configures transparent color, called **RGBA color**. RGBA color is supported by current versions of major browsers, including Internet Explorer (version 9 and later). Four values are required: red color, green color, blue color, and alpha (transparency). RGBA color does not use hexadecimal color values. Instead, decimal color values are configured; see the partial color chart in Figure 4.38 and the Web-Safe Color Palette in the Appendix for examples.

#FFFFFF rgb(255, 255, 255)	#FFFFCC rgb(255, 255, 204)	#FFFF99 rgb(255, 255, 153)	#FFFF66 rgb(255, 255, 102)
#FFFF33 rgb(255, 255, 51)	#FFFF00 rgb(255, 255, 0)	#FFCCFF rgb(255, 204, 255)	#FFCCCC rgb(255, 204, 204)
#FFCC99 rgb(255, 204, 153)	#FFCC66 rgb(255, 204, 102)	#FFCC33 rgb(255, 204, 51)	#FFCC00 rgb(255, 204, 0)
#FF99FF rgb(255, 153, 255)	#FF99CC rgb(255, 153, 204)	#FF9999 rgb(255, 153, 153)	#FF9966 rgb(255, 153, 102)

Figure 4.38 Hexadecimal and RGB decimal color values

The values for red, green, and blue must be decimal values from 0 to 255. The alpha value must be a number between 0 (transparent) and 1 (opaque). Figure 4.39 shows a web page with the text configured to be slightly transparent.

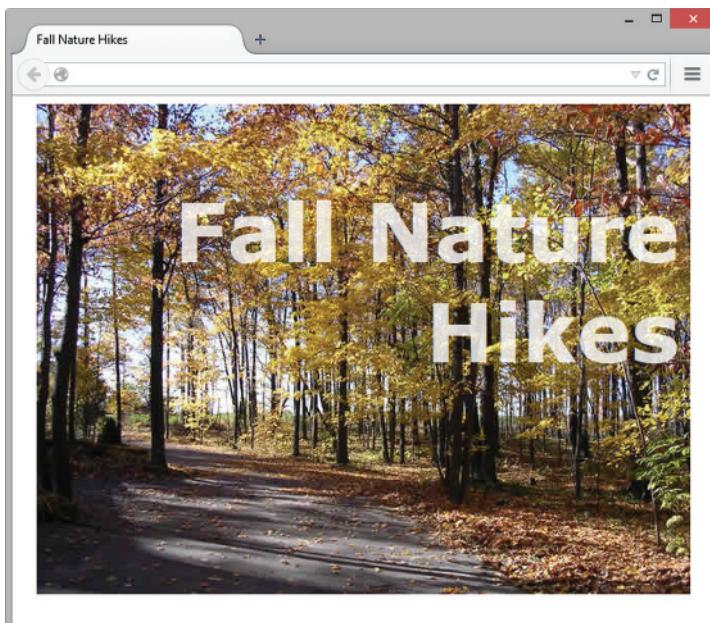


Figure 4.39 CSS3 RGBA color configures the transparent text



FAQ How is using RGBA color different from using the opacity property?

The `opacity` property applies to both the background and the text within an element. If you would like to specifically configure a semi-transparent background color, code the `background-color` property with RGBA color or HSLA color (described in the next section) values. If you would like to specifically configure semi-transparent text, code the `color` property with RGBA color or HSLA color values.



Hands-On Practice 4.14

In this Hands-On Practice you'll configure transparent text as you code the web page shown in Figure 4.39.

1. Create a new folder called rgbach4. Copy fall.jpg file from the chapter4/starters folder to your rgbach4 folder. Launch a text editor, and open the file you created in the previous Hands-On Practice (also located in the student files chapter4/4.13/index.html). Save the file with the name rgba.html in your rgbach4 folder.
2. Delete the current style declarations for the h1 element selector. You will create new style rules for the h1 selector to configure 10 pixels of right padding and right-aligned sans-serif white text that is 70% opaque, with a font size of 5em. Since not all browsers support RBGA color, you'll configure the color property twice. The first instance will be the standard color value that is supported by all modern browsers; the second instance will configure the RGBA color. Older browsers will not understand the RGBA color and will ignore it. Newer browsers will “see” both of the color style declarations and will apply them in the order they are coded, so the result will be transparent color. The CSS code is

```
h1 { color: #ffffff;  
     color: rgba(255, 255, 255, 0.7);  
     font-family: Verdana, Helvetica, sans-serif;  
     font-size: 5em;  
     padding-right: 10px;  
     text-align: right; }
```

3. Save the file. When you test your rgba.html file in a browser that supports RGBA color, it should look similar to the page shown in Figure 4.39. See the student files for a solution (chapter4/4.14/rgba.html). If you are using a nonsupporting browser such as Internet Explorer 8 (or earlier), you'll see white text instead of transparent text. While Internet Explorer versions 9 and later supports RGBA color, earlier versions support the proprietary filter property; an example is in the student files (chapter4/4.14/rbgaie.html).

CSS3 HSLA Color

For many years web designers have configured RGB color using either hexadecimal or decimal values on web pages. Recall that RGB color is based on hardware—the red, green, and blue light that is emitted by computer monitors. CSS3 introduced a new color notation system called **HSLA color**, based on a color wheel model, which stands for hue, saturation, lightness, and alpha. HSLA color is supported in the most recent versions of all major browsers, including Internet Explorer 9 and later versions.

Hue, Saturation, Lightness, and Alpha

When you work with HSLA color, think of a color wheel—a circle of color—with the color red at the top of the wheel as shown in Figure 4.40. Hue is the actual color which is represented by numeric values ranging from 0 to 360 (like the 360 degrees in a circle).

For example, red is represented by both the values 0 and 360, green is represented by 120, and blue is represented by 240. Set hue to 0 when configuring black, gray, and white. Saturation configures the intensity of the color and is indicated by a percentage value (full color saturation = 100%, gray=0%). Lightness determines the brightness or darkness of the color and is indicated by a percentage value (normal color = 50%, white=100%, black = 0%). Alpha represents the transparency of the color and has a value from 0 (transparent) to 1 (opaque). Note that you can omit the alpha value and use the `hsl` keyword instead of the `hsla` keyword.

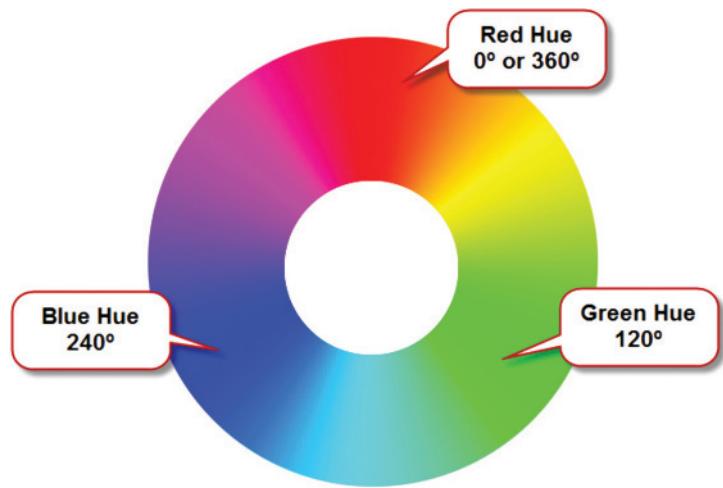


Figure 4.40 A color wheel.

HSLA Color Examples

Configure HSLA color as shown in Figure 4.41 with the following syntax:
`hsla(hue value, saturation value, lightness value, alpha value);`

- Red: `hsla(360, 100%, 50%, 1.0);`
- Green: `hsla(120, 100%, 50%, 1.0);`
- Blue: `hsla(240, 100%, 50%, 1.0);`
- Black: `hsla(0, 0%, 0%, 1.0);`
- Gray: `hsla(0, 0%, 50%, 1.0);`
- White: `hsla(0, 0%, 100%, 1.0);`

According to the W3C, an advantage to using HSLA color is that it is more intuitive to work with than the hardware-oriented RGB color. You can use a color wheel model to choose colors and generate the hue value from the degree placement on the circle. If you would like to use a tone of a color, which is a color with gray added, vary the saturation value. If you would like to use a shade or tint of a color, use the same hue value, but vary the lightness value to meet your needs. Figure 4.42 shows three shades of cyan blue configured using three different values for lightness: 25% (dark cyan blue), 50% (cyan blue), 75% (light cyan blue).

