

# HTML and CSS

---

Project Workbook  
Student Edition

Presented By  
**LK LearnKey®**

# HTML and CSS Project Workbook

## First Edition

LearnKey creates signature multimedia courseware. LearnKey provides expert instruction for popular computer software, technical certifications, and application development with dynamic video-based courseware and effective learning management systems. For a complete list of courses, visit <https://www.learnkey.com>.

All rights reserved. Unauthorized reproduction or distribution is prohibited.

# Table of Contents

<b>Introduction</b>	<b>1</b>
Best Practices Using LearnKey's Online Training	2
Using This Workbook	3
Skills Assessment	4
HTML and CSS Video Times	5
<b>Domain 1</b>	<b>6</b>
Fill-in-the-Blanks	7
Script and Noscript Tags	8
Link and Meta Tags	9
HTML Fundamentals	10
<b>Domain 2</b>	<b>11</b>
Fill-in-the-Blanks	12
Creating Style Sheets	13
Style Sheet Precedence	14
Valid Syntax for Rule Sets	15
Apply Classes	16
Apply IDs	17
Apply Pseudo Classes	18
<b>Domain 3</b>	<b>19</b>
Fill-in-the-Blanks	20
HTML Tables	22
HTML Headings, Paragraphs, and Line Breaks	23
Hr, Div, and Span Tags	24
HTML List Tags	25
Semantic Tags	26
Details and Summary Tags	27
Figure and Caption Tags	28
Image Links	29
Hyperlinks	30
Bookmarks	31
Relative and Absolute Links	32
Navigating Folder Hierarchies	33



Map and Area Tags	34
Action and Method Attributes	35
Accessibility	36
Input Types and Labels	37
The Select Tag	38
Textarea Tag	39
Button Tags	40
Output Tags	41
Option Tag	42
Datalist	43
Fieldset and Legend Tags	44
<b>Domain 4</b>	<b>45</b>
Fill-in-the-Blanks	46
Add Images	47
Image Layouts and Links	48
Video and Source Tags	49
Audio and Track Tags	50
Simple Iframe Implementations	51
<b>Domain 5</b>	<b>52</b>
Fill-in-the-Blanks	53
Height and Max-Width Used with an Element	55
Float Positioning	56
Relative and Absolute Positioning	57
Static and Fixed	58
Overflow and Alignment Settings	59
Display	60
Inline vs. Block	61
Visibility	62
Box Model Margins and Padding	63
Font Families in CSS	64
Font Colors in CSS	65
Font Styles and Weights in CSS	66
Link Colors in CSS	67
Font Variant	68



Text Alignment and Decoration	69
Indentation	70
Line Height and Word-Wrap	71
Letter-Spacing, Word-Spacing	72
Border Color and Style	73
Border Width	74
Backgrounds	75
Divs and Colors	76
Units of Measure	77
Viewport and Media Query Settings	78
Percentages and Pixels	79
Frameworks, Templates, and Max Width	80
Grids	81
<b>Domain 6</b>	<b>82</b>
Fill-in-the-Blanks	83
Reusing Rules and Rule Sets	84
CSS Comments	85
Multiple Browser Tests and Web Safe Fonts	86
Separation of Structure and Style	87
Text Alternatives	88
Color Contrast and Typography	89
Tab Order and Text Resizing	90
Text Hierarchy	91
Syntax Errors	92
Tag Mismatch	93
Cascading Issues	94
<b>Appendix</b>	<b>95</b>
Glossary	96
Objectives	99
<b>HTML and CSS Lesson Plan</b>	<b>101</b>
Domain 1 Lesson Plan	101
Domain 2 Lesson Plan	102
Domain 3 Lesson Plan	103
Domain 4 Lesson Plan	105



Domain 5 Lesson Plan

106

Domain 6 Lesson Plan

108

# HTML and CSS

---

## Introduction

Presented By  
**LK LearnKey®**

# Best Practices Using LearnKey's Online Training

LearnKey offers video-based training solutions that are flexible enough to accommodate private students and educational facilities and organizations.

