Thanks for downloading this chapter from Murach’s PHP and MySQL (2nd Edition). We hope it will show you how easy it is to learn from any Murach book, with its paired-pages presentation, its “how-to” headings, its practical coding examples, and its clear, concise style.

To view the full table of contents for this book, you can go to our website. From there, you can read more about this book, you can find out about any additional downloads that are available, and you can review our other books on related topics.

Thanks for your interest in our books!
What developers said about the previous edition

“AFTER reading several other books, I still was not able to get my webpage to work the way I wanted. After reading Murach’s book, it works perfectly.”

Harold E. Luse, posted online

“Compared to the stack of other PHP/MySQL titles on the market, I found Murach to be one of the easiest and fastest books to teach the subject.”

Mike Riley, Dr. Dobb’s Online

“Murach’s unique paired pages format is ingeniously simple yet amazingly effective. The result is a very fast read.”

Posted at an online bookseller

“Along with the great organization, ease of learning, and thorough coverage of the technologies and programming practice, this book is simply a pleasure to read. Any developer/techie/geek should enjoy it.”

Jeff Salter, Developer, California

“I can’t count how many PHP books I have purchased over the years trying to learn the language. Murach’s was the first book that helped me grasp the concepts and got me onto actually scripting in PHP.”

Posted at an online bookseller

“I would thoroughly recommend this book to all budding PHP web developers out there, including ASP.NET developers who wish to diversify into PHP.”

Jason Ong, ASPNET World

“This is my first Murach book, and I love it so much I now want to go and buy their CSS and JavaScript books.”

Posted at an online bookseller
Get started fast with PHP and MySQL

The six chapters in this section are designed to get you off to a fast start with PHP and MySQL. First, in chapter 1, you’ll learn how web programming with PHP works and what tools you can use to develop PHP applications. Then, in chapter 2, you’ll learn how to use PHP to develop applications that don’t require a database.

The next two chapters get you going with MySQL databases. In chapter 3, you’ll learn how a relational database works and how to use phpMyAdmin to work with a MySQL database. In chapter 4, you’ll learn how to develop PHP applications that get data from and store data in a MySQL database.

The last two chapters in this section round out your skillset. First, chapter 5 shows you how to structure and organize your code by using the MVC pattern. Then, chapter 6 presents the techniques that you will need for testing and debugging applications.

When you complete this section, you’ll have all the skills that you need for coding, testing, and debugging significant database applications. Then, you can enhance those skills by reading the chapters in any of the other sections.
Introduction to web development with PHP

This chapter introduces you to the concepts and terms that you need for developing web applications with PHP. This chapter also shows you how to deploy, edit, and test a PHP application. When you finish this chapter, you’ll have all the background you need for learning how to write the code for PHP applications.

The architecture of a web application
How a client-server architecture works
How static web pages are processed
How dynamic web pages are processed
A survey of web application software
Highlights in the history of PHP
Highlights in the history of MySQL
The Product Discount application
The user interface
The HTML file
The CSS file
The PHP file
How to edit and test a PHP application
How to edit a PHP page with a text editor
How to start and stop Apache and MySQL on your own computer
How to deploy a PHP application
How to run a PHP application
How to test and debug a PHP page
How to view the source code for a web page
How to use NetBeans to develop a PHP application
How to work with PHP projects and files
How to edit and test a PHP application
How to import and configure a PHP project
Perspective
The architecture of a web application

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 PHP programming, you need to have a basic understanding of how these components interact and where PHP fits into this architecture.

How a client-server architecture works

Web applications use a client-server architecture. This architecture consists of servers that share resources with clients over a network. Figure 1-1 shows the components of a simple client-server architecture.

A server can share resources such as files, printers, websites, databases, and e-mail. A web server is a server that shares websites, and a web browser is the client software used to access the web server.

A network is a communication system that allows clients and servers to communicate. A Network Interface Card (NIC) connects the computer to the network. This connection can either be wired or wireless. Ethernet is a common type of wired network. Wi-Fi is a common type of wireless network.

The network is responsible for getting information from one computer to another. This process is called routing. A router is a device that is connected to two or more networks. When information comes in from one network, the router determines which network is closest to the destination and sends the information out on that network.

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.

A Wide Area Network (WAN) consists of multiple LANs that have been connected together over long distances using routers. 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 together at large routers called Internet Exchange Points (IXP). This allows anyone connected to the Internet to exchange information with anyone else.

This figure shows an example of data crossing the Internet. In the diagram, data is being sent from the client in the top left to the server in the bottom right. Here, the data leaves the client’s LAN and enters the WAN owned by the client’s ISP. Next, the data is routed through an IXP 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 architecture of a web application

A server makes resources available to other computers called clients over a network. A server can share files, printers, websites, databases, or e-mail.

A network uses routers to get information from the sender to its destination.

A Local Area Network (LAN) directly connects computers that are near each other.

A Wide Area Network (WAN) uses routers to connect computers that are far from each other.

The Internet consists of many WANs that have been connected together at Internet Exchange Points (IXP). A list of IXPs can be found at http://en.wikipedia.org/wiki/IXP.

An Internet Service Provider (ISP) owns a WAN and leases access to this network. It connects its WAN to the rest of the Internet at one or more IXPs.

Figure 1-1 How a client-server architecture works
How static web pages are processed

A *static web page* is a web page that only changes when the web developer changes it. It is a plain text file that contains all the content to be displayed in the web browser. This web page is sent directly from the web server to the web browser when the browser requests it.

Figure 1-2 shows how a web server processes a request for a static web page. The process begins when a user requests a web page in a web browser. The user can either type in 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. In this figure, you can see the content of a simple HTTP request.

When the web server receives the HTTP request, it retrieves the requested web page from the disk drive and sends it back to the browser as an *HTTP response*. This response includes the *HTML (HyperText Markup Language)* for displaying the requested page. In this figure, you can see the HTTP response for a simple web page, which includes the HTML for the page.

When the browser receives the HTTP response, it uses the HTML to format the page and displays the page in the web browser. Then, the user can view the content. If the user requests another page, either by clicking a link or entering another web address in the browser’s address bar, the process begins again.

Incidentally, this process depends not only on the HTTP protocol but also on the *Transmission Control Protocol/Internet Protocol (TCP/IP)* suite of protocols. The protocols in TCP/IP let two computers communicate over the network.
How a web server processes a static web page

A simple HTTP request

GET / HTTP/1.1
Host: www.example.com

A simple HTTP response

HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 136
Server: Apache/2.2.3

<html>
<head>
<title>Example Web Page</title>
</head>
<body>
<p>This is a sample web page</p>
</body>
</html>

Two protocols that web applications depend upon

- **HyperText Transfer Protocol (HTTP)** is the protocol that web browsers and web servers use to communicate. It sets the specifications for HTTP requests and responses.

- **Transmission Control Protocol/Internet Protocol (TCP/IP)** is a suite of protocols that let two computers communicate over a network.

Description

- **HyperText Markup Language (HTML)** is the language used to design the web pages of an application.

- A **static web page** is an HTML document that’s stored on the web server and doesn’t change in response to user input. Static web pages have a filename with an extension of .htm or .html.

- 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 web page and sends it back to the browser in an **HTTP response**. This response includes the HTML document that’s stored in the file that was requested.
How dynamic web pages are processed

A dynamic web page is a page that’s created by a program or script that’s running on a server. This means that the page can be changed each time it is viewed.

