

# 1

## Introduction to web development

This chapter introduces you to the concepts and terms that you need to work with HTML and CSS. When you're finished with this chapter, you'll have the background you need for learning how to build websites.

<b>How web applications work.....</b>	<b>4</b>
The components of a web application .....	4
How static web pages are processed.....	6
How dynamic web pages are processed .....	8
A survey of web browsers and server-side scripting languages.....	10
How client-side JavaScript fits into web development .....	12
<b>An introduction to HTML and CSS .....</b>	<b>14</b>
The HTML for a web page .....	14
The CSS for a web page.....	16
A short history of the HTML and CSS standards.....	18
<b>Tools for web development .....</b>	<b>20</b>
Text editors for HTML and CSS .....	20
IDEs for web development.....	22
FTP programs for uploading files to the web server .....	24
<b>How to view deployed web pages .....</b>	<b>26</b>
How to view a web page .....	26
How to view the source code for a web page .....	28
<b>Five critical web development issues .....</b>	<b>30</b>
Users and usability.....	30
Cross-browser compatibility.....	32
User accessibility.....	34
Search engine optimization.....	36
Responsive Web Design .....	38
<b>Perspective .....</b>	<b>40</b>

## 4 Section 1 The essential concepts and skills

# How web applications work

The *World Wide Web*, or web, consists of many components that work together to bring a web page to your desktop over the *Internet*. Before you start web pages of your own, you should have a basic understanding of how these components work together.

## The components of a web application

The first diagram in figure 1-1 shows that web applications consist of *clients* and a *web server*. The clients are the computers, tablets, and mobile devices that use the web applications. They access the web pages through programs known as *web browsers*. The web server holds the files that make up a web application.

A *network* is a system that allows clients and servers to communicate. The Internet in turn is a large network that consists of many smaller networks. In a diagram like this, the “cloud” represents the network or Internet that connects the clients and servers.

In general, you don't need to know how the cloud works. But you should have a general idea of what's going on. That's why the second diagram in this figure gives you a conceptual view of the architecture of the Internet.

To start, networks can be categorized by size. A *local area network (LAN)* is a small network of computers that are near each other and can communicate with each other over short distances. Computers on a LAN are typically in the same building or in adjacent buildings. This type of network is often called an *intranet*, and it can be used to run web applications for use by employees only.

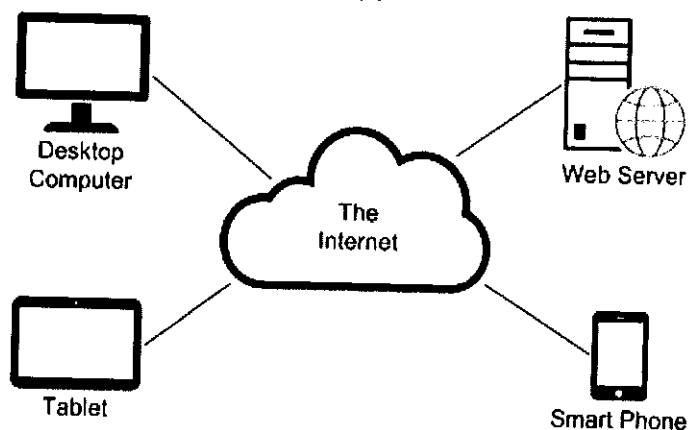
In contrast, a *wide area network (WAN)* consists of multiple LANs that have been connected together over long distances using *routers*. To pass information from one client to another, a router determines which network is closest to the destination and sends the information over that network. A WAN can be owned privately by one company or it can be shared by multiple companies.

An *Internet service provider (ISP)* is a company that owns a WAN that is connected to the Internet. An ISP leases access to its network to other companies that need to be connected to the Internet.

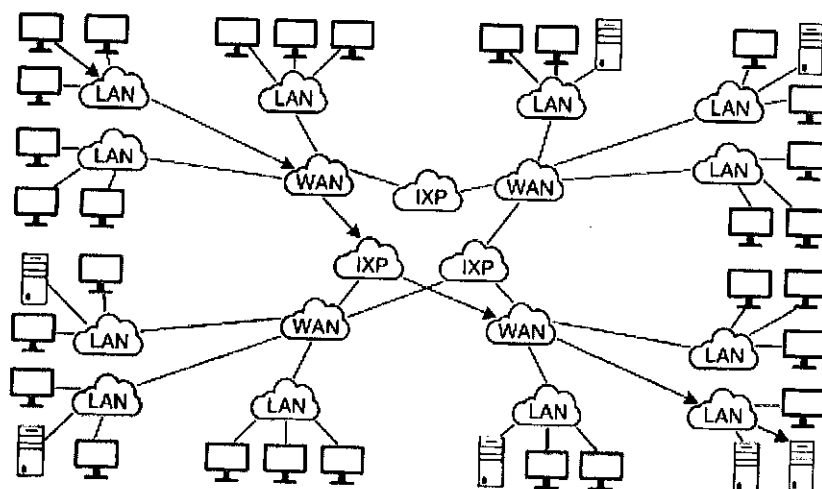
The Internet is a global network consisting of multiple WANs that have been connected together. ISPs connect their WANs at large routers called *Internet exchange points (IXP)*. This allows anyone connected to the Internet to exchange information with anyone else.

This diagram shows an example of data crossing the Internet. Here, data is being sent from the client in the top left to the server in the bottom right. First, the data leaves the client's LAN and enters the WAN owned by the client's ISP. Next, the data is routed through IXPs to the WAN owned by the server's ISP. Then, it enters the server's LAN and finally reaches the server. All of this can happen in less than 1/10th of a second.

### The components of a web application



### The architecture of the Internet



### Description

- A web application consists of clients, a web server, and a network. The *clients* use programs known as *web browsers* to request web pages from the web server. The *web server* returns the pages that are requested to the browser.
- A *local area network* (LAN) directly connects computers that are near each other. This kind of network is often called an *intranet*.
- A *wide area network* (WAN) consists of two or more LANs that are connected by *routers*. The routers route information from one network to another.
- The *Internet* consists of many WANs that have been connected at *Internet exchange points* (IXPs). There are several dozen IXPs located throughout the world.
- An *Internet service provider* (ISP) owns a WAN and leases access to its network. It connects its WAN to the rest of the Internet at one or more IXPs.

Figure 1-1 The components of a web application

6 Section 1 The essential concepts and skills

## How static web pages are processed

A *static web page* like the one at the top of figure 1-2 is a web page that only changes when the web developer changes it. This web page is sent directly from the web server to the web browser when the browser requests it.

The diagram in this figure shows how a web server processes a request for a static web page. This process begins when a client requests a web page using a web browser. Remember that a web browser is simply a program that runs on a client. To request a page using a web browser, the user can either type the address of the page into the browser's address bar or click a link in the current page that specifies the next page to load.

In either case, the web browser builds a request for the web page and sends it to the web server. This request, known as an *HTTP request*, is formatted using the *hypertext transfer protocol* (HTTP), which lets the web server know which file is being requested.

When the web server receives the HTTP request, it retrieves the requested file from the disk drive. This file contains the *HTML (HyperText Markup Language)* for the requested page. Then, the web server sends the file back to the browser as part of an *HTTP response*.

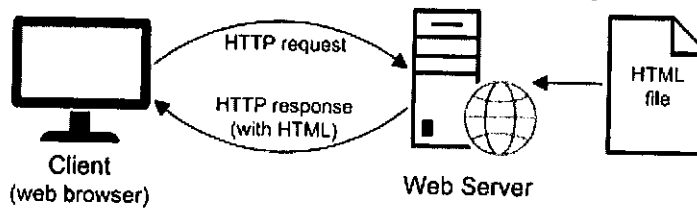
When the browser receives the HTTP response, it *renders* (translates) the HTML into a web page that is displayed in the browser. Then, the user can view the content. If the user requests another page, either by clicking a link or typing another web address into the browser's address bar, the process begins again.

In this book, you'll learn how to use HTML to create static web pages. You can spot these pages in a web browser by looking at the extension in the address bar. If the extension is .htm or .html, the page is a static web page.

### A static web page at <http://www.newtonforkranch.com>



### How a web server processes a static web page



### Description

- *Hypertext Markup Language (HTML)* is used to mark up web pages.
- A *static web page* is an HTML document that's stored on the web server and doesn't change. The filenames for static web pages have .htm or .html extensions.
- When the user requests a static web page, the browser sends an *HTTP request* to the web server that includes the name of the file that's being requested.
- When the web server receives the request, it retrieves the HTML for the web page and sends it back to the browser as part of an *HTTP response*.
- When the browser receives the HTTP response, it *renders* the HTML into a web page that is displayed in the browser.

Figure 1-2 How static web pages are processed

## How dynamic web pages are processed

A *dynamic web page* like the one in figure 1-3 is a page that's created by a program or script on the web server each time it is requested. This program or script is executed by an *application server* based on the data that's sent along with the HTTP request. In this example, the HTTP request identified the book that's shown. Then, the program or script for the requested page retrieved the image and data for that book from a *database server*.

The diagram in this figure shows how a web server processes a dynamic web page. The process begins when the user requests a page in a web browser. To do that, the user can either type the URL of the page in the browser's address bar, click a link that specifies the dynamic page to load, or click a button that submits a form that contains the data that the dynamic page should process.

In each case, the web browser builds an HTTP request and sends it to the web server. This request includes whatever data the application needs for processing the request. If, for example, the user has entered data into a form, that data will be included in the HTTP request.

When the web server receives the HTTP request, the server examines the file extension of the requested web page to identify the application server that should process the request. The web server then forwards the request to the application server that processes that type of web page.

Next, the application server retrieves the appropriate program or script from the hard drive. It also loads any form data that the user submitted. Then, it executes the script. As the script executes, it generates the HTML for the web page. If necessary, the script will request data from a database server and use that data as part of the web page it is generating.

