Installation, Configuration, and Web Application Basics

PENN FOSTER

Creating Web Pages : Installation, Configuration, and Web Application Basics

Lesson 1 Overview

In this lesson, you'll learn how to install and configure MySQL, Apache, and PHP, which you'll use throughout this course. You'll also review a brief history of the World Wide Web and discover how to use HTML and CSS to create a simple webpage. The lesson concludes with an introduction to both JavaScript and PHP.

1.1 Explain the web content delivery method

Installation and Configuration

READING ASSIGNMENT

Read this assignment. Then, read Chapter 1 and Appendices A through D of your textbook.

Understanding How the Web Works

At this point in your studies, you probably have a strong understanding of the World Wide Web and some of the technologies used to create webpages. Consider this assignment as a refresher—much of the rest of this course builds on this fundamental information.

Hypertext markup language (HTML) is the language for describing how text, graphics, and files containing other information are organized, linked, and displayed. It dictates how users will see content when they view it in **web browsers.** When the **web server** delivers the web content to the web browser, the web browser assembles web content components, arranges those parts according to HTML codes and commands, and displays the content on the user's monitor.

The appendices explain how to set up a local web server—Apache—for your personal development. If you want your web content to be accessible by others, you'll obtain a web hosting provider. Your textbook walks you through some of the aspects you should consider when selecting a provider, such as bandwidth, reliability, and more.

Every visitor to your website will likely use devices and configurations different than your own. That's why you should test how your websites and pages look on standard, portable, and mobile devices, as well as multiple browsers, such as the following:

- Apple Safari (www.apple.com/safari/)
- Google Chrome (www.google.com/chrome/)
- Microsoft Internet Explorer (www.microsoft.com/en-us/download/internet-explorer.aspx)
- Mozilla Firefox (www.mozilla.org/en-US/firefox/)

Once you have those browsers downloaded and installed, you'll create a sample HTML file in Notepad (Windows) or TextEdit (Mac). Make sure your code matches what's in Listing 1.1 in your textbook.

Then, you'll use a **file transfer protocol (FTP)** client to put your content on a web server to make it accessible to others. Your textbook explains how to select and use an FTP client. You'll also learn about basic file management, which involves naming and organizing directories. Doing so helps you find and maintain content in the future.

An **index file** (index.html) is the default page users see when they navigate to a specific directory in your website. It's used in subdirectories as well as the top-level directory (or **document root**) of your website.

Installation and Configuration

The four appendices at the end of your textbook walk you through the installation and configuration of the following:

- **XAMPP,** a cross-platform (X) installation package that includes Apache (A), MySQL (M, also known as MariaDB), PHP (P), and Perl (P)

Perl isn't discussed in your textbook, but you may come across this general-purpose programming language in the future.

- **MySQL,** a database management system
- **Apache,** which is web-server software
- **PHP,** an open-source, general-purpose scripting language

Your textbook includes installation instructions for the following operating systems:

- Linux/UNIX
- Windows
- Mac OS X

If you experience installation problems, visit the following sites to review the FAQs and troubleshooting advice:

- Apache Friends (www.apachefriends.org/index.html)
- MySQL Manual (dev.mysql.com/doc/)
- The Apache Software Foundation (httpd.apache.org/docs/)
- The PHP Group (www.php.net/manual/en/mongo.manual.php)

Key Points and Links

READING ASSIGNMENT

Key Points

- Hypertext markup language (HTML) is the language for describing how text, graphics, and files containing other information are organized, linked, and displayed.
- When the web server delivers the web content to the web browser, the browser assembles web content components, arranges those parts according to HTML codes and commands, and displays the content on the user's monitor.
- You should test how your websites and pages look on standard, portable, and mobile devices, as well as multiple browsers.
- Use a file transfer protocol (FTP) client to put content on a web server to make it accessible to others.
- XAMPP is a cross-platform (X) installation package that includes

Apache (local web-server software), MySQL (a database management system), PHP (an open-source, general-purpose scripting language), and Perl (a general-purpose programming language).

Links

- Apple Safari (www.apple.com/safari/)
- Google Chrome (www.google.com/chrome/)
- Microsoft Internet Explorer (www.microsoft.com/en-us/download/internet-explorer.aspx)
- Mozilla Firefox (www.mozilla.org/en-US/firefox/)
- Apache Friends (www.apachefriends.org/index.html)
- MySQL Manual (dev.mysql.com/doc/)
- The Apache Software Foundation (httpd.apache.org/docs/)
- The PHP Group (www.php.net/manual/en/mongo.manual.php)

Exercise: Installation and Configuration

Respond to the following based on your reading.