- Dark Cyan Blue:
`hsla(210, 100%, 25%, 1.0);`
- Cyan Blue:
`hsla(210, 100%, 50%, 1.0);`
- Light Cyan Blue:
`hsla(210, 100%, 75%, 1.0);`

Red <code>hsla(360, 100%, 50%, 1.0);</code>
Green <code>hsla(120, 100%, 50%, 1.0);</code>
Blue <code>hsla(240, 100%, 50%, 1.0);</code>
Black <code>hsla(0, 0%, 0%, 1.0);</code>
Gray <code>hsla(0, 0%, 50%, 1.0);</code>
White <code>hsla(0, 0%, 100%, 1.0);</code>

Figure 4.41 HSLA color examples.

<code>hsla(210, 100%, 25%, 1.0);</code>
<code>hsla(210, 100%, 50%, 1.0);</code>
<code>hsla(210, 100%, 75%, 1.0);</code>

Figure 4.42 Shades of cyan blue.



Hands-On Practice 4.15

In this Hands-On Practice you'll configure light yellow transparent text as you work with the web page shown in Figure 4.43.

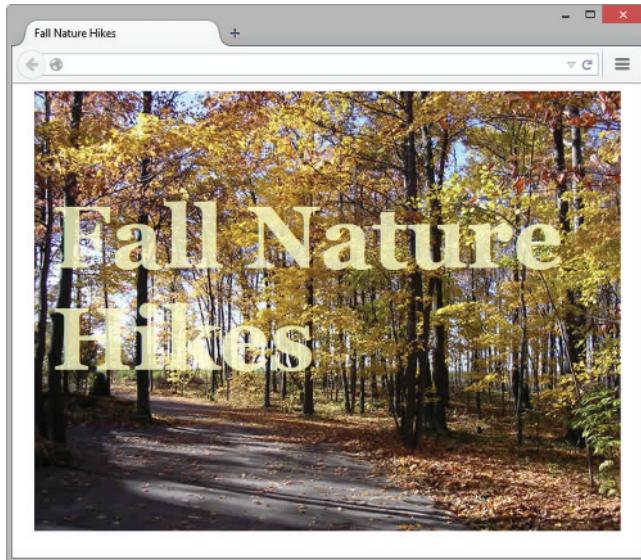


Figure 4.43 HSLA color.

1. Create a new folder called hslach4. Copy fall.jpg file from the chapter4/starters folder to your hslach4 folder. Launch a text editor and open the file you created in the previous Hands-On Practice (also located in the student files, chapter4/4.14/rgba.html). Save the file with the name hsla.html in your hslach4 folder.
2. Delete the style declarations for the h1 selector. You will create new style rules for the h1 selector to configure 20 pixels of padding and serif light yellow text with a 0.8 alpha value and a font size of 6em. Since not all browsers support HSLA color, you'll configure the color property twice. The first instance will be the standard color value that is supported by all modern browsers; the second instance will configure the HSLA color. Older browsers will not understand the HSLA color and will ignore it. Newer browsers will "see" both of the color style declarations and will apply them in the order they are coded, so the result will be transparent color. The CSS for the h1 selector is

```
h1 { color: #ffcccc;  
     color: hsla(60, 100%, 90%, 0.8);  
     font-family: Georgia, "Times New Roman", serif;  
     font-size: 6em;  
     padding: 20px; }
```

3. Save the file. When you test your hsla.html file in a browser that supports HSLA color it should look similar to the page shown in Figure 4.43. See the student files for a solution (chapter4/4.15/hsla.html). If you are using a nonsupporting browser such as Internet Explorer 8 (or earlier), you'll see solid text instead of transparent text.

CSS3 Gradients

CSS3 provides a method to configure color as a **gradient**, which is a smooth blending of shades from one color to another color. A CSS3 gradient background color is defined purely with CSS—no image file is needed! This provides flexibility for web designers, along with a savings in the bandwidth required to serve out gradient background image files.

Figure 4.34 displays a web page with a JPG gradient background image that was configured in a graphics application. The web page shown in Figure 4.44 (available at chapter4/gradient/index.html in the student files) does not use a JPG for the background; CSS3 gradient properties recreated the look of the linear gradient image.

The syntax for CSS3 gradients changed extensively while it was in draft status and you may find conflicting information about coding CSS3 gradients on the Web. The W3C CSS3 Image Values and Replaced Content Module was moved to Candidate Recommendation status in 2012. The W3C syntax described in this section is supported by modern browsers.

Linear Gradient Syntax

A **linear gradient** is a smooth blending of color in a single direction such as from top to bottom or from left to right. To configure a basic linear gradient, code the `linear-gradient` function as the value of the `background-image` property. Indicate the direction of the gradient by coding the keyword phrase “`to bottom`”, “`to top`”, “`to left`” or “`to right`”. Next, list the starting color and the ending color. The basic format for a two-color linear gradient that blends from white to green follows:

```
background-image: linear-gradient(to bottom, #FFFFFF, #00FF00);
```

Radial Gradient Syntax

A **radial gradient** is a smooth blending of color emanating outward from a single point. Code the `radial-gradient` function as the value of the `background-image` property to configure a radial gradient. List two colors as the values of the function. The first color will be displayed by default in the center of the element and gradually blend outward until the second color is displayed. The basic format for a two-color radial gradient that blends from white to blue follows:

```
background-image: radial-gradient(#FFFFFF, #0000FF);
```

CSS3 Gradients and Progressive Enhancement

It's very important to keep progressive enhancement in mind when using CSS3 gradients. Configure a “fallback” `background-color` property or `background-image` property which will be rendered by browsers that do not support CSS3 gradients. In Figure 4.44 the background color was configured to be same value as the ending gradient color.

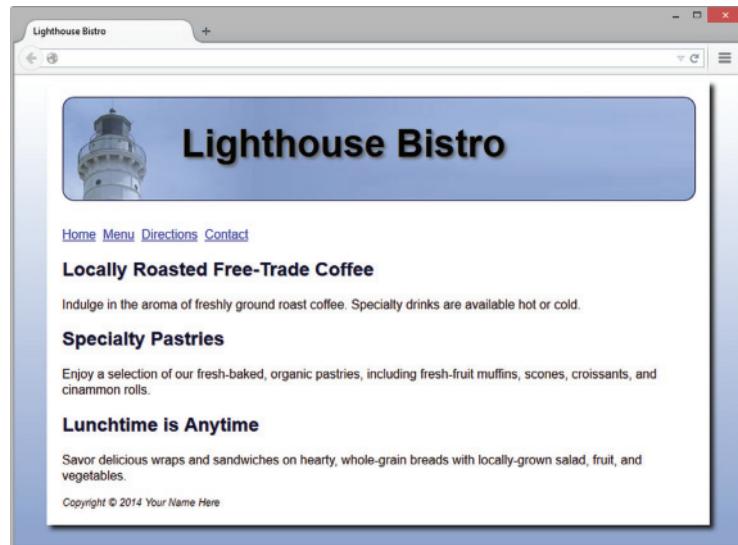


Figure 4.44 The gradient in the background was configured with CSS3 without an image file.



Hands-On Practice 4.16

You'll work with CSS gradient backgrounds in this Hands-On Practice. Create a new folder called gradientch4. Copy the chapter4/starter4.html file into your gradientch4 folder. Rename the file index.html. Launch a text editor and open the file.