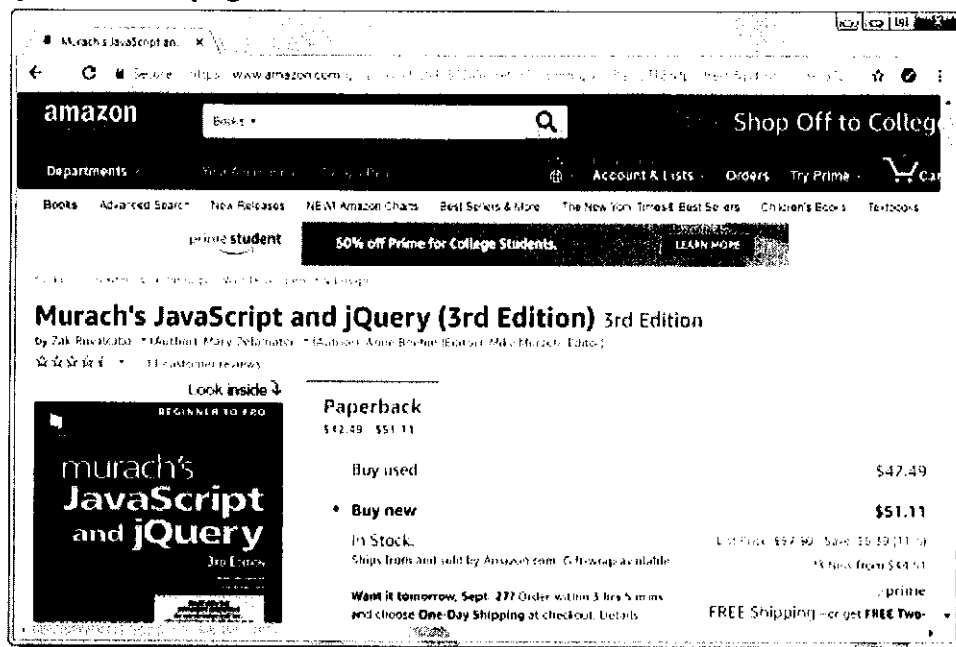
When the script is finished, the application server sends the dynamically generated HTML back to the web server. Then, the web server sends the HTML back to the browser in an HTTP response.

When the web browser receives the HTTP response, it renders the HTML and displays the web page. Note, however, that the web browser has no way to tell whether the HTML in the HTTP response was for a static page or a dynamic page. It just renders the HTML.

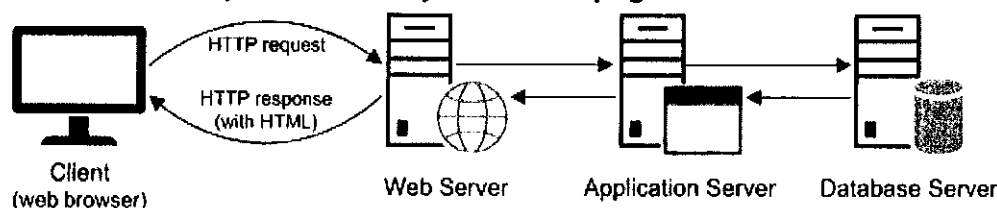
When the page is displayed, the user can view the content. Then, when the user requests another page, the process begins again. The process that begins with the user requesting a web page and ends with the server sending a response back to the client is called a *round trip*.

Dynamic web pages let you create interactive *web applications* that do all of the types of processing that you find on the Internet including eCommerce applications. Although you won't learn how to develop dynamic web pages in this book, you will learn how to create the HTML forms that send user data to the web server. Once you master HTML, you can learn how to use server-side technologies like JSP, ASP, or PHP to create the dynamic pages that a website needs.

## A dynamic web page at amazon.com



## How a web server processes a dynamic web page



### Description

- A *dynamic web page* is a web page that's generated by a server-side program or script.
- When a web server receives a request for a dynamic web page, it looks up the extension of the requested file to find out which *application server* should process the request.
- When the application server receives a request, it runs the specified script. Often, this script uses the data that it gets from the web browser to get the appropriate data from a *database server*. This script can also store the data that it receives in the database.
- When the application server finishes processing the data, it generates the HTML for a web page and returns it to the web server. Then, the web server returns the HTML to the web browser as part of an HTTP response.

Figure 1-3 How dynamic web pages are processed

## A survey of web browsers and server-side scripting languages

Figure 1-4 summarizes the six web browsers that are used the most today. Google's Chrome was released in 2008, grew quickly in popularity, and is now the most widely-used web browser. It has a clean, simple interface, it provides for searching directly from the address bar, and it has a large library of extensions and add-ons for developers. Plus, it's lightweight so it starts quickly and has a fast response time. Chrome is based on the WebKit rendering engine, and it's available for all major operating systems including Windows, Mac OS, and Linux.

Firefox is another widely used web browser. It is also available for all major operating systems. Firefox was built using source code from the original Netscape Navigator web browser.

Microsoft's Internet Explorer (IE) was once the most widely used web browser and is still widely used. However, with the rise of Chrome and Firefox, it isn't as widely used as it once was. In addition, Microsoft has released a newer browser named Edge that's designed to replace Internet Explorer. It's included with Windows 10.

Safari is the default web browser for Mac OS X. This browser was originally based on the same rendering engine as Chrome. As a result, they render pages similarly. Although Safari used to be available for Windows as well as Mac OS X, the Windows version is no longer supported.

Opera isn't widely used as a browser for desktop computers. However, it is more widely used as a browser for mobile devices.

Next, this figure summarizes the most common *scripting languages* for web servers. You use these languages along with a server-side technology to develop dynamic web pages. For instance, C# and Visual Basic are Microsoft languages that you can use to create ASP.NET web applications. Java is a free, open-source language that is commonly used to create Java Server Pages (JSPs). And PHP is another free, open-source language. To develop dynamic web pages, you need to choose the scripting language that you will use for *server-side processing*.

When you choose the scripting language, you also determine what web server you're going to need. For instance, Java Server Pages run on an Apache web server, which was developed by the Apache Software Foundation. It is an open-source software project that's available for free, and it runs on most operating systems, especially Linux systems. In contrast, ASP.NET web pages run on Microsoft's *Internet Information Services (IIS)*, which isn't open source and runs on Windows systems.



## Web browsers

Browser	Published by	Available on
Chrome	Google	All major operating systems
Firefox	Mozilla Corporation	All major operating systems
Internet Explorer	Microsoft	Windows systems prior to Windows 10
Edge	Microsoft	Windows 10
Safari	Apple	Macintosh
Opera	Opera Software	All major operating systems

## Server-side scripting languages

Language	Technology	Description
C#/VB	ASP.NET	Two free Microsoft languages that are commonly used to create server-side ASP.NET web applications. The web pages run on a Microsoft IIS web server and rely on the .NET Framework.
Java	JSP	A free, open-source language that is commonly used to create server-side Java Server Pages (JSPs). These pages typically run on an Apache web server and rely on the Java Runtime.
PHP	PHP	A free, open-source language that is commonly used to create server-side PHP web applications. PHP pages can run on an Apache or Microsoft IIS web server, as well as others.
Ruby	Ruby on Rails	A free, open-source language that is typically combined with the Rails framework to create server-side web applications.
Perl	CGI	A free, open-source language that was originally designed for use at the UNIX command line to manipulate text. It's commonly used to create server-side CGI web applications.
Python	Django/Flask	A free, open-source language that can be used to develop many types of applications. When combined with the Django or Flask framework, Python can be used to create server-side web applications.

## Description

- When you develop a website for general use, you need to test it on all of the web browsers listed above including all versions that are still in common use.
- To develop dynamic web pages, you use a server-side technology like ASP.NET, JSP, or PHP. Instructions are provided to each technology in the form of a programming language like C# or Visual Basic for ASP.NET, Java for JSP, and Perl for CGI.
- The scripting languages are designed to run on specific web servers. The two most popular web servers are Microsoft IIS (Internet Information Services) and Apache.

Figure 1-4 A survey of web browsers and server-side scripting languages

## How client-side JavaScript fits into web development

In contrast to the server-side processing that's done for dynamic web pages, *JavaScript* is a scripting language that provides for client-side processing. In the website in figure 1-5, for example, JavaScript is used to change the images that are shown without using server-side processing.

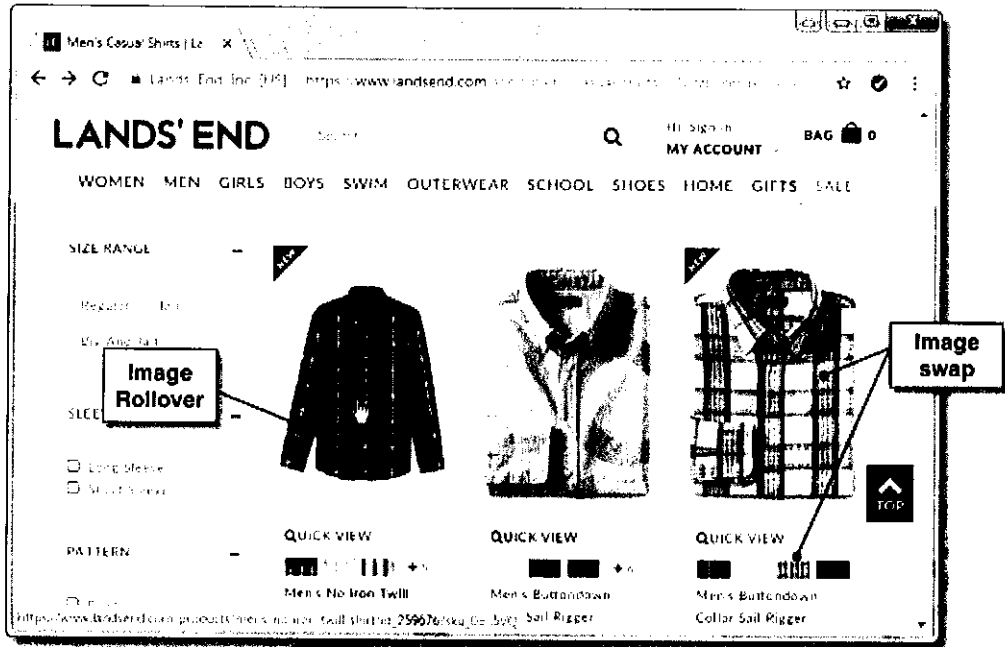
To make this work, all of the required images are loaded into the browser when the page is requested. Then, if the user clicks on one of the color swatches below a shirt, the shirt image is changed to the one with the right color. This is called an *image swap*. Similarly, if the user moves the mouse over a shirt, the image is replaced by a full-view image of the shirt. This is called an *image rollover*.

The diagram in this figure shows how JavaScript processing works. When a browser requests a web page, both the HTML and the related JavaScript are returned to the browser by the web server. Then, the JavaScript code is executed in the web browser by the browser's *JavaScript engine*. This takes some of the processing burden off the server and makes the application run faster. Often, JavaScript is used in conjunction with dynamic web pages, but it can also be used with static web pages.