1. Describe how a webpage's coding is translated, delivered, and displayed to the user.
2. Why would you need a web hosting provider? What should you consider when selecting one?
3. Why should you test your webpage on multiple browsers and devices?
4. Identify the four major browsers in which you should test your

webpage.

5. What should you use to put your content on a web server?
6. What programs are included in the XAMPP installation package? Briefly differentiate between them.

Exercise Answer Key:

Exercise: Installation and Configuration

1. When the web server delivers the web content to the web browser, the browser assembles content components, arranges those parts according to HTML codes and commands, and displays the content on the user's monitor.
2. If you want your web content to be accessible to others, you need to obtain a web hosting provider. Some of the aspects you should consider when selecting a provider include bandwidth and reliability.
3. Every visitor to your website will likely use devices and configurations different than your own.
4. Apple Safari, Google Chrome, Microsoft Internet Explorer, and Mozilla Firefox are the four major browsers in which you should test your webpage.
5. Use a file transfer protocol (FTP) client to put your content on a web server to make it accessible to others.
6. XAMPP is a cross-platform (X) installation package that includes Apache (A), MySQL (M, also known as MariaDB), PHP (P), and Perl (P). Apache is web-server software; MySQL is a database management system; PHP is an open-source, general-purpose scripting language; and Perl is a general-purpose programming language.

1.2 Identify basic HTML tags and CSS styles

HTML and Cascading Style Sheets

READING ASSIGNMENT

Read this assignment. Then, read Chapters 2 and 3 of your textbook.

To create a simple webpage, you'll start by writing a few paragraphs of text in a plain-text editor. Never use Microsoft Word or any other full-featured word-processing software when writing code. They're not meant for creating web content and can distort and corrupt your code. Use Listing 2.1 in your textbook as a guide and then save your text file with **.html** at the end of the filename.

Next, you'll add **HTML tags,** commands that start with **<** and end with **>**. These tags tell the web content how to display. HTML tags often have an **opening tag** that indicates when the command begins and a **closing tag** for when the command ends. The closing tag is identical to an opening tag except it starts with a forward slash (/). An **empty tag** or **empty element** issues an HTML command without adding any text to the page. For example, the empty tag **<br>** creates a line break.

HTML5 also includes tags to enhance the **semantics** (meaning) of the marked-up text, such as

- **<header>**, which contains introductory information
- **<aside>**, which is meant for highlighting elements such as pull quotes

- **<footer>**, which is the additional information that's often found at the bottom of the webpage

You'll fine-tune the visual display of your web content using **Cascading Style Sheets (CSS).** The concept of CSS is to create a style sheet document that specifies **style rules,** or directions regarding fonts, colors, and other characteristics that establish a unique look. Then you link every page that should share the look of the style sheet. This prevents you from needing to specify all those styles repeatedly in each separate document, especially if you choose to redesign the website in the future.

There are two types of style sheets:

- **Internal style sheets** are placed directed within a webpage. These employ the use of the **<style>** tag within the **<head>** tag. An internal style sheet would be used if a single webpage had a style that was different from the rest of the website.
- **External style sheets** exist in separate documents and are linked via a special tag. An external style sheet would list only formatting information. If a webpage requires the style rules set forth in the external style sheet, the webpage would include a link to the style sheet.

Take a look at Listing 2.5, which displays a single external style sheet. The style sheet sets style rules for the elements **body, h1,** and so on. The elements listed in this style sheet are called **selectors.** A selector is used to identify tags on a page to which you apply styles.

A CSS **style property** is a specific style that you can assign a value. Your textbook outlines some of the most important style properties used in CSS, including

- **display,** which dictates how an element is displayed with respect to other elements
- **border,** which establishes a visible boundary around an element
- **color** and **background-color,** which set the foreground and background color of an element, respectively
- **text-align** and **text-indent**, which control the alignment and indentation of text
- **font**, which precedes directions about a font's family, size, style, and weight
- **padding,** which controls the void space around an element

If you want certain text to look different from other text on your webpage, you'll create style classes. A **style class** is a custom set of formatting specifications that can be applied to any webpage element. Additionally, you can create **style IDs,** which are custom formatting specifications that can be applied to only one element on a webpage. It makes sense to apply a style ID to a webpage's title since it's used only once per time within each page.