The changes in the page can come from processing the form data that the user submits or by displaying data that’s retrieved from a database server. A database server stores information that’s organized in tables, and this information can be quickly retrieved by a database query.

Dynamic web pages enable web developers to create interactive web applications. As a result, users can purchase goods and services, search the web for information, and communicate with other users through forums, blogs, and social networking sites. Sites like these would be difficult or impossible to create without database-driven, dynamic web pages.

Figure 1-3 shows how a web server processes a dynamic web page with PHP. The process begins when the user requests a page in a web browser. The user can either enter 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. If the user submits form data, that data will be included in the HTTP request.

When the web server receives the HTTP request, it looks up the file extension of the requested web page to determine which server or program should process the request. For a PHP page, the web server forwards the request to the PHP interpreter, which is running on the web server.

The PHP interpreter gets the appropriate PHP 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 a web page as its output. The script may also 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 PHP interpreter passes the dynamically generated web page back to the web server. Then, the web server sends the page back to the browser in an HTTP response that includes the HTML for the page.

When the web browser receives the HTTP response, it formats and displays the web page. This is called rendering a page. Note, however, that the web browser has no way to tell whether the HTML in the HTTP response was from a static page or a dynamic page because all it receives is HTML.

When the page is displayed in the browser, 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.
How a web server processes a dynamic web page with PHP

**Description**

- A *dynamic web page* is a web page that’s generated by a server-side program or script. Often, the web page changes according to the information that’s submitted by the web browser to the server.

- When a web server receives a request for a dynamic web page, it uses the extension of the requested file to determine which server or program should process the request. If the extension is php, the web server calls the PHP interpreter to process the request and the data that’s submitted with the request.

- The PHP page can use the data that it gets from the web browser to access the appropriate data from a *database server*. The application can also store the data that it gets from the web browser in the database server.

- When the PHP interpreter finishes processing the PHP page, it generates an HTML page and returns it to the web server. The web server then returns the page to the web browser.

- The browser can’t tell whether the HTML that is returned to it was retrieved from a static web page or generated dynamically by the PHP interpreter. Either way, the browser simply displays *(renders)* the HTML that it receives.

- 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*.

---

*Figure 1-3  How dynamic web pages are processed*
A survey of web application software

Figure 1-4 summarizes the software for the four components of a web application: web browsers, web servers, server-side languages, and database servers. The first web browser was developed in 1991 by Tim Berners-Lee at the European Council for Nuclear Research (CERN) in Geneva, Switzerland. Since then, dozens of web browsers have been developed.

Google Chrome was released in 2008, grew quickly in popularity, and is now one of the most widely used web browsers. It is available for the Windows, Mac OS X, and Linux operating systems.

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. It is currently available only for Windows.

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.

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

The Apache web server, which was developed by the Apache Software Foundation, is the most widely used web server. It is an open-source software project that’s available for free. Although there are Apache versions for most modern operating systems, it is typically used on a Linux server.

The other widely used web server is Microsoft’s Internet Information Services (IIS). It is included as part of the Windows Server operating system.

Of the server-side languages, PHP, Perl, and Python can all be directly installed on an Apache or IIS web server. These languages are referred to as scripting languages.

In contrast, JSP and ASP.NET require an application server in addition to the web server. For example, JSP applications typically run on an application server such as Tomcat or Glassfish. And ASP.NET applications typically run on an application server for IIS. For more information about JSP and ASP.NET, please see our books on these subjects.

PHP is commonly used with the MySQL database server. This database server is an open-source database server that runs on all major operating systems. However, PHP can also be used with other database servers. Three of the most widely used database servers are Oracle, IBM’s DB2, and Microsoft’s SQL Server.
Web browsers

<table>
<thead>
<tr>
<th>Browser</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome</td>
<td>Published by Google. It is available for all major operating systems.</td>
</tr>
<tr>
<td>Firefox</td>
<td>Published by the Mozilla Corporation. It is available for all major operating systems.</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>Published by Microsoft. It is available for Windows.</td>
</tr>
<tr>
<td>Safari</td>
<td>Published by Apple. It is available for OS X.</td>
</tr>
<tr>
<td>Opera</td>
<td>Published by Opera Software. It is available for Windows and OS X.</td>
</tr>
</tbody>
</table>

Web servers

<table>
<thead>
<tr>
<th>Server</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache</td>
<td>An open-source web server that can run on any major operating system. It supports many server-side scripting languages and can interact with many different database servers. The most common configuration is known as LAMP, which consists of Linux, Apache, MySQL, and PHP.</td>
</tr>
<tr>
<td>IIS</td>
<td>Microsoft’s web server that only runs on Windows operating systems. It primarily supports ASP.NET web development and MS SQL Server.</td>
</tr>
</tbody>
</table>

Server-side languages

<table>
<thead>
<tr>
<th>Language</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHP</td>
<td>(PHP: Hypertext Processor) Typically used with the Apache web server but also available for IIS. It uses the .php file extension.</td>
</tr>
<tr>
<td>JSP</td>
<td>(JavaServer Pages) Requires an application server such as the Tomcat server that’s available from the Apache Software Foundation. JSP pages use the .jsp file extension and typically work with servlets that are written in Java.</td>
</tr>
<tr>
<td>ASP.NET</td>
<td>(Active Server Pages) Used with the Microsoft IIS web server. ASP.NET pages use the .aspx file extension and typically work with server-side code that’s written in C# or Visual Basic.</td>
</tr>
<tr>
<td>Perl</td>
<td>Originally developed for use at the UNIX command line to manipulate text, it was later used to build web applications. It uses the .pl file extension.</td>
</tr>
<tr>
<td>Python</td>
<td>Used to develop many types of applications in addition to web applications. It is typically used with the Apache web server. It uses the .py file extension.</td>
</tr>
</tbody>
</table>

Database servers

<table>
<thead>
<tr>
<th>Server</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>An open-source database that is available for all major operating systems.</td>
</tr>
<tr>
<td>Oracle</td>
<td>Oracle’s database server that is available for all major operating systems.</td>
</tr>
<tr>
<td>DB2</td>
<td>IBM’s database server that is available for all major operating systems.</td>
</tr>
<tr>
<td>MS SQL Server</td>
<td>Microsoft’s database server that is only available for the Windows operating system.</td>
</tr>
</tbody>
</table>
Highlights in the history of PHP

Figure 1-5 summarizes some of the highlights in the history of PHP. To start, PHP 2 was released in 1995 with a name of Personal Home Page. Then, in 1998, PHP 3 was released. With this release, the name of PHP was changed to PHP: Hypertext Processor.

In 2000, PHP 4 introduced the Zend Engine. This improved PHP’s performance and its popularity too. Finally, in 2004, PHP 5 introduced the Zend Engine II. In addition, PHP 5 introduced improved support for object-oriented programming, the PHP Data Objects extension, and other new features.

As this book goes to press, PHP 5.5 is the current version of PHP. As a result, that’s the version of PHP that’s used throughout this book. Most of the code in this book also works with other versions of PHP. However, if you’re using a different version of PHP, you may notice some differences.

