Linking Between Your Own Web Content ........................................... 141
Linking to External Web Content ...................................................... 144
Linking to an Email Address .............................................................. 145
Opening a Link in a New Browser Window ........................................ 146
Using CSS to Style Hyperlinks ............................................................ 147
Summary .......................................................................................... 151
Q&A ................................................................................................. 152
Workshop ......................................................................................... 153

HOUR 9: **Using Tables and Columns**
Creating a Simple Table ..................................................................... 156
Controlling Table Sizes ..................................................................... 160
Alignment and Spanning Within Tables .............................................. 163
Page Layout with Tables ..................................................................... 166
Using CSS Columns ........................................................................... 167
Summary .......................................................................................... 171
Q&A ................................................................................................. 173
Workshop ......................................................................................... 173

HOUR 10: **Creating Images for Use on the Web**
Choosing Graphics Software ............................................................... 175
The Least You Need to Know About Graphics .................................... 177
Preparing Photographic Images ......................................................... 178
Creating Banners and Buttons ............................................................ 184
Reducing or Removing Colors in an Image ........................................... 186
Creating Tiled Background Images ..................................................... 187
Creating Animated Web Graphics ....................................................... 189
Summary .......................................................................................... 190
Q&A ................................................................................................. 191
Workshop ......................................................................................... 191

HOUR 11: **Using Images in Your Website**
Placing Images on a Web Page ............................................................ 193
Describing Images with Text ............................................................... 195
Specifying Image Height and Width .................................................... 197
Aligning Images .................................................................................. 197
Turning Images into Links ................................................................... 201
Using Background Images ................................................................... 204
Using Imagemaps ............................................................................... 206
Summary .......................................................................................... 212
Q&A ................................................................................................. 214
Workshop ......................................................................................... 214

HOUR 12: **Using Multimedia in Your Website**
Linking to Multimedia Files .................................................................. 218
Embedding Multimedia Files ............................................................... 221

Using Pure HTML5 for Audio and Video Playback .................................. 224
Additional Tips for Using Multimedia .................................................. 227

HOUR 13: **Working with Margins, Padding, Alignment, and Floating**
Using Margins .................................................................................... 233
Padding Elements ............................................................................... 241
Keeping Everything Aligned ............................................................... 245
Understanding the *float* Property ..................................................... 246
Summary .......................................................................................... 249
Q&A ................................................................................................. 250
Workshop ......................................................................................... 250

HOUR 14: **Understanding the CSS Box Model and Positioning**
The CSS Box Model ............................................................................ 253
The Whole Scoop on Positioning .......................................................... 257
Controlling the Way Things Stack Up ................................................... 261
Managing the Flow of Text .................................................................. 264
Summary .......................................................................................... 265
Q&A ................................................................................................. 266
Workshop ......................................................................................... 266

HOUR 15: **Creating Fixed or Liquid Layouts**
Understanding Fixed Layouts .............................................................. 270
Understanding Liquid Layouts ............................................................. 272
Creating a Fixed/Liquid Hybrid Layout ............................................... 275
Considering a Responsive Web Design ................................................. 286
Summary .......................................................................................... 287
Q&A ................................................................................................. 288
Workshop ......................................................................................... 288

HOUR 16: **Using CSS to Do More with Lists**
HTML List Refresher ........................................................................... 291
How the CSS Box Model Affects Lists ................................................ 292
Placing List Item Indicators ................................................................. 295
Creating Image Maps with List Items and CSS .................................... 298
Summary .......................................................................................... 301
Q&A ................................................................................................. 302
Workshop ......................................................................................... 302
PART IV: Advanced Website Functionality and Management

HOUR 20: Creating Print-Friendly Web Pages
What Makes a Page Print-Friendly? 366
Applying a Media-Specific Stylesheet 369
Designing a Stylesheet for Print Pages 371
Viewing a Web Page in Print Preview 374
Summary 376
Q&A 377
Workshop 377

HOUR 21: Understanding Dynamic Websites and HTML5 Applications
Understanding the Different Types of Scripting 379
Including JavaScript in HTML 380
Displaying Random Content 382
Understanding the Document Object Model 387
Changing Images Based on User Interaction 388
Thinking Ahead to Developing HTML5 Applications 390
Summary 391
Q&A 392
Workshop 392

HOUR 22: Working with Web-Based Forms
How HTML Forms Work 395
Creating a Form 396
Accepting Text Input 401
Naming Each Piece of Form Data 402
Exploring Form Input Controls 405
Using HTML5 Form Validation 410
Submitting Form Data 412
Summary 413
Q&A 416
Workshop 416

HOUR 23: Organizing and Managing a Website
When One Page Is Enough 420
Organizing a Simple Site 422
Organizing a Larger Site 425
Writing Maintainable Code 429
Thinking About Version Control 432
Using HTML and CSS Frameworks 434
Summary 435
Q&A 436
Workshop 436

HOUR 24: Helping People Find Your Web Pages
Publicizing Your Website 439
Listing Your Pages with the Major Search Sites 441
Providing Hints for Search Engines 442
Additional Tips for Search Engine Optimization 448
Summary 450
Q&A 451
Workshop 451

Index 453
About the Author

Julie Meloni is a software development manager and technical consultant living in Washington, D.C. She has written several books and articles on web-based programming languages and database topics, including the bestselling *Sams Teach Yourself PHP, MySQL and Apache All in One*. 
We Want to Hear from You!

As the reader of this book, you are our most important critic and commentator. We value your opinion and want to know what we’re doing right, what we could do better, what areas you’d like to see us publish in, and any other words of wisdom you’re willing to pass our way.

You can email or write to let us know what you did or didn’t like about this book—as well as what we can do to make our books stronger.

Please note that we cannot help you with technical problems related to the topic of this book, and that due to the high volume of mail we receive, we might not be able to reply to every message.

When you write, please be sure to include this book’s title, edition number, and author, as well as your name and contact information.

Email: feedback@samspublishing.com
Mail: Sams Publishing
     800 East 96th Street
     Indianapolis, IN 46240 USA

Reader Services

Visit our website and register this book at informit.com/register for convenient access to any updates, downloads, or errata that might be available for this book.
This page intentionally left blank
Introduction

In 2012, more than 2.8 billion people had access to the Internet, including 255 million people in the United States alone. Throw in 567 million Chinese users, 69 million German users, 54 million British users, 76 million Russian users, and 99 million Brazilians, and you can see the meaning of the word world in the term World Wide Web. Many of these Internet users are also creating content for the web—you can be one of them! Although accurate measurements of the total number of web pages are difficult to come by, Google surpassed the 1 trillion mark of indexed pages in the middle of 2008.

In the next 24 hours, hundreds of millions of new pages will appear in accessible areas of the Internet. At least as many pages will be placed on private intranets, where they will be viewed by businesspeople connected via their local networks. Every one of those pages—like the billions of pages already online—will use Hypertext Markup Language (HTML).

As you complete the 24 one-hour lessons in this book, your web pages will be among those that appear on the Internet. These lessons also provide you with a foundation for developing one of the most valuable skills in the world today: mastery of HTML.

Can you really learn to create top-quality web pages yourself, without any specialized software, in less time than it takes to schedule and wait for an appointment with a highly paid design consultant? Can this relatively short, easy-to-read book really enable you to teach yourself state-of-the-art techniques for developing and publishing web content?

Yes. In fact, within the first two lessons in this book, someone with no previous HTML experience at all can have a web page ready to go online.

How can you learn the language of the web so quickly? By example. This book organizes lessons on the basics of HTML and CSS into simple steps and then shows you exactly how to tackle each step. Many of these HTML code examples are accompanied by pictures of the output produced by the code. You see how it’s done, you read a clear and concise explanation of how it works, and then you immediately do the same thing with your own page. A few minutes later, you’re on to the next step.

Soon you'll be marveling at your own impressive pages on the Internet.
**Beyond HTML**

This book covers more than just HTML because HTML isn’t the only thing you need to know to create web content today. The goal of this book is to give you a solid foundation in the skills you need to create a modern, standards-compliant website in just 24 short, easy lessons. This book covers the following key skills and technologies:

- HTML5 is the most current recommendation for web page creation. Every example in this book is validated HTML5.

- All the examples in the book have been tested for compatibility with the latest version of every major web browser. That includes Apple Safari, Google Chrome, Microsoft Internet Explorer, Mozilla Firefox, and Opera. You’ll learn from the start to be compatible with the past, yet ready for the future.

- The book has extensive coverage of the most current recommendation of Cascading Style Sheets (CSS), which allows you to carefully control the layout, fonts, colors, and formatting of every aspect of your web pages, including both text and images. When it comes to creating eye-popping web pages, CSS goes far beyond what traditional HTML pages could do by themselves.

- The technical stuff is not enough, so this book also includes the advice you need when setting up a website to achieve your goals. Key details—designing an effective page layout, posting your page to the Internet with FTP software, organizing and managing multiple pages, and getting your pages to appear high on the query lists at all the major Internet search sites—are all covered in enough depth to get you beyond the snags that often frustrate beginners.

Attention to many of these essentials are what made the first eight editions of this book bestsellers, and this updated edition is no different. All the examples have been updated, and significant portions of the content have been revised to match new examples and new technologies, fully embracing HTML5 and CSS3.

**Visual Examples**

Every example in this book is illustrated in two parts:

- The text you type to make an HTML page is shown first, with all HTML and CSS code highlighted.

- The resulting web page is shown as it will appear to users who view it with the world’s most popular web browsers.

You’ll often be able to adapt the example to your own pages without reading any of the accompanying text at all (although it is highly recommended that you read the rest of the text!).
All the examples in this book are standards compliant and work with Apple Safari, Google Chrome, Microsoft Internet Explorer, Mozilla Firefox, and Opera. Although all the screenshots were taken in Chrome, rest assured that all the code has been tested in all other browsers.

You will also find colors used within code examples and references to elements of code in the explanatory text. These colors highlight the different bits and pieces of code, both to enhance your familiarity with them and to call attention to their use.

- HTML tags appear in dark blue.
- HTML comments appear in grey.
- CSS elements appear in green.
- HTML attribute names appear in light blue.
- HTML attribute values appear in magenta.

Please note that the color of an item can change, depending on its context. For instance, when CSS elements are used within the style attribute of an HTML tag, they are color-coded as HTML attribute values (magenta) instead of CSS elements (green).

Special Elements

As you complete each lesson, margin notes help you immediately apply what you just learned to your own web pages.

Whenever a new term is used, it is clearly highlighted. No flipping back and forth to a glossary!

**TIP**
Tips and tricks to save you precious time are set aside in “Tip” boxes so that you can spot them quickly.

**NOTE**
Note boxes highlight interesting information you should be sure not to miss.

**CAUTION**
When you need to watch out for something, you’re warned about it in Caution boxes.
Q&A, Quiz, and Exercises

Every hour ends with a short question-and-answer session that addresses the kind of “dumb questions” everyone wishes they dared to ask. A brief but complete quiz lets you test yourself to be sure you understand everything presented in the hour. Finally, one or two optional exercises give you a chance to practice your new skills before you move on.
In the previous hour, you learned the basics of HTML, including how to set up a skeletal HTML template for all your web content. In this hour, you learn how to fine-tune the display of your web content using Cascading Style Sheets (CSS).

The concept behind stylesheets is simple: You create a stylesheet document that specifies the fonts, colors, spacing, and other characteristics that establish a unique look for a website. You then link every page that should have that look to the stylesheet instead of specifying all those styles repeatedly in each separate document. Therefore, when you decide to change your official corporate typeface or color scheme, you can modify all your web pages at once just by changing one or two entries in your stylesheet—you don’t have to change them in all your static web files. So a stylesheet is a grouping of formatting instructions that control the appearance of several HTML pages at once.

Stylesheets enable you to set a great number of formatting characteristics, including exact typeface controls, letter and line spacing, and margins and page borders, just to name a few. Stylesheets also enable you to specify sizes and other measurements in familiar units, such as inches, millimeters, points, and picas. In addition, you can use stylesheets to precisely position graphics and text anywhere on a web page, either at specific coordinates or relative to other items on the page.

In short, stylesheets bring a sophisticated level of display to the web. And they do so, if you’ll pardon the expression, with style.
How CSS Works

The technology behind stylesheets is called CSS, or Cascading Style Sheets. CSS is a language that defines style constructs such as fonts, colors, and positioning, which describe how information on a web page is formatted and displayed. CSS styles can be stored directly in an HTML web page or in a separate stylesheet file. Either way, stylesheets contain style rules that apply styles to elements of a given type. When used externally, stylesheet rules are placed in an external stylesheet document with the file extension .css.