Once you've created a style sheet, you can check it using a CSS validator, like WC3's CSS Validation Service. (jigsaw.w3.org/css-validator/)

Understanding the CSS Box Model and Positioning

Every element in HTML is considered a "box." The **CSS box model** describes the way in which every HTML box has the potential for padding, a border, and a margin, regardless if they're visible. The content is surrounded by **padding,** space that inherits the background color of the content area. The **border** exists on the outer edge of the padding, and the **margin** is the transparent area outside the border. Figure 3.1 illustrates these terms.

To know the true height and width of an element, all elements of the box model must be taken into account. For example, an image may be 250 pixels wide and 250 pixels high, but the padding and margin around that image change the size of the element and the amount of room it takes up on the page. The true size of the element will influence positioning.

Relative positioning is the default positioning of HTML elements: Elements are arranged from left to right, and when they reach the right edge of the screen, they move down to the next row. **Absolute positioning** allows you to set the exact position of HTML content on the page. A style rule can also indicate that the positioning is **static** (the default positioning according to the normal flow of content) or **fixed** (the element is "frozen" in place relative to the viewport).

While absolute positioning provides freedom to place elements anywhere, it may inadvertently take up the space another element is using. The **z-index** style property allows you to set the order of elements and control the way they overlap each other. The z-index property is used to assign a numeric value to each element's positioning so that higher z-index values appear on top of lower z-index values. You can also control how text flows around an element or prevent it from flowing near the element altogether. An **overflow** property handles text that doesn't fit within its rectangular area.

A **fixed layout** offers webpage designers precise control over the appearance of the content area, but this has gone out of favor with the advent of so many different screen sizes and resolutions. A **fluid layout** that doesn't use a specific width in pixels is preferable. **Responsive web design** builds on the adaptable nature of fluid layouts; the goal is to ensure the website is enjoyable and usable to all users, regardless of screen size.

A **fixed/fluid hybrid layout** is also desirable—it includes elements from both types of layouts. Your textbook explains how to create a basic hybrid layout.

Key Points and Links

READING ASSIGNMENT

Key Points

- HTML tags, which tell the web content how to display, often have an opening tag that indicates when the command begins and a closing tag for when the command ends.
- HTML5 includes tags to enhance the semantics (meaning) of the marked-up text.
- The concept of Cascading Style Sheets (CSS) is to create a style sheet document that specifies style rules, or directions regarding fonts, colors, and other characteristics that establish a unique look.
- The CSS box model describes the way in which every HTML box

has the potential for padding, a border, and a margin, regardless if they're visible.

- Relative positioning is when elements are arranged from left to right, and when they reach the right edge of the screen, they move down to the next row. Absolute positioning allows you to set the exact position of HTML content on the page.
- A fixed layout offers webpage designers precise control over the appearance of the content area, but a fluid layout that doesn't use a specific width in pixels is preferable. A fixed/fluid hybrid layout includes elements from both types of layouts.

Links

- WC3's CSS Validation Service (jigsaw.w3.org/css-validator/)

Exercise: HTML and Cascading Style Sheets

Respond to the following based on your reading.

1. Why should you never use Microsoft Word to write code?
2. Differentiate between opening and closing HTML tags.
3. Explain the purpose of cascading style sheets (CSS).
4. Name and describe the two types of style sheets.
5. Identify and define five style properties used in CSS.
6. What is the purpose of *style classes* and *style IDs?*
7. Define the four elements of the CSS box model.
8. Compare and contrast *relative positioning* and *absolute positioning.*
9. What does the **z-index** property do?

Exercise Answer Key:

Exercise: HTML and Cascading Style Sheets

1. Microsoft Word and other full-featured word-processing software aren't meant for creating web content and can distort and corrupt your code. Use a plain-text editor.
2. HTML tags often have an opening tag that indicates when the command begins and a closing tag for when the command ends. The closing tag is identical to an opening tag except it starts with a forward slash (/).
3. The concept of CSS is to create a style sheet document that specifies style rules, or directions regarding fonts, colors, and other characteristics that establish a unique look. Then you link every page that should share the look of the style sheet. This prevents you from needing to specify all those styles repeatedly in each separate document, especially if you choose to redesign the website in the future.
4. Internal style sheets are placed directed within a webpage. These employ the use of the <style> tag within the <head> tag. An internal style sheet would be used if a single webpage had a style that was different from the rest of the website. External style sheets exist in separate documents and are linked via a special tag. An external style sheet would list only formatting information. If a webpage requires the style rules set forth in the external style sheet, the webpage would include a link to the style sheet.
5. Some of the most important style properties used in

CSS include **display,** which dictates how an element is displayed with respect to other elements; **border,** which establishes a visible boundary around an element; **color** and **background-color,** which set the foreground and background colors of an element; **text-align** and **text-indent,** which control the alignment and indentation of text; **font,** which precedes directions about a font's family, size, style, and weight; and **padding,** which controls the void space around an element.