1. First, you will configure a linear gradient. Code embedded CSS in the head section. Configure the body of the web page to display a fallback orchid background color of #DA70D6 and a linear gradient background that blends white to orchid from top to bottom without repeating:

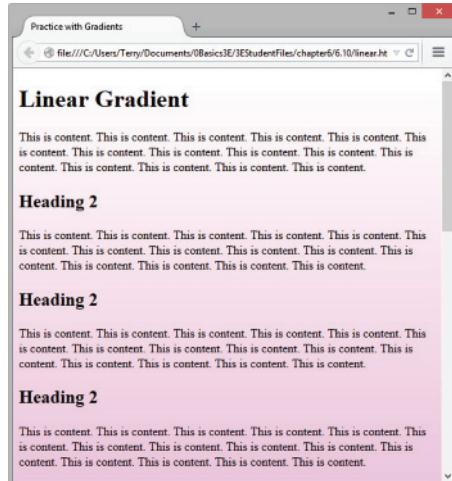


Figure 4.45 Linear gradient background.

```
body { background-color: #DA70D6;
background-image: linear-gradient(to bottom, #FFFFFF, #DA70D6);
background-repeat: no-repeat; }
```

2. Save your file and test it in a modern browser. The display should be similar to the results shown in Figure 4.45. The background gradient displays behind the page content, so scroll down the page to see the full gradient. Compare your work with the solution in the student files (chapter4/4.16/linear.html).
3. Next, you will configure a radial gradient. Edit the body section of the web page and code change the text within the h1 element to: Radial Gradient.
4. Edit the CSS and modify the value of the background-image property to configure a radial gradient linear gradient that blends white to orchid from center outward without repeating:

```
body { background-color: #DA70D6;
background-image: radial-gradient(#FFFFFF, #DA70D6);
background-repeat: no-repeat; }
```

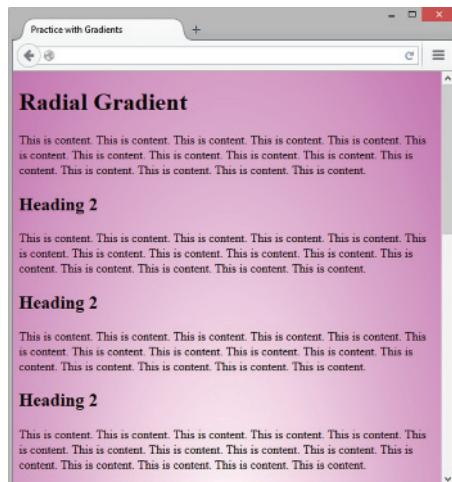


Figure 4.46 Radial gradient background.

5. Save your file and test it in a modern browser. The display should be similar to the results shown in Figure 4.46. Scroll down the page to see the full gradient. Compare your work with the solution in the student files (chapter4/4.16/radial.html).



Visit <http://css-tricks.com/css3-gradients> to delve deeper into CSS3 gradients. Experiment with generating CSS3 gradient code at <http://www.colorzilla.com/gradient-editor>, <http://www.css3factory.com/linear-gradients>, and <http://www.westciv.com/tools/gradients>.

Chapter Summary

This chapter has introduced the use of visual elements and graphics on web pages. The number-one reason visitors leave web pages is long download times. When using images, be careful to optimize the images for the Web, reducing both the size of the file and the dimensions of the image, in order to minimize download time.

You explored new HTML5 elements and many CSS properties in this chapter. When using the new CSS3 properties and HTML5 elements, be mindful of the concepts of progressive enhancement and accessibility. Verify that the pages display in an acceptable manner even if new techniques are not supported by the browser. Provide text alternatives to images with the `alt` attribute.

Visit the textbook's website at <http://www.webdevfoundations.net> for examples, the links listed in this chapter, and updated information.

Key Terms

	filter property	min-width property
alt attribute	gamma	opacity property
animated GIF	GIF images	padding property
area element	gradient	pixelation
aspect ratio	height attribute	PNG images
background-attachment property	hotspots	progress element
background-clip property	HSLA color	progressive enhancement
background-image property	horizontal rule <hr>	progressive JPEG
background-origin property	hspace attribute	radial gradient
background-position property	image element	RGBA color
background-repeat property	image link	resolution
background-size property	image map	sprite
border property	image optimization	src attribute
border-color property	image slicing	thumbnail image
border-radius property	interlaced image	transparency
border-style property	JPEG images	usemap attribute
border-width property	linear gradient	vspace attribute
box-shadow property	lossless compression	WebP image format
favicon	lossy compression	width attribute
figcaption element	max-width property	
figure element	meter element	

Review Questions

Multiple Choice

1. Which of the following graphic types can be made transparent?
 - a. GIF
 - b. JPG
 - c. BMP
 - d. PHOTO
2. Which of the following configures empty space between the content of the HTML element (typically text) and the border?
 - a. vspace property
 - b. padding property
 - c. margin property
 - d. border property
 - e. none of the above

- 3.** Which of the following creates an image link to the index.html page when the home.gif graphic is clicked?
-
 -

 -
 -

- 4.** What is the purpose of coding height and width attributes on an tag?
- They are required attributes and must always be included.
 - They help the browser render the page faster because it reserves the appropriate space for the image.
 - They help the browser display the image in its own window.
 - none of the above
- 5.** Which attribute specifies text that is available to browsers and other user agents that do not support graphics?
- alt
 - text
 - src
 - none of the above
- 6.** What is the term used to describe a square icon that is associated with a web page and is displayed in the browser address bar or tab?
- background
 - bookmark icon
 - favicon
 - logo
- 7.** Which of the following graphic types is best suited to photographs?
- GIF
 - JPG
 - BMP
 - PHOTO
- 8.** Which CSS property configures the background color?
- bgcolor
 - background-color
 - color
 - none of the above
- 9.** Which HTML tag configures a horizontal line on a web page?
- <line>
 -

 - <hr>
 - <border>
- 10.** Which of the following configures a graphic to repeat vertically down the side of a web page?
- hspace="10"
 - background-repeat:repeat;
 - valign="left"
 - background-repeat: repeat-y;

Fill in the Blank

- 11.** A background image will automatically be repeated, or _____, by a web browser.
- 12.** If your web page uses graphic links, include _____ at the bottom of the page to increase accessibility.
- 13.** A(n) _____ image is a smaller version of a larger image that usually links to the larger image.
- 14.** The _____ CSS3 property configures a drop-shadow effect on an HTML element.
- 15.** The _____ element displays a visual gauge of a numeric value within a known range.

Apply Your Knowledge

- 1. Predict the Result.** Draw and write a brief description of 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> 
</header>
<nav> Home <a href="about.html">About</a>
<a href="services.html">Services</a>
</nav>
<main><p>Our professional staff takes pride in its working
relationship with our clients by offering personalized services
that take their needs into account, develop their target areas, and
incorporate these items into a website that works.</p>
</main>
</body>
</html>
```

- 2. Fill in the Missing Code.** This web page contains an image link and should be configured so that the background and text colors have good contrast. The image used on this web page should link to a page called services.html. Some HTML attribute values, indicated by _, are missing. Some CSS style rules, indicated by _, are incomplete. The code follows:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CircleSoft Design</title>
<meta charset="utf-8">
<style>
body { _: _;
      color: _;
}
</style>
</head>
<body>
  <div>
    <a href="_">
      <br>Enter CircleSoft Design</a>
    </div>
  </body>
</html>
```

- 3. Find the Error.** This page displays an image called trillium.jpg. The image is 307 pixels wide by 200 pixels high. When this page is displayed, the image does not look right. Find the error. Describe any attributes that you would code in the tag to provide accessibility. The code follows:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Find the Error</title>
<meta charset="utf-8">
</head>
<body>