Highlights in the history of MySQL

Figure 1-5 also summarizes some of the highlights in the history of MySQL. To start, MySQL 3.23 was released in 1995. This version of MySQL became widely used by many websites. Then, in 2003, MySQL 4.0 introduced support for unions. In 2004, MySQL 4.1 introduced support for subqueries and prepared statements. In 2005, MySQL 5.0 introduced support for stored procedures, triggers, views, and transactions. In 2008, MySQL 5.1 introduced support for row-based replication and server log tables. In 2010, MySQL 5.5 changed the default storage engine to provide support for referential integrity by default. Finally, in 2011, MySQL 5.6 introduced new features for working with big data.

MySQL is owned and sponsored by a for-profit firm named MySQL AB. However, in 2008, Sun Microsystems acquired MySQL AB. Then, in 2009, Oracle Corporation acquired Sun Microsystems. As a result, Oracle now has a big say in the future of the MySQL database, which competes with the Oracle database.

Fearing that Oracle will stop developing MySQL, many of the original developers of MySQL left MySQL AB shortly after its acquisition by Oracle and began working on different forks of the open-source code. One of the most popular of these forks is MariaDB. As a result, even if Oracle stops or slows development of MySQL, developers can still consider using MariaDB or other forks of MySQL.
### Highlights in the history of PHP

<table>
<thead>
<tr>
<th>Version</th>
<th>Year</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1995</td>
<td>This version was named Personal Home Page.</td>
</tr>
<tr>
<td>3</td>
<td>1998</td>
<td>This version was renamed PHP: Hypertext Processor.</td>
</tr>
<tr>
<td>4</td>
<td>2000</td>
<td>This version introduced the Zend Engine.</td>
</tr>
<tr>
<td>5</td>
<td>2004</td>
<td>This version introduced the Zend Engine II. It includes improved support for object-oriented programming, the PHP Data Objects extension, and other new features.</td>
</tr>
<tr>
<td>6</td>
<td>--</td>
<td>PHP6 has been in development since October 2006 but as of this writing has not yet been released.</td>
</tr>
</tbody>
</table>

### Highlights in the history of MySQL

<table>
<thead>
<tr>
<th>Version</th>
<th>Year</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.23</td>
<td>1995</td>
<td>The original version of MySQL is released.</td>
</tr>
<tr>
<td>4.0</td>
<td>2003</td>
<td>This version introduced support for unions.</td>
</tr>
<tr>
<td>4.1</td>
<td>2004</td>
<td>This version introduced support for subqueries and prepared statements.</td>
</tr>
<tr>
<td>5.0</td>
<td>2005</td>
<td>This version introduced support for stored procedures, triggers, views, and transactions.</td>
</tr>
<tr>
<td>5.1</td>
<td>2008</td>
<td>This version introduced support for row-based replication and server log tables.</td>
</tr>
<tr>
<td>5.5</td>
<td>2010</td>
<td>This version changed the default storage engine to provide support for referential integrity by default.</td>
</tr>
<tr>
<td>5.6</td>
<td>2011</td>
<td>This version introduced support for working with big data.</td>
</tr>
</tbody>
</table>

### Description

- MySQL is owned and sponsored by MySQL AB, a for-profit firm.
- In 2008, Sun Microsystems acquired MySQL AB.
- In 2009, Oracle Corporation acquired Sun Microsystems.
- In 2009, many of the original developers of MySQL left MySQL AB and begin working on different forks of the open-source code. One of the most popular of these forks is MariaDB.
- In 2012, several tech writers reported that Oracle was holding back MySQL Server test cases and no longer synchronizing their changes with the public source repository. Also in 2012, several Linux distributions, as well as Wikipedia and Google, began to replace MySQL with MariaDB.
The Product Discount application

To give a better idea of how a PHP application works, this topic presents a simple application. This application consists of three files: an HTML file, a CSS file, and a PHP file.

This simple application shows how PHP can get data that’s submitted by a user, process that data, and return the processed data to the user. It also shows how HTML, CSS, and PHP work together in a web application. In chapter 2, you’ll learn how to write the PHP code for an application like this.

The user interface

Figure 1-6 shows the user interface for this application. On the first page of the application, the user enters a product description, price, and discount percent, and then clicks the Calculate Discount button. This submits the data to the web server as part of an HTTP request.

When the web server receives the request, it passes the request to the PHP interpreter, which processes the request. It then generates the HTML for the second page of this application and sends it to the web server, which passes the page back to the browser.
The first page (index.html)

The second page (the response)

Figure 1-6  The user interface for the Product Discount application
The HTML file

This book assumes that you already have a basic understanding of HTML because it is essential to the use of PHP. If you don’t have that background, we recommend that you get Murach’s *HTML5 and CSS3*. If you read just the first six chapters and chapter 10, you’ll have all the skills you need for this book.

Once you’re familiar with HTML, you shouldn’t have any problem understanding the code in figure 1-7, which is the HTML for the first page of this application. It consists of a form that contains three labels, three text boxes, and a submit button. When the button is clicked, an HTTP request for the `display_discount.php` file is created. Because the form specifies the POST method, the request also includes the data that the user entered into the text boxes.

This HTML code is typical of the HTML that’s used in the applications for this book, which is generally quite simple. So if you understand this code, you probably don’t need to enhance your HTML skills. Also, if you’re working with a team of developers and a web designer is responsible for the HTML of your website, you may not need to completely understand this code. Here, though, are a few highlights that you should be aware of.

First, the `<link>` tag loads the CSS file named `main.css` that’s shown in the next figure. That file is used to format both of the pages in this application.

Second, the first three `<label>` and `<input>` tags create the three labels and text boxes that let the user enter data. Then, the last `<label>` and `<input>` tag creates the button that submits the data to the server when the user clicks on it. The last `<label>` tag is blank, but is necessary for formatting. Later, you’ll see that PHP refers to the data in the text boxes by their name attributes.

Third, the `<main>` and `<div>` tags are used by the CSS that formats the web page. The id attributes for the div tags are “`data`”, and “`buttons`” (even though this form only has one button). As you will see in the next figure, these tags and ids are used by the CSS.

Fourth, when creating PHP applications, the best practice is to separate the HTML, CSS, and PHP code as shown by this application. This reduces code duplication and makes the application easier to maintain. As a result, this technique is commonly used in professional web applications, and it is used throughout this book.
The HTML file (index.html)

```html
<!DOCTYPE html>
<html>
<head>
    <title>Product Discount Calculator</title>
    <link rel="stylesheet" type="text/css" href="main.css">
</head>
<body>
<main>
    <h1>Product Discount Calculator</h1>
    <form action="display_discount.php" method="post">
        <div id="data">
            <label>Product Description:</label>
            <input type="text" name="product_description"><br>
            <label>List Price:</label>
            <input type="text" name="list_price"><br>
            <label>Discount Percent:</label>
            <input type="text" name="discount_percent"><span>%</span><br>
        </div>
        <div id="buttons">
            <label>&nbsp;</label>
            <input type="submit" value="Calculate Discount"><br>
        </div>
    </form>
</main>
</body>
</html>
```

Figure 1-7 The HTML file for the Product Discount application
The CSS file

Figure 1-8 shows the CSS (Cascading Style Sheets) file for the Product Discount application. Here again, this book assumes that you have a basic understanding of CSS. If you don’t, please get a copy of Murach’s HTML5 and CSS3, and read chapters 4 through 6.