6. For certain text to look different from other text on your webpage, create style classes. A style class is a custom set of formatting specifications that can be applied to any webpage element. Style IDs are custom formatting specifications that can be applied to only one element on a webpage. It makes sense to apply a style ID to a webpage's title since it's used only once within each page.
7. The CSS box model describes the way in which every HTML box has the potential for padding, a border, and a margin, regardless if they're visible. The content is surrounded by padding, space that inherits the background color of the content area. The border exists on the outer edge of the padding, and the margin is the transparent area outside the border.
8. Relative positioning is the default positioning of HTML elements: Elements are arranged from left to right, and when they reach the right edge of the screen, they move down to the next row. Absolute positioning allows you to set the exact position of HTML content on the page.
9. The **z-index** property is used to assign a numeric value to each element's positioning so that higher z-index values appear on top of

lower z-index values.

1.3 Compare and contrast JavaScript and PHP

Introducing JavaScript and PHP

READING ASSIGNMENT

Read this assignment. Then, read Chapters 4 and 5 of your textbook.

Introducing JavaScript

Whereas HTML and CSS are used to format webpages, **scripting languages** such as JavaScript are used to add **dynamism** (action) and interactivity. For example, JavaScript allows you to animate images, display messages to the user, validate contents of a form, make calculations, and more. Listing 4.1 in your textbook shows an HTML document with a simple script that provides a user with the current date and time.

Scripts can be placed in the body of the page, in the header, within an HTML tag, or in a separate file entirely. A separate JavaScript file (written in a text editor and saved with a **.js** extension) is recommended for more complicated scripts. You can add an external script to a webpage by specifying the file name in the **<script>** tag.

Scripts are **event handlers.** In other words, they typically respond to events, such as a button being clicked. In that example, the script would indicate what happens when the button is clicked. In Listing 4.2, the script's event response is to display a message to the user.

A **statement** is the basic unit of programs such as JavaScript; it's a section of code that performs a single action. **Functions** are groups of statements that are treated as a single unit. A statement that uses a function is a **function call.**

Other important terms used in JavaScript include the following:

- **Variables** are containers that can store a number, a string of text, or another value.
- **Objects** store two or more pieces of data at once, which are called **properties** of the object, and **methods,** which are functions that work with the object's data.
- **Conditional statements** use the keyword **if**; if a certain event happens, it will trigger an action.
- **Loops** are groups of statements that repeat a certain number of times and use the keyword **for**.

Though JavaScript is a simple language, its **syntax** (rules that define how you use the language) must be used correctly. Some basic rules are as follows:

- JavaScript is **case sensitive;** meaning that whether a letter is capitalized or not matters. Keywords and object name are lowercase, built-in objects are capitalized, and methods are a combination of capitals and lowercase.
- Names for variables, objects, and function names must begin with a letter or underscore and can't include **reserved words**, such as keywords and object names like **if** or **window**.
- Blank space or **whitespace** is ignored by JavaScript but makes the

script more readable.

You can—and should—include comments about your script to explain it to other programmers or remind yourself about it. Here are some other best practices:

- Use one statement per line.
- Use a semicolon at the end of a statement.
- Use separate JavaScript files whenever possible.
- Avoid being browser-specific.
- Keep JavaScript optional—don't rely on it to perform an essential function on your site.

JSON (pronounced "Jason") refers to **JavaScript Object Notation**. It's not part of the core JavaScript language. Instead, it's used as a general-purpose syntax for exchanging data in a string format. In short, it structures information into a hierarchy and groups the information into an object, which you can then use in a JavaScript script.

Most JavaScript errors are the result of mistyped statements. You can configure your browser to help you diagnose JavaScript problems.