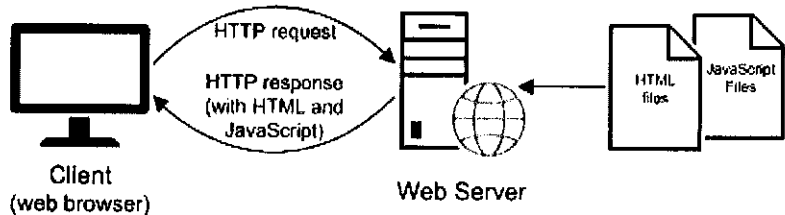
Besides image swaps and rollovers, there are many other uses for JavaScript. For instance, another common use is to validate the data that the user enters into an HTML form before it is sent to the server for processing. This saves unnecessary trips to the server. Other common uses of JavaScript are to rotate headlines or products in one area of a web page and provide animation. In fact, whenever you see a portion of a web page cycle through a series of text blocks or images, that's probably being done by JavaScript.

In this book, you won't learn how to code JavaScript. However, you will learn how to use existing JavaScript routines in chapter 19 of this book. There, you'll learn how to use JavaScript and a JavaScript library known as *jQuery* to enhance your web pages with features like image swaps, image rollovers, and data validation.

### A web page with image swaps and rollovers



### How JavaScript fits into this architecture



### Three of the many uses of JavaScript

- Data validation
- Image swaps and rollovers
- Slide shows

### Description

- *JavaScript* is a *client-side scripting language* that is run by the *JavaScript engine* of a web browser.
- When the browser requests an HTML page that contains JavaScript or a link to a JavaScript file, both the HTML and the JavaScript are loaded into the browser.
- Because JavaScript runs on the client, not the server, it provides functions that don't require a trip back to the server. This can help an application run more efficiently.

Figure 1-5 How client-side JavaScript fits into web development

14      *Section 1    The essential concepts and skills*

## **An introduction to HTML and CSS**

To develop a web page, you use HTML to define the content and structure of the page. Then, you use CSS to format that content. The topics that follow introduce you to HTML and CSS.

### **The HTML for a web page**

*HyperText Markup Language (HTML)* is used to define the content and structure of a web page. In figure 1-6, for example, you can see the HTML for a web page followed by a browser that shows how that page is displayed in the Chrome browser. Although you're going to learn how to code every aspect of an HTML page in this book, here's a brief introduction to what's going on.

The code for the entire page is called an *HTML document*. This document starts with a *DOCTYPE declaration* that is followed by *tags* that identify the *HTML elements* within the document. The *opening tag* for each element consists of the element name surrounded by angle brackets, as in `<html>`. And the *closing tag* consists of a left angle bracket, a forward slash, the element name, and the right angle bracket, as in `</html>`.

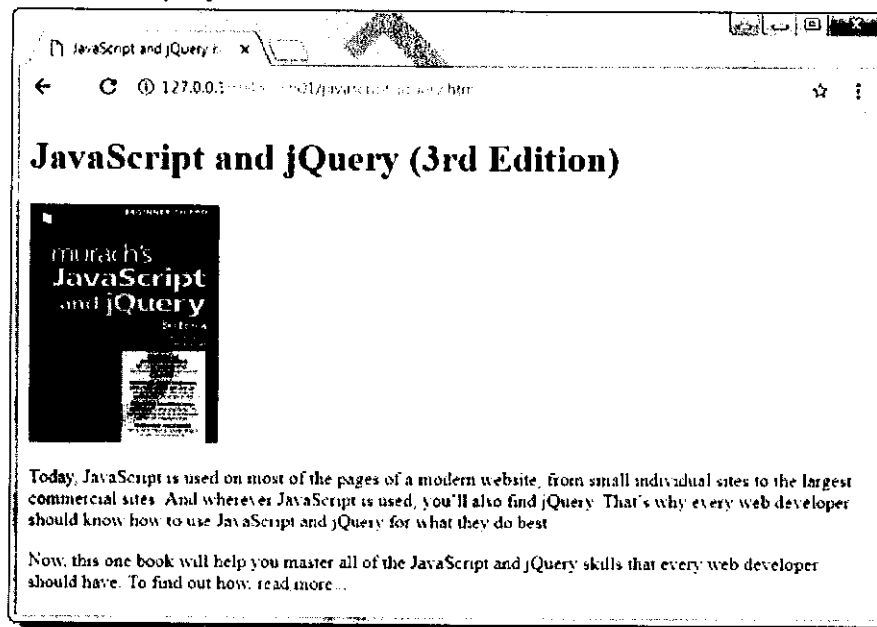
The basic structure of an HTML document consists of head and body elements that are coded within the `html` element. The head section contains elements that provide information about the document. The body section contains the elements that will be displayed in the web browser. For instance, the title element in the head section provides the title that's shown in the tab for the page in the web browser, while the `h1` element in the body section provides the heading that's displayed in the browser window.

Many elements can be coded with *attributes* that identify the element and define the way the content in the element is displayed. These attributes are coded within the opening tag, and each attribute consists of an attribute name, an equals sign, and the attribute value. For instance, the `<img>` tag in this example has two attributes named `src` and `alt`. In this case, the `src` attribute provides the name of the image file that should be displayed and the `alt` attribute provides the text that should be displayed if the image can't be found.

### The code for an HTML file named javascript\_jquery.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>JavaScript and jQuery book</title>
  </head>
  <body>
    <h1>JavaScript and jQuery (3rd Edition)</h1>
    
    <p>Today, JavaScript is used on most of the pages of a modern
      website, from small individual sites to the largest commercial
      sites. And wherever JavaScript is used, you'll also find jQuery.
      That's why every web developer should know how to use JavaScript
      for what it does best and jQuery for what it does best. </p>
    <p>Now, this one book will help you master all of the JavaScript and
      jQuery skills that every web developer should have. To find out
      how, <a href="">read more...</a></p>
  </body>
</html>
```

### The HTML displayed in a web browser



### Description

- **HTML (HyperText Markup Language)** is used to define the structure and content of a web page.

Figure 1-6 The HTML for a web page

## The CSS for a web page

Not long ago, HTML documents were coded so the HTML not only defined the content and structure of the web page but also the formatting of that content. However, this mix of structural and formatting elements made it hard to edit, maintain, and reformat the web pages.

Today, however, *Cascading Style Sheets (CSS)* let you separate the formatting from the content and structure of a web page. As a result, the formatting that was once done with HTML should now be done with CSS.

In most cases, the best way to apply a *style sheet* to an HTML document is to code the styles in a separate file called an *external style sheet*. Then, to apply the style sheet, you code a link element in the head section of the HTML document as shown at the top of figure 1-7. Here, the href attribute of the tag says that the style sheet in the file named book.css should be applied to the HTML document.

After this link element, you can see the CSS that's in the book.css file. This is followed by a browser that shows how the web page is displayed after the style sheet has been applied to it. If you compare this to the browser in the previous figure, you can see that the page is now centered with a border around it, the font for the text has been changed, there's less spacing between paragraphs, and the text is displayed to the right of the book image. This gives you a quick idea of how much you can do with CSS.

Although you're going to learn how to code CSS in chapters 4, 5, and 6, here's a brief introduction to how the CSS works. First, this CSS file consists of four *style rules*. Each of these style rules consists of a *selector* and one or more *declarations* enclosed in braces { }. The selector identifies one or more HTML elements, and the declarations specify the formatting for the elements.

For instance, the first style rule applies to the body element. Its first declaration says that the font family for the content should be Arial, Helvetica, or the default sans-serif type, in that order of preference. Then, the second declaration says that the font-size should be 100% of the browser's default font size. These declarations set the base font and font size for the elements that are coded within the body.

The third declaration for the body sets its width to 560 pixels. Then, the fourth declaration sets the top and bottom margins to zero and the left and right margins to auto, which centers the page in the browser window. Finally, the fifth declaration sets the padding within the body to 1 em (a unit that you'll learn about in chapter 4), and the sixth declaration adds a solid, navy border to the body.

Similarly, the second style rule formats the h1 element in the HTML with a larger font size and the navy color. The third style rule formats the image by floating it to the left so the <p> elements are displayed to its right. And the fourth style rule changes the spacing between <p> elements.

You will of course learn all of the details for coding style rules in this book, but this should give you an idea of what's going on. In short, the HTML defines the content and structure of the document, and the CSS defines the formatting of the content. This separates the content from the formatting, which makes it easier to create and maintain web pages.

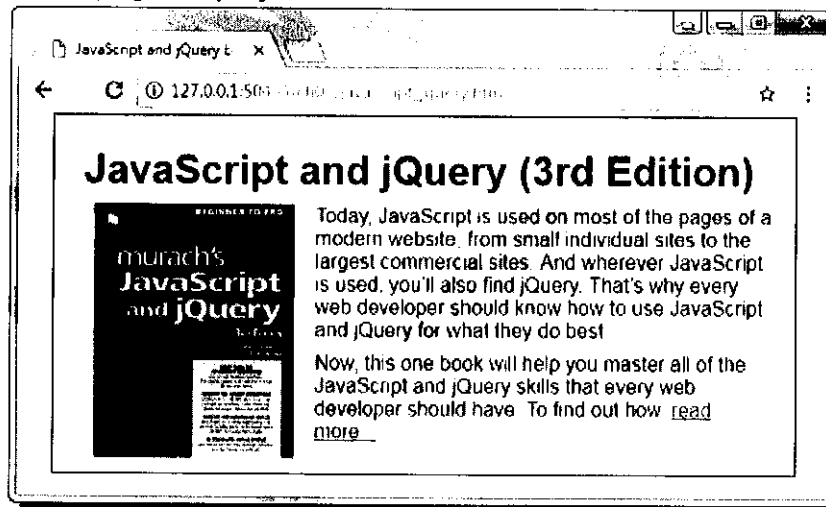
### The element in the head section of the HTML file that links it to the CSS file

```
<link rel="stylesheet" href="book.css">
```

### The code for the CSS file named book.css