Once you’re familiar with CSS, you won’t have any trouble understanding the code in this figure. Here, eight rule sets are applied to the HTML elements for both pages of the application. The first rule set applies to the <body> element and the second applies to the <main> element. The third and fourth apply to the <h1> and <label> elements. The fifth and sixth apply to the <input> and <span> elements that are coded in the “data” <div> tag. The seventh applies to the <input> elements that are coded in the “buttons” <div> tag. And the eighth applies to the <br> tag.

Because CSS is incidental to PHP applications, this is the only time that we present the CSS for an application in this book. Note, however, that the CSS for every application is available with the download for this book.

Like the HTML for this application, this CSS is typical of the CSS used for the other applications in this book, which is quite simple. Also, if you are working with a team of developers and a web designer is responsible for the CSS for your website, you may not need to use CSS at all. In that case, just realize that the CSS does the formatting for the pages.
The CSS file (main.css)

```css
body {
    font-family: Arial, Helvetica, sans-serif;
    margin: 1em;
    padding: 0;
}

main {
    width: 450px;
    margin: 0 auto;
    padding: 1.5em;
    background: white;
    border: 2px solid navy;
}

h1 {
    color: navy;
}

label {
    width: 10em;
    padding-right: 1em;
    float: left;
}

#data input {
    float: left;
    width: 15em;
    margin-bottom: .5em;
}

#data span {
    padding-left: .25em;
}

#buttons input {
    float: left;
    margin-bottom: .5em;
}

br {
    clear: left;
}
```

Figure 1-8 The CSS file for the Product Discount application
The PHP file

Figure 1-9 shows the PHP file for the Product Discount application. In the next chapter, you’re going to learn the details for writing PHP code like this. But if you have some programming experience, the explanation that follows should help you understand what’s happening right now.

To start, the PHP file gets the data that’s stored in the text boxes of the HTML file. To do that, it uses the $POST array to retrieve the data from the text boxes. This works because the HTML file specifies the POST method for the form that is submitted. To retrieve a value from the $POST array, this code specifies the name attributes that were used in the <input> tags for the text boxes. For example, a value of “product_description” gets the data that the user entered into the Product Description text box.

After the PHP gets the data for each text box, it stores it in a variable with a name that starts with a dollar ($) sign. For instance, the product description is stored in a variable named $product_description.

After it stores the data in variables, the PHP code calculates the discount amount and discount price by operating on the data in the variables. To do that, the first statement uses the multiplication operator (*) to calculate the discount amount and store it in the $discount variable. Then, the second statement uses the subtraction operator (-) to calculate the discount price and store it in a variable named $discount_price.

These calculations are possible because PHP automatically converts string values to numeric data types if the string value contains numeric characters only. For example, if the user enters 399.95 for the list price, PHP stores this variable as a number, not as a string. As a result, you can perform arithmetic calculations on this variable.

After performing the calculations, the PHP code formats four of the values that the page needs to display. To do that, it uses the number_format function to format three of the numeric variables so they display two decimal places, and it formats the fourth numeric variable so it displays one decimal place. It also uses the concatenation operator (.) to add a dollar sign ($) to three of the variables and a percent sign (%) to one of the variables. This automatically converts the numeric values to string values.

After formatting the numeric values, the PHP code escapes the string for the product description. To do that, it uses the htmlspecialchars function to convert any special characters that are used by HTML tags to HTML entities. This prevents a type of security vulnerability known as a JavaScript injection attack. In general, it’s a good practice to escape any user input that hasn’t been thoroughly validated.

The rest of the PHP file looks much like a normal HTML page. However, it uses five PHP echo statements to display the data that’s stored in the PHP variables within the five <span> tags on the page.
The PHP file (display_discount.php)

```php
// get the data from the form
$product_description = $_POST['product_description'];
$list_price = $_POST['list_price'];
$discount_percent = $_POST['discount_percent'];

// calculate the discount
$discount = $list_price * $discount_percent * .01;
$discount_price = $list_price - $discount;

// apply currency formatting to the dollar and percent amounts
$list_priceFormatted = "$".number_format($list_price, 2);
$discount_percentFormatted = number_format($discount_percent, 1)."%";
$discountFormatted = "$".number_format($discount, 2);
$discount_priceFormatted = "$".number_format($discount_price, 2);

// escape the unformatted input
$product_description_escaped = htmlspecialchars($product_description);

<!DOCTYPE html>
<html>
<head>
<title>Product Discount Calculator</title>
<link rel="stylesheet" type="text/css" href="main.css">
</head>
<body>
<main>
<h1>Product Discount Calculator</h1>
<label>Product Description:</label>
<span><?php echo $product_description_escaped; ?></span>
<br>
<label>List Price:</label>
<span><?php echo $list_price_formatted; ?></span>
<br>
<label>Standard Discount:</label>
<span><?php echo $discount_percent_formatted; ?></span>
<br>
<label>Discount Amount:</label>
<span><?php echo $discount_formatted; ?></span>
<br>
<label>Discount Price:</label>
<span><?php echo $discount_price_formatted; ?></span>
<br>
</main>
</body>
</html>
```

Figure 1-9 The PHP file for the Product Discount application
How to edit and test a PHP application

Now that you know the components that make up a PHP application, you’re ready to learn how to edit and test a PHP application.

How to edit a PHP page with a text editor

Although you can use any text editor to enter and edit HTML, CSS, and PHP files, using a text editor that’s designed for working with HTML, CSS, and PHP can speed development time and help reduce coding errors. Some features to look for in a text editor are syntax highlighting and auto-completion.

If you’re using Windows, we recommend that you use Notepad++ as your editor because it provides these features. Although there are many other free and commercial text editors, Notepad++ provides all of the features that you’ll need for the purposes of this book.

If you’re using Mac OS X, we recommend that you use TextWrangler as your editor. This is a free editor that provides syntax highlighting and FTP access, although it doesn’t provide auto-completion. Here again, though, you have other choices like commercial editors that do provide auto-completion such as BBEdit (the commercial version of TextWrangler).

To illustrate the use of a text editor for HTML, CSS, and PHP files, figure 1-10 shows Notepad++ as it’s being used to edit a PHP file. This editor provides tabs so you can edit more than one file at the same time. In addition, it color codes the syntax of the statements to reflect different coding elements. If you experiment with Notepad++, you’ll find that it has many other capabilities that this brief summary doesn’t present.

Before you start using Notepad++, you may want to take the time to change the style for comments because the default style is too small on some systems. This skill is summarized in this figure.

Then, when you start a new file, you should let Notepad++ know what language you’re working with. To do that, you can either save the file with the .html, .css, or .php extension, or you can use the Language menu to select the language. Once you do that, this editor uses the appropriate color coding.
Chapter 1  Introduction to web development with PHP

Notepad++ with three tabs open

How to open, save, close, and start files

• To open a file, use the Open button in the toolbar. Or, right-click on the file in the Windows Explorer and select the Edit with Notepad++ command.

• To save the current file, use the Save button in the toolbar or press Ctrl+S. Or, to save all open files, use the Save All button.

• To close the current file, use the Close button in the toolbar. Or, to close all open files, use the Close All button.

• To start a new file in a new tab, use the New button in the toolbar.

How to change the style for comments

• Start the Settings→Style Configurator command, and select php in the language list and COMMENT in the style list. Then, change the font name and font size in the drop-down lists to the blank entries at the top of the lists.

• Repeat this for COMMENTLINE for the PHP language, for COMMENT for the HTML language, and for COMMENT for the CSS language.