Introducing PHP

PHP is a server-side scripting language that operates on a web server. In other words, it's like JavaScript in that it's a scripting language and can automate tasks. However, the program codes that have been previously discussed relied on the user's browser. PHP language is executed on the server before it gets to the user's browser. This means that the user's browser doesn't need to have a Java plug-in to experience the functionality of a script.

When writing PHP, you need to tell the PHP engine to execute your command; otherwise, your code will be mistaken for HTML. Table 5.1 in your textbook shows how to start and end your PHP tags. Like JavaScript, PHP uses statements to represent instructions and comments to annotate scripts.

Key Points and Links

READING ASSIGNMENT

Key Points

- Scripting languages such as JavaScript are used to add dynamism and interactivity.
- Scripts are event handlers; they typically respond to events, such as a button being clicked.
- JavaScript Object Notation (JSON) structures information into a hierarchy and groups the information into an object, which can then be used in a JavaScript script.
- PHP is a server-side scripting language that operates on a web server. It's like JavaScript in that it's a scripting language and can automate tasks.
- When writing PHP, you need to tell the PHP engine to execute your command; otherwise, your code will be mistaken for HTML.

Exercise: Introducing JavaScript and PHP

Respond to the following based on your reading.

1. What is the purpose of JavaScript?
2. Where can JavaScript scripts be placed?
3. What does it mean that scripts are "event handlers"?
4. What is a *statement,* and how is it used in relation to functions?
5. What are the basic syntax rules for JavaScript?
6. Name three best practices when it comes to using JavaScript.
7. What is the purpose of JSON?
8. How does PHP differ from JavaScript?

Exercise Answer Key:
Exercise: Introducing JavaScript and PHP

1. Whereas HTML and CSS are used to format webpages, scripting languages such as JavaScript are used to add dynamism and interactivity. For example, JavaScript allows you to animate images, display messages to the user, validate contents of a form, make calculations, and more.
2. Scripts can be placed in the body of the page, in the header, within an HTML tag, or in a separate file entirely. A separate JavaScript file (written in a text editor and saved with a .js extension) is recommended for more complicated scripts.
3. Scripts typically respond to an event, such as a button being clicked.

4. A statement is the basic unit of programs such as JavaScript; it's a section of code that performs a single action. Functions are groups of statements that are treated as a single unit. A statement that uses a function is a function call.
5. JavaScript is case sensitive—whether a letter is capitalized or not matters. Names for variables, objects, and function names must begin with a letter or underscore and can't include reserved words, such as keywords and object names like *if* or *window*. Blank space or whitespace is ignored by JavaScript but makes the script more readable.
6. JavaScript best practices include the following: use comments about your script to explain it to other programmers or remind yourself about it; use one statement per line; use a semicolon at the end of a statement; use separate JavaScript files whenever possible; avoid being browser-specific; and keep JavaScript optional—don't rely on it to perform an essential function on your site.
7. JSON is used as a general-purpose syntax for exchanging data in a string format. In short, it structures information into a hierarchy and groups the information into an object, which you can then use in a JavaScript script.
8. PHP is a server-side scripting language that operates on a web server. In other words, it's like JavaScript in that it's a scripting language and can automate tasks. However, the programmed codes that have been previously discussed relied on the user's browser. PHP language is executed on the server before it gets to the user's browser. This means that the user's browser doesn't need to have a Java plug-in to experience the functionality of a script.

1.4 Create a webpage using HTML and CSS

INT210 Graded Project 1

READING ASSIGNMENT

Your project must be submitted as a zipped/compressed (*.zip) file that includes the following files:

- HTML file of your webpage's content
- CSS file of your webpage's external style sheet
- Screenshot(s) in JPEG format (*.jpg) of how your webpage displays in your browser
- A Rich Text Format (*.rtf) or Microsoft Word (*.doc) file that lists the following information:
  - Your name
  - Your student ID number
  - The exam number
  - Your email address

For information on how to take and save a screenshot on your computer, visit the link that's most applicable to you:

- Digital Trends "How to Take a Screenshot on a PC" (www.digitaltrends.com/computing/how-to-take-a-screenshot-on-pc/)
- Digital Trends "How to Take a Screenshot on a Mac" (www.digitaltrends.com/computing/how-to-take-a-screenshot-on-a-mac/)

For information on how to compress a file, visit the link that's most applicable to you:

- Microsoft "Zip and Unzip Files" (support.microsoft.com/en-us/help/14200/windows-compress-uncompress-zip-files)
- Macworld "How to Zip Files on Mac" (www.macworld.co.uk/how-to/mac-software/zip-files-mac-3676176/)

Save your compressed file as **[Your Name]_INT210_GradedProject1.** Your project will be individually graded by your instructor and therefore may take up to five to seven days to grade. To submit your graded project, follow these steps:

- Log into your student portal.
- Click **Take Exam** next to the lesson you're working on.
- Find the exam number for your project at the top of the Project Upload page.
- Follow the instructions provided to complete your exam.