</body>
</html>
```

Hands-On Exercises

1. Write the HTML to place an image called primelogo.gif on a web page. The image is 100 pixels high by 650 pixels wide.
2. Write the HTML to create an image for the schaumburghthumb.jpg image. It is 100 pixels high by 150 pixels wide. The image should link to a larger image called schaumburg.jpg. There should be no border on the image.
3. Write the HTML to create a nav element that contains three images used as navigation links. Table 4.5 provides information about the images and their associated links.

Table 4.5

Image Name	Link Page Name	Image Height	Image Width
homebtn.gif	index.html	50	200
productsbtn.gif	products.html	50	200
orderbtn.gif	order.html	50	200

4. Experiment with page backgrounds. Locate the twocolor.gif file in the student files chapter4/starters folder. Design a web page that uses this file as a background image that repeats down the left side of the browser window. Save your file as bg1.html.
5. Experiment with page backgrounds. Locate the twocolor1.gif file in the student files chapter4/starters folder. Design a web page that uses this file as a background image that repeats across the top of the browser window. Save your file as bg2.html.
6. Visit one of your favorite websites. Note the colors used for background, text, headings, images, and so on. Write a paragraph that describes how the site uses color for these elements. Code a web page that uses colors in a similar manner. Save your file as color.html.

7. Practice with CSS.
 - a. Write the CSS for an HTML selector footer with the following characteristics: a light-blue background color, Arial font, dark-blue text color, 10 pixels of padding, and a narrow, dashed border in a dark-blue color.
 - b. Write the CSS for an id named `notice` that is configured to 80% width and centered.
 - c. Write the CSS to configure a class that will produce a headline with a dotted line underneath it. Choose a color that you like for the text and dotted line.
 - d. Write the CSS to configure an h1 element selector with drop-shadow text, a 50% transparent background color, and sans-serif font that is 4em in size.
 - e. Write the CSS to configure an id named `feature` with small, red, Arial font; a white background; a width of 80%; and a drop shadow.
8. Design a new web page about you. Use CSS to configure a background color and text color for the page. Include the following on your web page:
 - Your name
 - A description of your favorite hobbies and activities
 - A photo of yourself (be sure to optimize the image for display on the Web)Save the page as `yourlastname.html`.
9. Design a web page that provides a list of resources for free stock photographs. The list should contain at least five different websites. Use your favorite graphic sites, the sites suggested in this chapter, or sites you have found on the Web. Save the page as `freegraphics.html`.
10. Visit the textbook's website at <http://webdevfoundations.net/8e/chapter4.html> and follow the link to the Adobe Fireworks or Adobe Photoshop tutorial. Follow the instructions to create a logo banner. Hand in the printouts described in the tutorial to your instructor.

Web Research

1. Providing access to the Web for all people is an important issue. Visit the W3C's Web Accessibility Initiative and explore its WCAG 2.0 Quick Reference at <http://www.w3.org/WAI/WCAG20/quickref>. View additional pages at the W3C's site as necessary. Explore the checkpoints that are related to the use of color and images on web pages. Create a web page that uses color, uses images, and includes the information that you discovered.
2. This chapter has introduced you to several new CSS3 properties. Choose one of them to research further. Create an example web page that demonstrates the use of the property. Use one of the following sites as a starting point to determine the current browser support of the property, and include a summary of this information (along with the URLs of the resources you used) in your web page:
 - <http://www.quirksmode.org/css/contents.html>
 - <http://www.findmebyip.com/litmus>
 - <http://www.impressivewebs.com/css3-click-chart>

Focus on Web Design

Visit a website that interests you. Print the home page or one other pertinent page from the site. Write a one-page summary and reaction to the website you chose to visit. Address the following topics:

- a. What is the purpose of the site?
- b. Who is the intended audience?
- c. Do you believe the site reaches its audience?
- d. Was this site useful to you? Why or why not?
- e. List the colors and/or graphics that are used on the home page of this website: background, backgrounds of page sections, text, logo, navigation buttons, and so on.
- f. How does the use of color and graphics enhance the website?



WEBSITE CASE STUDY

Using Graphics & Visual Elements

Each of the case studies in this section continues throughout most of the text. In this chapter, we add images to the websites, create a new page, and modify existing pages.

JavaJam Coffee House

See Chapter 2 for an introduction to the JavaJam Coffee House Case Study. Figure 2.30 shows a site map for the JavaJam website. The Home page and Menu page were created in earlier chapters. Using the existing website as a starting point, you will modify the design of the pages and create a new page, the Music page. You have five tasks in this case study:

1. Create a new folder for this JavaJam case study, and obtain the starter image files.
2. Modify the Home page to display the winding road image as shown in Figure 4.47.
3. Modify the Menu page, shown in Figure 4.48, to be consistent with the Home page.
4. Create a new Music page, as shown in Figure 4.49.
5. Modify the style rules in the `javajam.css` file as needed.



Figure 4.47 New JavaJam Home page

Hands-On Practice Case

Task 1: Create a folder on your hard drive or portable storage device called javajam4. Copy all the files from your Chapter 3 javajamcss folder into the javajam4 folder. Obtain the images used in this case study from the student files. The images are located in the chapter4/casestudystarters/javajam folder. The images are background.gif, greg.jpg, gregthumb.jpg, javajamlogo.jpg, melanie.jpg, melaniethumb.jpg, mugs.jpg, and windingroad.jpg. Save them in your javajam4 folder.

Task 2: The Home Page. Launch a text editor, and open the index.html file from your javajam4 folder. Modify the index.html file to look similar to the web page shown in Figure 4.47.

1. Replace the “Relax at JavaJam” text contained within the h2 element with “Follow the Winding Road to JavaJam.”
2. Code an img tag for the windingroad.jpg image above the h2 element in the main content area. Be sure to include the alt, height, and width attributes. Also configure the image to appear to the right of the text content by coding the align="right" attribute on the tag. *Note:* The W3C HTML validator will indicate that the align attribute is invalid. We’ll ignore the error for this case study. In Chapter 6 you’ll learn to use the CSS float property (instead of the align property) to configure this type of layout.
3. Add a paragraph with the following text below the h2 element:

“We’re a little out of the way, but take a drive down Route 42 to JavaJam today! Indulge in our locally roasted free-trade coffee and home-made pastries. You’ll feel right at home at JavaJam!”

4. Configure the text “JavaJam Coffee House features:” within an h3 element below the paragraph and above the unordered list.

Save and test your new index.html page. It will be similar to Figure 4.47, but you’ll notice that a few final touches (including the background image and logo image) are missing; you’ll configure these with CSS in Task 5.

Task 3: The Menu Page. Launch a text editor, and open the menu.html page from your javajam4 folder. Modify the menu.html file to look similar to the web page shown in Figure 4.48.



Figure 4.48 JavaJam menu.html

1. Code an img tag for the mugs.jpg image above the h2 element in the main content area. Be sure to include the alt, height, and width attributes. Also configure the image to appear to the right of the text content by coding the align="right" attribute on the tag. Note: The W3C HTML validator will indicate that the align attribute is invalid. We'll ignore the error for this case study. In Chapter 6, you'll learn to use the CSS float property (instead of the align property) to configure this type of layout.

2. Add a paragraph with the following text below the h2 element:

"Indulge in our locally roasted free-trade coffee and enjoy the aroma, the smooth taste, the caffeine! Join our Mug Club and get a 10% discount on each cup of coffee you purchase — ask the barista for details." Save and test your new menu.html page. It will be similar to Figure 4.48, but missing a few final touches (see Task 5).

Task 4: The Music Page. Use the Menu page as the starting point for the Music page. Launch a text editor, and open the menu.html file in the javajam4 folder. Save the file as music.html. Modify the music.html file to look similar to the Music page, as shown in Figure 4.49:



Figure 4.49 JavaJam music.html

1. Change the page title to an appropriate phrase.
2. Delete the image and description list from the page.
3. Configure "Music at JavaJam" as the text within the h2 element.
4. Configure the following as the text within the paragraph element:

"The first Friday night each month at JavaJam is a special night. Join us from 8 pm to 11 pm for some music you won't want to miss!"

5. The rest of the content in the page will consist of two areas describing music performances. The area describing each music performance consists of an h4 element, a div assigned to the class named details, and an image link.

January Music Performance:

- Configure an h4 element with the following text: January
- Code an opening div tag. Assign the div to the class named `details`.
- Configure the melaniethumb.jpg as an image link to melanie.jpg. Code appropriate attributes on the `` tag.
- Configure the following text within the div after the image link:
Melanie Morris entertains with her melodic folk style.

February Music Performance:

- Configure an h4 element with the following text: February
- Code an opening div tag. Assign the div to the class named `details`.
- Configure the gregthumb.jpg as an image link to greg.jpg. Code appropriate attributes on the `` tag.
- Configure the following text within the div after the image:
Tahoe Greg is back from his tour. New songs. New stories.

Save the music.html file. If you test your page in a browser, you'll notice that it looks different from Figure 4.49—you still need to configure style rules.

Task 5: Configure the CSS. Open javajam.css in a text editor. Edit the style rules as follows:

1. Modify the body element selector style rules. Configure background.gif as the background image.
2. Modify the style rules for the `wrapper` id. Configure the background color to be #FEF6C2. Configure a minimum width of 900px (use `min-width`). Configure a maximum width of 1280px (use `max-width`). Use the `box-shadow` property to configure a drop-shadow effect.
3. Modify the header element selector style rules. Remove the `text-align` declaration. Code a declaration to set the height to 150px. Configure declarations to display the javajamlogo.jpg image as a background image without repeating.
4. Modify the h1 element selector style rules. Remove the `line-height` declaration. Configure declarations to set the top padding to 45px, left padding to 220px, and font size to 3em.
5. Modify the nav element selector style rules. Configure declarations for 1.5em font size size and 10px of top padding.
6. Code a new style rule to prevent the hyperlinks in the nav area from displaying the default underline. Use `nav a { text-decoration: none; }`
7. Modify the footer element selector style rules. Configure declarations for 10px of bottom padding and a solid 2px top border (use #221811 as the color).
8. Add a new style rule for the h4 element selector that configures a background color (#D2B48C), font size (1.2em), left padding (10px), and bottom padding (5px).
9. Add a new style rule for the main element selector to configure 2em of padding on the left, right, and bottom. Also configure `display: block;` (this will prevent rendering issues in versions of Internet Explorer that do not support the HTML5 main element).

10. Add a new style rule for the class named `details` to add 20% left and right padding. Notice how this rule adds empty space on either side of the music performance description and image on the `music.html` page.
11. Add a new style rule for the `img` element selector that configures 10px left padding and 10 px right padding.

Save the `javajam.css` file. Test your pages (`index.html`, `menu.html`, and `music.html`) in a browser. If your images do not appear or your image links do not work, examine your work carefully. Use Windows Explorer or Mac Finder to verify that the images are saved in your `javajam4` folder. Examine the `src` attribute on the `` tags to be sure you spelled the image names correctly. Another useful troubleshooting technique is to validate the HTML and CSS code. See Chapters 2 and 3 for Hands-On Practice exercises that describe how to use these validators.

Fish Creek Animal Hospital

See Chapter 2 for an introduction to the Fish Creek Animal Hospital Case Study.

Figure 2.34 shows a site map for Fish Creek. The Home page and Services page were created in earlier chapters. Using the existing website as a starting point, you will modify the design of the pages and create a new page, the Ask the Vet page. You have five tasks in this case study:

1. Create a new folder for this Fish Creek case study, and obtain the starter image files.
2. Modify the Home page to display a logo image and navigation image links as shown in Figure 4.50.
3. Modify the Services page to be consistent with the Home page.
4. Create a new Ask the Vet page, as shown in Figure 4.51.
5. Modify the style rules in the `fishcreek.css` file as needed.

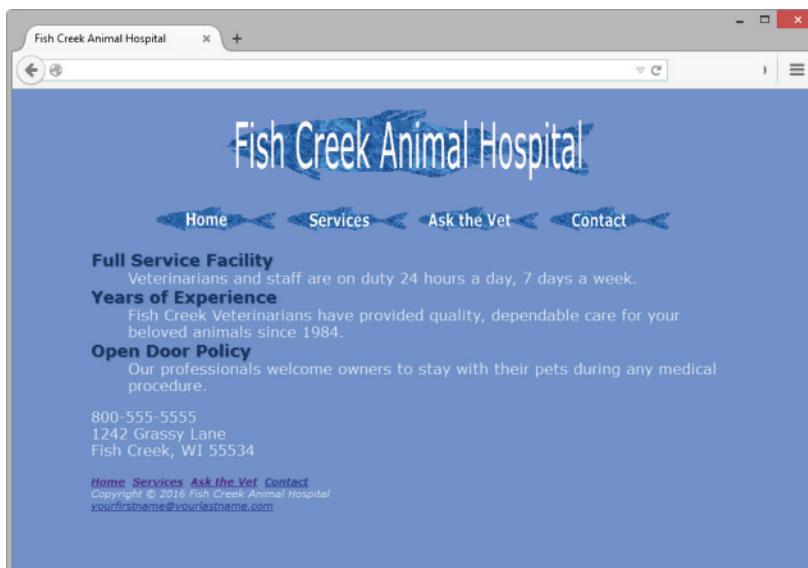


Figure 4.50 New Fish Creek Home page

Hands-On Practice Case

Task 1: Create a folder on your hard drive or portable storage device called fishcreek4. Copy all the files from your Chapter 3 fishcreekcss folder into the fishcreek4 folder. Obtain the images used in this case study from the student files. The images are located in the chapter4/casestudystarters/fishcreek folder. The images are fishcreeklogo.gif, home.gif, services.gif, askthevet.gif, and contact.gif. Save the files in your fishcreek4 folder.

Task 2: The Home Page. Launch a text editor, and open the index.html file from your fishcreek4 folder. Modify the index.html file to look similar to the web page shown in Figure 4.50.

1. Replace the “Fish Creek Animal Hospital” text contained within the h1 element with the fishcreeklogo.gif. Be sure to include the alt, height, and width attributes on the tag for the graphic.
2. Update the navigation area.
 - Since you will be replacing the top navigation with image links, it’s a good idea to provide for accessibility by including a set of text navigation links in the footer section of the web page. Copy the nav element, and paste it inside the footer area above the copyright line.
 - Refer to Figure 4.50, and replace the top navigation text hyperlinks with image links. The home.gif should link to index.html. The services.gif should link to services.html. The askthevet.gif should link to askvet.html. The contact.gif should link to contact.html. Use appropriate attributes on the tag: alt, height, and width.

Save and test your new index.html page. It will be similar to Figure 4.50, but you’ll notice that a few final touches (like the text shadow on the categories) are missing; you’ll configure these with CSS in Task 5.

Task 3: The Services Page. Launch a text editor, and open the services.html page from your fishcreek4 folder. Replace the “Fish Creek Animal Hospital” heading with the fishcreeklogo.gif. Configure the navigation areas in a similar way as the home page. Save and test your new services.html page.

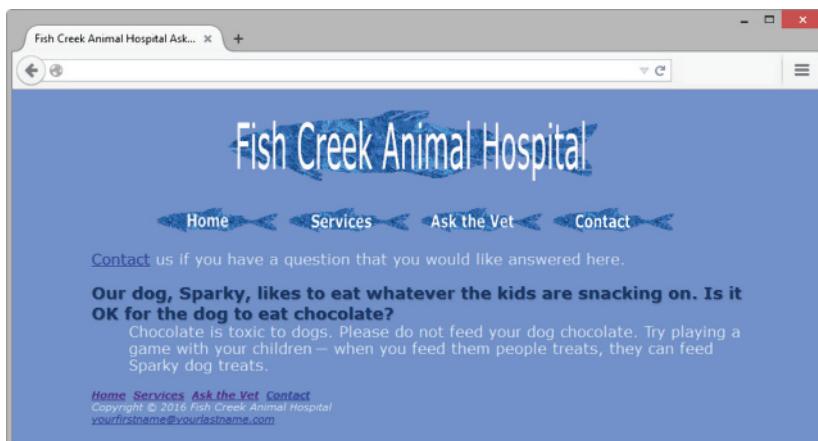


Figure 4.51 Fish Creek askvet.html

Task 4: The Ask the Vet Page. Use the Services page as the starting point for the Ask the Vet page. Launch a text editor, and open the services.html file in the fishcreek4 folder. Save the file as askvet.html. Modify the askvet.html file to look similar to the Ask the Vet page, as shown in Figure 4.51:

1. Change the page title to an appropriate phrase.
2. Delete the unordered list from the page.
3. The page content consists of a paragraph of text followed by a description list that contains a question and an answer.

a. Configure the text in the paragraph as follows:

Contact us if you have a question that you would like answered here.

b. The word “Contact” should link to the contact.html page.

c. The description list displays the question and answer. The `<dt>` element configures the question. Assign the `<dt>` element to the `category` class used on the Services page. The `<dd>` element configures the answer. The content of the description list follows:

Question: Our dog, Sparky, likes to eat whatever the kids are snacking on. Is it OK for the dog to eat chocolate?”

Answer: Chocolate is toxic to dogs. Please do not feed your dog chocolate. Try playing a game with your children — when you feed them people treats, they can feed Sparky dog treats.

d. *Hint:* See Appendix C, “Special Characters,” for the character code to display the em dash (—).

Save the askvet.html file. If you test your page in a browser, you’ll notice that it looks different from Figure 4.51—you still need to configure style rules.

Task 5: Configure the CSS. Open fishcreek.css in in a text editor. Edit the style rules as follows:

1. Modify the style rules for the `wrapper` id. Configure a minimum width of 700px (use `min-width`). Configure a maximum width of 1024px (use `max-width`).
2. Modify the style rules for the `h1` element selector. Delete the `line-height` style declaration. Add a new declaration to center the `h1` content (use `text-align:center`).
3. Modify the style rules for the `nav` element selector. Add a new declaration to center the text (use `text-align: center`).
4. Modify the `category` class to display text with a drop shadow (use `text-shadow: 1px 1px 1px #667`).
5. Add a new style rule for the `img` element selector to display no border.
6. Add a new style rule for the navigation in the footer area (use `footer nav` as the selector) to overrule the previous `nav` style rule and configure text to be left-aligned (use `text-align: left`).

Save the fishcreek.css file. Test your pages (`index.html`, `services.html`, and `askvet.html`) in a browser. If your images do not appear or your image links do not work, examine your work carefully. Use Windows Explorer or Mac Finder to verify that the images are saved in your `fishcreek4` folder. Examine the `src` attribute on the `` tags to be sure you

spelled the image names correctly. Another useful troubleshooting technique is to validate the HTML and CSS code. See Chapters 2 and 3 for Hands-On Practice exercises that describe how to use these validators.

Pacific Trails

See Chapter 2 for an introduction to the Pacific Trails Case Study. Figure 2.38 shows a site map for Pacific Trails. The Home page and Yurts page were created in earlier chapters. Using the existing website as a starting point, you will modify the design of the pages to display a large image on each page, as indicated in the wireframe in Figure 4.52. You will also create a new page, the Activities page. You have five tasks in this case study:

1. Create a new folder for this Pacific Trails case study, and obtain the starter image files.
2. Modify the Home page to display a logo image and scenic photograph as shown in Figure 4.53.
3. Modify the Yurts page to be consistent with the Home page.
4. Create a new Activities page, as shown in Figure 4.54.
5. Modify the style rules in the pacific.css file as needed.



Figure 4.52 New Pacific Trails wireframe

Hands-On Practice Case

Task 1: Create a folder on your hard drive or portable storage device called pacific4. Copy all the files from your Chapter 3 pacificcss folder into the pacific4 folder. Obtain the images used in this case study from the student files. The images are located in the chapter4/casestudystarters/pacific folder. The images are sunset.jpg, coast.jpg, yurt.jpg, trail.jpg, and background.jpg. Save the files in your pacific4 folder.

Task 2: The Home Page. Launch a text editor, and open the index.html file from your pacific4 folder. Modify the index.html file to look similar to the web page shown in Figure 4.53. Configure a div element to display the coast.jpg image. Code an opening div tag assigned to the id named `homehero` after the closing nav tag. Next, code a closing div tag. As shown in the wireframe in Figure 4.52, this div is located between the nav element and the main element. There is no HTML or text content for this div. The purpose of this div is to display a large image (configured with CSS in Task 5).

Save and validate your new index.html page. It will not yet be similar to Figure 4.53; you'll configure CSS in Task 5.

Task 3: The Yurts Page. Launch a text editor, and open the yurts.html page from your pacific4 folder. Configure a div element to display the yurt.jpg image. Code an opening div tag assigned to the id named `yurthero` after the closing nav tag. Next, code a closing div tag. As shown in the wireframe in Figure 4.52, this div is located between the nav element and the main element. There is no HTML or text content for this div. The purpose of this div is to display a large image (configured with CSS in Task 5). Save and validate your new yurts.html page.

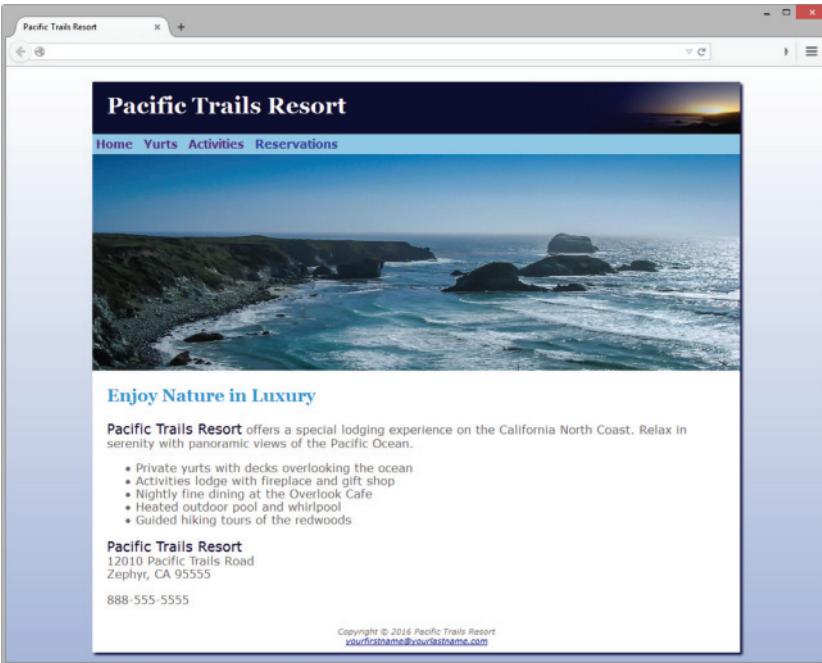


Figure 4.53 New Pacific Trails Resort Home page



Figure 4.54 Pacific Trails Resort activities.html

Task 4: The Activities Page. Use the Yurts page as the starting point for the Activities page. Launch a text editor, and open the yurts.html file in the pacific4 folder. Save the file as activities.html. Modify the activities.html file to look similar to the Activities page, as shown in Figure 4.54:

1. Change the page title to an appropriate phrase.
2. Change the text in the <h2> to be “Activities at Pacific Trails”.
3. Modify the div assigned to the id `yurthero`. Replace `yurthero` with `trailhero`.
4. Delete the description list from the page.
5. Configure the following text, using h3 tags for the headings and paragraph tags for the sentences:

Hiking

Pacific Trails Resort has 5 miles of hiking trails and is adjacent to a state park. Go it alone or join one of our guided hikes.

Kayaking

Ocean kayaks are available for guest use.

Bird Watching

While anytime is a good time for bird watching at Pacific Trails, we offer guided bird-watching trips at sunrise several times a week.

6. Configure a span element to contain the phrase “Pacific Trails Resort” in the first paragraph on the page. Assign the span to the class named `resort`.

Save the activities.html file. If you test your page in a browser, you’ll notice that it looks different from Figure 4.54; you still need to configure style rules.

Task 5: Configure the CSS. Open pacific.css in in a text editor. Edit the style rules as follows:

1. Modify the body element selector style rules. Configure background.jpg as the background image.
2. Modify the style rules for the `wrapper` id. Configure the background color to be `#fffffff`. Configure a minimum width of 700px (use `min-width`). Configure a maximum width of 1024px (use `max-width`). Use the `box-shadow` property to configure a drop-shadow effect.
3. Modify the style rules for the `h1` element selector. Configure the sunset.jpg as a background image that displays on the right and does not repeat. Also configure 20 pixels of left padding. Configure a 72-pixel height (the same height as the background image).
4. Modify the style rules for the `nav` element selector. Configure 5 pixels of padding.
5. Modify the style rules for the `footer` element selector. Configure 10 pixels of padding.
6. Add a new style rule for the `h3` element selector to display `#000033` text color.
7. Add a new style rule for the `main` element selector that configures 20 pixels of left and right padding. Also configure `display: block;` (this will prevent rendering issues in versions of Internet Explorer that do not support the HTML5 main element).

8. Add a new selector for an id named `homehero`. Code declarations to configure 300px height and to display the `coast.jpg` background image to fill the space (use `background-size: 100% 100%;`) without repeating.
9. Add a new selector for an id named `yurthero`. Code declarations to configure 300px height and to display the `yurt.jpg` background image to fill the space (use `background-size: 100% 100%;`) without repeating.
10. Add a new selector for an id named `trailhero`. Code declarations to configure 300px height and to display the `trail.jpg` background image to fill the space (use `background-size: 100% 100%;`) without repeating.
11. Code a new style rule to prevent the hyperlinks in the nav area from displaying the default underline. Use `nav a { text-decoration: none; }`
12. Have you noticed extra empty space between the logo header area and the navigation? Let's do something about that. We'll need to use the CSS margin property, which you'll explore in depth in Chapter 6. Modify the style rules for the `h1` selector to set the bottom margin to 0 with the following code:

```
margin-bottom: 0;
```

Save the `pacific.css` file. Test your pages (`index.html`, `yurts.html`, and `activities.html`) in a browser. The Home page (`index.html`) should look similar to Figure 4.53. The new Activities page (`activities.html`) should look similar to Figure 4.54. If your images do not appear, examine your work carefully. Use Windows Explorer or Mac Finder to verify that the images are saved in your `pacific4` folder. Examine the `src` attribute on the `` tags to be sure you spelled the image names correctly. Another useful troubleshooting technique is to validate the HTML and CSS code. See Chapters 2 and 3 for Hands-On Practice exercises that describe how to use these validators.

Path of Light Yoga Studio

See Chapter 2 for an introduction to the Path of Light Yoga Studio Case Study. Figure 2.42 shows a site map for Path of Light Yoga Studio. The Home page and Classes page were created in earlier chapters. Using the existing website as a starting point, you will modify the design of the pages and create a new page, the Schedule page. You have five tasks in this case study:

1. Create a new folder for this Path of Light Yoga Studio case study, and obtain the starter image files.
2. Modify the Home page to display as shown in Figure 4.55.
3. Modify the Classes page to display as shown in Figure 4.57.
4. Create a new Schedule page, as shown in Figure 4.58.
5. Modify the style rules in the `yoga.css` file as needed.

Hands-On Practice Case

Task 1: Create a folder on your hard drive or portable storage device called `yoga4`. Copy all the files from your Chapter 3 `yogacss` folder into the `yoga4` folder. Obtain the images used in this case study from the student files. The images are located in the `chapter4/casestudystarters/yoga` folder. The images are `lilyheader.jpg`, `yogadoor.jpg`, `yogalounge.jpg`, and `yogamat.jpg`. Save the files in your `yoga4` folder.

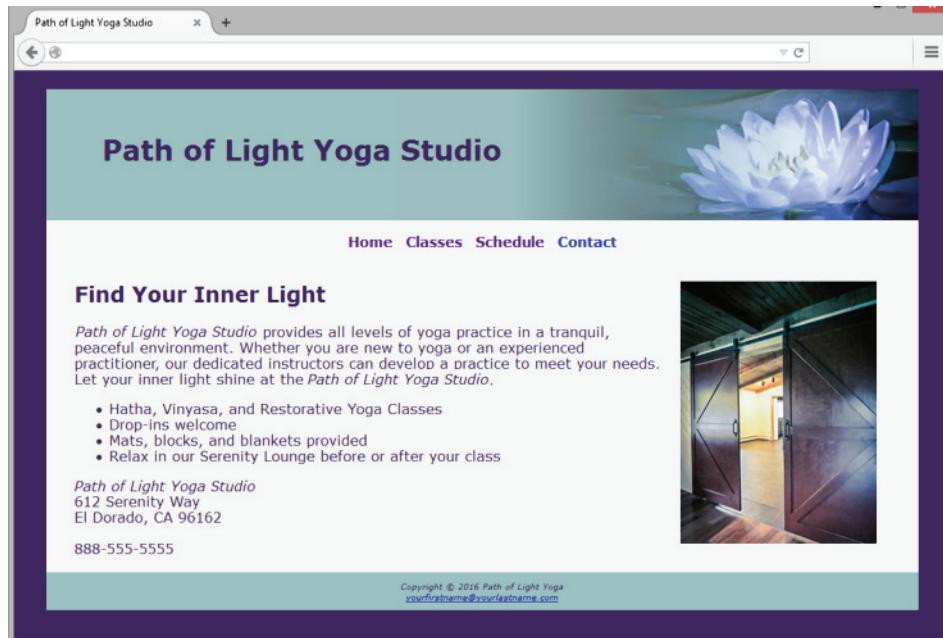


Figure 4.55 Path of Light Yoga Studio Home page

Task 2: The Home Page. Launch a text editor, and open the index.html file from your yoga4 folder. Modify the index.html file to look similar to the web page shown in Figure 4.55.

Code an img tag for the yogadoor.jpg image above the h2 element in the main content area. Be sure to include the alt, height, and width attributes. Also configure the image to appear to the right of the text by coding the align="right" attribute on the tag. Note: The W3C HTML validator will indicate that the align attribute is invalid. We'll ignore the error for this case study. In Chapter 6, you'll learn to use the CSS float property (instead of the align property) to configure this type of layout.

Save and test your new index.html page. It will be similar to Figure 4.55 but you'll notice that a few final touches (including dark page background and lily image in the header) are missing; you'll configure these with CSS in Task 5.

Task 3: The Classes Page. It's common for the content pages of a website to have a slightly different structure than the home page. The wireframe shown in Figure 4.56 depicts the structure of the Classes and Schedule pages. Launch a text editor, and open the classes.html page from your yoga4 folder. Configure a div element to display the yogamat.jpg image. As shown in the wireframe in Figure 4.56, this div is located within the main element. Code an opening div tag after the opening main tag. Assign the div to an id named hero. Code an img tag for the yogamat.jpg image. Be sure to include the alt, height, and width attributes. Next, code a closing div tag. Save and test your new classes.html page. If you test your page in a browser, you'll notice that it looks a bit different from Figure 4.57; you still need to configure style rules.