A style rule is a formatting instruction that can be applied to an element on a web page, such as a paragraph of text or a link. Style rules consist of one or more style properties and their associated values. An internal stylesheet is placed directly within a web page, whereas an external stylesheet exists in a separate document and is simply linked to a web page via a special tag—more on this tag in a moment.

The cascading part of the name CSS refers to the manner in which stylesheets rules are applied to elements in an HTML document. More specifically, styles in a CSS stylesheet form a hierarchy in which more specific styles override more general styles. It is the responsibility of CSS to determine the precedence of style rules according to this hierarchy, which establishes a cascading effect. If that sounds a bit confusing, just think of the cascading mechanism in CSS as being similar to genetic inheritance, in which general traits are passed from parents to a child, but more specific traits are entirely unique to the child. Base style rules are applied throughout a stylesheet but can be overridden by more specific style rules.

A quick example should clear things up. Take a look at the following code to see whether you can tell what’s going on with the color of the text:

```html
<div style="color:green">
This text is green.
</div>

<p style="color:blue">This text is blue.</p>
<p>This text is still green.</p>
```

In the previous example, the color green is applied to the <div> tag via the color style property. Therefore, the text in the <div> tag is colored green. Because both <p> tags are children of the <div> tag, the green text style cascades down to them. However, the first <p> tag overrides...
the color style and changes it to blue. The end result is that the first line (not surrounded by a paragraph tag) is green, the first official paragraph is blue, and the second official paragraph retains the cascaded green color.

If you made it through that description on your own and came out on the other end unscathed, congratulations—that’s half the battle. Understanding CSS isn’t like understanding rocket science, and the more you practice, the more it will become clear. The real trick is developing the aesthetic design sense that you can then apply to your online presence through CSS.

Like many web technologies, CSS has evolved over the years. The original version of CSS, known as Cascading Style Sheets Level 1 (CSS1), was created in 1996. The later CSS 2 standard was created in 1998, and CSS 2 is still in use today; all modern web browsers support CSS 2. The latest version of CSS is CSS3, which builds on the strong foundation laid by its predecessors but adds advanced functionality to enhance the online experience. Throughout this book, you learn core CSS, including new elements of CSS3 that are applicable to the basic design and functionality that this text covers. So when I talk about CSS throughout the book, I’m referring to CSS3.

You’ll find a complete reference guide to CSS at http://www.w3.org/Style/CSS/. The rest of this hour explains the basics of putting CSS to good use.

**A Basic Stylesheet**

Despite their intimidating power, stylesheets can be simple to create. Consider the web pages shown in Figures 3.1 and 3.2. These pages share several visual properties that can be put into a common stylesheet:

- They use a large, bold Verdana font for the headings and a normal-size and -weight Verdana font for the body text.
- They use an image named logo.gif floating within the content and on the right side of the page.
- All text is black except for subheadings, which are purple.
- They have margins on the left side and at the top.
- They include vertical space between lines of text.
- They include a footer that is centered and in small print.
FIGURE 3.1
This page uses a stylesheet to fine-tune the appearance and spacing of the text and images.

About BAWSI
The Bay Area Women’s Sports Initiative (BAWSI) is a public benefit, nonprofit corporation with a mission to create programs and partnerships through which women athletes bring health, hope, and influence to our community. Founded in 2005 by Olympic and World Cup soccer stars Brandi Chastain and Julie Foudy and Marlene Bjornlund, former general manager of the San Jose CyberRays women’s professional soccer team, BAWSI provides a meaningful path for women athletes to become a more visible and valued part of the Bay Area sports culture.

BAWSI’s History
The concept of BAWSI was inspired by one of the most spectacular achievements in women’s sports history and born out of one of its biggest disappointments...

[continue reading]

Copyright © 2009–2013 BAWSI (www.bawsi.org). All rights reserved. Used with permission.

FIGURE 3.2
This page uses the same stylesheet as the one in Figure 3.1, thus maintaining a consistent look and feel.

About BAWSI: The History, continued
In the summer of 1999, the U.S. Women’s national soccer team defeated China to win the Women’s World Cup in the Rose Bowl, the climax of a three-week, 16-team tournament that remains history’s most successful women’s sports event, drawing more than 500,000 spectators. It also represented the most visible achievement of the women’s sports movement in the United States since the 1977 enactment of federal Title IX, which required schools and colleges to provide equal sports opportunities to girls and women. Athletes like Brandi Chastain and Julie Foudy, who led that team to the World Cup, spawned the Women’s United Soccer Association in 2001, the world’s first women’s professional soccer league. But in 2003, after only three seasons, the league folded, and the organizations created by some of the most accomplished athletes and administrators in women’s sports were disbanded.

The Bay Area Women’s Sports Initiative - BAWSI - arose from the remnants of that effort, created to ensure that the WUSA would not be the zenith of a short-lived golden age of women’s sports but rather another step toward women’s full achievement in the sports world. BAWSI’s programs began with a critical look at the unmet needs in our community and took shape after imagining a profound change for a better world.

[return to beginning]

Copyright © 2005–2013 BAWSI (www.bawsi.org). All rights reserved. Used with permission.
Listing 3.1 shows the CSS used in a stylesheet to specify these properties.

Listing 3.1   A Single External Stylesheet

```css
body {
    font-size: 10pt;
    font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;
    color: black;
    line-height: 14pt;
    padding-left: 5pt;
    padding-right: 5pt;
    padding-top: 5pt;
}

h1 {
    font: 14pt Verdana, Geneva, Arial, Helvetica, sans-serif;
    font-weight: bold;
    line-height: 20pt;
}

p.subheader {
    font-weight: bold;
    color: #593d87;
}

img {
    padding: 3pt;
    float: right;
}

a {
    text-decoration: none;
}

a:link, a:visited {
    color: #8094d6;
}

a:hover, a:active {
    color: #FF9933;
}

footer {
    font-size: 9pt;
    font-style: italic;
    line-height: 12pt;
    text-align: center;
    padding-top: 30pt;
}
```
This might initially appear to be a lot of code, but if you look closely, you’ll see that there isn’t a lot of information on each line of code. It’s fairly standard to place individual style rules on their own line, to help make style sheets more readable, but that is a personal preference; you could put all the rules on one line as long as you kept using the semicolon to separate each rule (more on that in a bit). Speaking of code readability, perhaps the first thing you noticed about this stylesheet code is that it doesn’t look anything like normal HTML code. CSS uses a syntax all its own to specify stylesheets.

Of course, the listing includes some familiar HTML tags (although not all tags require an entry in the stylesheet). As you might guess, body, h1, p, img, a, and footer in the stylesheet refer to the corresponding tags in the HTML documents to which the stylesheet will be applied. The curly braces after each tag name describe how all content within that tag should appear.

In this case, the stylesheet says that all body text should be rendered at a size of 10 points, in the Verdana font (if possible), and with the color black, with 14 points between lines. If the user does not have the Verdana font installed, the list of fonts in the stylesheet represents the order in which the browser should search for fonts to use: Geneva, then Arial, and then Helvetica. If the user has none of those fonts, the browser will use whatever default sans serif font is available. Additionally, the page should have left, right, and top padding of 5 points each.

Any text within an <h1> tag should be rendered in boldface Verdana at a size of 14 points. Moving on, any paragraph that uses only the <p> tag will inherit all the styles indicated by the body element. However, if the <p> tag uses a special class named subheader, the text will appear bold and in the color #593d87 (a purple color).

The pt after each measurement in Listing 3.1 means points (there are 72 points in an inch). If you prefer, you can specify any stylesheet measurement in inches (in), centimeters (cm), pixels (px), or “widths of a letter m,” which are called ems (em).

You might have noticed that each style rule in the listing ends with a semicolon (;). Semicolons are used to separate style rules from each other. It is therefore customary to end each style rule with a semicolon so that you can easily add another style rule after it. Review the remainder of the stylesheet in Listing 3.1 to see the presentation
formatting applied to additional tags. Don’t worry—you’ll learn more about each of these types of entries throughout the lessons in this book.

To link this stylesheet to HTML documents, include a `<link />` tag in the `<head>` section of each document. Listing 3.2 shows the HTML code for the page shown in Figure 3.1. It contains the following `<link />` tag:

```html
<link rel="stylesheet" type="text/css" href="styles.css" />
```

This assumes that the stylesheet is stored under the name `styles.css` in the same folder as the HTML document. As long as the web browser supports stylesheets—and all modern browsers do—the properties specified in the style sheet will apply to the content in the page without the need for any special HTML formatting code. This meets one of the goals of HTML, which is to provide a separation between the content in a web page and the specific formatting required to display that content.

**LISTING 3.2  HTML Code for the Page Shown in Figure 3.1**

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <title>About BAWSI</title>
  <link rel="stylesheet" type="text/css" href="styles.css" />
</head>
<body>
  <section>
    <header>
      <h1>About BAWSI</h1>
    </header>
    <p><img src="logo.gif" alt="BAWSI logo"/>The Bay Area Women's Sports Initiative (BAWSI) is a public benefit, nonprofit corporation with a mission to create programs and partnerships through which women athletes bring health, hope and wholeness to our community. Founded in 2005 by Olympic and World Cup soccer stars Brandi Chastain and Julie Foudy and Marlene Bjornsrud, former general manager of the San Jose CyberRays women's professional soccer team, BAWSI provides a meaningful path for women athletes to become a more visible and valued part of the Bay Area sports culture.</p>
    <p class="subheader">BAWSI's History</p>
  </section>
</body>
</html>
```
The concept of BAWSI was inspired by one of the most spectacular achievements in women’s sports history and born out of one of its biggest disappointments...

The code in Listing 3.2 is interesting because it contains no formatting of any kind. In other words, nothing in the HTML code dictates how the text and images are to be displayed—no colors, no fonts, nothing. Yet the page is carefully formatted and rendered to the screen, thanks to the link to the external stylesheet, styles.css. The real benefit to this approach is that you can easily create a site with multiple pages that maintain a consistent look and feel. And you have the benefit of isolating the visual style of the page to a single document (the stylesheet) so that one change impacts all pages.

Starting from scratch, create a new text document called mystyles.css and add some style rules for the following basic HTML tags: <body>, <p>, <h1>, and <h2>. After creating your stylesheet, make a new HTML file that contains these basic tags. Play around with different style rules, and see for yourself how simple it is to change entire blocks of text in paragraphs with one simple change in a stylesheet file.

A CSS Style Primer

You now have a basic knowledge of CSS stylesheets and how they are based on style rules that describe the appearance of information in web pages. The next few sections of this hour provide a quick overview of some of the most important style properties and allow you to get started using CSS in your own stylesheets.
CSS includes various style properties that are used to control fonts, colors, alignment, and margins, to name just a few. The style properties in CSS can be generally grouped into two major categories:

- **Layout properties**, which consist of properties that affect the positioning of elements on a web page, such as margins, padding, and alignment.
- **Formatting properties**, which consist of properties that affect the visual display of elements within a website, such as the font type, size, and color.

### Basic Layout Properties

CSS layout properties determine how content is placed on a web page. One of the most important layout properties is the `display` property, which describes how an element is displayed with respect to other elements. The `display` property has four basic values:

- **block**—The element is displayed on a new line, as in a new paragraph.
- **list-item**—The element is displayed on a new line with a list-item mark (bullet) next to it.
- **inline**—The element is displayed inline with the current paragraph.
- **none**—The element is not displayed; it is hidden.

Understanding the `display` property is easier if you visualize each element on a web page occupying a rectangular area when displayed—the `display` property controls the manner in which this rectangular area is displayed. For example, the `block` value results in the element being placed on a new line by itself, whereas the `inline` value places the element next to the content just before it. The `display` property is one of the few style properties that can be applied in most style rules. Following is an example of how to set the `display` property:

```css
display:block;
```

You control the size of the rectangular area for an element with the `width` and `height` properties. As with many size-related CSS properties, `width` and `height` property values can be specified in several different units of measurement:

- **in**—Inches
- **cm**—Centimeters

**NOTE**

The `display` property relies on a concept known as *relative positioning*, which means that elements are positioned relative to the location of other elements on a page. CSS also supports *absolute positioning*, which enables you to place an element at an exact location on a page, independent of other elements. You’ll learn more about both of these types of positioning in Hour 14, “Understanding the CSS Box Model and Positioning.”
You can mix and match units however you choose within a stylesheet, but it's generally a good idea to be consistent across a set of similar style properties. For example, you might want to stick with points for font properties and pixels for dimensions. Following is an example of setting the width of an element using pixel units:

```css
width: 200px;
```

### Basic Formatting Properties

CSS formatting properties to control the appearance of content on a web page, as opposed to controlling the physical positioning of the content. One of the most popular formatting properties is the `border` property, which establishes a visible boundary around an element with a box or partial box. Note that a border is always present in that space is always left for it, but the border does not appear in a way that you can see unless you give it properties that make it visible (like a color). The following `border` properties provide a means of describing the borders of an element:

- `border-width`—The width of the border edge
- `border-color`—The color of the border edge
- `border-style`—The style of the border edge
- `border-left`—The left side of the border
- `border-right`—The right side of the border
- `border-top`—The top of the border
- `border-bottom`—The bottom of the border
- `border`—All the border sides

The `border-width` property establishes the width of the border edge. It is often expressed in pixels, as the following code demonstrates:

```css
border-width: 5px;
```
Not surprisingly, the `border-color` and `border-style` properties set the border color and style. Following is an example of how these two properties are set:

```css
border-color: blue;
banner-style: dotted;
```

The `border-style` property can be set to any of the following basic values (you learn about some more advanced border tricks later in this book):

- `solid` — A single-line border
- `double` — A double-line border
- `dashed` — A dashed border
- `dotted` — A dotted border
- `groove` — A border with a groove appearance
- `ridge` — A border with a ridge appearance
- `inset` — A border with an inset appearance
- `outset` — A border with an outset appearance
- `none` — No border
- `hidden` — Effectively the same as `none`

The default value of the `border-style` property is `none`, which is why elements don’t have a border unless you set the border property to a different style. Although `solid` is the most common border style, you will also see the other styles in use.

The `border-left`, `border-right`, `border-top`, and `border-bottom` properties enable you to set the border for each side of an element individually. If you want a border to appear the same on all four sides, you can use the single `border` property by itself, which expects the following styles separated by a space: `border-width`, `border-style`, and `border-color`. Following is an example of using the `border` property to set a border that consists of two (double) red lines that are a total of 10 pixels in width:

```css
border: 10px double red;
```

Whereas the color of an element’s border is set with the `border-color` property, the color of the inner region of an element is set using the `color` and `background-color` properties. The `color` property sets the color
of text in an element (foreground), and the `background-color` property sets the color of the background behind the text. Following is an example of setting both color properties to predefined colors:

```html
color:black;
background-color:orange;
```

You can also assign custom colors to these properties by specifying the colors in hexadecimal (covered in more detail in Hour 7, “Working with Colors and Borders”) or as RGB (Red Green Blue) decimal values:

```html
background-color:#999999;
color:rgb(0,0,255);
```

You can also control the alignment and indentation of web page content without too much trouble. This is accomplished with the `text-align` and `text-indent` properties, as the following code demonstrates:

```html
text-align:center;
text-indent:12px;
```

When you have an element properly aligned and indented, you might be interested in setting its font. The following basic font properties set the various parameters associated with fonts (you’ll learn about some more advanced font usage in Hour 6, “Working with Fonts”):

- **font-family**—The family of the font
- **font-size**—The size of the font
- **font-style**—The style of the font (normal or italic)
- **font-weight**—The weight of the font (normal, lighter, bold, bolder, and so on)

The `font-family` property specifies a prioritized list of font family names. A prioritized list is used instead of a single value to provide alternatives in case a font isn’t available on a given system. The `font-size` property specifies the size of the font using a unit of measurement, usually points. Finally, the `font-style` property sets the style of the font, and the `font-weight` property sets the weight of the font. Following is an example of setting these font properties:

```html
font-family: Arial, sans-serif;
font-size: 36pt;
font-style: italic;
font-weight: normal;
```
Now that you know a whole lot more about style properties and how they work, refer back at Listing 3.1 and see whether it makes a bit more sense. Here’s a recap of the style properties used in that style sheet, which you can use as a guide for understanding how it works:

- **font**—Lets you set many font properties at once. You can specify a list of font names separated by commas; if the first is not available, the next is tried, and so on. You can also include the words `bold` and/or `italic` and a font size. Alternatively, you can set each of these font properties separately with `font-family`, `font-size`, `font-weight`, and `font-style`.

- **line-height**—Also known in the publishing world as *leading*. This sets the height of each line of text, usually in points.

- **color**—Sets the text color using the standard color names or hexadecimal color codes (see Hour 7 for more details).

- **text-decoration**—Useful for turning off link underlining—simply set it to `none`. The values of `underline`, `italic`, and `line-through` are also supported. Hour 8, “Using External and Internal Links,” covers applying styles to links in more detail.

- **text-align**—Aligns text to the `left`, `right`, or `center`, along with justifying the text with a value of `justify`.

- **padding**—Adds padding to the left, right, top, and bottom of an element; this padding can be in measurement units or a percentage of the page width. Use `padding-left` and `padding-right` if you want to add padding to the left and right of the element independently. Use `padding-top` or `padding-bottom` to add padding to the top or bottom of the element, as appropriate. You learn more about these style properties in Hour 13, “Working with Margins, Padding, Alignment, and Floating,” and Hour 14, “Understanding the CSS Box Model and Positioning.”

### Using Style Classes

This is a “teach yourself” book, so you don’t have to go to a single class to learn how to give your pages great style—although you do need to learn what a style class is. Whenever you want some of the text on your pages to look different from the other text, you can create what amounts to a custom-built HTML tag. Each type of specially formatted text you define is called a *style class*. A style class is a custom set of formatting specifications that can be applied to any element in a web page.
Before showing you a style class, I need to take a quick step back and clarify some CSS terminology. First off, a CSS **style property** is a specific style that you can assign a value, such as `color` or `font-size`. You associate a style property and its respective value with elements on a web page by using a selector. A **selector** is used to identify tags on a page to which you apply styles. Following is an example of a selector, a property, and a value all included in a basic style rule:

```
h1 { font: 36pt Courier; }
```

In this code, `h1` is the selector, `font` is the style property, and `36pt Courier` is the value. The selector is important because it means that the font setting will be applied to all `h1` elements in the web page. But maybe you want to differentiate between some of the `h1` elements—what then? The answer lies in style classes.

Suppose you want two different kinds of `<h1>` headings for use in your documents. You create a style class for each one by putting the following CSS code in a stylesheet:

```
h1.silly { font: 36pt Comic Sans; }
h1.serious { font: 36pt Arial; }
```

Notice that these selectors include a period (.) after `h1`, followed by a descriptive class name. To choose between the two style classes, use the `class` attribute, like this:

```
<h1 class="silly">Marvin's Munchies Inc. </h1>
<p>Text about Marvin's Munchies goes here. </p>
```

Or you could use this:

```
<h1 class="serious">MMI Investor Information</h1>
<p>Text for business investors goes here.</p>
```

When referencing a style class in HTML code, simply specify the class name in the `class` attribute of an element. In the previous example, the words *Marvin's Munchies Inc.* would appear in a 36-point Comic Sans font, assuming that you included a `<link />` to the stylesheet at the top of the web page and that the user has the Comic Sans font installed. The words *MMI Investor Information* would appear in the 36-point Arial font instead. You can see another example of classes in action in Listing 3.2: look for the `subheader <p>` class.

What if you want to create a style class that can be applied to any element instead of just headings or some other particular tag? In your
CSS, simply use a period (.) followed by any style class name you make up and any style specifications you choose. That class can specify any number of font, spacing, and margin settings all at once. Wherever you want to apply your custom tag in a page, just use an HTML tag plus the `class` attribute, followed by the class name you created.

For example, the stylesheet in Listing 3.1 includes the following style class specification:

```css
p.subheader {
    font-weight: bold;
    color:#593d87;
}
```

This style class is applied in Listing 3.2 with the following tag:

```html
<p class="subheader">
```

Everything between that tag and the closing `</p>` tag in Listing 3.2 appears in bold purple text.

What makes style classes so valuable is how they isolate style code from web pages, effectively allowing you to focus your HTML code on the actual content in a page, not how it is going to appear on the screen. Then you can focus on how the content is rendered to the screen by fine-tuning the stylesheet. You might be surprised by how a relatively small amount of code in a stylesheet can have significant effects across an entire website. This makes your pages much easier to maintain and manipulate.

### Using Style IDs

When you create custom style classes, you can use those classes as many times as you like—they are not unique. However, in some instances, you want precise control over unique elements for layout or formatting purposes (or both). In such instances, look to IDs instead of classes.

A *style ID* is a custom set of formatting specifications that can be applied only to one element in a web page. You can use IDs across a set of pages, but only once per time within each page.

For example, suppose you have a title within the body of all your pages. Each page has only one title, but all the pages themselves

---

**TIP**

You might have noticed a change in the coding style when a style rule includes multiple properties. For style rules with a single style, you’ll commonly see the property placed on the same line as the rule, like this:

```css
p.subheader { font-weight: bold; }
```

However, when a style rule contains multiple style properties, it’s much easier to read and understand the code if you list the properties one per line, like this:

```css
p.subheader {
    font-weight: bold;
    color:#593d87;
}
```
include one instance of that title. Following is an example of a selector with an ID indicated, plus a property and a value:

```
p#title {font: 24pt Verdana, Geneva, Arial, sans-serif}
```

Notice that this selector includes a hash mark, or pound sign (#), after `p`, followed by a descriptive ID name. When referencing a style ID in HTML code, simply specify the ID name in the `id` attribute of an element, like so:

```
<p id="title">Some Title Goes Here</p>
```

Everything between the opening and closing `<p>` tags will appear in 24-point Verdana text—but only once on any given page. You often see style IDs used to define specific parts of a page for layout purposes, such as a header area, footer area, main body area, and so on. These types of areas in a page appear only once per page, so using an ID rather than a class is the appropriate choice.

### Internal Stylesheets and Inline Styles

In some situations, you want to specify styles that will be used in only one web page. You can enclose a stylesheet between `<style>` and `</style>` tags and include it directly in an HTML document. Stylesheets used in this manner must appear in the `<head>` of an HTML document. No `<link />` tag is needed, and you cannot refer to that stylesheet from any other page (unless you copy it into the beginning of that document, too). This kind of stylesheet is known as an internal stylesheet, as you learned earlier in the hour.

Listing 3.3 shows an example of how you might specify an internal stylesheet.

**Listing 3.3  A Web Page with an Internal Stylesheet**

```html
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Some Page</title>

    <style type="text/css">
      footer {
        font-size: 9pt;
      }
    
```
In the listing code, the footer style class is specified in an internal stylesheet that appears in the head of the page. The style class is now available for use within the body of this page. In fact, it is used in the body of the page to style the copyright notice.

Internal stylesheets are handy if you want to create a style rule that is used multiple times within a single page. However, in some instances, you might need to apply a unique style to one particular element. This calls for an inline style rule, which allows you to specify a style for only a small part of a page, such as an individual element. For example, you can create and apply a style rule within a <p>, <div>, or <span> tag via the style attribute. This type of style is known as an inline style because it is specified right there in the middle of the HTML code.

Here’s how a sample style attribute might look:

```html
<p style="color:green">
  This text is green, but <span style="color:red">this text is red</span>.
  Back to green again, but...
</p>
<p>
  ...now the green is over, and we're back to the default color for this page.
</p>
```

This code makes use of the <span> tag to show how to apply the color style property in an inline style rule. In fact, both the <p> tag and the <span> tag in this example use the color property as an inline style. What’s important to understand is that the color:red style property overrides the color:green style property for the text between the <span> and </span> tags. Then in the second paragraph, neither of the color styles applies because it is a completely new paragraph that adheres to the default color of the entire page.

NOTE

<span> and </span> are dummy tags that do nothing in and of themselves except specify a range of content to apply any style attributes that you add. The only difference between <div> and <span> is that <div> is a block element and, therefore, forces a line break, whereas <span> doesn’t. Therefore, you should use <span> to modify the style of any portion of text that is to appear in the middle of a sentence or paragraph without any line break.

CAUTION

Using inline styles isn’t considered a best practice when used beyond page-level debugging or the process of trying out new things in a controlled setting. The best practice of all is having your pages link to a centrally maintained stylesheet so that changes are immediately reflected in all pages that use it.
Validate Your Stylesheets

Just as it is important to validate your HTML or XHTML markup, it is important to validate your stylesheet. You can find a specific validation tool for CSS at http://jigsaw.w3.org/css-validator/. As with the validation tool discussed in Hour 2, “Structuring an HTML Document,” you can point the tool to a web address, upload a file, or paste content into the form field provided. The ultimate goal is a result like the one in Figure 3.3: valid!

Summary

In this hour, you learned that a stylesheet can control the appearance of many HTML pages at once. It can also give you extremely precise control over the typography, spacing, and positioning of HTML elements. You also learned that, by adding a style attribute to almost any HTML tag, you can control the style of any part of an HTML page without referring to a separate stylesheet document.

You learned about three main approaches to including stylesheets in your website: a separate stylesheet file with the extension .css that is linked to in the <head> of your documents, a collection of style rules placed in the head of the document within the <style> tag, and rules placed directly in an HTML tag via the style attribute (although the latter is not a best practice for long-term use).
Table 3.1 summarizes the tags discussed in this hour. Refer to the CSS stylesheet standards at http://www.w3c.org for details on what options can be included after the `<style>` tag or the `style` attribute.

<table>
<thead>
<tr>
<th>Tag/Attributes</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;style&gt;...&lt;/style&gt;</code></td>
<td>Allows an internal stylesheet to be included within a document. Used between <code>&lt;head&gt;</code> and <code>&lt;/head&gt;</code>.</td>
</tr>
</tbody>
</table>

**Attribute**

- **type=**"contenttype"  
  The Internet content type. (Always "text/css" for a CSS stylesheet.)

- `<link />`  
  Links to an external stylesheet (or other document type). Used in the `<head>` section of the document.

**Attribute**

- **href=**"url"  
  The address of the stylesheet.

- **type=**"contenttype"  
  The Internet content type. (Always "text/css" for a CSS stylesheet.)

- **rel=**"stylesheet"  
  The link type. (Always "stylesheet" for stylesheets.)

- `<span>...</span>`  
  Does nothing but provide a place to put `style` or other attributes. (Similar to `<div>...</div>`, but does not cause a line break.)

**Attribute**

- **style=**"style"  
  Includes inline style specifications. (Can be used in `<span>`, `<div>`, `<body>`, and most other HTML tags.)
Q&A

Q. Say I link a stylesheet to my page that says all text should be blue, but there’s a `<span style="font-color:red">` tag in the page somewhere. Will that text display as blue or red?

A. Red. Local inline styles always take precedence over external stylesheets. Any style specifications you put between `<style>` and `</style>` tags at the top of a page also take precedence over external stylesheets (but not over inline styles later in the same page). This is the cascading effect of stylesheets that I mentioned earlier in the hour. You can think of cascading style effects as starting with an external stylesheet, which is overridden by an internal stylesheet, which is overridden by inline styles.

Q. Can I link more than one stylesheet to a single page?

A. Sure. For example, you might have a sheet for formatting (text, fonts, colors, and so on) and another one for layout (margins, padding, alignment, and so on)—just include a `<link />` for both. Technically, the CSS standard requires web browsers to give the user the option to choose between stylesheets when multiple sheets are presented via multiple `<link />` tags. However, in practice, all major web browsers simply include every stylesheet unless it has a `rel="alternate"` attribute. The preferred technique for linking in multiple stylesheets involves using the special `@import` command. The following is an example of importing multiple stylesheets with `@import`:

```plaintext
@import url(styles1.css);
@import url(styles2.css);
```

Similar to the `<link />` tag, the `@import` command must be placed in the head of a web page. You learn more about this handy little command in Hour 20, “Creating Print-Friendly Web Pages,” when you learn how to create a stylesheet specifically for printing web pages.
Workshop

The Workshop contains quiz questions and exercises to help you solidify your understanding of the material covered. Try to answer all questions before looking at the “Answers” section that follows.

Quiz

1. What code would you use to create a stylesheet to specify 30-point blue Arial headings and all other text in 10-point blue Times Roman (or the default browser font)?

2. If you saved the stylesheet you made for Question 1 as corporate.css, how do you apply it to a web page named intro.html?

3. How many different ways are there to ensure that style rules can be applied to your content?

Answers

1. Your stylesheet would include the following:

```html
h1 { font: 30pt blue Arial; }  
body { font: 10pt blue "Times New Roman"; }
```

2. Put the following tag between the `<head>` and `</head>` tags of the intro.html document:

```html
<link rel="stylesheet" type="text/css" href="corporate.css"/>
```

3. Three: externally, internally, and inline.

Exercises

- Using the stylesheet you created earlier in this hour, add some style classes to your stylesheet. To see the fruits of your labor, apply those classes to the HTML page you created as well. Use classes with your `<h1>` and `<p>` tags to get the feel for things.

- Develop a standard stylesheet for your website, and link it into all your pages. (Use internal stylesheets and/or inline styles for pages that need to deviate from it.) If you work for a corporation, chances are, it has already developed font and style specifications for printed materials. Get a copy of those specifications, and follow them for company web pages, too.
Be sure to explore the official stylesheet specs at http://www.w3.org/Style/CSS/, and try some of the more esoteric style properties not covered in this hour.
<table>
<thead>
<tr>
<th>Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acrobat (Adobe), converting PDF files, 375</td>
</tr>
<tr>
<td>Adjust Hue/Lightness/Saturation tool (GIMP), 182</td>
</tr>
<tr>
<td>aligning images and text, 86</td>
</tr>
<tr>
<td>block-level elements, 87-89</td>
</tr>
<tr>
<td>tables, 163-165</td>
</tr>
<tr>
<td>text-align property (CSS), 59</td>
</tr>
<tr>
<td>text-align style rule, 88-89</td>
</tr>
<tr>
<td>text-decoration property (CSS), 59</td>
</tr>
<tr>
<td>alternate text, 195-196</td>
</tr>
<tr>
<td>Amazon.com, website organization, 425-426</td>
</tr>
<tr>
<td>analogous color schemes, 120</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Small Orange web hosting provider, 11</td>
</tr>
<tr>
<td>anchor tags (HTML), 203</td>
</tr>
<tr>
<td>anchor locations, linking to, 138-141</td>
</tr>
<tr>
<td>naming, 140</td>
</tr>
<tr>
<td>web pages, identifying locations within, 138</td>
</tr>
<tr>
<td>absolute addresses, web pages and, 136-137</td>
</tr>
<tr>
<td>absolute links, web pages and, 136</td>
</tr>
<tr>
<td>absolute positioning, 257</td>
</tr>
<tr>
<td>display property (CSS), 55</td>
</tr>
<tr>
<td>positioning property, 258-261</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ASCII text</th>
</tr>
</thead>
<tbody>
<tr>
<td>formatted text, 102</td>
</tr>
<tr>
<td>web content, creating, 30, 37</td>
</tr>
<tr>
<td>&lt;aside&gt; semantic element (HTML5), 69-72, 79</td>
</tr>
<tr>
<td>&lt;aside&gt; tags, HTML5, 41</td>
</tr>
<tr>
<td>.asp file extensions, 31</td>
</tr>
<tr>
<td>attributes, 135</td>
</tr>
<tr>
<td>default values, 88</td>
</tr>
<tr>
<td>defining, 86</td>
</tr>
<tr>
<td>HTML style attribute, text-align style rule and, 88-89</td>
</tr>
<tr>
<td>audio</td>
</tr>
<tr>
<td>&lt;audio&gt; element, embedding multimedia files, 224-225</td>
</tr>
<tr>
<td>HTML5 audio playback, 224-225</td>
</tr>
<tr>
<td>auto image loading, web browsers, 24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;b&gt; tags (HTML), 102, 105-106</td>
</tr>
<tr>
<td>backgrounds</td>
</tr>
<tr>
<td>background-image style property, 204</td>
</tr>
<tr>
<td>background-position style property, 205</td>
</tr>
<tr>
<td>background-repeat style property, 205</td>
</tr>
<tr>
<td>color</td>
</tr>
<tr>
<td>background-color property (CSS), 58</td>
</tr>
</tbody>
</table>
background-color style property, 204
CSS and, 125-127
images, 204-205
tiled backgrounds, 187-189
bad website examples, 122
bandwidth, web hosting providers, 10
banners, creating, 184-186
BAWSI.org, website organization, 428
Bing, listing websites with, 442
block-level elements, aligning text in, 87-89
block value (CSS display property), 55
blogs, 24
<body> tags, 31, 34, 37
boldface text, 105-106
Bootstrap framework, 434
borders
CSS, 254
border-bottom property, 56-57
border-color property, 56-57
border-left property, 56-57
border property, 56-57
border-right property, 56-57
border-style property, 56-57
border-top property, 56-57
border-width property, 56-57
content, 254
create, 158-159
spacing, 166
box model (CSS), 253, 256
borders, 254
content, 254
lists and, 292-295
margins, 254
padding, 254
sizing elements, 254-255
<br> tags, 36-37, 108
browsers
140 cross-browser color names, 121
Chrome, 13
CSS support, 54
development of, 6
differences in, accounting for, 341
distributing, 23
Firefox, 13
helper applications, defining, 219
history of, 6
HTML development, 6
images, auto image loading, 24
Internet Explorer, 13, 256
links, opening in new browser windows, 146-147
margins and, 292-294
non-viewable window areas, 270
Opera, 13
servers, basic browser server interaction, 7-9
text, adjusting font size settings, 25
vendor prefixes, 341
web content, displaying, 92
websites
comparing, 30
testing, 12-13, 30
bulleted lists, 94
buttons, creating, 184-186
buying domain names, 11
categorizing elements, 75
CD-ROM, transferring photos to, 178
cells (tables)
creating, 156
sizing, 161-162
character entity, 103
check boxes (forms), 405-406
child (nested) tags, 92
Chrome web browser (Google), 13
clarity, coding for, 431
Classic FTP FTP client, 15
server connections, 18
website connections, 16-17
clear property (CSS), text flow and, 264
client-side scripting, 380, 390
clip art, 176
closing slashes (HTML tags), 36
closing tags (HTML), 33-34
cm value (CSS height/width properties), 55
color
140 cross-browser color names, 121
analogous color schemes, 120
background color
background-color style property, 204
CSS and, 125-127
best practices, 119-121
border color (tables), CSS and, 125-127
color property (CSS), 59
color style rule (CSS), fonts and, 109
color theory, 120
color wheel, 120
Colorblind Web Page Filter tool, 127
complementary color schemes, 121
graphics, adjusting color in, 182
hexadecimal color codes, 122-124
links and, 122
monitors and, 122
rounded corners, creating, 128-130
tables and, 165-166
text
CSS and, 125-127
formatting in style sheets, 48
triadic color schemes, 121
using, 119-121
W3C color standards, 121
columns
CSS, 167-171
fixed/liquid hybrid layouts
defining columns in, 277-279
setting column height in, 280-284
comments
JavaScript code comments, 381, 384
websites, maintaining code via, 429-430
complementary color schemes, 121
compression
graphics, 177
JPEG files, 183-184
concatenation operators
(JavaScript), 385
conceptualizing web pages, 69-74
control panels, selecting web hosting providers, 11-12
copyrights
copyright symbol, 103
graphics and, 176
corners (rounded), creating, 128-130
Creative Commons licenses, 176
cropping images, 178-179
CSS
align property, 233, 245
Bootstrap framework, 434
box model, 253, 256
borders, 254
content, 254
lists and, 292-295
margins, 254
padding, 254
sizing elements, 254-255
browser support for, 54
cascading component of, 48
clear property, text flow and, 264
color, specifying via CSS, 125-127
columns, 167-171
creating, 49, 52-54
CSS Zen Garden layouts, 270
CSS1, 49
CSS2, 49
CSS3, 49
definition of, 47-48
<div> tags, 48, 63
DOCTYPE declarations, 256
external style sheets, 48, 51-54
float property, 233, 246-249, 264
formatting properties, 55
background-color property, 58
border-bottom property, 56-57
border-color property, 56-57
border-left property, 56-57
border property, 56-57
border-right property, 56-57
border-style property, 56-57
border-top property, 56-57
border-width property, 56
color property, 59
font-family property, 58
font property, 59
font-size property, 58
font-style property, 58
font-weight property, 58
line-height property, 59
padding property, 59
text-align property, 58-59
text-decoration property, 59
text-indent property, 58
Foundation framework, 434
hidden (rollover) text, 327-328
HTML documents, linking to, 53
image maps, creating, 298-301
inline styles, 63
internal style sheets, 48, 62-63
layouts
elastic layouts, 288
fixed layouts, 270-271
fixed/liquid hybrid layouts, 275-286
layout properties, 55
liquid layouts, 270-275
responsive web design, 286-287
<link> tag, 53
links, styling, 147-151
lists
horizontal navigation, 316-319
list-style-image property, 292
list-style-position property, 292, 295-297
list-style-type property, 292
navigation lists, 306-319
vertical navigation, 306-315
margin property, 233-241
media-specific style sheets, 369-371
mouse actions
events and event handling, 329-336
rollover (hidden) text, 327-328
ToolTip, 323-326
overflow property, text flow and, 264
padding property, 233, 241-244
positioning
absolute positioning, 257-261
overlapping elements, 261-263
positioning property, 257
relative positioning, 257-259
z-index property, 261-263
print pages, designing style sheets for, 371-374
reference guide online resource, 49
rollover (hidden) text, 327-328
selectors, 60
<span> tags, 63
style classes, 59-61
style IDs, 61-62
style properties, 60
style rules, 48, 52
color style rule, 109
font-family style rule, 109
font-size style rule, 109
font-weight style rule, 105-106
multiple style properties in, 61
viewing, 54
style sheets
creating, 51-52
formatting properties, 57
tags, 52
elements and, 48
selectors, 60
text
formatting color, 48
italic text, 59
line-through text, 59
sizing, 53
strikethrough text, 59
underline text, 59
ToolTip, 323-326
validating, 64
z-index property, 261-263
CSS3

2D transformations
rotating elements, 344-346
scaling elements, 346-348
skewing elements, 348
translating elements, 342-344
3D transformations
applying perspective, 350
creating/manipulating cubes, 354-358
moving (translating) elements, 353
rotating elements, 353
vanishing points, 350
animation, 360-361
font-face feature, 112-114
transitions, 359-360
cubes, CSS3 3D transformations, 354-358
current line, text flow and, 264
cursors, creating ToolTips, 323-326
customer service, web hosting providers, 10
Cyberduck FTP client, 15

E

editors (blog), 24
elastic layouts, 288
elements
2D transformations (CSS3)
rotating elements, 344-346
scaling elements, 346-348
skewing elements, 348
translating elements, 342-344
3D transformations (CSS3)
applying perspective, 350
creating/manipulating cubes, 354-358
moving (translating) elements, 353
rotating elements, 353
vanishing points, 350
block-level elements, aligning text in, 87-89
categorizing, 75
definition of, 48
flow content, 75
form elements, grouping, 404
rounded corners, creating, 128-130
transitions (CSS3), 359-360
<em> tags (HTML), 105-106

email addresses
email address encoders, 146
linking to, 145-146
validating, 411
embedding multimedia files, 221-224
emphasized text. See italic text
empty tags (HTML), 34
ESPN.com, website organization, 423-424
European languages, formatting text for, 102-104
events and event handling
event attributes, 329
onclick event and mouse actions, 330-336
external style sheets (CSS), 48, 51-54

F

Fetch FTP client, 15
Firebug, validating web content, 43
Firefox web browser, 13
FireFTP FTP client, 15-16
FireZilla FTP client, 16
fixed layouts, 270-271
fixed/liquid hybrid layouts, 285-286
column height, setting, 280-284
columns, defining in, 277-279
minimum width, setting, 279
structure of, 275-277
Flickr, 190, 195
float property (CSS), 233, 246-249, 264
flow content elements, 75
flowing text, 264
fluid layouts. See liquid layouts
folders (web content), 135-137
fonts (text)
Arial font, 109
CSS
color style rule, 109
font-family property, 58
font-family style rule, 109
font property, 59
font-size property, 58
font-size style rule, 109
font-style property, 58
font-weight property, 58
font weight style rule, 105-106
<font> tags (HTML), 102, 108
font-face feature (CSS3), 112-114
foreign languages, 102-104
HTML, customizing in, 108-112
resumes, creating, 110-112
sans-serif font, 109
sizing, style sheets, 53
special characters, 102-104
Times Roman font, 109
web browsers, adjusting font size settings, 25
web fonts, 112-114
<footer> semantic element (HTML5), 69-72, 80-81
<footer> tags, HTML5, 41
foreign languages, formatting text for, 102-104
formatting
CSS, formatting properties, 55-56
style sheets
text color, 48
text size, 53
text
boldface text, 105-106
bulleted lists, 94
customizing fonts in HTML, 108-112
definition lists, 90-92, 292
foreign languages, 102-104
italic text, 105-106
multitiered lists, 94-97
nested lists, 92, 292
older HTML tags, 102
ordered lists, 89-92, 291
outlines, 92-93
resumes, creating, 110-112
special characters, 102-104
style sheets, 48, 53
subscription text, 106
superscript text, 106
unordered lists, 90-92, 291
web fonts, 112-114
web page creation, 37-38
formatting properties (CSS)
background-color property, 58
border-bottom property, 56-57
border-color property, 56-57
border-left property, 56-57
border property, 56-57
border-right property, 56-57
border-style property, 56-57
border-top property, 56-57
border-width property, 56
color property, 59
text-align property, 58
font-color property, 58
font-family property, 58
font property, 59
font-size property, 58
font-style property, 58
font-weight property, 58
line-height property, 59
padding property, 59
text-align property, 58-59
text-decoration property, 59
text-indent property, 58
forms, 395
check boxes, 405-406
creating, 396-401
data
hidden data, 404-405
labeling, 403
naming, 402
submitting, 412-413
grouping elements, 404
HTML5, 397, 410-411
passwords, 402
pull-down pick lists, 407-408
radio buttons, 407
scrolling lists, 407-408
selection lists, 407-408
sending, 397
text areas, 409-410
text fields, 409-410
text input, accepting, 401-402
user input, accepting, 401-402
validating, HTML5 form validation, 410-411
forward slashes (/), HTML and, 136
Foundation framework, 434
frames (animated graphics), 190
frameworks
Bootstrap, 434
Foundation, 434
HTML5 Boilerplate, 434
FTP, 14, 33
Classic FTP FTP client, 15
server connections, 18
website connections, 16-17
Cyberduck, 15
Fetch, 15
FireFTP, 15-16
FireZilla, 16
FTP clients, 16
selecting, 15
using, 17-18
G
Gickr, animated graphics, 189
GIF files, 186
animated graphics, 189-190
tiled backgrounds, 187-189
transparent images, 187
GIMP, 176