Our course content is presented by top experts in their respective fields and provides clear and comprehensive information. The full line of LearnKey products has been extensively reviewed to meet superior standards of quality. Our course content has also been endorsed by organizations such as Certiport, CompTIA®, Cisco, and Microsoft. However, it is the testimonials given by countless satisfied customers that truly set us apart as leaders in the information training world.

LearnKey experts are highly qualified professionals who offer years of job and project experience in their subjects. Each expert has been certified at the highest level available for their field of expertise. This expertise provides the student with the knowledge necessary to obtain top-level certifications in their chosen field.

Our accomplished instructors have a rich understanding of the content they present. Effective teaching encompasses presenting the basic principles of a subject and understanding and appreciating organization, real-world application, and links to other related disciplines. Each instructor represents the collective wisdom of their field and within our industry.

## Our Instructional Technology

Each course is independently created based on the manufacturer's standard objectives for which the course was developed.

We ensure that the subject matter is up-to-date and relevant. We examine the needs of each student and create training that is both interesting and effective. LearnKey training provides auditory, visual, and kinesthetic learning materials to fit diverse learning styles.

## Course Training Model

The course training model allows students to undergo basic training, building upon primary knowledge and concepts to more advanced application and implementation. In this method, students will use the following toolset:

**Pre-assessment:** The pre-assessment is used to determine the student's prior knowledge of the subject matter. It will also identify a student's strengths and weaknesses, allowing the student to focus on the specific subject matter he/she needs to improve the most. Students should not necessarily expect a passing score on the pre-assessment as it is a test of prior knowledge.

**Video training sessions:** Each training course is divided into sessions or domains and lessons with topics and subtopics. LearnKey recommends incorporating all available external resources into your training, such as student workbooks, glossaries, course support files, and additional customized instructional material. These resources are located in the folder icon at the top of the page.

**Exercise labs:** Labs are interactive activities that simulate situations presented in the training videos. Step-by-step instructions and live demonstrations are provided.

**Post-assessment:** The post-assessment is used to determine the student's knowledge gained from interacting with the training. In taking the post-assessment, students should not consult the training or any other materials. A passing score is 80 percent or higher. If the individual does not pass the post-assessment the first time, LearnKey recommends incorporating external resources, such as the workbook and additional customized instructional material.

**Workbook:** The workbook has various activities, such as fill-in-the-blank worksheets, short answer questions, practice exam questions, and group and individual projects that allow the student to study and apply concepts presented in the training videos.

# Using This Workbook

This project workbook contains practice projects and exercises to reinforce the knowledge you have gained through the video portion of the **HTML and CSS** course. The purpose of this workbook is twofold. First, get you further prepared to pass the HTML and CSS exam, and second, to get you job-ready skills and increase your employability in the area of HTML5 and CSS.

The projects within this workbook follow the order of the video portion of this course. To save your answers in this workbook, you must first download a copy to your computer. You will not be able to save your answers in the web version. You can complete the workbook exercises as you go through each section of the course, complete several of these at the end of each domain, or complete them after viewing the entire course. The key is to go through these projects to strengthen your knowledge of this course.

Each project is based upon a specific video (or videos) in the course and specific test objectives. The materials you will need for this course include:

- LearnKey's **HTML and CSS** courseware.
- The course project files. All applicable project files are located in the support area where you downloaded this workbook.

## For Teachers

LearnKey is proud to provide extra support to instructors upon request. For your benefit as an instructor, we also provide an instructor support .zip file containing answer keys, completed versions of the workbook project files, and other teacher resources. This .zip file is available within your learning platform's admin portal.