Be sure to keep a backup copy of any files you submit to the school!

Introduction

You'll apply the concepts of Lesson 1 to create a personal webpage using HTML and CSS.

Instructions

Review Chapters 2 and 3 in your book to ensure you understand how to use and apply HTML tags and CSS styles.

Then, you'll create a personal webpage using HTML and CSS. The content of your webpage will be about yourself and your interests.

Your webpage should include the following:

- Three paragraphs preceded by a header that introduces the topic of the paragraph
- A footer that contains the text *All content © [year]* followed by your name
- Three links that pertain to the content of your webpage
- Three images that are positioned according to the CSS box model

You'll also need to create a single external style sheet document that establishes a unique look for your webpage. Establish style rules for the following properties:

- Headers
- Body text
- Footer
- Images
- Links

Work only in a plain-text editor. Save your webpage as an .html file and your style sheet as a .css file. Be sure to link the webpage to the style sheet using the **<link>** tag.

Grading Criteria

The following rubric will be used to grade your project.

INT210 Graded Project 1					
Grading Criteria	**Exemplary**	**Proficient**	**Fair**	**Poor**	**Not Evident**

Webpage Content (HTML File)	The webpage includes all of the following required content: three paragraphs preceded by a header that introduces the topic of the paragraph, a footer that contains the text *All content © [year]* followed by the student’s name, three links that pertain to the content of the webpage, and three images that are positioned according to the CSS box model. (40 points)	The webpage includes most of the following required content: three paragraphs preceded by a header that introduces the topic of the paragraph, a footer that contains the text *All content © [year]* followed by the student’s name, three links that pertain to the content of the webpage, and three images that are positioned according to the CSS box model. (39–26 points)	The webpage includes some of the following required content: three paragraphs preceded by a header that introduces the topic of the paragraph, a footer that contains the text *All content © [year]* followed by the student’s name, three links that pertain to the content of the webpage, and three images that are positioned according to the CSS box model. (25–13 points)	The webpage includes few of the following required content: three paragraphs preceded by a header that introduces the topic of the paragraph, a footer that contains the text *All content © [year]* followed by the student’s name, three links that pertain to the content of the webpage, and three images that are positioned according to the CSS box model. (12–1 points)	The webpage doesn’t include the following required content: three paragraphs preceded by a header that introduces the topic of the paragraph, a footer that contains the text *All content © [year]* followed by the student’s name, three links that pertain to the content of the webpage, and three images that are positioned according to the CSS box model. (0 points)
Webpage Style (CSS File)	The single external style sheet document includes style rules for the following properties: headers, body text, footer, images, and links. (30 points)	The single external style sheet document includes style rules for most of the following properties: headers, body text, footer, images, and links. (29–20 points)	The single external style sheet document includes style rules for some of the following properties: headers, body text, footer, images, and links. (19–10 points)	The single external style sheet document includes style rules for a few of the following properties: headers, body text, footer, images, and links. (9–1 points)	The single external style sheet document lacks style rules for the following properties: headers, body text, footer, images, and links. (0 points)

Functionality	The webpage displays and functions as intended. Code is neat, clear, and correct. The overall design of the webpage shows thoughtful consideration and a strong understanding of the lesson's material. (30 points)	The webpage displays and functions as intended. For the most part, code is neat, clear, and correct. The overall design of the webpage shows consideration and a good understanding of the lesson's material. (29–20 points)	The webpage displays and functions, but not as intended. Some of the code is neat, clear, and correct. The overall design of the webpage shows some consideration and a fair understanding of the lesson's material. (19–10 points)	The webpage displays and functions, but not as intended. For the most part, code is messy, unclear, and incorrect. The overall design of the webpage shows little consideration and a weak understanding of the lesson's material. (9–1 points)	The webpage doesn't display nor function. Code is messy, unclear, and incorrect. The overall design of the webpage lacks consideration and shows a poor understanding of the lesson's material. (0 points)