```
body {  
    font-family: Arial, Helvetica, sans-serif;  
    font-size: 100%;  
    width: 560px;  
    margin: 0 auto;  
    padding: 1em;  
    border: 1px solid navy;  
}  
h1 {  
    margin: 0;  
    padding: .25em;  
    font-size: 200%;  
    color: navy;  
}  
img {  
    float: left;  
    margin: 0 1em 1 em 1em;  
}  
p {  
    margin: 0;  
    padding-bottom: .5em;  
}
```

### The web page displayed in a web browser



### Description

- *Cascading Style Sheets (CSS)* are used to control how web pages are displayed by specifying the fonts, colors, borders, spacing, and layout of the pages.

Figure 1-7 The CSS for a web page

## A short history of the HTML and CSS standards

As figure 1-8 shows, HTML is a standardized language that has been around since the early 1990's. During the 90's, three different versions of HTML were adopted. Then, XHTML 1.0 was adopted in January 2000.

*XHTML (eXtensible HTML)* is a modified version of HTML that uses the same syntax and supports the same elements as HTML 4.01, but relies on the strict rules and guidelines that govern how XML is written. This allows XHTML to be read and manipulated by automated tools. This also allows XHTML editors to identify errors in the document structure more easily.

Today, HTML 5 along with its first revision, HTML 5.1, replaces the old HTML and XHTML standards with a single standard. Although HTML 5.0 provided many new features when it was released, all modern browsers support the features of both HTML 5.0 and HTML 5.1.

As this figure also shows, CSS is a standardized language that goes back to 1996. However, it wasn't accepted as the right way to format HTML pages until more recently. Today, CSS 3.0 is supported by all modern browsers, and it is widely accepted as the right way to format web pages. Note that there are no plans for CSS 4.0.

In this book, you're going to learn how to use the new features of HTML 5.0 and HTML 5.1 as well the new features of CSS 3. Of course, you will also learn how to use the HTML and CSS from previous specifications that is included in the current specifications. Please note that from this point forward we'll refer to HTML 5 and CSS 3 as *HTML5* and *CSS3*.

What about browsers that don't support the new features of HTML5 and CSS3? This book will also show you how to get your web pages to work in those older browsers. Often, though, you won't have to do anything because the new features will just be ignored by those browsers with no harm done. If, for example, you use CSS3 to provide for rounded corners or shadows on HTML boxes, these features will improve the graphics in some browsers, but they'll be ignored in other browsers.

This figure also presents two websites that you ought to become familiar with. The first is for the *World Wide Web Consortium*, which is commonly referred to as *W3C*. This is the group that develops the standards, and this site should be one of your primary sources for HTML and CSS information.

The second website is for the *Web Hypertext Application Technology Working Group (WHATWG)*. This is a community of people interested in evolving HTML and related technologies, and this site should be another primary source for HTML and CSS information.



### Highlights in the development of the HTML standards

Version	Description
HTML 1.0	A draft specification released in January 1993 that was never adopted as a standard.
HTML 2.0	Adopted in November 1995.
HTML 4.0	Adopted in December 1997. It formalized new features that were used by web browsers and deprecated older features.
HTML 4.01	Adopted in December 1999 and updated through May 2001.
XHTML 1.0	Adopted in January 2000 and revised in August 2002. It reformulates HTML 4 using the syntax of XML, which makes it easier to parse the web page. This allows automated tools to find errors in a web page.
XHTML 1.1	Adopted in May 2001. The control of the presentation of content is now done through CSS.
HTML 5	Released as a working draft in January 2008. Originally, it defined an HTML version called HTML 5 and an XHTML version called XHTML 5. Today, the draft has been enhanced into a new HTML specification that replaces both HTML 4 and XHTML 1.
HTML 5.1	Released as the first minor revision of HTML 5 in April of 2013. It added the main element and removed the hgroup element. Accepted as a W3C standard in November of 2016.

### Highlights in the development of the CSS standards

Version	Description
1.0	Adopted in December 1996.
2.0	Adopted in May 1998.
2.1	First released as a candidate standard in February 2004, it returned to working draft status in June 2005. It became a candidate standard again in July 2007.
3.0	A modularized version of CSS with the earliest drafts in June 1999. Builds on CSS 2.1 module by module, and each module is accepted as a standard independently.

### Two websites that you should become familiar with

- The *World Wide Web Consortium (W3C)* is an international community in which member organizations, a full-time staff, and the public work together to develop Web standards. Its website address is: [www.w3.org](http://www.w3.org).
- The *Web Hypertext Application Technology Working Group (WHATWG)* is a community of people interested in evolving HTML and related technologies. Its website address is: [www.whatwg.org](http://www.whatwg.org).

### Description

- Today, all modern browsers support HTML 5.1 and CSS 3.0.

Figure 1-8 A short history of the HTML and CSS standards

## Tools for web development

To create and edit the HTML and CSS files for a website, you need either a text editor or an IDE for web development. To deploy a website on the Internet, you also need an FTP program to upload the files from your computer or network server to the web server. You'll learn about these tools next.

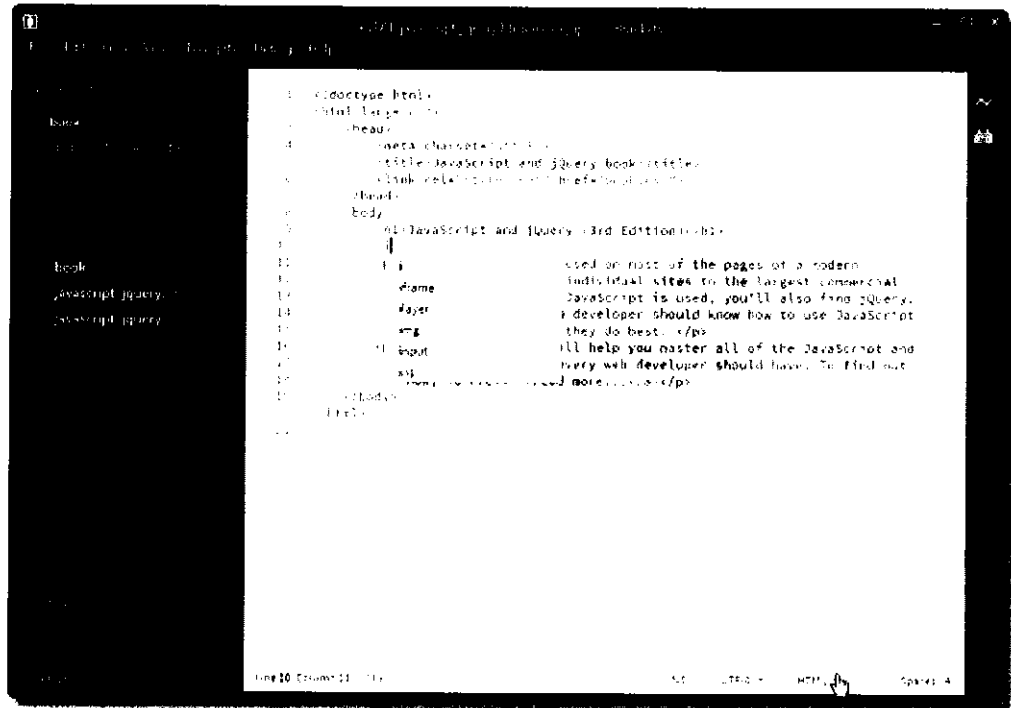
### Text editors for HTML and CSS

A *text editor* lets you enter and edit text. Although you can use any text editor to enter and edit HTML and CSS files, a better editor can speed development time and reduce coding errors. For that reason, we recommend Brackets. It is a free editor that runs on both Windows and Mac OS systems, and it has many excellent features. For instance, Brackets provides syntax highlighting and auto-completion lists that let you select an item after you enter the first characters.

Brackets, auto-completion feature, called *code hints*, is illustrated in figure 1-9. Here, the developer has entered the left angle bracket and the first letter of an `<img>` tag, and the auto-completion list shows the words that start with that letter. At that point, the developer can move the cursor to the right word in the list and press the Enter key to enter that word into the code. Then, when the developer enters the right angle bracket, Brackets adds the ending tag so the developer can type the content between the two tags.

In the next chapter, you'll learn how to use Brackets to develop your web pages. That will help you appreciate the many features that it provides. Please note, however, that you can use any text editor that you like for developing your web pages. Or, you can use an IDE like one of those in the next figure.

Brackets with the auto-completion feature



Five free text editors that you can use with this book

Editor	Runs on
Brackets	Windows, Macintosh, Linux
Atom 1	Windows, Macintosh, Linux
Visual Studio Code	Windows, Macintosh, Linux
Notepad++	Windows
TextMate	Macintosh

Description

- A *text editor* lets you enter and edit the HTML and CSS files for a web application.
- Some common features of a text editor for HTML and CSS are syntax highlighting and auto-completion.
- Today, there are many text editors for developing web applications with HTML and CSS. Most of them are either free or inexpensive.
- In chapter 2, we show how to use Brackets because it's free; it runs on Windows, Macintosh, and Linux; and it has many excellent features. However, you can use whichever text editor or IDE you prefer as you develop your web pages.

Figure 1-9 Text editors for HTML and CSS

## IDEs for web development

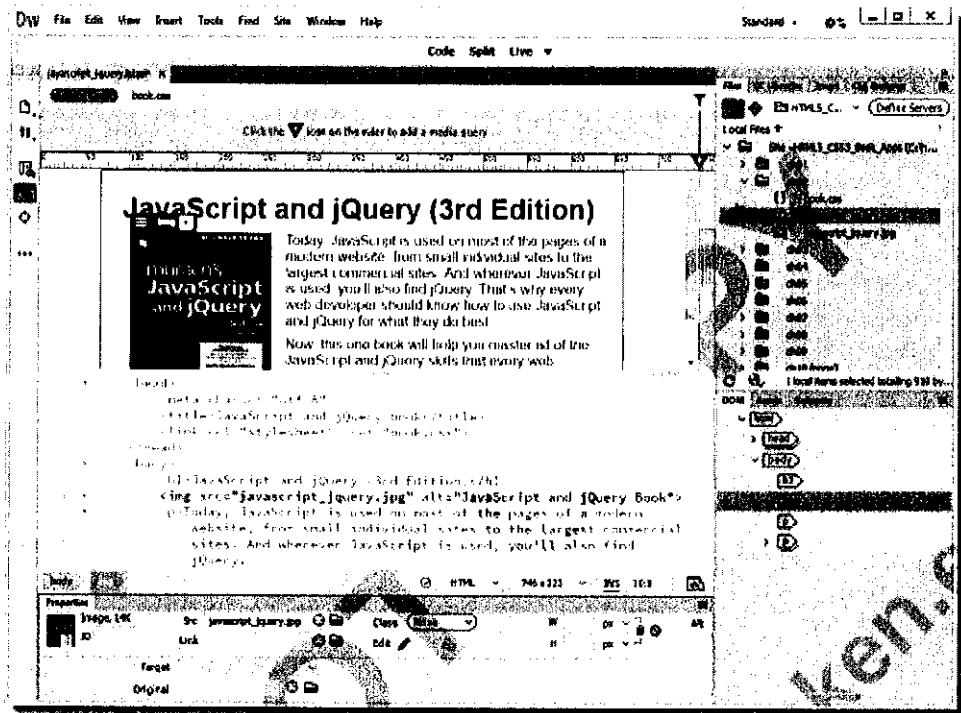
After you get some experience with a text editor like Brackets, you may be interested in an *Integrated Development Environment (IDE)* for web development. For instance, Adobe Dreamweaver has long been the most popular commercial IDE for web development.

The latest release of Dreamweaver is Dreamweaver CC 2018. As you can see in figure 1-10, it lets you edit HTML or CSS code in one panel of its window while it shows you how the web page will look in another panel. Dreamweaver CC also lets you generate code instead of entering it by dragging the symbols for common elements onto the HTML document. It provides the starting code for an HTML document whenever you start a new file. It helps you manage the folders and files for the website. It provides an FTP program that uploads the pages from your development server to your Internet server. And it has many other features.

In general, an IDE like Dreamweaver can help you make dramatic improvements in your productivity. That's why professional web developers often use IDEs for web development. Note, however, that IDEs vary considerably in features and price. For instance, some IDEs offer features that let you integrate your HTML and CSS development with JavaScript and server-side scripting languages like PHP, Perl, and Python. So before you buy an IDE, you need to find the IDE that best suits your requirements and budget.