### Notes

- Extra teacher notes, when applicable, are in the Project Details box within each exercise.
- Exam objectives are aligned with the course objectives listed in each project, and project file names correspond with these numbers.
- The Finished folder in each domain has reference versions of each project. These can help you grade projects.
- Short answers may vary but should be similar to those provided in this workbook.
- Teachers may consider asking students to add their initials, student ID, or other personal identifiers at the end of each saved project.
- Refer to your course representatives for further support.

We value your feedback about our courses. If you have any questions, comments, or concerns, please let us know by visiting <https://about.learnkey.com>.

# Skills Assessment

**Instructions:** Rate your skills on the following tasks from 1-5 (1 being needs improvement, 5 being excellent).

Skills	1	2	3	4	5
Construct markup that uses metadata elements.					
Construct well-formed page markup.					
Analyze and implement inline styles, internal (embedded) style sheets, and external style sheets.					
Construct and analyze rule sets.					
Construct and analyze markup to structure content and organize data.					
Construct and analyze markup that uses HTML5 semantic elements.					
Construct and analyze markup that implements navigation.					
Construct and analyze markup that uses form elements.					
Construct and analyze markup that displays images.					
Construct and analyze markup that plays video and audio.					
Construct and analyze styles that position content.					
Construct and analyze styles that format text.					
Construct and analyze styles that format backgrounds and borders.					
Construct and analyze styles that create a simple responsive layout.					
Construct well-formed HTML and CSS markup that conforms to industry best practices.					
Apply accessibility principles and evaluate content accessibility.					
Evaluate the structural integrity of HTML and CSS markup.					

# HTML and CSS Video Times

<b>Domain 1</b>	<b>Video Time</b>
Introduction	00:02:03
Markup with Metadata	00:08:13
Well-Formed Markup	00:11:23
<b>Total Time</b>	<b>00:21:39</b>

<b>Domain 2</b>	<b>Video Time</b>
Define Style Sheets	00:06:08
CSS Rule Sets	00:09:13
<b>Total Time</b>	<b>00:15:21</b>

<b>Domain 3</b>	<b>Video Time</b>
Structure Content Through Markup	00:17:10
Semantic HTML Elements	00:10:01
Navigation Elements	00:15:15
Form Elements	00:24:37
<b>Total Time</b>	<b>01:07:03</b>

<b>Domain 4</b>	<b>Video Time</b>
Image, SVG, and Canvas Elements	00:05:10
Video and Audio Formats	00:08:34
<b>Total Time</b>	<b>00:13:44</b>

<b>Domain 5</b>	<b>Video Time</b>
Position Text with CSS	00:28:43
Format Text with CSS	00:23:37
Backgrounds and Borders	00:14:18
Responsive Layouts	00:13:01
<b>Total Time</b>	<b>01:19:39</b>

<b>Domain 6</b>	<b>Video Time</b>
Construct HTML and CSS	00:10:56
Accessibility Principles and Evaluate Content	00:12:15
Accessibility	
Structural Integrity of HTML and CSS Markup	00:04:43
Course Conclusion	00:01:33
<b>Total Time</b>	<b>00:29:27</b>

# HTML and CSS

---

Domain 1

Presented By  
**LK LearnKey®**

# Fill-in-the-Blanks

**Instructions:** While watching Domain 1, fill in the missing words according to the information presented by the instructor. [References are found in the brackets.]

## Markup with Metadata

1. Out of the four main parts of an HTML document, the \_\_\_\_\_ tag contains information that describes a webpage but does not necessarily show on a webpage. [Script Tag]
2. The noscript tag lets a user know that the browser used does not support the \_\_\_\_\_ in the script tag code. [Noscript Tag]
3. Finish this code: <link rel="stylesheet" href="test.css" \_\_\_\_\_. [Link Tag]
4. \_\_\_\_\_ is often described as data defining data. [Meta Tags]