Description

• You can use many different text editors for editing HTML, CSS, and PHP code. For Windows, we recommend Notepad++. For Mac OS X, we recommend TextWrangler.

Figure 1-10  How to edit a PHP file with a text editor
How to start and stop Apache and MySQL on your own computer

As you have already learned, PHP applications run on an Apache web server that has a PHP interpreter. As a result, to run the applications for this book on your own computer or a local server, you need to install Apache and PHP.

To do that, we recommend that you download and install the XAMPP software package. This is a single download that includes Apache, MySQL, and PHP. It is free, open-source, cross-platform, and easy to install. For detailed instructions on how to install it, please refer to the appendixes for this book.

Once you’ve installed the XAMPP package, you need to make sure that both Apache and MySQL are running before you test your applications. To do that, you can click on the Start buttons in the XAMPP Control Panel as shown in figure 1-11. If you want to start the Apache and MySQL servers each time your computer starts, you can use the check box to its left to install it as a service. For more information about that, please refer to the appendixes for this book. In this figure, for example, the MySQL server is installed as a service. As a result, it will start every time this computer starts.

Please note that the XAMPP Control Panel shown in this figure is for a Windows system. If you’re using a different operating system such as Mac OS X, the XAMPP Control Panel will look different than the one shown in this figure, and it will work a little differently too. However, it will still include buttons that allow you to start and stop the Apache and MySQL servers.
The XAMPP Control Panel

How to start the XAMPP control panel

- On a Windows system, select the XAMPP Control Panel item from the Windows Start menu or double-click on the XAMPP icon on your desktop.
- On a Mac OS X system, open the Applications folder, open the XAMPP folder, and double-click the manager-osx application.
- On a Linux system, open a terminal, use the cd command to change to the opt/lampp directory, and use the sudo command to execute the manager-linux-x64.run or manager-linux-x86.run file.

How to start and stop Apache or MySQL

- Click on its Start or Stop button.
- To start Apache or MySQL automatically when your computer starts, please refer to the appendixes.

Description

- XAMPP is a free, open-source web server package that includes Apache, MySQL, and an interpreter for PHP.
- XAMPP is available for Windows, Linux, and Mac OS X systems (the X in XAMPP stands for cross-platform).
- To install XAMPP, please refer to the appendixes.
How to deploy a PHP application

When you *deploy* an application, you make it accessible from a browser. To do that on your own computer or on a local server when running Windows, you copy all of the directories and files for the application to the `\xampp\htdocs` directory for the Apache server because that’s where Apache looks for applications.

This is illustrated by the first directory structure of figure 1-12. Here, the top-level directory for a guitar store application has been copied to the `htdocs` directory. In this example, only four subdirectories are shown, including a directory for CSS files and a directory for storing the images that are required by the application. In practice, though, a large application is likely to contain many subdirectories at several different levels.

Note here, that the `htdocs` directory is called the *document root directory*, while the `guitar_store` directory is called the *application root directory*. This just means that `htdocs` is the root directory for all applications, while `guitar_store` is the root directory for one application.

For this book, you’ll be working with many small applications that are designed to help you learn. If you install these applications as described in the appendix, they will be in two directories that are subordinate to the `htdocs` directory: `book_apps` and `ex_starts`. This is illustrated by the second directory structure in this figure. For instance, `ch01_product_discount` in the `book_apps` directory is the directory for the application in this chapter. And `ch02_ex1` in the `ex_starts` directory is the directory for the application for the first exercise in chapter 2.

To deploy an application on an Internet server, you copy the application directories and files from the local server to your root directory on the Internet server. To do that, you can use an *FTP program*, which uses *File Transfer Protocol* to upload the files from the local server to the Internet server. To make this work, of course, the Internet server must be running Apache and a PHP interpreter.

In practice, PHP applications are usually developed and tested on a local server before they are uploaded to an Internet server. To make that manageable, the directory structure on the local server is exactly the same as (it “mirrors”) the directory structures on the Internet server. That also makes it easier to maintain and enhance a web application later on.
The file structure for a PHP application on a local server

```
<table>
<thead>
<tr>
<th>xampp</th>
</tr>
</thead>
<tbody>
<tr>
<td>htdocs (the document root directory)</td>
</tr>
<tr>
<td>guitar_store (the application root directory)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
```

The file structure for the book applications and exercise starts

```
<table>
<thead>
<tr>
<th>xampp</th>
</tr>
</thead>
<tbody>
<tr>
<td>htdocs</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
```

How to deploy a PHP application on a local server
- Copy all of the directories and files for an application to the \xampp\htdocs directory on the server. This is where Apache looks for PHP applications.

How to deploy the downloadable applications on a local server
- Copy the book_apps and ex_starts directories and all their contents to the \xampp\htdocs directory on the server.

How to deploy an application on an Internet server
- Use an FTP (File Transfer Protocol) program to upload the tested directories and files to the htdocs directory of the Apache web server.

Description
- To deploy a PHP application so you can run it, you need to store the directories and files for the application in the htdocs directory of an Apache server that’s on your own computer, a local server, or an Internet server.
- The files for a PHP application usually include HTML, CSS, PHP, and image files. These files are stored in appropriate directories.
- Usually, the directories and files on the local server mirror the directories and files that get uploaded to the Internet server.
How to run a PHP application

To access a web page on the Internet, you can enter a Uniform Resource Locator (URL) into the address bar of your browser. This URL consists of the four components shown in figure 1-13.

In most cases, the protocol is HTTP. If you omit the protocol, the browser uses HTTP as the default.

The domain name 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, most browsers let you omit “www.” from the domain name.

The third component is the path where the file resides on the server. The path lists the directories that contain the file. A forward slash is used to separate names in the path and to represent the server’s top-level directory at the beginning of the path. In the first example in this figure, the path is “/books/”.

The last component is the filename. In this example, the file is named index.htm. If you omit the filename, though, the web server will search for a default file in the path you specify. By default, the Apache web server provides for default filenames of index.htm, index.html, and index.php.

If you want to view a page that’s running on your computer, you can use the localhost keyword as the domain name. In this figure, for example, the second set of examples uses the localhost keyword to access the Apache server that’s running on the same computer as the web browser.

If you omit the filename in a URL and there is no default file in the directory that you specify, Apache displays a list of the directories and files within that directory. This is illustrated by the last example in this figure. Here, the book_apps directory doesn’t contain a default file. Instead, it contains one directory for each of the applications presented in this book. As a result, Apache displays an index page that lists these directories in the browser.

This is a good way to access the book applications for this book. Then, you can start any application in the book_apps directory by clicking on its link. If you also get an index of the applications in the ex_starts directory and if you bookmark both of these indexes, you’ll always have an easy way to run a book application or an exercise.

Incidentally, when you create the directories and filenames for your applications, they should only contain lowercase letters, numbers, the period, and the underscore character because the names in the path may be case sensitive on some servers. Then, if a URL contains a directory named “Images”, but the directory on the server is actually named “images”, the web server will report that it cannot find the file.
The components of an HTTP URL

http://www.murach.com/books/index.htm

protocol   domain name   path     filename

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, one of the default filenames for the Apache web server
  will be used: index.htm, index.html, or index.php.
- If you omit the filename and there is no default file, Apache will display an index of
  the files and directories in the path.