At the highest level, an IDE for web development can include all of the programs that are required for developing a website. IDEs like this are often referred to as *suites*. For example, a suite of Adobe Creative Cloud products might include Photoshop CC for editing photos and images, Illustrator CC for creating and editing illustrations, Animate CC for adding animation and interactivity to web pages, and several other programs that are related to web development.

## Adobe Dreamweaver CC



## Popular IDEs for web development

IDE	Runs on
Adobe Dreamweaver CC	Windows and Macintosh
Microsoft Visual Studio	Windows
WebStorm	Windows, Macintosh, and Linux
Eclipse	Windows, Macintosh, and Linux
NetBeans	Windows, Macintosh, Linux, and more
Aptana Studio 3	Windows, Macintosh, and Linux
Cloud9	The Web

## Description

- An *Integrated Development Environment (IDE)* goes beyond text editing to provide other features for the development of websites.

Figure 1-10 IDEs for web development

## FTP programs for uploading files to the web server

If you want to *deploy* (or *publish*) your website on the Internet, you need to transfer the folders and files for your website from your computer or network to a web server with Internet access. To do that, you use an *FTP program* like one of those listed in figure 1-11. This type of program uses *File Transfer Protocol* to transfer files to or from the web server.

IDEs like Dreamweaver CC typically include FTP programs. If you're using a text editor, you may be able to find a plugin for an FTP program. Then, you can access that FTP program from your text editor. For example, eqFTP is a plugin that can be used with Brackets. By the way, Brackets calls these *extensions*, and you'll learn how to install them in chapter 2.

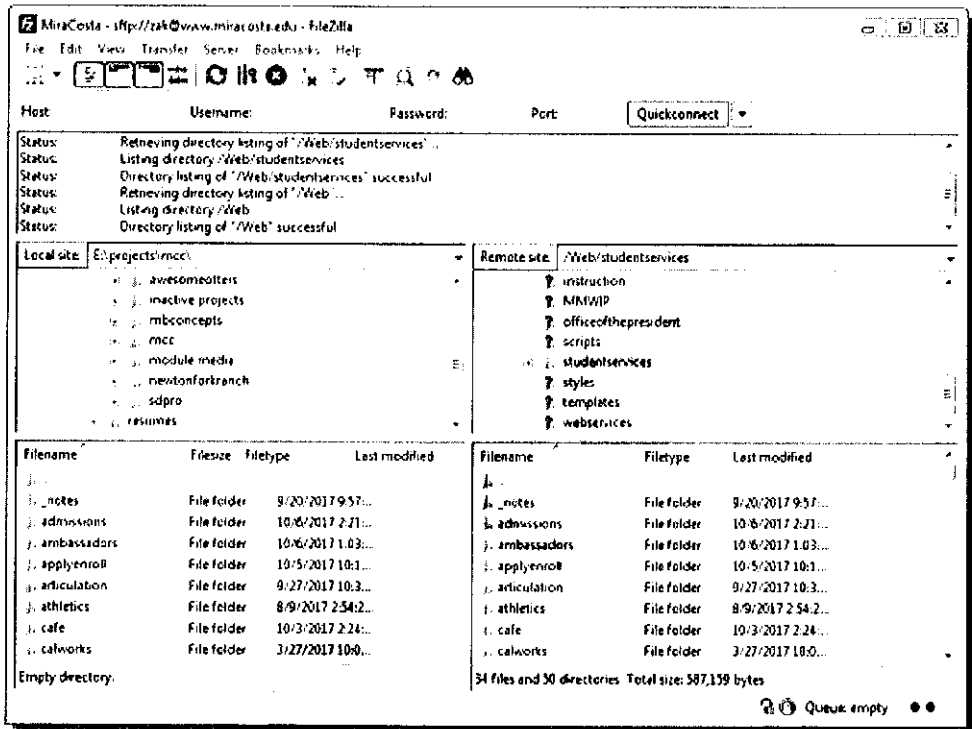
If you don't already have an Internet web server, one option is to find an *Internet Service Provider (ISP)* that provides *web hosting*. If you search the web, you'll be able to find many ISPs that provide web hosting, often for a small monthly fee.

If you're going to use dynamic web pages on your website, you need to find an ISP that supports the server-side technology and the database that you're going to use. If, for example, you're going to use PHP with a MySQL database, which is a common combination, you need to find an ISP that supports that.

When you select a web host, you get an *IP address* like 64.71.179.86 that uniquely identifies your website (IP stands for Internet Protocol). Then, you can get a *domain name* like www.murach.com. To do that, you can use any number of companies on the Internet, and sometimes you can get the domain name from your ISP. Until you get your domain name, you can use the IP address to access your site.

After you get a web host, you use your FTP program to upload the files for your website to the web server of the web host. Then, you can test your website on the Internet. When you're through testing, you can announce your website to the world and let it go live.

# FileZilla as it is used to upload files to the web server



## Some popular FTP programs

Program	Description
FileZilla	A free program for Windows, Macintosh, and Linux.
FTP Voyager	An inexpensive program for Windows.
CuteFTP	An inexpensive program for Windows and Macintosh.
Fetch	An inexpensive program for Macintosh.

## Description

- To *deploy* (or *publish*) a website on the Internet, you need to transfer the folders and files for the website from your computer or network to a web server on the Internet. To do that, you use an *FTP program* that uses *File Transfer Protocol*.
- An FTP program not only lets you transfer files from a client to a web server but also from a web server to a client.
- Most IDEs have built-in FTP programs. For instance, both Dreamweaver CC and Aptana Studio 3 provide FTP.
- If you're using a text editor, you typically have to use a separate FTP program or add a plugin FTP program to your editor.
- For more information on deploying an application to a web server, please see chapter 18.

Figure 1-11 FTP programs for uploading files to the web server

## How to view deployed web pages

Next, you'll learn how to view a web page in a web browser and how to view the source code for a web page that's displayed in the browser. These are valuable skills as you test your own web pages or study the pages on other sites.

### How to view a web page

Figure 1-12 shows you how to view a web page on the Internet. One way is to enter a *uniform resource locator (URL)* into the address bar of your browser. The other is to click on a link on a web page that requests another page.

As the diagram at the start of this figure shows, the URL for an Internet page consists of four components. In most cases, the *protocol* is HTTP. If you omit the protocol, the browser uses HTTP as the default.

The second component is the *domain name* that identifies the web server that the HTTP request will be sent to. The web browser uses this name to look up the address of the web server for the domain. Although you can't omit the domain name, you can often omit the "www." from the domain name.

The third component is the *path* where the file resides on the server. The path lists the folders that contain the file. Forward slashes are used to separate the names in the path and to represent the server's top-level folder at the start of the path. In this example, the path is "/studentservices/scholarships".

The last component is the name of the file. In this example, the file is named `index.html`. If you omit the filename, the web server will search for a default document in the path. Depending on the web server, this file will be named `index.html`, `default.htm`, or some variation of the two.

If you want to view an HTML page that's on your own computer or an intranet, you can use one of the three techniques summarized in this figure. First, your text editor or IDE should make it easy to view a page that you're working on. Second, you can find the file in your file explorer, and then double-click or right-click on it. Third, if the browser has a menu bar, you can use the File→Open or File→Open File command.

At the bottom of this figure, you can see our naming recommendations for your folders and files. In general, we recommend that folder and filenames should only contain lowercase letters, numbers, underscores or hyphens, and the period. In the examples in this book, you'll see the author's preference, which is to use underscores instead of hyphens to separate the words in a name. But some developers use hyphens instead of underscores.

The other recommendation is to create names that clearly indicate the contents of your folders and web pages. This can improve search engine optimization (SEO), which you'll learn more about in a moment.

Incidentally, Linux/Apache web servers are case-sensitive. Then, if a URL specifies a folder named "Images", but the folder on the server is actually named "images", the web server will report that it cannot find the file. By using lowercase letters only, you avoid this problem.

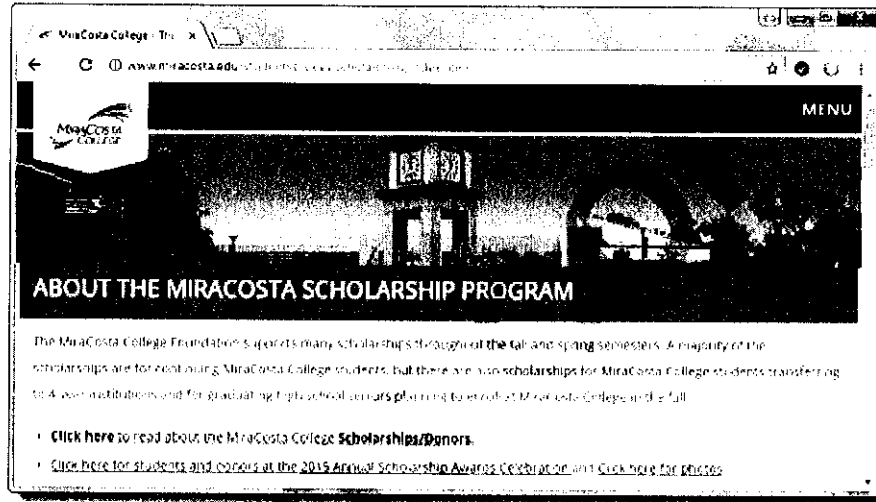


## The components of an HTTP URL

`http://www.modulemedia.com/ourwork/index.html`

protocol	domain name	path	filename
----------	-------------	------	----------

`http://www.miracosta.edu/student-services/scholarships/index.html`



## What happens if you omit parts of a URL

- If you omit the protocol, the default of `http://` will be used.
- If you omit the filename, the default document name for the web server will be used. This is typically `index.html`, `default.htm`, or some variation.

## Two ways to access a web page on the Internet

- Enter the URL of a web page into the browser's address bar.
- Click on a link in the current web page to load the next web page.

## Three ways to access a web page on your own server or computer

- Use the features of your text editor or IDE.
- Find the file in your file explorer. Then, double-click on it to open it in your default browser. Or, right-click on it and use the Open With command to select the browser.
- If your browser has a menu bar, use the File→Open or File→Open File command.

## Naming recommendations for your own folders and files

- Create names for folders and files that consist of lowercase letters, numbers, underscores or hyphens, and the period.
- Use filenames that clearly indicate what a page contains. This is good for search engine optimization (see figure 1-17).

Figure 1-12 How to view a web page

## How to view the source code for a web page

When a web page is displayed by a browser, you can use the techniques in figure 1-13 to view the HTML code for the page. If, for example, you're using the Chrome browser, you can right-click the page and then select the View Page Source command to see the HTML code in a separate window. In this example, the HTML code for the page in the previous figure is shown.

Viewing the source code can be useful when you're testing an application. But you can also use this technique to view the HTML for the pages of other sites on the Internet. This can be a good way to learn how other sites work. Although some sites use various techniques to hide their code, a lot of the code for Internet sites is available.

If the CSS for an HTML page is stored in an external file, you can sometimes use the Chrome or Firefox browser to open that file by clicking on the path in the HTML code. In the code in this figure, for example, you can click on `styles/layout.css` to access the second of the six CSS files that this site uses. This lets you analyze how the CSS code works.