## Well-Formed Markup

5. Besides HTML5, other DOCTYPES you may find on a webpage are HTML 4, 4.0.1 or \_\_\_\_\_. [DOCTYPE Declaration and Html Tag]
6. The two most common pieces of information in the head tag, besides meta information, are scripts and \_\_\_\_\_. [Head Tag]
7. The body tag contains information that \_\_\_\_\_ on a webpage. [Body Tag]
8. A /> symbol at the end of a tag indicates a tag to where the closing portion does not need to be \_\_\_\_\_. [Closing Tags]
9. Entities are pieces of HTML code which display \_\_\_\_\_ and characters. [Entities]
10. The opening and closing tags for an HTML comment are \_\_\_\_\_ and \_\_\_\_\_. [Comments]

# Script and Noscript Tags

Script tags are tags which hold programming code in a language other than HTML, usually JavaScript. These are often found in the body of an HTML document but can be in the head section if the script is meant to run on a webpage but not display the information contained in the script.

The noscript tag is used to display text that will show if the browser being used to display a webpage does not support the scripting language used within a script tag.

## Purpose

Upon completing this project, you will be able to use script and noscript tags within an HTML document. Note that when you are called upon to open a file throughout this workbook, you will want to open it in your code editor and not a browser unless specifically told to do so.

## Steps for Completion

1. Open the **11a-scripts.html** file in the MusicStore1 student folder in Visual Studio or a similar code editor.
2. On the line below the words, Script Test, add an opening and closing script tag, indicating that JavaScript will be the script language.
3. In between the opening and closing script tag, add an alert indicating that the browser supports JavaScript.
4. Below the closing script tag, add an opening and closing noscript tag.
5. Inside of the noscript tags, add a message indicating the browser does not support JavaScript.
6. Save your changes.
7. Preview your webpage. You should see an alert message indicating that the browser supports JavaScript (so long as you are using a browser that does support it).

## Project Details

### Project file

11a-scripts.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 1

**Topic:** Markup with Metadata

**Subtopic:** Script Tag; Noscript Tag

### Objectives covered

#### 1 HTML Fundamentals

**1.1** Construct markup that uses metadata elements

**1.1.a** Script

**1.1.b** Noscript

# Link and Meta Tags

Link tags are used to link HTML files to files such as style sheets (CSS files) and script files such as JavaScript files. The link tag is not used for hyperlinks. The anchor tag (A tag) is used for hyperlinks.

Part of proper HTML markup in a document is using metadata, which is often described as data defining data. Meta tags have four functions: define character types, define webpage content, define how the webpage is viewed, and indicate if the language used for the page can be translated. Meta tags, as they do not display on the webpage itself, are always listed in the head section of a webpage.

## Purpose

Upon completing this project, you will be able to add link tags and meta tags to an HTML document.

## Steps for Completion

1. Open the **11c-scripts.html** file in the MusicStore1 student folder in your code editor.
2. On the line below the title tag, add a link tag that links to a stylesheet named **test.css**
3. On the line below the opening head tag, add a meta tag to define the charset as **utf-8**
4. On the next line, define a meta tag that uses the keywords **music, store, and instruments**
5. On the next line, define a meta tag for the viewport with the width set to the device width and the initial scale set to **1.0**
6. On the next line, define a meta tag that will not allow translation through Google.
7. Save your changes.
8. Preview your webpage. You should see a light green background (due to the style sheet) and a message indicating that either the browser supports JavaScript or it does not.
9. Close your browser.
10. Close your open files in your code editor, saving changes if prompted.

## Project Details

### Project file

11c-scripts.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 1

**Topic:** Markup with Metadata

**Subtopic:** Link Tag; Meta Tags

### Objectives covered

#### 1 HTML Fundamentals

**1.1** Construct markup that uses metadata elements

**1.1.c** Style

**1.1.d** Link

# HTML Fundamentals

HTML documents have four areas, the DOCTYPE, HTML, head, and body. The DOCTYPE indicates the type of document. The HTML area opens and closes the document's code. The head contains titles, links, and meta tags that do not need to show on the webpage. The title is the name of your webpage and is used by search engines.