Adjust Hue/Lightness/Saturation tool, 182
banners, creating, 184-186
buttons, creating, 184-186
images
adjusting color, 182
cropping, 178-180
JPEG compression, 183-184
resizing, 181
Red Eye Removal, 182
Git website version control, 433
Google
Chrome web browser, 13
Images, 190
listing websites with, 442
Picasa, 176
searches, 8
graphics
Adobe Photoshop, 176
aligning
horizontal alignment, 197-199
vertical alignment, 199-201
animated graphics, 189-190
backgrounds
  background-image style property, 204
graphics, 204-205
tiled backgrounds, 187-189
banners, creating, 184-186
buttons, creating, 184-186
CD-ROM, transferring graphics to, 178
changing based on user interaction, 388-389
clip art, 176
color, adjusting, 182
compression, 177
copyrights and, 176
Creative Commons licenses, 176
cropping, 178-179
file sizes, 177
Flickr, 190, 195
GIF files, 186
  animated graphics, 189-190
tiled backgrounds, 187-189
transparent images, 187
GIMP, 176
  adjusting image color, 182
  banners, 184-186
  buttons, 184-186
  cropping images, 178-180
  JPEG compression, 183-184
  resizing images, 181
Google Images, 190
Google Picasa, 176
height/width, specifying, 197
image maps, 206-211, 298-301
JPEG files
  compression, 183-184
tiled backgrounds, 187-189
links, turning graphics into, 201-204
Pixlr, 176
PNG files, 187
Red Eye Removal, 182
republishing, 195
resizing, 181
resolution, 177
responsive images, 197
software, choosing, 175-176
storing, 194
text descriptions, 195-196
transparent graphics, 187
uses of, 177
web pages
  grabbing graphics from, 176
  placing graphics on web pages, 193-195
Windows Media Video, linking to, 219
groove value (CSS border-style properties), 57
grouping form elements, 404

H
<head> tags, 31, 34-37
<header> semantic element (HTML5), 69-75
<header> tags, HTML5, 40
heading tags (HTML), 38-39
headings (tables), creating, 156
height
  CSS box model, adjusting in, 254-255
  fixed/liquid hybrid layouts, setting column height in, 280-284
  height property (CSS), 55
  images, specifying height in, 197
Hello World HTML file, creating, 13-14
help
  CSS reference guide online resource, 49
  web hosting providers, selecting, 10
helper applications, defining, 219
hexadecimal color codes, 122-124
hidden form data, 404-405
hidden (rollover) text, displaying, 327-328
horizontal image alignment, 197-199
horizontal navigation, 316-319
horizontal rule tags (HTML), 37
<hr> tags, 37

HTML
attributes, 135
default values, 88
defining, 86
  style attribute and text-align style rule, 88-89
<audio> element, embedding multimedia files, 224-225
Bootstrap framework, 434
character entity, 103
CSS
  box model, 253-256, 292-295
  external style sheets, 53-54
  linking style sheets to HTML documents, 53
development of, 5-6
element, rounded corners, 128-130
<embed> element, embedding multimedia files, 223
files
  creating, 13-14, 31-40
  index pages, 22
  managing, 19-21
  organizing, 19-21
  transferring, 14-18, 33
  viewing, 33
forms, 395
  check boxes, 405-406
  creating, 396-401
  grouping elements, 404
  hidden data, 404-405
  labeling form data, 403
  naming form data, 402
  passwords, 402
  pull-down pick lists, 407-408
  radio buttons, 407
  scrolling lists, 407-408
  selection lists, 407-408
  sending, 397
  submitting form data, 412-413
  text areas, 409-410
  text fields, 409-410
  text input, 401-402
  user input, 401-402
forward slashes (/), 136
FTP, 14
   selecting FTP clients, 15-16
   using FTP clients, 16-18
graphics, image maps, 209-211
Hello World sample file, creating, 13-14
history of, 5-6
HTML-compatible word processors, creating HTML files, 31
.html file extensions, 31
images, placing on web pages, 193-195
JavaScript in
   comments in HTML files, 381
   displaying random content in HTML files, 382-386
   scripting in HTML files, 380-381
layouts
   elastic layouts, 288
   fixed layouts, 270-271
   fixed/liquid hybrid layouts, 275-286
   liquid layouts, 270-275
   responsive web design, 286-287
links
   absolute links, 136
   anchor tags, 138-141
   images as, 147
   linking between web content, 141-144
   linking to email addresses, 145-146
   linking to external web content, 144
   opening links in new browser windows, 146-147
   relative-root links, 137
   styling via CSS, 147-151
lists
   bulleted lists, 94
   definition lists, 90-92, 292
   list-style-type style rule, 94-97
   multitiered lists, 94-97
   nested lists, 92, 292
   ordered lists, 89-92, 291
   outlines, building via lists, 92-93
   unordered lists, 90-92, 291
marked up text, 7
<meta> tags, 444-448
Notepad, creating HTML files, 30
<object> element, embedding multimedia files, 221-223
outlines, building via lists, 92-93
pseudoclasses, 147-151
rounded corners, creating, 128-130
style rules
   default values, 88
   list-style-type style rule, 94-97
tables
   aligning within, 163-165
   borders, 158-159
   cells, creating, 156
   cells, sizing, 161-162
   color in, 165-166
   creating, 156-160
   headings, creating, 156
   images in, 166
   page layout via, 166-167
   rows, creating, 156
   sizing, 160-162
   spacing borders, 166
   spanning within, 165
   uses for, 156
tags, 9, 32
   <a> tags, 138-141, 203
   attributes, 86, 135
   <b> tags, 102-106
   <body> tag, 31, 34, 37
   <br> tag, 36-37, 108
   child (nested) tags, 92
   closing slashes, 36
   closing tags, 33-34
   <div> tags, 203
   <em> tags, 105-106
   empty tags, 34
   <font> tags, 102, 108
   formatting and older HTML tags, 102
   <head> tag, 31, 34-37
   heading tags, 38-39
   horizontal rule tag, 37
<hr> tag, 37
<html> tag, 31, 34-37
</i> tags, 102, 105-106
<img> tags, 193-195
line breaks, 35-37
naming files with, 31
nested (child) tags, 31
opening tags, 33-34
<p> tag, 35-37
paragraphs, 35-37
<title> tag, 31, 34-39
<tbody> tags, 160
<td> tags, 156, 159
<tfoot> tags, 160
<th> tags, 156-159
<tr> tags, 160
</table> tags, 156-157
</u> tags, 106
xml: lang attribute, 34
xmlns attribute, 34
text, formatting
   boldface text, 105-106
   customizing fonts, 108-112
   definition lists, 292
   foreign languages, 102-104
   italic text, 105-106
   nested lists, 292
   older HTML tags, 102
   ordered lists, 291
   resumes, creating, 110-112
   special characters, 102-104
   subscript text, 106
   superscript text, 106
   unordered lists, 291
   web fonts, 112-114
   whitespace, 35
validating, 42-43
</video> element, embedding multimedia files, 226-227
web content

absolute addresses, 136-137
absolute links, 136
anchor tags, 138-141
creating, 6-7
delivering, 7-9
directories, 135-136
folders, 135-137
images as links, 147
linking between, 141-144
linking to email addresses, 145-146
linking to external web content, 144
opening links in new browser windows, 146-147
organizing, 135-136
publishing locally, 23
publishing to blogs, 24
relative addresses, 136-137
relative-root addresses, 136
relative-root links, 137
selecting web hosting providers, 10-12, 30
styling links via CSS, 147-151
website architectures, creating, 137

whitespace, 35
Word, creating HTML files, 31
WYSIWYG editors, creating HTML files, 31

HTML Validator, 43

HTML5

application development, 390-391
client-side scripting, 390
centralizing web pages, 69-74
forms, 397, 410-411
HTML5 Boilerplate framework, 434
JavaScript, 390

multimedia

audio playback, 224-225
video playback, 224-227
Outline tool, 74

semantic elements, 40-41, 73
<article>, 69-72, 77
<aside>, 69-72, 79
<footer>, 69-72, 80-81
<header>, 69-75
<nav>, 69-72, 78-79
<section>, 69-72, 75-77
server-side scripting, 390
tags
<article> tags, 41
<aside> tags, 41
<footer> tags, 41
<header> tags, 41
<nav> tags, 41
<section> tags, 41

<aside> tags, 41
<header> tags, 41

semantic elements, 40-41

hyperlinks. See links

<i> tags (HTML), 102, 105-106

images

aligning
horizontal alignment, 197-199
vertical alignment, 199-201
animated images, 189-190
auto image loading, web browsers, 24
backgrounds
images, 204-205
tiled backgrounds, 187-189
banners, creating, 184-186
buttons, creating, 184-186
CD-ROM, transferring images to, 178
changing based on user interaction, 388-389
clip art, 176
color, adjusting, 182
compression, 177
copyrights and, 176
Creative Commons licenses, 176
cropping, 178-179
file sizes, 177
Flickr, 190, 195
GIF files, 186
animated images, 189-190
tiled backgrounds, 187-189
transparent images, 187
Google Images, 190

Google Picasa, 176
height/width, specifying, 197
image maps, 206-211, 298-301
JPEG files
compression, 183-184
tiled backgrounds, 187-189
links
images as, 147
turning images into links, 201-204
Pixlr, 176
PNG files, 187
Red Eye Removal, 182
republishing, 195
resizing, 181
resolution, 177
responsive images, 197
storing, 194
tables and, 166
text descriptions, 195-196
transparent images, 187
uses of, 177
web pages
grabbing images from, 176
placing images on web pages, 193-195

Windows Media Video, linking to, 219

<img> tags (HTML), 193-195

in value (CSS height/width properties), 55

indenting
code, 431
text, web page creation, 38

index pages, HTML file management, 22

inline styles (CSS), 63

inline value (CSS display property), 55

input controls, forms
check boxes, 405-406
radio buttons, 407
selection lists, 407-408
text areas, 409-410
text fields, 409-410

inset value (CSS border-style properties), 57

internal style sheets (CSS), 48, 62-63
Mashable.com, publicizing websites via 461

Internet Explorer, 13, 256
ISP, selecting, 29
italic text, 59, 105-106

JavaScript
arrays, 382-384
client-side scripting, 380
comments, 381, 384
concatenation operators, 385
DOM, 387
events and event handling
event attributes, 329
onclick event, 330-336
HTML files
comments, 381
displaying random content in, 382-386
scripting in, 380-381
HTML5, 390
images, changing based on user interaction, 388-389
JPEG files
compression, 183-184
tiled backgrounds, 187-189
.jsp file extensions, 31

K - L
keyframes (CSS3 animation), 360
labeling form data, 403
languages (foreign), formatting text for, 102-104
layouts
elastic layouts, 288
fixed layouts, 270-271
fixed/liquid hybrid layouts, 285-286

defining columns in, 277-279
setting column height, 280-284
setting minimum width, 279
structure of, 275-277
hidden (rollover) text, displaying, 327-328
layout properties (CSS), 55
liquid layouts, 272-275
responsive web design, 286-287
rollover (hidden) text, displaying, 327-328
web resources, 270
leading (text), line-height property (CSS), 59
line breaks, web page creation, 35-37
line-height property (CSS), 59
line-through text, style sheets, 59
links
absolute links, 136
anchor tags
identifying locations within web pages via, 138
linking to anchor locations, 138-141
naming, 140
color and, 122
e-mail addresses, 145-146
images, 147
linking to Windows Media Video, 219
turning images into links, 201-204
<link> tags (CSS), 53
media-specific style sheets to web pages, 370-371
multimedia files, 218-221
opening in new browser windows, 146-147
relative-root links, 137
styling via CSS, 147-151
web content
linking between, 141-144
linking to external web content, 144
web pages, linking media-specific style sheets to, 370-371
Windows Media Video, linking images to, 219
liquid layouts, 270-275
lists
bulleted lists, 94
CSS box model and, 292-295
definition lists, 90-92, 292
list item indicators, placing, 295-297
list-item value (CSS display property), 55
list-style-image property (CSS), 292
list-style-position property (CSS), 292, 295-297
list-style-type property (CSS), 292
list-style-type style rule, 94-97
multitiered lists, 94-97
navigation lists
horizontal navigation, 316-319
primary navigation, 306
regular lists versus, 305-306
secondary navigation, 306
vertical navigation, 306-315
nested lists, 92, 292
ordered lists, 90-92, 292
unordered lists, 90-92, 291
loading web content, 25
LunarPages web hosting provider, 11

M
managing
domain names, 11
HTML files, 19
document roots, 20-21
index pages, 22
web page headings, 38-39
websites
coding clarity, 431
comments, 429-430
documenting code, 429-430
indenting code, 431
maintainable code, 429-431
version control, 432-433
margins
browsers and, 292-294
CSS box model, 254
margin property (CSS), 233-241
marked up text, HTML, 7
Mashable.com, publicizing websites via, 441
media-specific style sheets, 369-371

<meta> tags, 444-448
Microsoft Internet Explorer. See
Internet Explorer

Microsoft Word. See Word (MS)
MIME types, multimedia files, 222-223
mm value (CSS height/width
properties), 56
monitors
color and, 122
resolution, 184

mouse actions
events and event handling
event attributes, 329
onclick event, 330-336
rollover (hidden) text, displaying,
327-328
ToolTips, creating, 323-326

moving (translating) elements, CSS3
3D transformations, 353
Mozilla Firefox web browser. See
Firefox web browser

multimedia, 228

audio
<audio> element, 224-225
HTML5 playback, 224-225
creating, 218
defining, 217
<embed> element, 223
embedding files, 221-224
linking to files, 218-221
MIME types, 222-223
<object> element, 221-223
QuickTime support, 221
streaming files, 221
video
<video> element, 226-227
video files, HTML5 playback,
224-227

multitiered lists, 94-97

N

naming
anchor tags, 140
files with HTML tags, 31
form data, 402


<nav> semantic element (HTML5),
69-72, 78-79
<nav> tags, HTML5, 41

navigation lists
horizontal navigation, 316-319
primary navigation, 306
regular lists versus, 305-306
secondary navigation, 306
vertical navigation, 306-308, 311
multilevel vertical navigation,
312-315
single-level vertical navigation,
309-310

nested (child) tags, 92
nested lists, 92, 292

non-viewable window areas
(browsers), 270

none value, CSS
border-style properties, 57
display property, 55

Notepad, creating HTML files, 30

O

<object> element, embedding
multimedia files, 221-223
onclick events (JavaScript), 329-336
onmousedown events (JavaScript), 329
onmouseup events (JavaScript), 329
online resources, CSS
browser support, 54
CSS reference guide, 49