If you're using Edge, the HTML source code for a page is displayed in the Debugger tab of the Developer Tools. Then, to view the CSS in an external file, you can hold the Ctrl key down and click on the path in the HTML code.

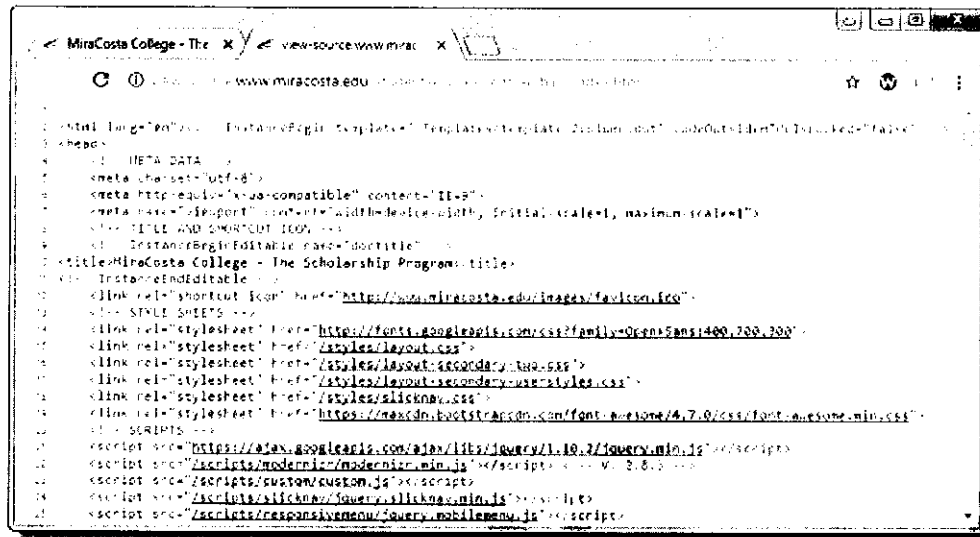
If you're using Internet Explorer, you can view the CSS code in an external file by entering the URL for the CSS file in the browser's address bar. For instance, for the first CSS file that's identified by the link element in the HTML code in this figure, you can enter this address into the browser:

<http://www.miracosta.edu/styles/layout.css>

Then, when you press the Enter key, the file is opened. In chapter 3, you'll learn more about the relative addresses that are used in HTML code so you'll be able to determine what their URLs are.

When you view the HTML code for a web page, keep in mind that it may include embedded CSS code or JavaScript code. Beyond that, you'll find that all but the simplest sites are quite complicated. Once you finish this book, though, you should be able to figure out how the HTML and CSS for most sites work. And you'll learn a lot by studying how the best sites are coded.

## The HTML source code for the page in figure 1-12



### How to view the HTML source code for a web page

- In Chrome, right-click the page and select the View Page Source command.
- In Internet Explorer or Edge, right-click the page and select the View Source command.
- In Chrome or Internet Explorer, the source code is displayed in a new browser tab or a separate window.
- In Edge, the source code is displayed in the Debugger tab of the Developer Tools. Also, if the View Source command isn't displayed when you right-click on the page, you may have to push the F12 key first to open the Developer Tools.

### How to view the CSS code in an external CSS file

- In Chrome, click on the link that refers to the CSS file.
- In Edge, hold down the Ctrl key and click on the link that refers to the CSS file.
- In Internet Explorer, enter the URL for the CSS file in the address bar of your web browser.

### Description

- Viewing the source code for a page is sometimes useful when debugging. But it also provides a way to see how other websites are coded.
- If the CSS for the page is stored in the HTML file, you can also see that code.
- If the CSS for the page is stored in external files, you can view that code by using the techniques above.

Figure 1-13 How to view the source code for a web page

## Five critical web development issues

Whenever you develop a web application, you should be aware of the issues that are presented in the next five figures. Then, as you progress through this book, you will be given guidelines for coding the tags and attributes that help provide for cross-browser compatibility, user accessibility, and search engine optimization.

### Users and usability

Before you design a website, you need to think about who your users are going to be and what they are going to expect. After all, it is your users who are going to determine the success of your website.

What do users want when they reach a website? They want to find what they're looking for as quickly and easily as possible. And when they find it, they want to extract the information or do the task as quickly and easily as possible.

How do users use a web page? They don't read it in an orderly way, and they don't like to scroll any more than they have to. Instead, they scan the page to see if they can find what they're looking for or a link to what they're looking for. Often, they click quickly on a link to see if it gives them what they want, and if it doesn't, they click on the Back button to return to where they were. In fact, users click on the Back button more than 30% of the time when they reach a new page.

If the users can't find what they're looking for or get too frustrated, they leave the site. It's that simple. For some websites, more than 50% of first-time visitors to the home page leave without ever going to another page.

In web development terms, what the users want is *usability*. This term refers to how easy it is to use a website, and usability is one of the key factors that determines the effectiveness of a website. If a site is easy to use, it has a chance to be effective. If it isn't easy to use, it probably won't be effective.

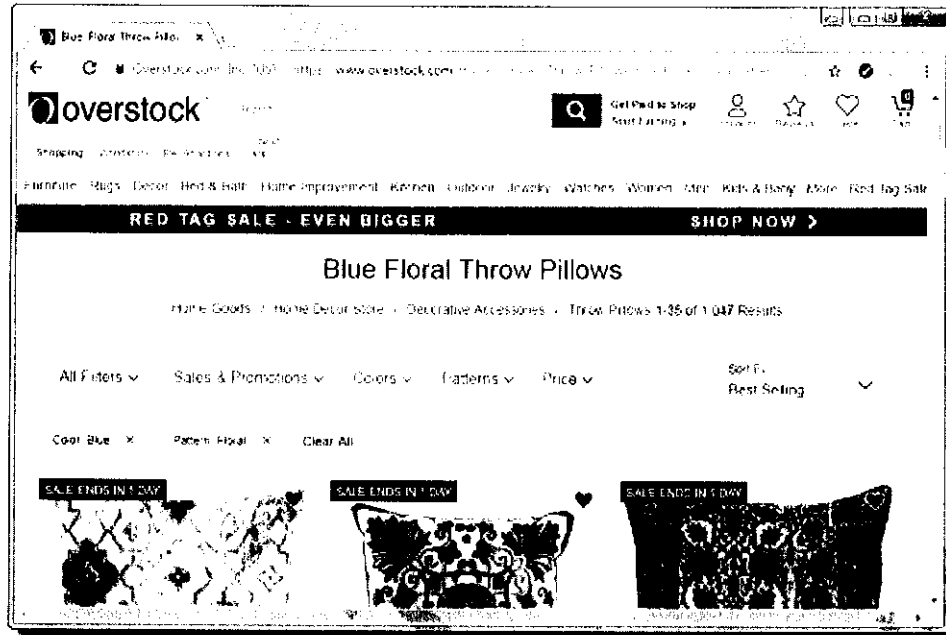
Figure 1-14 presents one page of a website that has a high degree of usability, and it presents four guidelines for improving usability. First, you should try to present the essential information "above the fold." This term refers to what's shown on the screen when a new page is displayed, which is analogous to the top half of a newspaper. This reduces the need for scrolling, and it gives the page a better chance for success.

Second, you should try to group related items and limit the number of groups on each page. That will make the page look more manageable and will help people find what they're looking for.

Third, you should adhere to the current conventions for the header. For instance, the header should include a logo and tagline that identify the website. It should also include a navigation bar and links to utilities. If your site requires a search function, it should also be in the header.

Fourth, you should adhere to the current conventions for navigation. For instance, the home page should be displayed when the user clicks on the logo, and the shopping cart should be displayed when the user clicks the cart icon. In addition, it should be clear what other items on the page are clickable.

## A website that is easy to use



### What website users want

- To find what they're looking for as quickly and easily as possible
- To get information or do a task as quickly and easily as possible

### How website users use a web page

- They scan the page to find what they're looking for or a link to what they're looking for, and they don't like to scroll. If they get frustrated, they leave.
- They often click on links and buttons with the hope of finding what they're looking for, and they frequently click on the Back button when they don't find it.

### Four guidelines for improving usability

- Present as much critical information as possible "above the fold" so the user has to scroll less.
- Group related items and limit the number of groups on each page.
- Include a header that identifies the site and provides a navigation bar and links to utilities.
- Use current navigation conventions, like including a logo that goes to your home page when clicked and a cart icon that goes to your shopping cart when clicked.

### Description

- *Usability* refers to how easy it is to use a website, and usability is a critical requirement for an effective website.

Figure 1-14 The issue of usability

If you look at the website in figure 1-14, you can see that it has implemented these guidelines. All of the critical information is presented above the fold. The page is divided into a header and three other well-defined groups. It's also easy to tell where to click on the page. That's true even though this is a large website with hundreds of product categories and thousands of products.

## Cross-browser compatibility

If you want your website to be used by as many visitors as possible, you need to make sure that your web pages are compatible with as many browsers as possible. That's known as *cross-browser compatibility*. That means you should test your applications on as many browsers as possible, including the six browsers summarized in figure 1-15 as well as older versions of Internet Explorer.

The table in this figure shows the current release numbers of these browsers and their ratings for HTML5 support. To get an updated version of this information, you can go to the website at [www.html5test.com](http://www.html5test.com). When you access this website, it will also provide details of how the browser that you're using supports HTML5. If a browser doesn't support an HTML5 or CSS3 feature that's presented in this book, this book will show you the workaround that you need to use.

In general, Internet Explorer gives web developers the most problems because it's the least standard. In contrast, the other five browsers generally support the same features so if a web page runs on one of them, it will also run on the others. The other five browsers also provide for automatic updates, but IE hasn't always done that. As a result, the other five browsers should always be up-to-date, which means you shouldn't have to test your web pages in older versions of these browsers.

In the past, it's been difficult to test web pages in older versions of Internet Explorer because (1) you couldn't get them anymore and (2) you couldn't put more than one version on a single system even if you could get them. Fortunately, newer versions of IE have made it easier to test in older versions by providing tools that let you emulate those versions. To use these tools, you press F12 from the browser window to display the Developer Tools window. Then, you press Ctrl+8 or click the Emulation tab at the top of the window to display the emulation options. Finally, you select the version of IE you want to use from the Document Mode drop-down list.

If you're a student, you probably won't need to test your web pages in old versions of Internet Explorer. In fact, for the exercises in this book, you just need Chrome and your default browser (IE or Edge on a Windows system or Safari on a Mac). For production applications, though, you need to test your pages on all the browsers that your users are likely to use.