The body contains the elements that will be visible on the webpage. All four areas of an HTML document have opening and closing syntax vital to the functionality of the document.

Regarding symbols and comments, symbols are HTML entities that start with an ampersand and end with a semicolon. For example, &copy; is an entity that represents a copyright. Comments start with a <!-- and end with a -->. Webpages ignore text inside comments. The comments often serve as explanations of parts of a webpage, notes to a fellow developer, or perhaps placeholders for future text.

## Purpose

Upon completing this project, you will be able to create and use proper syntax for all four areas of an HTML document. You will also be able to use entities and comments within a webpage.

## Steps for Completion

1. Create a new HTML document using your HTML editor. Name the file **index.html** and save it to your MusicStore1 student folder.
2. Add the four areas of an HTML document.
3. In the head section, add a title to name the webpage. Use: **THE Music Store** for the title. Close the title tag.
4. Add text to the body section of the webpage. Type: **Welcome to THE Music Store**
5. Below the welcome text, add a comment for text that will be added later. The comment should have the text, **Main text goes here**
6. To the left of the word THE in THE Music Store in the body, add two entities to represent two less than symbols.
7. To the right of the word, Store, in THE Music Store in the body, add two entities to represent two greater than symbols.
8. Save your changes.
9. Preview the webpage in a web browser. You should see a webpage with the following text: Welcome to <<**THE Music Store>>**
10. Close your web browser.
11. Close all open files within your editor.

## Project Details

### Project file

N/A

### Estimated completion time

10-15 minutes

### Video reference

#### Domain 1

**Topic:** Well-Formed Markup

**Subtopic:** DOCTYPE Declaration and Html Tag; Head Tag; Body Tag; Closing Tags; Entities; Comments

### Objectives covered

**1** HTML Fundamentals

**1.2** Construct well-formed page markup

**1.2.a** DOCTYPE declaration

**1.2.b** Html

**1.2.c** Head

**1.2.d** Body

**1.2.e** Proper syntax

**1.2.f** Closing tags and commonly used symbols

# HTML and CSS

---

Domain 2

Presented By  
**LK LearnKey®**

# Fill-in-the-Blanks

**Instructions:** While watching Domain 2, fill in the missing words according to the information presented by the instructor. [References are found in the brackets.]

## Define Style Sheets

1. An \_\_\_\_\_ sheet is created for and applied to a single webpage. [Inline, Internal, and External Styles]
2. The order of precedence in style sheets is that \_\_\_\_\_ styles take precedence over \_\_\_\_\_ styles which take precedence over \_\_\_\_\_ styles. [Style Sheet Precedence]

## CSS Rule Sets

3. Style elements are placed inside of \_\_\_\_\_ brackets. [Valid Syntax]
4. A class is a style that can be applied to any \_\_\_\_\_ at any time. [Apply Classes]
5. An id is a selector comparable to a class but it should only be used once per \_\_\_\_\_. [Apply IDs]
6. A pseudo class is a class that is specific to a \_\_\_\_\_. [Apply Pseudo Classes]

# Creating Style Sheets

Style sheets, better known as Cascading Style Sheets (CSS), are used to control formatting within a webpage and, as is the case with external style sheets, part, or all of a website. Style sheets consist of one or more combinations of an HTML tag or class paired with one or more formatting properties. There are three types of style sheets one can use in web design:

- External: a style sheet that is created as a separate file and then applied to some or all pages in a website.
- Internal: a style sheet that is created and applied to a single webpage.
- Inline: a style applied directly to an HTML tag on a webpage.

The cascading part of Cascading Style Sheets works in this manner: inline styles will take precedence over internal styles and internal styles will take precedence over external styles. For example, an inline style for a single paragraph (p) tag could set the color to navy. An internal style sheet could set paragraphs to maroon. An external style sheet could set paragraphs to gray. In this instance, all paragraphs except the navy one would be maroon.