onload events (JavaScript), 329
onmousedown events
(JavaScript), 329
onmousemove events
(JavaScript), 329
onmouseout events (JavaScript), 330
onmouseover events (JavaScript), 330
onmouseup events
(JavaScript), 330

opening tags (HTML), 33-34

Opera web browser, 13
ordered lists, 89-92, 291

organizing
HTML files, 19
document roots, 20-21
index pages, 22
web content, 135-136
web page headings, 38-39
websites
Amazon.com, 425-426
BAWSI.org, 428
ESPN.com, 423-424
larger websites, 425-429
Peet’s Coffee & Tea, 426-427
simple websites, 422-425
single-page websites, 420-421

Outline tool (HTML5), 74
outlines, building via lists, 92-93
outset value (CSS border-style
properties), 57
overflow property (CSS), text flow
and, 264
overlapping elements, z-index
property (CSS), 261-263

P

<p> tags, 35-37
padding
browsers and, 292-294
CSS box model, 254
padding property (CSS), 59, 233,
241-244

paragraphs, web page creation, 35-37
parent folders, 137
passwords, forms, 402
PDF files, converting, 375
Peet’s Coffee & Tea, website
organization, 426-427

peripherals
monitors
color and, 122
resolution, 184
mouses
events and event handling,
329-336
rollover (hidden) text, 327-328
ToolTips, 323-326
perspective, CSS3 3D transformations, 350
photos. See also images
aligning, 199
  horizontal alignment, 197-198
  vertical alignment, 200-201
background-image style property, 204
background photos, 204-205
changing based on user interaction, 388-389
cropping, 178-179
Flickr, 190, 195
Google Images, 190
height/width, specifying, 197
image maps, 206-211, 298-301
links, turning images into, 201-204
Red Eye Removal, 182
republishing, 195
resizing, 181
responsive images, 197
storing, 194
text descriptions, 195-196
web pages, placing photos on, 193-195
Windows Media Video, linking to, 219
Photoshop (Adobe), 176
PHP, 31, 396
Picasa (Google), 176
pictures. See images
Pixlr, 176
plain text, 30, 37, 102
plug-ins, defining, 219
PNG files, 187
pop-up windows, 146
positioning
  absolute positioning, 257-261
  overlapping elements, z-index property (CSS), 261-263
  positioning property, 257
  relative positioning, 257-259
<pre> tags (HTML), 106-108
prefixes (vendor), web browsers and, 341
pricing web hosting providers, 10
primary navigation, 306
printing
  print-friendly web pages, 365
criteria for print-friendliness, 366-369
designing style sheets for print pages, 371-374
media-specific style sheets, 369-371
reviewing content for print-friendliness, 366
Print Preview, viewing web pages in, 374-375
pseudoclasses, 147-151
pt value (CSS height/width properties), 56
publishing web content
  blog publication, 24
  local publication, 23
pull-down pick lists (forms), 407-408
pull quotes, 79
purchasing domain names, 11
px value (CSS height/width properties), 56
Q - R
QuickTime, browser support for, 221
quotes (pull), 79
radio buttons (forms), 407
random content, displaying in HTML files via JavaScript, 382-386
Red Eye Removal, 182
registered trademark symbol, 103
relative addresses, web pages and, 136-137
relative positioning, 55, 257-259
relative-root addresses, web pages and, 136
relative-root links, web pages and, 137
reliability, web hosting providers, 10
republishing images, 195
resizing images, 181
resolution
  graphics, 177
  screen, 184
responsive images, 197
responsive web design, 286-287
ridge value (CSS border-style properties), 57
robots, web page searches, 441
rollover (hidden) text, displaying, 327-328
rotating elements, CSS3
  2D transformations, 344-346
  3D transformations, 353
rounded corners, creating, 128-130
rows (tables), creating, 156
Safari web browser, 13
sans-serif font (text), 109
saving files with HTML tags, 31
scaling
  elements (CSS3 2D transformations), 346-348
  images, 181
screen resolution, 184
scripting
  client-side scripting, 380, 390
  PHP, 396
  server-side scripting, 380, 390
scrolling lists (forms), 407-408
searches
  Google searches, 8
  search engines
    SEO, 439, 448-450
    spamming, 443
    URL creation, 449
    websites, listing with search engines, 441-448
  web content, <meta> tags (HTML), 444-448
secondary navigation, 306
<section> semantic element (HTML5), 69-72, 75-77
<section> tags

- HTML, 143
- HTML5, 41

Selection lists (forms), 407-408

Selectors (CSS), 60

Semantic elements, 40-41, 73
- <article>, 69-72, 77
- <aside>, 69-72, 79
- <footer>, 69-72, 80-81
- <header>, 69-75
- <nav>, 69-72, 78-79
- Recommendations, 70
- <section>, 69-72, 75-77

Servers
- Browsers, basic browser server interaction, 7-9
- Document roots, 18-21
- FTP client connections, 18
- Server-side scripting, 380, 390
- Space, 10
- Uptime, 10
- Web hosting providers, selecting, 10

Single-page websites, 420-421

Sizing
- Elements (CSS box model), 254-255
- Images, 181
- Tables, 160-162
- Text
  - Font-size style rule (CSS), 109
  - Stylesheets, 53

Skeleton pages. See templates

Skewing elements (CSS3 2D transformations), 348

Solid value (CSS border-style properties), 57

Source editors, blogs, 24

Spamming search engines, 443

<span> tags (CSS), 63

Spanning with tables, 165

Special characters (symbols), 102-104

Spiders, web page searches, 441

Streaming multimedia files, 221

<strike> tags (HTML), 106

Strikethrough text, stylesheets, 59

<strong> tags (HTML), 105-106

Strong text. See boldface text

Style attribute, text-align style rule and, 88-89

Style classes (CSS), 59-61

Style IDs (CSS), 61-62

Style properties (CSS), 60

Style rules

- CSS, 48, 52
  - Color style rule, fonts and, 109
  - Font-family style rule, 109
  - Font-size style rule, 109
  - Font-weight style rule, 105-106
  - Multiple style properties in, 61
  - Viewing, 54
- Default values, 88
- List-style-type style rule, 94-97
- Text-align, 87-89

Stylesheets

- Align property, 233, 245
- Bootstrap framework, 434
- Box model, 253, 256
  - Borders, 254
  - Content, 254
  - Lists and, 292-295
  - Margins, 254
  - Padding, 254
  - Sizing elements, 254-255
- Browser support for, 54
- Clear property, text flow and, 264
- Color, specifying via style sheets, 125-127
- Creating, 49-54
- CSS tags, 52
- Definition of, 47
- <div> tags, 63
- DOCTYPE declarations, 256
- External style sheets, 48, 51-54
- Float property, 233, 246-249, 264
- Font-face feature, 112-114
- Formatting properties, 55
  - Background-color property, 58
  - Border-bottom property, 56-57
  - Border-color property, 56-57
  - Border-left property, 56-57
  - Border property, 56-57
  - Border-right property, 56-57
  - Border-style property, 56-57
- Border-top property, 56-57
- Border-width property, 56
- Color property, 59
- Font-family property, 58
- Font property, 59
- Font-size property, 58
- Font-style property, 58
- Font-variant property, 58
- Font-weight property, 58
- Line-height property, 59
- Padding property, 59
- Text-align property, 58-59
- Text-decoration property, 59
- Text-indent property, 58

Foundation framework, 434

Hidden (rollover) text, 327-328

HTML documents, linking to, 53

Image maps, creating, 298-301

Inline styles, 63

Internal style sheets, 48, 62-63

Italic text, 59

Layouts
- Elastic layouts, 288
- Fixed layouts, 270-271
- Fixed/liquid hybrid layouts, 275-286
- Responsive web design, 286-287

Line-through text, 59

Links, styling, 147-151

Lists
  - Horizontal navigation, 316-319
  - List-style-image property, 292
  - List-style-position property, 292, 295-297
  - List-style-type property, 292
- Navigation lists, 306-319
- Vertical navigation, 306-315

Margin property, 233-241

Media-specific style sheets, 369-371

Mouse actions
  - Events and event handling, 329-336
  - Rollover (hidden) text, 327-328
  - ToolTips, 323-326

Overflow property, text flow and, 264
padding property, 233, 241-244
positioning
absolute positioning, 257-261
overlapping elements, 261-263
positioning property, 257
relative positioning, 257-259
z-index property, 261-263
print pages, designing style sheets for, 371-374
rollover (hidden) text, 327-328
selectors, 60
<span> tags, 63
strikethrough text, 59
style classes, 59-61
style IDs, 61-62
style properties, 60
style rules, 48, 52
  color style rule, 109
  font-family style rule, 109
  font-size style rule, 109
  font weight style rule, 105-106
multiple style properties in, 61
viewing, 54
text
  formatting color, 48
  sizing, 53
ToolTip, 323-326
underline text, 59
validating, 64
z-index property, 261-263
<table> tags (HTML), 106
<sub> tags, 106
subscript text, 106
Subversion website version control, 433
<br> tags (HTML), 106
superscript text, 106
support, selecting web hosting providers, 10
symbols (special characters), 102-104

T

tables
  aligning within, 163-165

borders
  creating, 158-159
  spacing, 166
  specifying color via CSS, 125-127
cells
  creating, 156
  sizing, 161-162
color in, 165-166
  creating, 156-160
CSS columns, 167-171
headings, creating, 156
images in, 166
  page layout via, 166-167
rows, creating, 156
sizing, 160-162
spanning within, 165
<table> tags (HTML), 156
uses for, 156
tags
  child (nested) tags, 92
CSS, 52
  <div> tags, 63
  elements and, 48
  <link> tag, 53
  selectors, 60
  <span> tags, 63
  <div> tags, 63, 203, 330-336
HTML, 9, 32
  <> tags, 138-141, 203
  attributes, 86, 135
  <b> tags, 102, 105-106
  <body> tag, 31, 34, 37
  <br> tags, 36-37, 108
  closing slashes, 36
  closing tags, 33-34
  <div> tags, 203
  <em> tags, 105-106
  empty tags, 34
  <font> tags, 102, 108
  <head> tag, 31, 34-37
  heading tags, 38-39
  horizontal rule tag, 37
  <hr> tag, 37
  <html> tag, 31, 34-37
  <i> tags, 102, 105-106
  <img> tags, 193-195
  line breaks, 35-37

naming files with, 31
older HTML tags, formatting and, 102
opening tags, 33-34
<p> tag, 35-37
paragraphs, 35-37
<pre> tags, 106-108
<p> tags, 106
<sub> tags, 106
<table> tags, 156
<thead> tags, 156
<tr> tags, 156-157
<th> tags, 156, 159
<title> tags, 31, 34-39
<tt> tags, 156-157
<ul> tags, 106
xml: lang attribute, 34
xmlns attribute, 34
HTML5
  <article> tags, 41
  <aside> tags, 41
  <footer> tags, 41
  <header> tags, 40
  <nav> tags, 41
  <section> tags, 41
  semantic elements, 40-41
<meta> tags, 444-448
nested (child) tags, 92
<section> tags, 41, 143
<span> tags, 63
<sub> tags (HTML), 106
subscript text, 106
Subversion website version control, 433
<sup> tags (HTML), 106
superscript text, 106
symbols (special characters), 102-104

testing
  web content, 24-25
  websites, multiple web browsers, 12-13, 30
text
  aligning, 86
    block-level elements, 87-89
  tables, 163-165
text-align style rule, 87-89
alternate text, 195-196
transformation (CSS3)
2D transformations
  rotating elements, 344-346
  scaling elements, 346-348
  skewing elements, 348
  translating elements, 342-344
3D transformations
  applying perspective, 350
  creating/manipulating cubes, 354-358
  moving (translating) elements, 353
  rotating elements, 353
  vanishing points, 350
transitions (CSS3), 359-360
translating elements (CSS3)
2D transformations, 342-344
3D transformations, 353
transparent images, 187
triadic color schemes, 121

U - V
<u> tags (HTML), 106
underline text, style sheets, 59
unordered lists, 90-92, 291
unregistered trademark symbol, 103
uptime, servers and, 10
URL, creating, 449
USB drivers, distributing web browsers, 23
user input in forms, 401-402
validating
  email addresses, 411
  forms in HTML5, 410-411
  style sheets, 64
  web content, 42-43
vanishing points, CSS3 3D transformations, 350
VBScript (Microsoft), client-side scripting, 380
vendor prefixes, web browsers and, 341
version control, websites and, 432-433
vertical image alignment, 199-201
vertical navigation, 306-308, 311
  multilevel vertical navigation, 312-315
  single-level vertical navigation, 309-311
video
  HTML5 video playback, 224-227
  <video> element, embedding multimedia files, 226-227
viewing
  CSS style rules, 54
  HTML files, 33
  web pages, 33
visual editors, blogs, 24

W

W3C color standards, 121
W3C CSS Validator, 64
W3C Validation Service, 42-43
web browsers
  Chrome, 13
  CSS support, 54
  development of, 6
  differences in, accounting for, 341
  distributing, 23
  Firefox, 13
  helper applications, defining, 219
  history of, 6
  HTML development, 6
  images, auto image loading, 24
  Internet Explorer, 13
  Opera, 13
  plug-ins, defining, 219
  popularity of, 30
  QuickTime support, 221
  Safari, 13
  servers, basic browser server interaction, 7-9
text, adjusting font size settings, 25
  vendor prefixes, 341
  web content, displaying, 92
  websites
    comparing, 30
testing, 12-13, 30
web content
  absolute addresses, 136-137
  aligning, align property (CSS), 233, 245
  browsers, displaying in, 92
clear property (CSS), text flow and, 264
color
  140 cross-browser color names, 121
  best practices, 119-121
  Colorblind Web Page Filter tool, 127
  hexadecimal color codes, 122-124
  using, 119-121
  W3C color standards, 121
creating, 6-7
  ASCII text, 30, 37
  comparing web content HTML codes, 40
  formatting text, 37-38
  HTML tags, 31-37
  indenting text, 38
  line breaks, 35-37
  organizing content via headings, 38-39
  overview of, 32-33
  paragraphs, 35-37
  plain text, 30, 37
  templates, 35
CSS box model, 253, 256
  borders, 254
  content, 254
  lists and, 292-295
  margins, 254
  padding, 254
  sizing elements, 254-255
delivering, 7-9
directories, 135-136
float property (CSS), 233, 246-249, 264
folders, 135-137
forms, 395
  check boxes, 405-406
  creating, 396-401
  grouping elements, 404
hidden data, 404-405
labeling form data, 403
naming form data, 402
passwords, 402
pull-down pick lists, 407-408
radio buttons, 407
scrolling lists, 407-408
selection lists, 407-408
sending, 397
submitting form data, 412-413
text areas, 409-410
text fields, 409-410
text input, 401-402
user input, 401-402

graphics
  adjusting color, 182
  Adobe Photoshop, 176
  aligning graphics, 197-201
  animated graphics, 189-190
  background graphics, 204-205
  banners, 184-186
  buttons, 184-186