Key Points and Links

READING ASSIGNMENT

Links

- Digital Trends "How to Take a Screenshot on a PC" (www.digitaltrends.com/computing/how-to-take-a-screenshot-on-pc/)
- Digital Trends "How to Take a Screenshot on a Mac" (www.digitaltrends.com/computing/how-to-take-a-screenshot-on-a-mac/)
- Microsoft "Zip and Unzip Files" (support.microsoft.com/en-us/help/14200/windows-compress-uncompress-zip-files)
- Macworld "How to Zip Files on Mac" (www.macworld.co.uk/how-to/mac-software/zip-files-mac-3676176/)

Lesson 1 Review

Self-Check

1. Which of the following dictates how users will see content when they view it in web browsers?

a. HTML
b. JSON
c. Apache
d. PHP

2. Which of the following delivers web content to the browser?

a. Server
b. JSON
c. Hosting provider
d. Scripts

3. _______ is a local web server.

a. Perl
b. Apache
c. PHP
d. MySQL

4. Which of the following is the default page users see when they navigate to a specific directory in your website?

a. index.html
b. firstpage.js
c. index.js
d. firstpage.html

5. Which of the following should you use to put your content on a web server to make it accessible to others?

a. CSS

b. FTP client

c. JSON

d. PHP script

6. An HTML closing tag is similar to an opening tag, although it starts with which of the following characters?

a. /

b. >

c. $

d. ?

7. What are the two types of style sheets?

a. Internal and external

b. Fixed and fluid

c. Local and global

d. Hybrid and responsive

8. Which of the following is an HTML5 tag that enhances the semantics of marked-up text?

a. <p>

b. <br>

c. <footer>

d. <a href>

9. Which of the following style properties allows you to set the order of elements and control the way they overlap each other?

a. **z-index**

b. **block**

c. **inline**

d. **span**

10. _______ positioning is the default positioning of HTML elements.

a. Relative

b. Absolute

c. Static

d. Fixed

11. The _______ of an element will influence positioning.

a. true size

b. border

c. margin

d. padding

12. An element's content is surrounded by

a. a border.

b. padding.

c. a margin.

d. pixels.

13. Which of the following style properties controls the void space around an element?

a. **display**

b. **border**

c. **padding**

d. **text-align**

14. Which of the following syntax rules for JavaScript is *true?*

a. Methods must be lowercase.

b. Capital and lowercase letters can be used interchangeably.

c. Names for variables can begin with an underscore.

d. Keywords and object names must be capitalized.

15. A/An _______ is the basic unit of programs such as JavaScript; it's a section of code that performs a single action.

a. function

b. object

c. statement

d. variable

16. _______ structures information into a hierarchy and groups the information into an object, which you can then use in a JavaScript script.

a. PHP

b. CSS

c. JSON

d. MySQL

17. Which of the following is a JavaScript best practice?

a. Be browser-specific.

b. Include JavaScript statements within the body of the page whenever possible.

c. Use a colon at the end of a statement.

d. Use one statement per line.

18. When writing PHP, you need to tell the PHP engine to execute your command; otherwise, your code will be mistaken for

a. CSS.

b. HTML.

c. JavaScript.

d. JSON.

19. Most JavaScript errors are the result of

a. unconfigured browsers.

b. PHP interference.

c. server downtime.

d. mistyped statements.

20. _______ allows you to animate images, display messages to the user, validate contents of a form, make calculations, and more.

a. Apache
b. JavaScript
c. PHP
d. MySQL

Self-Check Answer Key

1. HTML
   Explanation: Hypertext markup language (HTML) is the language for describing how text, graphics, and files containing other information are organized, linked, and displayed. It dictates how users will see content when they view it in web browsers.
   Reference: Section 1.1

2. Server
   Explanation: When the web server delivers the web content to the web browser, the browser assembles the content components, arranges those parts according to HTML codes and commands, and displays the content on the user's monitor.
   Reference: Section 1.1

3. Apache
   Explanation: Apache is a local web server.
   Reference: Section 1.1

4. index.html

Explanation: An index file (index.html) is the default page users see when they navigate to a specific directory in your website.
Reference: Section 1.1

5. FTP client
   Explanation: Use a file transfer protocol (FTP) client to put your content on a web server to make it accessible to others.
   Reference: Section 1.1

6. /
   Explanation: The closing tag is identical to an opening tag except it starts with a forward slash (/).
   Reference: Section 1.2