## Purpose

Upon completing this project, you will be able to create a style sheet and apply it to a webpage.

## Steps for Completion

1. In your code editor, create a new style sheet file, naming the file **main.css**
2. If necessary, add the body tag with a set of curly brackets to the style sheet.
3. Inside the curly brackets, add the font-family property and set the font families to Segoe UI, Tahoma, and sans-serif.
4. Save your changes.
5. Open the **21a-index.html** file in the MusicStore4 student folder.
6. In the head section, create a link to the stylesheet you have created (main.css).
7. Save your changes.
8. Preview the webpage in a browser. You should see the text in either Segoe UI or Tahoma for a font or a matching sans-serif font.

## Project Details

### Project file

21a-index.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 2

**Topic:** Define Style Sheets

**Subtopic:** Inline, Internal, and External Styles

### Objectives covered

#### 2 CSS Fundamentals

**2.1** Analyze and implement inline styles, internal (embedded) style sheets, and external style sheets

**2.1.a** When to use inline styles, internal (embedded) style sheets, or external style sheets

# Style Sheet Precedence

As mentioned in the previous project, inline styles will take precedence over internal styles and internal styles will take precedence over external styles. For the exam, make sure you understand this cascading order of styles sheets and be able to identify what format will apply to HTML code depending upon the style sheet situation.

## Purpose

Upon completing this project, you will better understand the cascading aspect of Cascading Style Sheets.

## Steps for Completion

1. Open the **21b-main.css** file in the MusicStore2 student folder.
2. Add the paragraph tag as a tag to be styled in the style sheet.
3. For the paragraph tag, set the color to navy.
4. Open the **21b-index.html** and **21b-books.html** files from the MusicStore4 student folder.
5. Preview both pages in a web browser. The paragraph text on both pages should be navy blue.
6. Return to the **21b-books.html** file in the code editor.
7. Add an internal style sheet to the page.
8. In the internal style sheet, add a selector for the paragraph tag, setting the color to maroon.
9. Save your changes.
10. Preview both the **21b-books.html** and **21b-index.html** pages in a web browser. The paragraph text should be maroon in the **21b-books.html** file and navy blue in the **21b-index.html** file.
11. Return to the **21b-books.html** file in the code editor.
12. Add an inline style to the paragraph, Look at all of these great books!
13. In the inline style, set the color to fuchsia.
14. Save your changes.
15. Preview the webpage in a web browser. The paragraph text color should be maroon, with the exception of the text, Look at all of these great books!, which should be fuchsia.

## Project Details

### Project file

21b-main.css  
21b-books.html  
21b-index.html

### Estimated completion time

15 minutes

### Video reference

#### Domain 2

**Topic:** Define Style Sheets

**Subtopic:** Style Sheet Precedence

### Objectives covered

#### 2 CSS Fundamentals

**2.1** Analyze and implement inline styles, internal (embedded) style sheets, and external style sheets

**2.1.b** Precedence when using a combination of inline styles and style sheets

# Valid Syntax for Rule Sets

A rule set is a combination of a selector which can be a tag, class, or id, and a property and value pair set in a style sheet. For each rule set, the selector is listed first and the property and value pairs are all contained within a set of curly brackets. The property is listed, a colon is added, the value is added, and then a semicolon finishes the pair.

Properties never contain spaces. Where one would think there is a space, a dash is used. For example, font family has a property name of font-family.

## Purpose

Upon completing this project, you will better understand CSS rule sets and be able to apply a rule set to a CSS style sheet.

## Steps for Completion

1. Open the **22a-main.css** file in the MusicStore2 student folder.
2. Within the body rule set, add a background color property with the value set to **#fab66b**
3. Save your changes.
4. Open the **22a-index.html** file in the MusicStore2 folder.
5. Preview the webpage in a browser. The webpage background should be a peach color.