  choosing software, 175-176
  clip art, 176
  compression, 177
  copyrights and, 176
  Creative Commons licenses, 176
cropping, 178-179
delivering, 7-9
directories, 135-136
delivering, 7-9
float property (CSS), 233, 246-249, 264
folders, 135-137
forms, 395
  check boxes, 405-406
  creating, 396-401
  grouping elements, 404
hidden data, 404-405
labeling form data, 403
naming form data, 402
passwords, 402
pull-down pick lists, 407-408
radio buttons, 407
scrolling lists, 407-408
selection lists, 407-408
sending, 397
submitting form data, 412-413
text areas, 409-410
text fields, 409-410
text input, 401-402
user input, 401-402

W3C color standards, 121
W3C CSS Validator, 64
W3C Validation Service, 42-43
web browsers
  Chrome, 13
  CSS support, 54
  development of, 6
differences in, accounting for, 341
distributing, 23
Firefox, 13
helper applications, defining, 219
history of, 6
HTML development, 6
images, auto image loading, 24
Internet Explorer, 13
Opera, 13
plug-ins, defining, 219
popularity of, 30
QuickTime support, 221
Safari, 13
servers, basic browser server interaction, 7-9
text, adjusting font size settings, 25
  vendor prefixes, 341
  web content, displaying, 92
  websites
    comparing, 30
    testing, 12-13, 30
web content
  absolute addresses, 136-137
  aligning, align property (CSS), 233, 245
  browsers, displaying in, 92
clear property (CSS), text flow and, 264
color
  140 cross-browser color names, 121
  best practices, 119-121
  Colorblind Web Page Filter tool, 127
  hexadecimal color codes, 122-124
  using, 119-121
  W3C color standards, 121
creating, 6-7
  ASCII text, 30, 37
  comparing web content HTML codes, 40
  formatting text, 37-38
  HTML tags, 31-37
  indenting text, 38
  line breaks, 35-37
  organizing content via headings, 38-39
  overview of, 32-33
  paragraphs, 35-37
  plain text, 30, 37
  templates, 35
CSS box model, 253, 256
  borders, 254
  content, 254
  lists and, 292-295
  margins, 254
  padding, 254
  sizing elements, 254-255
delivering, 7-9
directories, 135-136
float property (CSS), 233, 246-249, 264
folders, 135-137
forms, 395
  check boxes, 405-406
  creating, 396-401
  grouping elements, 404
hidden data, 404-405
labeling form data, 403
naming form data, 402
passwords, 402
pull-down pick lists, 407-408
radio buttons, 407
scrolling lists, 407-408
selection lists, 407-408
sending, 397
submitting form data, 412-413
text areas, 409-410
text fields, 409-410
text input, 401-402
user input, 401-402

graphics
  adjusting color, 182
  Adobe Photoshop, 176
  aligning graphics, 197-201
  animated graphics, 189-190
  background graphics, 204-205
  banners, 184-186
  buttons, 184-186
  choosing software, 175-176
  clip art, 176
  compression, 177
  copyrights and, 176
  Creative Commons licenses, 176
cropping, 178-179
file sizes, 177
Flickr, 190, 195
GIFs, 186-190
GIMP, 176-186
Google Images, 190
Google Picasa, 176
grabbing from web pages, 176
image maps, 206-211
JPEG files, 183-184, 187-189
Pixlr, 176
placing graphics on web pages, 193-195
PNG files, 187
Red Eye Removal, 182
republishing, 182
resizing, 181
resolution, 177
specifying height/width, 197
text descriptions, 195-196
tiled backgrounds, 187-189
transparent graphics, 187
turning graphics into links, 201-204
uses of, 177
hidden (rollover) text, displaying, 327-328
links
absolute links, 136
anchor tags, 138-141
email addresses, 145-146
images as, 147
linking between, 141-144
linking to external web
content, 144
opening in new browser
windows, 146-147
relative-root links, 137
styling via CSS, 147-151
lists
CSS box model and, 292-295
horizontal navigation, 316-319
navigation lists, 305-319
placing list item indicators,
295-297
vertical navigation, 306-315
loading, timing, 25
managing
coding clarity, 431
comments, 429-430
documenting code, 429-430
indenting code, 431
maintainable code, 429-431
version control, 432-433
margins
browsers and, 292-294
margin property (CSS),
233-241
organizing, 135-136
larger websites, 425-429
simple websites, 422-425
single-page websites, 420-421
overflow property (CSS), text flow
and, 264
padding
browsers and, 292-294
padding property (CSS), 233,
241-244
print-friendly web pages, 365
criteria for print-friendliness,
366-369
designing style sheets for print
pages, 371-374
media-specific style sheets,
369-371
reviewing content for print-
friendliness, 366
Print Preview, viewing web content
in, 374-375
publishing
locally, 23
to blogs, 24
random content, displaying via
JavaScript, 382-386
relative addresses, 136-137
relative-root addresses, 136
responsive web design, 286-287
rollover (hidden) text, displaying,
327-328
searches
<meta> tags (HTML), 444-448
search engines, listing web
content with, 441-448
stylesheets
creating, 49-54
definition of, 47
external style sheets, 48-54
formatting properties, 55-59
formatting text color, 48
inline styles, 63
internal style sheets, 48,
62-63
layout properties, 55
linking to HTML documents, 53
selectors, 60
sizing text, 53
style classes, 59-61
style IDs, 61-62
style properties, 60
style rules, 48, 52-54, 61
validating, 64
tables
aligning within, 163-165
borders, 158-159
cells, creating, 156
cells, sizing, 161-162
color in, 165-166
creating, 156-160
CSS columns, 167-171
headings, creating, 156
images in, 166
page layout via, 166-167
rows, creating, 156
sizing, 160-162
spacing borders, 166
spanning within, 166
uses for, 156
testing, 24-25
text
flowing, 264
formatting, 102-114, 291-292
transferring, FTP, 33
validating, 42-43
viewing, 9, 33
web hosting providers, selecting,
10-12, 30
website architectures,
creating, 137
web fonts, 112-114
web hosting providers
A Small Orange, 11
bandwidth, 10
control panels, 11-12
customer service, 10
DailyRazor, 11
domain names, managing, 11
LunarPages, 11
pricing, 11
reliability, 10
selecting, 10-12, 30
server space, 10
web pages
absolute addresses, 136-137
align property (CSS), 233, 245
clear property (CSS), text flow
and, 264
color
140 cross-browser color
names, 121
best practices, 119-121
Colorblind Web Page Filter
tool, 127
hexadecimal color codes,
122-124
web pages

graphics
- adjusting color, 182
- Adobe Photoshop, 176
- aligning graphics, 197-201
- animated graphics, 189-190
- background graphics, 204-205
- banners, 184-186
- buttons, 184-186
- choosing software, 175-176
- clip art, 176
- compression, 177
- copyrights and, 176
- Creative Commons licenses, 176
- cropping, 178-179
- file sizes, 177
- Flickr, 190, 195
- GIFs, 186-190
- GIMP, 176-186
- Google Images, 190
- Google Picasa, 176
- grabbing from web pages, 176
- image maps, 206-211
- JPEG files, 183-184, 187-189
- Pixlr, 176
- placing graphics on web pages, 193-195
- PNG files, 187
- Red Eye Removal, 182
- republishing, 195
- resizing, 181
- resolution, 177
- specifying height/width, 197
- text descriptions, 195-196
- tiled backgrounds, 187-189
- transparent graphics, 187
- turning graphics into links, 201-204
- uses of, 177
- hidden (rollover) text, displaying, 327-328

links
- absolute links, 136
- anchor tags, 138-141
- email addresses, 145-146
- images as, 147
- linking between web pages, 141-144
- linking to external web pages, 144
- opening in new browser windows, 146-147
- relative-root links, 137
- styling via CSS, 147-151

lists
- CSS box model and, 292-295
- horizontal navigation, 316-319
- navigation lists, 305-319
- placing list item indicators, 295-297
- vertical navigation, 306-315

loading, timing, 25

managing
- coding clarity, 431
- comments, 429-430
- documenting code, 429-430
- indenting code, 431
- maintainable code, 429-431
- version control, 432-433

margins
- browsers and, 292-294
- margin property (CSS), 233, 241-244

multimedia
- <audio> element, 224-225
- audio/video playback via HTML5, 224-227
- creating, 218
- defining, 217
- embedding files, 221-224
- linking to files, 218-221
- MIME types, 222-223
- <object> element, 221-223
- QuickTime support, 221
- streaming files, 221
- <video> element, 226-227

organizing
- larger websites, 425-429
- simple websites, 422-425
- single-page websites, 420-421

overflow property (CSS), text flow and, 264

padding
- browsers and, 292-294
- padding property (CSS), 233, 241-244
print-friendly web pages, 365
  criteria for print-friendliness, 366-369
designing style sheets for print
  pages, 371-374
media-specific style sheets, 369-371
reviewing content for print-
  friendliness, 366
Print Preview, viewing web pages
  in, 374-375
random content, displaying via
JavaScript, 382-386
relative addresses, 136-137
relative-root addresses, 136
responsive web design, 286-287
robots, 441
rollover (hidden) text, displaying, 327-328
search engines, listing web pages
  with, 441-448
spiders, 441
style sheets
  creating, 49-54
  definition of, 47
external style sheets, 48, 51-54
formatting properties, 55-59
formatting text color, 48
inline styles, 63
internal style sheets, 48, 62-63
layout properties, 55
linking to HTML documents, 53
selectors, 60
sizing text, 53
style classes, 59-61
style IDs, 61-62
style properties, 60
style rules, 48, 52-54, 61
validating, 64
tables
  aligning within, 163-165
  borders, 158-159
  cells, creating, 156
  cells, sizing, 161-162
  color in, 165-166
  creating, 156-160
CSS columns, 167-171
headings, creating, 156
images in, 166
page layout via, 166-167
rows, creating, 156
sizing, 160-162
spacing borders, 166
spanning within, 165
uses for, 156
text
  flowing, 264
  formatting, 102-114, 291-292
transferring, FTP, 33
URL, creating, 449
validating, 42-43
viewing, 33
website architectures, 137
web servers
  document roots, 18-21
  FTP client connections, 18
web sockets, 391
websites
  aligning, 233, 245
  architectures, creating, 137
  bad website examples, 122
clear property (CSS), text flow
  and, 264
color
  140 cross-browser color
  names, 121
  best practices, 119-121
  Colorblind Web Page Filter
tool, 127
  hexadecimal color codes,
  122-124
  using, 119-121
  W3C color standards, 121
comparing, 30
connecting to, Classic FTP FTP
  client, 17
CSS box model, 253, 256
  borders, 254
  content, 254
  lists and, 292-295
  margins, 254
  padding, 254
  sizing elements, 254-255
dynamic websites, 379
  changing images based on
  user interaction, 388-389
  client-side scripting, 380
  DOM, 387
  JavaScript, 380-389
  server-side scripting, 380
  VBScript, 380
float property (CSS), 233, 246-249, 264
floating elements, 233
forms, 395
  check boxes, 405-406
  creating, 396-401
  grouping elements, 404
  hidden data, 404-405
  labeling form data, 403
  naming form data, 402
  passwords, 402
  pulling down pick lists, 407-408
  radio buttons, 407
  scrolling lists, 407-408
  selection lists, 407-408
  sending, 397
  submitting form data, 412-413
  text areas, 409-410
  text fields, 409-410
  text input, 401-402
user input, 401-402
graphics
  adjusting color, 182
Adobe Photoshop, 176
aligning graphics, 197-201
animated graphics, 189-190
background graphics, 204-205
banners, 184-186
buttons, 184-186
choosing software, 175-176
clip art, 176
copyrights and, 176
Creative Commons
  licenses, 176
cropping, 178-179
file sizes, 177
Flickr, 190, 195
GIF files, 186-190
GIMP, 176-186
Google Images, 190
Google Picasa, 176
grabbing from web pages, 176
imagemaps, 206-211
JPEG files, 183-184, 187-189
Pixlr, 176
placing graphics on web pages, 193-195
PNG files, 187
Red Eye Removal, 182
republishing, 195
resizing, 181
resolution, 177
specifying height/width, 197
text descriptions, 195-196
tiled backgrounds, 187-189
transparent graphics, 187
turning graphics into links, 201-204
uses of, 177
links
absolute links, 136
anchor tags, 138-141
e-mail addresses, 145-146
images as, 147
linking between web pages, 141-144
 linking to external web pages, 144
opening in new browser windows, 146-147
relative-root links, 137
styling via CSS, 147-151
lists
CSS box model and, 292-295
horizontal navigation, 316-319
navigation lists, 305-319
placing list item indicators, 295-297
vertical navigation, 306-315
managing
coding clarity, 431
comments, 429-430
documenting code, 429-430
indenting code, 431
maintainable code, 429-431
version control, 432-433
margins
browsers and, 292-294
margin property (CSS), 233-241
organizing, 420
Amazon.com, 425-426
BAWSI.org, 428
ESPN.com, 423-424
larger websites, 425-429
Peet’s Coffee & Tea, 426-427
simple websites, 422-425
single-page websites, 420-421
overflow property (CSS), text flow and, 264
padding
browsers and, 292-294
padding property (CSS), 233, 241-244
publicizing, 439-441
search engines, listing websites with, 441-448
SEO, 439, 448-450
single-page websites, 420-421
style sheets
creating, 49-54
definition of, 47
external style sheets, 48, 51-54
formatting properties, 55-59
formatting text color, 48
inline styles, 63
internal style sheets, 48, 62-63
layout properties, 55
linking to HTML documents, 53
selectors, 60
sizing text, 53
style classes, 59-61
style IDs, 61-62
style properties, 60
style rules, 48, 52-54, 61
validating, 64
testing, 12-13, 30
text, flowing, 264
traffic, increasing, 439-441
URL, creating, 449
web content
absolute addresses, 136-137
directories, 135-136
folders, 135-137
organizing, 135-136
relative addresses, 136-137
relative-root addresses, 136
web pages
print-friendly web pages, 365-375
viewing in Print Preview, 374-375
whitespace, HTML, 35
width
CSS box model, adjusting width in, 254-255
fixed/liquid hybrid layouts, setting minimum width in, 279
images, specifying width in, 197
width property (CSS), 55
windows (browsers), non-viewable window areas, 270
Windows Media Video, linking to images, 219
Word (MS), creating HTML files, 31
WordPress Theme Gallery, layouts and, 270
WWW (World Wide Web), HTML development, 6
WYSIWYG editors, creating HTML files, 31
X - Y - Z
XHTML
defining, 7
function of, 53
goal of, 53
xml: lang attribute (HTML tags), 34
xmlns attribute (HTML tags), 34
z-index property (CSS), 261-263
Zen Garden layouts (CSS), 270