7. Internal and external
   Explanation: There are two types of style sheets: internal and external.
   Reference: Section 1.2

8. <footer>
   Explanation: HTML5 includes tags to enhance the *semantics* (meaning) of the marked-up text, such as **<header>, <aside>,** and **<footer>.**
   Reference: Section 1.2

9. **z-index**
   Explanation: The **z-index** style property allows you to set the order

of elements and control the way they overlap each other.
Reference: Section 1.2

10. Relative
Explanation: Relative positioning is the default positioning of HTML elements: Elements are arranged from left to right, and when they reach the right edge of the screen, they move down to the next row.
Reference: Section 1.2

11. true size
Explanation: The true size of the element will influence positioning.
Reference: Section 1.2

12. padding.
Explanation: An element's content is surrounded by padding, space that inherits the background color of the content area. The border exists on the outer edge of the padding, and the margin is the transparent area outside the border.
Reference: Section 1.2

13. **padding**
Explanation: The **padding** style property controls the void space around an element.
Reference: Section 1.2

14. Names for variables can begin with an underscore.
Explanation: Though JavaScript is a simple language, its syntax

(rules that define how you use the language) must be used correctly. Some basic rules are as follows:

- JavaScript is case sensitive; whether a letter is capitalized or not matters. Keywords and object name are lowercase, built-in objects are capitalized, and methods are a combination of capitals and lowercase.
- Names for variables, objects, and function names must begin with a letter or underscore and can't include reserved words, such as keywords and object names.
- Blank space or whitespace is ignored by JavaScript but makes the script more readable.

Reference: Section 1.3

15. statement
    Explanation: A statement is the basic unit of programs such as JavaScript; it's a section of code that performs a single action.
    Reference: Section 1.3

16. JSON
    Explanation: JSON is used as a general-purpose syntax for exchanging data in a string format. In short, it structures information into a hierarchy and groups the information into an object, which you can then use in a JavaScript script.
    Reference: Section 1.3

17. Use one statement per line.

Explanation: JavaScript best practices include the following:

- Use one statement per line.
- Use a semicolon at the end of a statement.
- Use separate JavaScript files whenever possible.
- Avoid being browser-specific.
- Keep JavaScript optional—don't rely on it to perform an essential function on your site.

Reference: Section 1.3

18. HTML.

    Explanation: When writing PHP, you need to tell the PHP engine to execute your command; otherwise, your code will be mistaken for HTML.

    Reference: Section 1.3

19. mistyped statements.

    Explanation: Most JavaScript errors are the result of mistyped statements.

    Reference: Section 1.3

20. JavaScript

    Explanation: JavaScript allows you to animate images, display messages to the user, validate contents of a form, make calculations, and more.

    Reference: Section 1.3

Flash Cards

1. Term: Hypertext Markup Language (HTML)

Definition: The language for describing how text, graphics, and files containing other information are organized, linked, and displayed

2. Term: Index File

Definition: The default page users see when they navigate to a specific directory in a website

3. Term: Web Server

Definition: Delivers the web content to the web browser

4. Term: Web Browser

Definition: Assembles web content components, arranges those parts according to HTML codes and commands, and displays the content on the user's monitor

5. Term: File Transfer Protocol (FTP)

Definition: A method of putting content on a web server to make it accessible to others

6. Term: Responsive Web Design

Definition: Builds on the adaptable nature of fluid layouts; the goal is to ensure a website is enjoyable and usable to all users, regardless of screen size

7. Term: Relative Positioning

Definition: The default positioning of HTML elements; elements are arranged from left to right, and when they reach the right edge of the screen, they move down to the next row

8. Term: Style Class
Definition: A custom set of formatting specifications that can be applied to any webpage element

9. Term: Style Rules
Definition: Directions regarding fonts, colors, and other characteristics that establish a unique look

10. Term: Style Property
Definition: A specific style that can be assigned a value

11. Term: Statement
Definition: The basic unit of programs such as JavaScript; a section of code that performs a single action

12. Term: Syntax
Definition: Rules that define how a language is used

13. Term: JavaScript Object Notation (JSON)
Definition: A general-purpose syntax for exchanging data in a string format; structures information into a hierarchy and groups the information into an object, which you can then use in a JavaScript script

14. Term: Function Call

Definition: A statement that uses a function

15. Term: PHP

Definition: An open-source, general-purpose, server-side scripting language that operates on a web server