**The current browsers and their HTML5 ratings (perfect score is 555)**

Browser	Release	HTML5 Test Rating
Google Chrome	63	528
Opera	45	518
Mozilla Firefox	57	486
Edge	16	476
Apple Safari	11	452
Internet Explorer	11	312

**The website for these ratings**

<http://www.html5test.com>

**Guidelines for cross-browser compatibility**

- Test your web pages on all of the major browsers, including older versions of Internet Explorer that are still in common use.
- Use the features of HTML5 and CSS3 as you develop your web pages, but use the workarounds so those features will work in older browsers that are still in use.

**How to test your web pages in older versions of Internet Explorer**

- One of the problems in testing different versions of Internet Explorer is that you can't install more than one version on a system at the same time.
- The solution is to use the emulation tools that became available with Internet Explorer 9. These tools let you emulate several older versions of Internet Explorer.

**Description**

- As a web developer, you want your web pages to work on as many different web browsers as possible. This is referred to as *cross-browser compatibility*.
- When you develop a website for general use, you need to test it on the current release of all of the web browsers listed above plus older versions of Internet Explorer that are still in common use.
- Although Internet Explorer is still a commonly-used browser, it gives web developers the most problems because it is the least standard and because it hasn't always provided for automatic updates.
- In this book, you'll be alerted to cross-browser compatibility problems, and you'll learn to use the workarounds that you need for these browsers. Eventually, all browsers will support HTML5 and CSS3 so the workarounds won't be necessary.
- When you access [www.html5test.com](http://www.html5test.com), it will automatically rate the browser you're using.

Figure 1-15 The issue of cross-browser compatibility

**34**      *Section 1    The essential concepts and skills*

## **User accessibility**

The third major issue is *user accessibility*, or just *accessibility*. This refers to the qualities that make a website accessible to as many users as possible, especially disabled users.

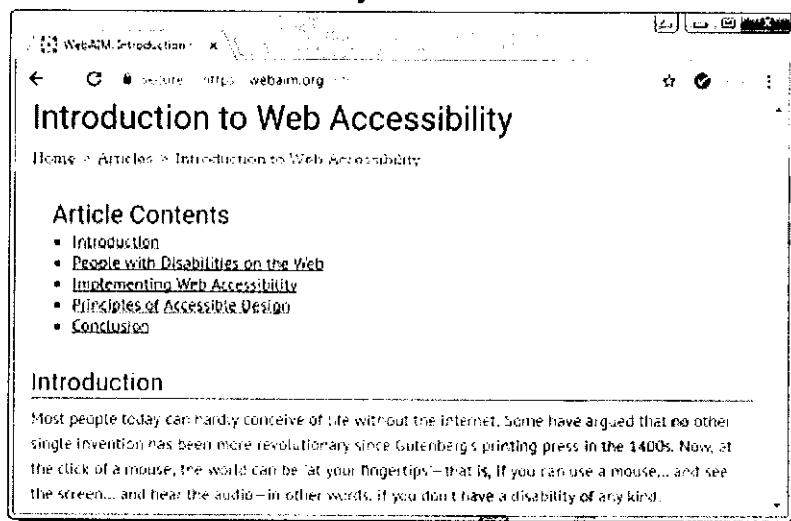
For instance, visually-impaired users may not be able to read text that's in images, so you need to provide other alternatives for them. Similarly, users with motor disabilities may not be able to use the mouse, so you need to make sure that all of the content and features of your website can be accessed through the keyboard.

To a large extent, this means that you should develop your websites so the content is still usable if images, CSS, and JavaScript are disabled. A side benefit of doing that is that your site will also be more accessible to search engines, which rely primarily on the text portions of your pages.

In this book, you will be given guidelines for providing accessibility as you learn the related HTML. However, there's a lot more to accessibility than that. As a result, we recommend that you learn more on your own by going to the sites that are identified in this figure.



## Articles on Web accessibility on the WebAIM site



## Accessibility laws that you should be aware of

- The Americans with Disabilities Act (ADA).
- Sections 504 and 508 of the federal Rehabilitation Act.
- Section 255 of the Telecommunications Act of 1996.

## Types of disabilities

- Visual
- Hearing
- Motor
- Cognitive

## Information sources

- The WebAIM website provides a good starting point for learning about accessibility at <http://www.webaim.org>.
- The World Wide Web Consortium (W3C) provides a full set of accessibility guidelines at <http://www.w3.org/TR/WCAG>.
- W3C also provides a specification called WAI-ARIA (Web Accessibility Initiative—Accessible Rich Internet Applications) that shows how to make rich internet applications more accessible to the disabled at <http://www.w3.org/TR/wai-aria>.

## Description

- **Accessibility** refers to the qualities that make a website accessible to users, especially disabled users.
- As you go through this book, you'll be given guidelines for coding the elements and attributes that provide accessibility. However, there's a lot more to accessibility than that.

Figure 1-16 The issue of user accessibility

## Search engine optimization

*Search engine optimization*, or *SEO*, refers to the goal of optimizing your website so your pages rank higher in search engines like Google, Bing, and Yahoo. In figure 1-17, for example, what causes the links that are shown to be returned by Google?

Since SEO is critical to the success of a website, this book presents SEO guidelines as you learn the related HTML. For instance, the title element in the head section of an HTML document is one of the most important elements for SEO. In this book, then, you'll learn how to code this element and all elements like it that affect SEO.

The HTML5 structural elements may also affect the way that search engines rank pages. In particular, these elements should help the search engines find pages that are more relevant to the search term. In this book, you'll learn how to code these elements in the best way for SEO.

Note, however, that SEO goes way beyond the way HTML pages are coded. So here again, you need to do some independent research. To start, you can search the Internet for information on SEO, but you might also want to buy a book or two on the subject.



## Responsive Web Design

The term *Responsive Web Design*, or *RWD*, was first coined by Ethan Marcotte in an article in the May 2010 issue of *A List Apart Magazine*. It refers to the theory and practice of creating websites that adapt gracefully to all viewing mediums, from desktop computers to mobile phones. The idea is to provide a website that is easy to read and navigate and that requires a minimum amount of resizing and scrolling.

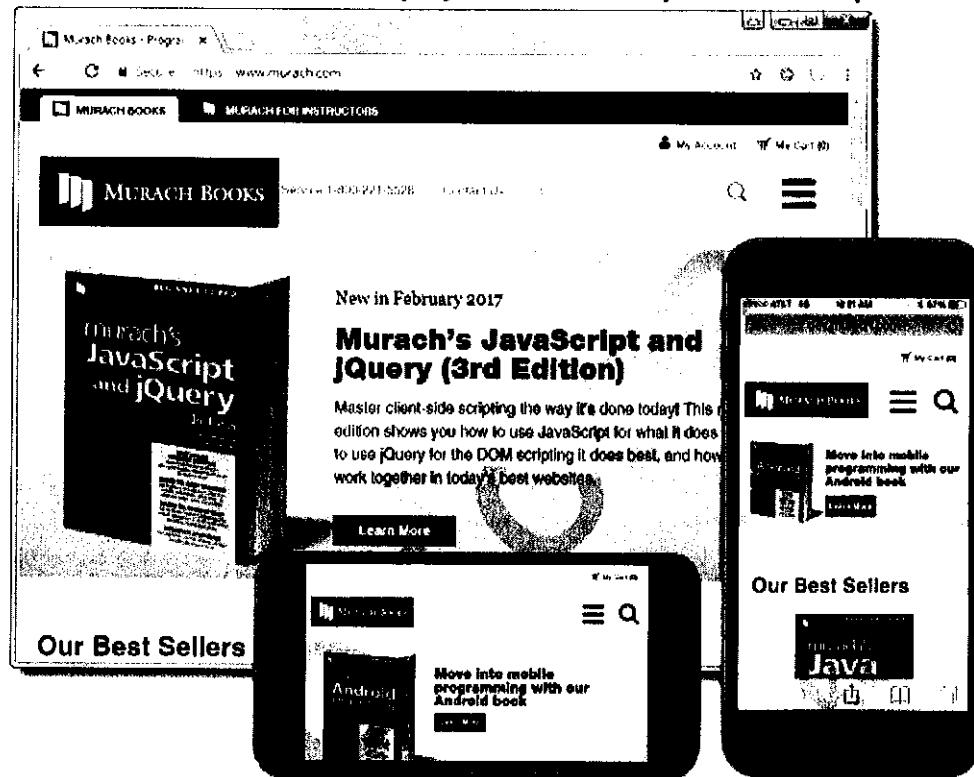
According to Marcotte, the layout of a website that's designed with RWD in mind should adapt to the viewing environment by using fluid, proportion-based grids, flexible images, and CSS3 media queries. His idea for RWD has now become the standard for web development. In fact, RWD has become a cost-effective alternative to the development of mobile applications for specific platforms like iOS and Android, called *native mobile applications*. In addition, several popular open-source web design frameworks such as Skeleton, Foundation, and Twitter Bootstrap incorporate RWD principles as a base for developing responsive websites.

So why should you use Responsive Web Design as you develop your websites? Most importantly, because statistics prove that mobile devices are being used more every day to access the Web. Some of these statistics are listed in figure 1-18. Consider, for example, that one third of consumers worldwide are using smartphones. Also consider that of those users, 80% do some shopping on their smartphones.

The increase in smartphone sales is due in part to the increase in the screen size of some phones. That includes iPhone Plus models with a 5.5 inch display, Galaxy Note models with displays up to 6.3 inches, and Nexus models with displays up to 5.96 inches. These phones are sometimes referred to as "phablets" because they're so large, and their sales have started cutting into the tablet market.

This figure also illustrates how a site that uses RWD adapts to the size and orientation of the screen. Here, you can see the home page of the Murach Books website in a desktop browser. Because it uses RWD, the layout of the page changes when it's displayed on a device with a smaller screen size. In this figure, for example, you can see how the page appears in an iPhone 8 in both portrait and landscape orientation. Note that the basic look-and-feel of the page remains the same across the different screen sizes. That way, users won't feel that they're visiting completely different sites from different devices. That's the beauty of Responsive Web Design!

### The Murach Books website displayed on a desktop and a smartphone



### Statistics that prove the need for Responsive Web Design

- Over half of the world's population has a mobile phone subscription.
- Over one third of consumers worldwide are using smartphones.
- 60% of all Internet access is being done on mobile devices.
- 80% of all smartphone users access retail content on their device at least once a month.
- 40% of users will choose a different search result if the first one isn't mobile friendly.
- As of April 2015, Google began prioritizing responsive websites in its search engine results.

### Description

- *Responsive Web Design* refers to websites that are designed to adapt gracefully to the screen size.
- Typically, the overall look-and-feel of a website will remain consistent from one screen size to the next.
- Media queries, scalable images, and flexible layouts are the backbone of Responsive Web Design.

Figure 1-18 The Issue of Responsive Web Design

## Perspective

Now that you know the concepts and terms that you need for developing websites with HTML and CSS, you're ready to learn how to develop a web page. So in the next chapter, you'll learn how to enter, edit, test, and validate a web page. After that, you'll be ready to learn all the details of HTML and CSS that you need for developing the pages of a website.

## Terms

World Wide Web	closing tag
Internet	attribute
client	CSS (Cascading Style Sheets)
web browser	style sheet