Figure 4.56 New Path of Light Yoga Studio wireframe

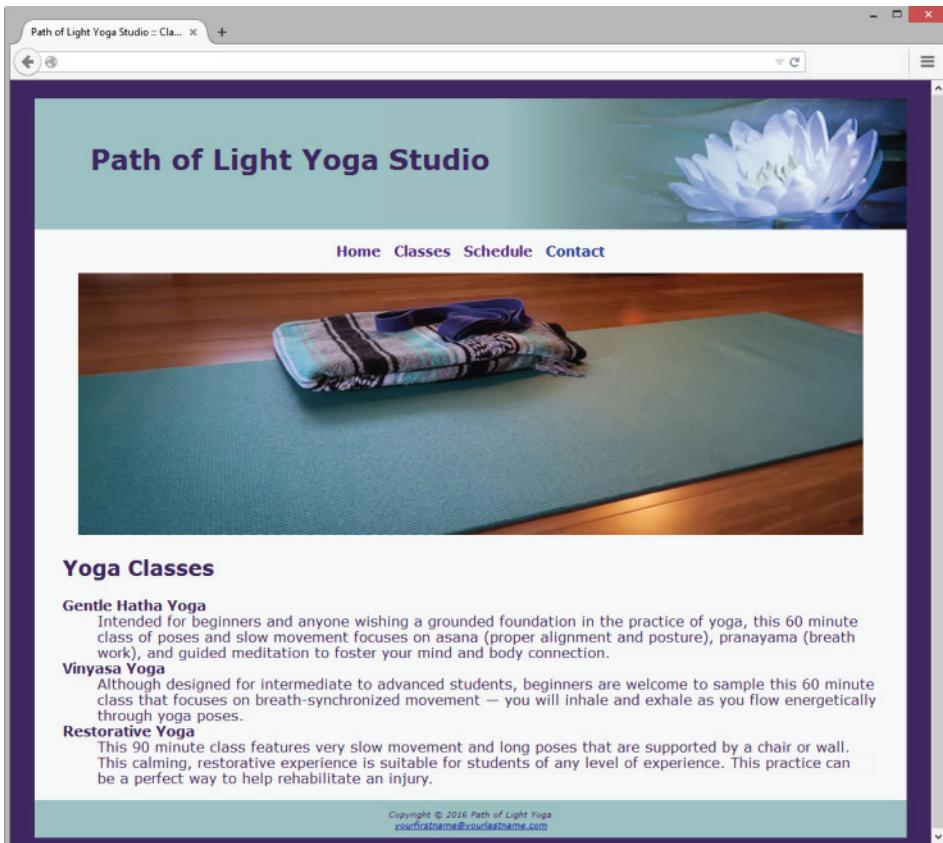


Figure 4.57 Path of Light Yoga Studio Classes Page

Task 4: The Schedule Page. Use the Classes page as the starting point for the Schedule page. Launch a text editor, and open the classes.html file in the yoga4 folder. Save the file as schedule.html. Modify your file to look similar to the Schedule page, as shown in Figure 4.58.

1. Change the title to an appropriate phrase.
2. Modify the img tag to display the yogalounge.jpg image. Configure appropriate alt text.
3. Replace the h2 element text “Yoga Classes” with “Yoga Schedule”.
4. Delete the description list.
5. Configure content for the Schedule Page page.
 - Configure a paragraph element that contains the following text:

Mats, blocks, and blankets provided. Please arrive 10 minutes before your class begins. Relax in our Serenity Lounge before or after your class.
 - Configure an h3 element with the following text:

Monday — Friday
 - Configure an unordered list with the following text:
 - 9:00am Gentle Hatha Yoga
 - 10:30am Vinyasa Yoga
 - 5:30pm Restorative Yoga
 - 7:00pm Gentle Hatha Yoga

- Configure an h3 element with the following text:
Saturday & Sunday
- Configure an unordered list with the following text:
10:30am Gentle Hatha Yoga
Noon Vinyasa Yoga
1:30pm Gentle Hatha Yoga
3:00pm Vinyasa Yoga
5:30 pm Restorative Yoga

Save the schedule.html file. If you test your page in a browser, you'll notice that it looks different from Figure 4.58; you still need to configure style rules.

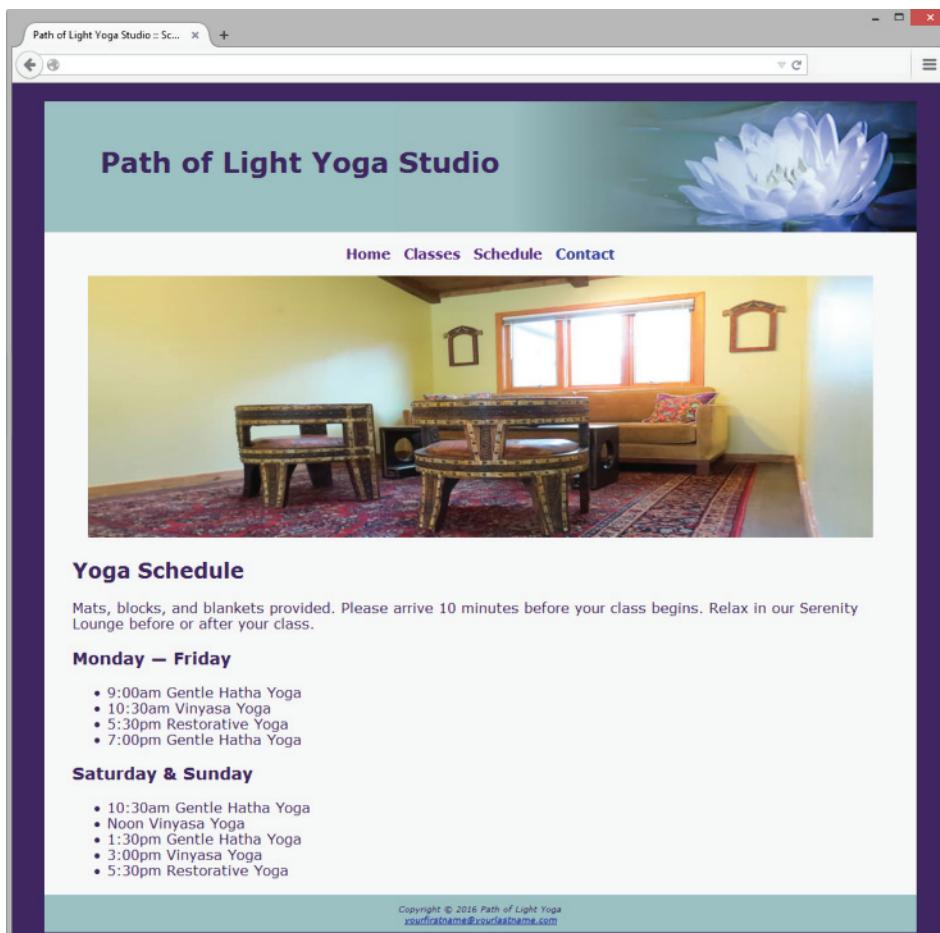


Figure 4.58 Path of Light Yoga Studio schedule.html

Task 5: Configure the CSS. Open yoga.css in a text editor. Edit the style rules as follows:

1. Modify the style rules for the body element selector to configure a very dark background color (#3F2860).
2. Modify the style rules for the #wrapper id. Configure #F5F5F5 as the background color. Configure a minimum width of 1000px (use min-width) and, a maximum width of 1280px (use max-width).

3. Modify the style rules for the header element selector. Remove the `text-align` declaration. Configure `lilyheader.jpg` as a background image that displays on the right without repeating. Set height to 150px.
4. Modify the style rules for the `h1` element selector. Remove the `line-height` declaration. Configure 50px top padding and 2em left padding.
5. Modify the style rules for the `nav` element selector. Configure 1em padding.
6. Modify the style rules for the `footer` element selector. Configure 1em padding.
7. Configure styles for the `main` element selector. Set left and right padding to 2em. You may need to nudge Internet Explorer to display the page as intended by adding the `display: block;` declaration (see Chapter 6).
8. Configure styles for the `img` element selector. Set left and right padding to 1em.
9. Configure styles for an id selector named `hero`. Set `text-align` to center.

Save the `yoga.css` file. Test your pages (`index.html`, `classes.html`, and `schedule.html`) in a browser. Your pages should be similar to Figures 4.55, 4.57, and 4.58. If your images do not appear or your image links do not function, examine your work carefully. Use Windows Explorer or Mac Finder to verify that the images are saved in your `yoga` folder. Examine the `src` attributes on the `` tags to be sure you spelled the image names correctly. Another useful troubleshooting technique is to validate the HTML and CSS code. See Chapters 2 and 3 for Hands-On Practice exercises that describe how to use these validators.