URLs for requesting pages from an Internet web server

A request for a specific page

http://www.murach.com/books/phps/index.htm

A request for the default (home) page of a website

http://www.murach.com/

URLs for requesting applications from a local web server

A request for the default page in an application directory

http://localhost/book_apps/ch01_product_discount/

A request for a directory that doesn’t contain a default page

http://localhost/book_apps/

An index of the applications in the book_apps directory

![Index of /book_apps](image)

Description

- To request a web page or website on the Internet, enter the URL for the page or site
  in the address bar of your web browser.
- To request a web page or application on a local server, use localhost as the domain name.
- To make it easy to run the book applications and exercise starts for this book, get an
  index of the book_apps and ex_starts directories, and bookmark those pages.
How to test and debug a PHP page

When you finish editing a file for a web page, you need to save it. Then, you can test the web page or application by running it in a browser using the techniques of the last figure.

When the page is displayed in your web browser, you can test it by entering any required data and performing the actions indicated by the controls on the page. In figure 1-14, for example, the Firefox browser shows the first page of the web application that’s stored in the ch01_product_discount directory after you enter some data. This of course is the default page (index.html) of the Product Discount application that you saw earlier. After you enter the data, you click the Calculate Discount button to get the results.

Then, if the application doesn’t produce the right results or if it doesn’t produce any results, your code has one or more errors (or bugs). At that point, you need to find and fix those errors, which is commonly referred to as debugging. Sometimes, these bugs are syntax errors like omitting a semicolon at the end of a statement; sometimes they are logical errors like omitting a line of required code; and sometimes these errors are caused by statements that the PHP interpreter can’t execute. No matter what, though, you need to find and fix the errors.

Usually, when PHP encounters a statement that it can’t execute, the browser displays an error message as shown in this figure. In this case, the message shows that the error occurred as the PHP interpreter was trying to execute the statement on line 12 of the file named display_discount.php. As a result, you can begin by taking a closer look at the code near line 12 of that file.

In this case, though, the error occurred because the user entered invalid data for the list price and the discount percent. Here, the second entry starts with a dollar sign, which is illegal in a numeric field, and the third entry ends with a percent sign, which is illegal in a numeric field. In chapter 2, you’ll learn how to fix that type of problem by checking the data for validity before processing it.

When you find the causes of the error, you need to edit the file to fix the errors and save the file. Then, to test the application again, you can return to the web browser and click the Reload or Refresh button. This reloads the edited file. At that point, you can check whether the problems have been fixed.
The Product Discount application in the Chrome browser

An error displayed in the Chrome browser

How to test a PHP page for the first time
1. Make sure the Apache and MySQL servers are running.
2. Start a web browser and enter the URL for the application as shown in the last figure.
3. Test the page by entering both valid and invalid data, clicking on all links, and so on.

How to retest a PHP page after you change the source code
- Click the Reload or Refresh button in the browser. Then, test the page again.

Description
- If you get an error message, read the message to help determine the cause of the error. Then, modify the PHP code and retest the page.
- If you discover other types of errors that don’t lead to error messages, you also modify the code and retest the page.
- In chapter 6, you’ll learn more about finding errors (bugs) and fixing them (debugging).

Figure 1-14 How to test and debug a PHP 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 1-15 to view the HTML code for the page in a separate window. If, for example, you use the Chrome browser to view a web page, you can right-click the page and then select the View Page Source command to see the HTML code for that page.

In this figure, for example, you can see the HTML code that’s returned to the browser by the PHP file of the Product Discount application. Note that this HTML doesn’t include any PHP code. It only contains the HTML code that has been generated by the PHP code.

Why would you want to view the source code? When the page for an application isn’t displayed correctly and you want to find out whether the problem is the generated HTML code or something else. If, for example, the PHP code generates the code for an HTML table, you can check to see whether that HTML has been generated correctly.
The source code for a PHP page

![Source code of a PHP page]

How to view the source code for a page in Chrome and Firefox

- Right-click the page and then select the View Page Source command.

How to view the source code for a page in Internet Explorer

- Right-click the page and then select the View Source command.

Description

- When you view the source code for a web page in a web browser, the HTML code is opened in a separate window.
- In the source code, you see the HTML that has been generated by the PHP for the page. You can also see that all of the PHP has been removed from the page so it consists only of HTML.
- When testing an application that isn’t displaying the information correctly in the browser, you can view the source code to see whether the PHP has correctly generated the HTML code.
How to use NetBeans to develop a PHP application

Although a text editor is adequate for developing simple PHP applications like the ones for this book, most professional PHP developers use an Integrated Development Environment (IDE) that’s designed for developing PHP applications. However, an IDE can also be useful as you’re learning how to develop PHP applications.

If you’re interested, we recommend the NetBeans IDE for PHP. This free IDE runs on all major operating systems, and it provides many features that make it easier to edit and test PHP applications. To learn how to install this IDE, please refer to the appendixes. To learn how to get started with this IDE, please read the three topics that follow.

How to work with PHP projects and files

When you use NetBeans, all of the directories and files for an application are grouped within a project. You can also have more than one project open at the same time. In figure 1-16, the Projects tab of NetBeans shows that two projects are open: ch01_product_discount and ch02_future_value.

When you work with NetBeans, you can start commands by using the menus, clicking on the toolbar buttons, or right-clicking on an object and using the resulting shortcut menu, just as you would with any application. In this figure, for example, you can see how to open a project, start a new project, or close a project by using the toolbar buttons, but you can get the same results by using the commands in the File menu.

To work with the files in a project, you can use the Projects tab to view the files for the project. Then, you can open any file by double-clicking on the filename in the Projects tab. In this figure, for example, three of the files for the ch01_product_discount project are open in the text editor, and the file named display_discount.php is displayed. To start a new file, you can select the project in the Projects tab and click the New File button in the toolbar.

Incidentally, when you create a NetBeans project, NetBeans adds a subdirectory named nbproject to the application root directory. It uses this subdirectory and its files to manage the project. However, this has no effect on the operation of the project, and you can still edit the files using a standard text editor.

In addition, the applications you download from our website are Netbeans projects that contain an nbprojects subdirectory, so you should be able to open them by using the Open Project button and then navigating to the \xampp\htdocs\ folder.
NetBeans with three files open

How to work with projects

- To open a project, use the Open Project button in the toolbar and navigate to the project you want to open. This is how you will open the downloaded book applications.
- To start a new project, use the New Project button in the toolbar.
- To close a project, right-click on the project in the Projects tab and select the Close command from the resulting menu.

How to work with files

- To open a file, use the Projects tab to navigate to the file and then double-click the file.
- To start a new file, select the project and click the New File button in the toolbar.

Description

- NetBeans is an Integrated Development Environment (IDE) for developing PHP applications that makes it easier to create, edit, and test all of the HTML, CSS, and PHP files that you need for a web application.
- A NetBeans project consists of a top-level directory that contains the subdirectories and files for an application.
- When you create a NetBeans project, NetBeans adds an nbproject subdirectory that contains the extra files that NetBeans needs for managing the project.

Mac OS X note

- To enable right-clicking with Mac OS X, you can edit the system preferences for the mouse.

Figure 1-16  How to use NetBeans to work with projects and files
**How to edit and test a PHP application**

Figure 1-17 shows how to edit and test a PHP application. In general, the PHP editor works like any other text editor so you shouldn’t have any trouble using it. However, its auto-completion feature does an even better job of offering the variable names that you might want to use.