web server	external style sheet
network	style rule
local area network (LAN)	selector
intranet	declaration
wide area network (WAN)	XHTML (eXtensible HTML)
router	W3C (World Wide Web Consortium)
Internet service provider (ISP)	WHATWG (Web Hypertext
Internet exchange point (IXP)	Application Technology Working
static web page	Group)
HTTP request	text editor
HTTP (HyperText Transfer Protocol)	IDE (Integrated Development
HTML (HyperText Markup	Environment)
Language)	suite
HTTP response	deploy
render a web page	publish
dynamic web page	FTP program
application server	FTP (File Transfer Protocol)
database server	FTP plugin
round trip	ISP (Internet Service Provider)
web application	web hosting
scripting language	IP address
server-side processing	domain name
Apache web server	URL (Uniform Resource Locator)
IIS (Internet Information Services)	protocol
JavaScript	domain name
client-side processing	path
image swap	usability
image rollover	cross-browser compatibility
JavaScript engine	user accessibility
HTML document	SEO (Search Engine Optimization)
DOCTYPE declaration	RWD (Responsive Web Design)
HTML element	native mobile application
tag	
opening tag	

## Summary

- A web application consists of clients, a web server, and a network. *Clients* use *web browsers* to request web pages from the web server. The *web server* returns the requested pages.
- A *local area network (LAN)* connects computers that are near to each other. This is often called an *intranet*. In contrast, a *wide area network (WAN)* uses routers to connect two or more LANs. The *Internet* consists of many WANs.
- To request a web page, the web browser sends an *HTTP request* to the web server. Then, the web server retrieves the HTML for the requested page and sends it back to the browser in an *HTTP response*. Last, the browser *renders* the HTML into a web page.
- A *static web page* is a page that is the same each time it's retrieved. The file for a static page has .html or .htm as its extension, and its HTML doesn't change.
- The HTML for a *dynamic web page* is generated by a server-side program or script, so its HTML can change from one request to another.
- *JavaScript* is a *scripting language* that is run by the *JavaScript engine* of a web browser. It provides for *client-side processing*.
- *HTML (HyperText Markup Language)* is the language that defines the structure and contents of a web page. *CSS (Cascading Style Sheets)* are used to control how the web pages are formatted.
- To develop web pages, you can use a *text editor* like Brackets or an *Integrated Development Environment (IDE)* like NetBeans.
- To *deploy* (or *publish*) a website on the Internet, you need to transfer the folders and files for your site from your computer to a web server with Internet access. To do that, you use an *FTP program* that uses *File Transfer Protocol*.
- To view a web page on the Internet, you can enter the URL (*Uniform Resource Locator*) into a browser's address bar. A URL consists of the protocol, domain name, path, and filename.
- To view a web page that's on your own computer or server, you can use the features of your text editor or IDE. You can locate the file in your file explorer and double-click or right-click on it. Or, you can use your browser's File→Open or File→Open File command.
- To view the HTML for a web page, right-click on the page and select View Source or View Page Source. Then, to view the CSS for a page, you can click or Ctrl+click on its link in the source code or enter its URL in the address bar.
- Five critical issues for web development are *usability*, *cross-browser compatibility*, *user accessibility*, *search engine optimization (SEO)*, and *Responsive Web Design (RWD)*.

## Before you do the exercises for this book...

Before you do the exercises for this book, you should download and install the Chrome browser. You should also download and install the applications for this book. The procedures for installing the software and applications for this book are described in appendix A.

### Exercise 1-1 Visit some Internet websites

In this exercise, you'll visit some Internet websites and view the source code for those sites.

#### Visit the website in figure 1-2 with Chrome

1. Start Chrome.
2. Enter [www.newtonforkranch.com](http://www.newtonforkranch.com) into the address bar and press the Enter key. That should display the home page for this website. Here, JavaScript is used to rotate the images at the top of the page.
3. If you're using a Windows system, enter "newtonforkranch" into the address bar, hold down the Ctrl key, and press the Enter key. If this works, it will add www. and .com to your entry. This is a quick way to enter the URL for a .com address.
4. Use the technique in figure 1-13 to view the source code for the home page. Here, the five link elements with a rel attribute value of "stylesheet" identify the CSS files that do the formatting for the page. This is followed by five script elements that identify JavaScript files and two script elements that contain JavaScript code.
5. If you scroll through this code, it probably looks overwhelming, even though this site is relatively simple. By the time you complete this book, though, you should understand the HTML and CSS that it uses.
6. Click on the underlined value of the href attribute in the second link element. That should open the first CSS file for this page. This shows how easy it is to access the HTML and CSS code for many (but not all) sites.

#### Visit other websites

7. Go to [www.landsend.com](http://www.landsend.com), find a page like the one in figure 1-5, and experiment with the image swaps and rollovers. Those are done by JavaScript after all of the images are loaded with the page.
8. Use Chrome to visit other websites and view the source code for those sites. When you're through experimenting, go to the next exercise.



## Exercise 1-2 View the application for this chapter

In this exercise, you'll visit the book page that was used as an example in figures 1-6 and 1-7. You'll do that with your default browser, and if Chrome isn't your default browser, you'll also do it with Chrome.

### Open the page in your default browser and Chrome

1. Use your file explorer to locate this HTML file:  
`c:\murach\html5_css3_4\book_apps\ch01\javascript_jquery.html`  
Then, double-click on the file to open it in your default browser.
2. If Chrome isn't your default browser, use your file explorer to locate the file in step 1 again. This time, right-click on it and use the **Open With** command to open the file in Chrome.
3. In Chrome, right-click on the page and choose **View Page Source** to display the source code for this page.
4. In Chrome, click on `book.css` in the link element in the HTML code to display the CSS file for this page.

### If Edge is your default browser...

5. If Edge is your default browser, switch to it. Then, right-click on the page and select the **View Source** command. This should display the source code for the page in the **Debugger** tab of the **Developer Tools**.
6. Hold down the **Ctrl** key and click on `book.css` in the link element to see whether that displays the CSS file for the page. (In our testing, that hasn't always worked.)

## Exercise 1-3 View other applications and examples

1. Start Chrome if it isn't already open.
2. Use your file explorer to locate this file:  
`c:\murach\html5_css3_4\book_apps\ch07\town_hall\index.html`  
Then, drag the `index.html` file to the browser window to display the web page.
3. Click on the link for Scott Sampson to see that page. This is the website that's presented at the end of chapter 7, and this gives you some idea of what you'll be able to do when you complete that chapter. Note, however, that only the Scott Sampson link has been implemented for this website.
4. Open this file in the Chrome browser:  
`c:\murach\html5_css3_4\book_examples\ch16\06_animations\index.html`  
This is the example for figure 16-6 in chapter 16. Note that the figure number is in the folder name this time. This is true for all of the examples that require more than one file.

**44**      *Section 1    The essential concepts and skills*

5. Open this file in the Chrome browser.  
`c:\murach\html5_css3_4\book_examples\ch16\06_animations\index.html`  
This is the example for figure 16-6 in chapter 16. Note that the figure number is in the folder name this time. This is true for all of the examples that require more than one file.

**Exercise 1-4      Learn more about HTML5, accessibility, SEO, and RWD**

1. Go to [www.w3.org](http://www.w3.org). This is the website for the group that develops the HTML5 and CSS3 standards. It provides all sorts of useful information including HTML5 documentation.
2. Go to [www.whatwg.org](http://www.whatwg.org). This is the website for a community that is interested in evolving HTML and its related technologies. It also provides all sorts of useful information including HTML5 documentation.
3. Go to [www.html5test.com](http://www.html5test.com), and view the HTML5 rating for your browser. Then, review the other browser data that this site provides.
4. Go to [www.webaim.org](http://www.webaim.org). Then, review the information about accessibility that this site provides.
5. Use Google to search for “search engine optimization”. Then, click on the first google.com link for this subject to get more information about it. You may also want to download and print the PDF on SEO that Google offers.
6. Use Google to search for “responsive web design”. Then, click on one or more of the links to see what type of information is available.