## Project Details

### Project file

22a-main.css

22a-index.html

### Estimated completion time

5 minutes

### Video reference

#### Domain 2

**Topic:** CSS Rule Sets

**Subtopic:** Valid Syntax

### Objectives covered

2 CSS Fundamentals

**2.2** Construct and analyze rule sets

**2.2.a** Valid syntax for the CSS rule set

# Apply Classes

A CSS class is a style that can be applied to any tag at any time. A class is created in a CSS style sheet but applied to an HTML tag. Classes all start with a dot and can have any name, so long as the name is valid. For example, a class name cannot have spaces.

## Purpose

Upon completing this project, you will be able to create a CSS class in a CSS style sheet and apply the class to an HTML tag.

## Steps for Completion

1. Open the **22b-main.css** file in the MusicStore2 student folder.
2. Below the current rule sets, add a class named **sale**
3. Within the sale class, add a rule set to set the font weight to **bold**
4. Within the sale class, add a rule set to set the background color to **aqua**
5. Save your changes.
6. Open the **22b-books.html** file in the MusicStore2 student folder.
7. Add an opening and closing span tag around the words, Special sale!, near the end of the page.
8. Apply the sale class to the open span tag.
9. Save your changes.
10. Preview the webpage in a browser. You should see the words, Special sale!, highlighted in aqua at the bottom of the webpage.

## Project Details

### Project file

22b-main.css

22b-books.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 2

**Topic:** CSS Rule Sets

**Subtopic:** Apply Classes

### Objectives covered

#### 2 CSS Fundamentals

**2.2** Construct and analyze rule sets

**2.2.b** Selectors, including class, id, elements, and pseudo-class

# Apply IDs

In HTML, ids have multiple purposes. First, any scripts used to enhance an HTML element will refer to the element by ID. Secondly, IDs can be added to CSS files and then used in a similar fashion as are classes. However, an ID can only be used once per webpage in this scenario. Classes can be used an unlimited number of times per webpage.

Whereas classes start with a dot, IDs start with a pound symbol (#).

## Purpose

Upon completing this project, you will be able to create a CSS ID and apply the ID to an HTML tag.

## Steps for Completion

1. Open the **22b2-main.css** file in the MusicStore2 student folder.
2. Below the current list of rule sets, add an id selector with the name, **sidenote**
3. Within the selector, add a rule set that sets the background color to **#dbd9d9**
4. Save your changes.
5. Open the **22b2-spotlight.html** file in the MusicStore2 student folder.
6. Locate the aside tag that starts with the text, Come in and sign up for a free kazoo demonstration.
7. Add the id attribute of **sidenote** to the aside tag.
8. Save your changes.
9. Preview the webpage in a browser. You should see a gray background on the aside text.

## Project Details

### Project file

22b2-main.css

22b2-spotlight.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 2

**Topic:** CSS Rule Sets

**Subtopic:** Apply IDs

### Objectives covered

2 CSS Fundamentals

**2.2** Construct and analyze rule sets

**2.2.b** Selectors, including class, id, elements, and pseudo-class

# Apply Pseudo Classes

A CSS pseudo class applies specifically to an HTML tag. A tag can have multiple pseudo classes that can be applied to that specific tag. For example, the anchor tag (a) has multiple pseudo classes, with the link, visited, and hover pseudo classes as some examples. A pseudo class is added, after a colon, to a tag in a rule set.

## Purpose

Upon completing this project, you will be able to create a CSS pseudo class and apply the pseudo class to a specific HTML tag.

## Steps for Completion

1. Open the **22b3-main.css** file in the MusicStore2 student folder.
2. Add a rule set for the a tag.
3. Within the rule set, set the text decoration property to **none**
4. Add a rule set for the a tag with the hover pseudo-class.
5. Within the rule set, set the text decoration property to underline and the font weight to **bold**
6. Save your changes.
7. Open the **22b3-books.html** file in the MusicStore4 student folder.
8. Preview the webpage in a browser. The hyperlink will not be underlined on the webpage.
9. Hover your mouse over the hyperlink. It should turn bold and become underlined while the mouse is over the hyperlink and should return to not being bold and not underlined when the mouse moves away from the link.