In addition, the PHP editor does more error checking as you type than most text editors. In this figure, for example, you can see a red circle error icon at the start of line 10 because the previous statement doesn’t end with a semicolon. This as-you-type error checking is also done for HTML and CSS code. The benefit, of course, is that you find and fix errors before you test the application, which is far more efficient than finding and fixing them later on.

The PHP editor also checks for other issues that aren’t errors but that might cause problems. In this figure, for example, you can see yellow triangle warning icons on lines 3, 4, and 5. If you hover your mouse over the icon, as shown in the figure, the PHP editor gives you details about what the issue is.

In this figure, the warning icons display a message that says that you shouldn’t access the global $_POST array directly. That’s because doing so can lead to security vulnerabilities if you don’t validate the data input or escape the data output. However, since this is only a warning, this code will run. As a result, you can ignore this warning for now. Later in this book, you’ll learn how to validate data input and how to escape data output.

After you’ve edited the files for an application, NetBeans also makes it easier to test them. To test the project that’s currently selected in the Projects window, for example, you just press the F6 key. This opens the default file for the application in the default browser of your system so the application is ready for testing. You can use the other techniques in this figure to run a project that isn’t currently selected in the Projects window or to run a specific file.
How to edit a PHP file

- Use normal editing techniques as you enter PHP code.
- When you see an auto-completion list, you can highlight an entry and press the Enter key to enter it into your code or you can double-click on it.
- If you see a red error icon at the start of a line, you should fix whatever errors the line contains before you test the application.
- If you see a yellow warning icon at the start of a line, you can hover the mouse over the icon to read the warning. Then, you can fix the issue if you want.

How to test a PHP application

- To run the current project, click on the Run Project button in the toolbar or press F6.
- To run other projects, right-click on the project and select the Run command.
- To run a file, right-click on the file and select the Run command.

Description

- The auto-completion feature of NetBeans provides lists of possible entries after you enter the starting characters for an entry.
- NetBeans uses a red error icon to identify lines that contain errors.
- NetBeans uses a yellow warning icon to identify lines that might not follow best practices.
- To remove some types of warnings, select the Tools→Options item and click on the Hints tab.
How to import and configure a PHP project

If you’ve created an application without using NetBeans, you need to import the application into a NetBeans project before you can edit and test the project. To do that, you can use the techniques described in figure 1-18.

It’s important to note that this only applies to applications created outside of NetBeans. For example, if you used Notepad++ to create the application for this chapter, you’d need to follow the importing procedure to edit and test the project in NetBeans. However, because the downloaded book applications were created in NetBeans, you don’t have to import them. You should be able to work with the downloaded book application projects in NetBeans just by opening them, as described in figure 1-16.

As part of the importing procedure, NetBeans prompts you to check the run configuration for the project. In this figure, for example, the second dialog box shows the configuration for an application named ch01_product_discount as it is being imported into NetBeans. Here, you need to make sure that the Project URL text box contains the correct path for running the project.

If the directory for the application is stored within the htdocs directory, NetBeans usually sets the Project URL correctly. As a result, you don’t need to change it. However, if you want to store the application in another directory, such as the book_apps directory, you can edit this URL. In addition, if you rename or copy an existing NetBeans project, the Project URL in the run configuration may no longer be correct. Then, when you run the application from NetBeans, the browser may display the wrong application. In that case, you can fix the problem by editing the run configuration as shown in this figure.

So be forewarned. If you only work with the book applications and exercise starts that you download from our website and install as shown in the appendix, you shouldn’t have any problems with run configurations. But as you import and copy applications of your own, keep the run configuration in mind.
The dialog box for starting a new project

The dialog box for configuring a project

How to check the run configuration for an existing project

- Right-click on a project in the Projects tab and select the Properties command. Then, click on Run Configuration in the Categories list and check the Project URL.

How to import a project

- To import a project, use the New Project command, but select PHP Application with Existing Sources in the Projects list. This will step you through the import procedure.
- In the third step, you are asked to check the run configuration. Here, you need to make sure the URL for running the project is correct.

Description

- Before you can run an existing application with NetBeans, you need to import it. Then, NetBeans creates the files it needs for managing the project in its nbproject directory.
- When you import a new Project, you need to check its configuration to make sure the URL for running the project is correct. You also need to check this URL if you copy a NetBeans project from one directory to another and then open it in NetBeans.

Figure 1-18 How to use NetBeans to import and configure a PHP project
Section 1  Get started fast with PHP and MySQL

Perspective

Now that you know how PHP applications work and how to edit and run them, you’re ready to learn how to code them. That, of course, is what you’ll learn to do in the next chapter.

Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>World Wide Web</td>
<td>PHP interpreter</td>
</tr>
<tr>
<td>Internet</td>
<td>round trip</td>
</tr>
<tr>
<td>web application</td>
<td>server-side language</td>
</tr>
<tr>
<td>client-server architecture</td>
<td>scripting language</td>
</tr>
<tr>
<td>client</td>
<td>application server</td>
</tr>
<tr>
<td>server</td>
<td>PHP: Hypertext Preprocessor (PHP)</td>
</tr>
<tr>
<td>web server</td>
<td>Cascading Style Sheets (CSS)</td>
</tr>
<tr>
<td>network</td>
<td>rule set</td>
</tr>
<tr>
<td>Network Interface Card (NIC)</td>
<td>XAMPP</td>
</tr>
<tr>
<td>router</td>
<td>deploy an application</td>
</tr>
<tr>
<td>Local Area Network (LAN)</td>
<td>document root directory</td>
</tr>
<tr>
<td>intranet</td>
<td>application root directory</td>
</tr>
<tr>
<td>Wide Area Network (WAN)</td>
<td>FTP (File Transfer Protocol) program</td>
</tr>
<tr>
<td>Internet Service Provider (ISP)</td>
<td>Uniform Resource Locator (URL)</td>
</tr>
<tr>
<td>Internet Exchange Point (IXP)</td>
<td>URL protocol</td>
</tr>
<tr>
<td>static web page</td>
<td>domain name</td>
</tr>
<tr>
<td>HyperText Transfer Protocol (HTTP)</td>
<td>path</td>
</tr>
<tr>
<td>HTTP request</td>
<td>bug</td>
</tr>
<tr>
<td>HTTP response</td>
<td>debug</td>
</tr>
<tr>
<td>HyperText Markup Language (HTML)</td>
<td>Integrated Development Environment (IDE)</td>
</tr>
<tr>
<td>dynamic web page</td>
<td>project</td>
</tr>
<tr>
<td>database server</td>
<td>import an application</td>
</tr>
</tbody>
</table>

Summary

- A **web application** uses an architecture that 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 that are connected at **Internet Exchange Points**.

- To request a web page, the web browser sends an **HTTP request** to the web server that includes the name of the requested file. Then, the web server retrieves the HTML for the requested web 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 web 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. When a web server receives a request for a web page, it uses the extension of the requested file to find out which server or script should process the request.

• HTTP requests for web pages with .php as the extension are processed by the PHP interpreter that runs on the web server. This interpreter processes the request along with any data that is passed with the request. Then, it generates the HTML for the HTTP response.

• HTML (HyperText Markup Language) is the language that defines the structure and contents of a web page. CSS (Cascading Style Sheets) is used to control how the web pages are formatted. For this book, you need to be familiar with both HTML and CSS, but the emphasis is on PHP.

• XAMPP is a free, open-source, cross-platform web server package that includes Apache, MySQL, and the PHP interpreter. It is commonly used for PHP applications, and it’s easy to install.

• To deploy a PHP application on your own computer or a local server, you need to copy all of its directories and files to the document root directory of Apache (\xampp\htdocs).

• To deploy a PHP application on the Internet, you need to transfer the directories and files from your computer or local server to a web server with Internet access. To do that, you can use an FTP program that uses File Transfer Protocol.

• To run a web application, you can enter the URL (Uniform Resource Locator) of the application’s directory into a browser’s address bar. Then, Apache looks for the default file in that directory and runs it. If it can’t find a default file, Apache displays an index of the subdirectories.

• If PHP can’t execute a statement while you’re testing an application, an error message is displayed in the browser. Then, you need to debug the application.

• To view the HTML for a web page, you can use your browser’s View Page Source or View Source command. This can be useful when you want to see whether the PHP application generated the correct HTML.

• To develop web pages, you can use a text editor like Notepad++ or TextWrangler. You can also use an Integrated Development Environment (IDE) that combines text editing with other development functions.

• NetBeans is an IDE that makes it easier to edit and test PHP applications. NetBeans treats each application as a project. You can import an existing application into a NetBeans project and then configure it so it will run right.
Before you do the exercises for this book...

Before you do the exercises for this book, you should download and install the software and applications for this book as described in the appendix.

Exercise 1-1  Test the Product Discount application

In this exercise, you’ll run the Product Discount application that’s described in this chapter. You’ll also use your browser to view its source code.

Start the application

1. If necessary, start the Apache web server as shown in figure 1-11.
2. Start your web browser.
3. Run the Product Discount application by entering this URL into the browser’s address bar and pressing the Enter key:
   
   http://localhost/book_apps/ch01_product_discount/

   If this application runs, it means that you have installed the software and downloadable applications correctly.

Test the application

4. Enter valid values in the three text boxes. Then, click the Calculate Discount button. This should display the results of the calculation on the second page of the application.
5. Look in the browser’s address bar and note that the URL points to a PHP file.
6. View the source code for this web page as shown in figure 1-15. Note that this source code only contains HTML, not PHP code.
7. Click on the browser’s Back button to return to the first page of the application, enter invalid numeric values like “xx” and “yy” in the second and third text boxes, and click the Calculate Discount button. Then, read the error message that’s displayed. You’ll learn how to handle errors like this later in this book.
8. Click the browser’s Back button to return to the first page of the application. Then, run the application again with valid data.
Exercise 1-2  Run the book and exercise applications

In this exercise, you’ll create and bookmark index pages for the book applications and for the applications that are the starting points for the exercises. Then, you’ll run some of these applications.

**Run some of the book applications**
1. Enter this URL into your browser’s address bar and press the Enter key
   
   `http://localhost/book_apps/
   
   This should display an index of the book applications.
2. Bookmark this page so you can quickly access it whenever you need it.
3. Run the Future Value application for chapter 2 by clicking on the ch02_future_value link in the index. This should start that application. Then, click on the Back button to return to the index.
4. If necessary, start MySQL as shown in figure 1-11. Then, go back to your browser and start the Product Manager application by clicking on ch04_product_manager. If it starts, this shows that you’ve installed MySQL and the downloadable MySQL databases correctly as shown in the appendices.
5. Click the Back button and start the Guitar Shop application in chapter 5. This should give you some idea of the types of applications that you’ll be able to develop by the time you complete section 1 of this book.

**Run some of the exercise applications**
6. Enter this URL into your browser’s address bar and press the Enter key
   
   `http://localhost/ex_starts/
   
   This should display an index of the applications that are used as the starting points for the exercises.
7. Bookmark this page so you can quickly access it whenever you need it.
8. Click on the link for ch02_ex1. This is the link for the application that you’ll start with in exercise 1 of chapter 2. It is the Product Discount application of chapter 1, which you’ll enhance in this exercise.
9. You are now ready to run any of the book applications or exercise starts with relative ease.

Exercise 1-3  Check out NetBeans

We recommend using NetBeans to do the exercises for this book. This exercise will help you get started with NetBeans. However, if prefer to use a text editor, you can skip to exercise 1-4.

1. Open NetBeans and then open this project as summarized in figure 1-16:
   
   `\xampp\htdocs\ex_starts\ch02_ex1`
2. Open the three files for this application as summarized in figure 1-16.
3. Review the three files and see how the color coding highlights the syntax in the HTML, CSS, and PHP files.
4. Delete the semicolon at the end of one of the PHP statements and note the red error icon that appears at the start of the line. Then, re-enter the semicolon.
5. Start a new line in the PHP portion of the PHP file, type $, and note the auto-completion list that’s displayed. Then, delete the code for the line that you started.
6. Run the application as shown in figure 1-17. This should start the application in your default browser.
7. Check the configuration for the application as shown in 1-18, and note that the Project URL is correct for this application.

**Exercise 1-4 Check out your text editor**

This exercise shows you how to use a text editor to do the exercises in this book. If you’re going to use NetBeans, you can skip this exercise.

1. Review the directories and files for the applications in the \xampp\htdocs directory. Note that each application includes an nbproject directory that’s used by NetBeans, but you can ignore that directory if you aren’t using NetBeans.
2. Use your text editor to open the html, css, and php files in this directory:

   \xampp\htdocs\ex_starts\ch02_ex1

   If you’re using Notepad++ on a Windows system, you can select the three files in the Windows Explorer, right-click on them, and select the Edit with Notepad++ item.
3. Review the three files and see how the color coding highlights the syntax in the HTML, CSS, and PHP files.
4. Delete the semicolon at the end of one of the PHP statements and see whether the error is highlighted. Then, re-enter the semicolon.
5. Start a new line in the PHP portion of the PHP file by typing $d and see whether an auto-completion list is displayed. If it isn’t, check the settings for your text editor to see whether you need to turn this feature on. Then, delete the code for the line that you started.
6. If you’re using Notepad++, you may want to change the style for comments as shown in figure 1-10.
7. If your text editor can’t open more than one file at a time or if it doesn’t provide syntax highlighting and auto-completion, you should probably switch to a different text editor or to NetBeans.
How to build your PHP and MySQL skills

The easiest way is to let *Murach’s PHP and MySQL (2nd Edition)* be your guide! So if you’ve enjoyed this chapter, I hope you’ll get your own copy of the book today. You can use it to:

- Teach yourself how to use PHP and MySQL to create the kind of database-driven web applications that are found on a majority of websites today, including sites like Facebook and Twitter
- Pick up new skills whenever you want to or need to by focusing on material that’s new to you
- Look up coding details or refresh your memory on forgotten details when you’re in the middle of developing a web application
- Loan to your colleagues who are always asking you questions about server-side programming

To get your copy, you can order online at [www.murach.com](http://www.murach.com) or call us at 1-800-221-5528 (toll-free in the U.S. and Canada). And remember, when you order directly from us, this book comes with my personal guarantee:

**100% Guarantee**

*You must be satisfied. Each book you buy directly from us must outperform any competing book or course you’ve ever tried, or send it back within 60 days for a full refund…no questions asked.*

Thanks for your interest in Murach books!

Mike