## Project Details

### Project file

22b3-main.css

22b3-books.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 2

**Topic:** CSS Rule Sets

**Subtopic:** Apply Pseudo Classes

### Objectives covered

#### 2 CSS Fundamentals

**2.2** Construct and analyze rule sets

**2.2.b** Selectors, including class, id, elements, and pseudo-class

# HTML and CSS

---

Domain 3

Presented By  
**LK LearnKey®**

# Fill-in-the-Blanks

**Instructions:** While watching Domain 3, fill in the missing words according to the information presented by the instructor. [References are found in the brackets.]

## Structure Content Through Markup

1. \_\_\_\_\_ offers more flexibility in laying out webpages than do tables. [Table Tags]
2. HTML has \_\_\_\_\_ levels of heading tags. [Heading Tags]
3. The P tag is used to create \_\_\_\_\_. [P Tag]
4. A br tag is the equivalent of a \_\_\_\_\_ return in Word or similar programs. [Br Tag]
5. The hr tag provides a \_\_\_\_\_ used to help break up sections of a webpage. [Hr Tag]
6. To define and format block level elements, use a \_\_\_\_\_ tag. Use the \_\_\_\_\_ tag to define and format inline elements. [Span Tag]
7. Unordered lists are lists with \_\_\_\_\_ points. [Ul and Li Tags]
8. Ordered lists are lists with \_\_\_\_\_ points. [Ol and Li Tags]

## Semantic HTML Elements

9. Semantic tags are used to help \_\_\_\_\_ webpages. [Header Tag]
10. The Nav tag is used to help spot navigation \_\_\_\_\_ on webpages. [Nav Tag]
11. Section tags help organize \_\_\_\_\_ webpages. [Section Tag]
12. An \_\_\_\_\_ tag helps to specify a self-contained piece of content, while an \_\_\_\_\_ tag helps mark off a side story or related secondary content. [Article and Aside Tags]
13. A footer tag helps one be consistent with content at the \_\_\_\_\_ area of a webpage. [Footer Tag]
14. A \_\_\_\_\_ tag surrounds a \_\_\_\_\_ tag, which contains content that can be hidden or shown at will. [Detail and Summary Tags]
15. A figure tag is used in conjunction with a \_\_\_\_\_ tag to line up a caption with an image. [Figure Tag]
16. A caption tag is usually used on \_\_\_\_\_ rather than on images. [Caption Tag]

## Navigation Elements

17. Images should have \_\_\_\_\_ text to display in case an image does not render properly. [Image Links]
18. The A tag is primarily used for \_\_\_\_\_. [The A Tag]
19. By setting the target property of a hyperlink to \_\_\_\_\_, the browser is instructed to open that link in a new tab or window. [Targets]
20. Bookmarks allow one to navigate to specific \_\_\_\_\_ within a webpage, often a longer webpage. [Bookmarks]
21. A \_\_\_\_\_ link points to a file or page within a website, while an \_\_\_\_\_ link points to a page on a different website. [Relative vs. Absolute Links]

22. To indicate that a link is in a folder above the current folder in a hierarchical path, two \_\_\_\_\_ are needed in the code for the link. [Folder Hierarchy Navigation]
23. The map tag creates an image map that links to an image, while the \_\_\_\_\_ tag defines the different areas within the image map. [Map]
24. The \_\_\_\_\_ attribute contains the hyperlink target for a selectable area. [Area]

### Form Elements

25. A form element's \_\_\_\_\_ attribute directs a browser on how to send form data to a server or retrieve data from a server, while its \_\_\_\_\_ attribute indicates which URL form data is sent to upon submission. [Action and Method Attributes]
26. A \_\_\_\_\_ request takes submitted data and inserts it into a database, while a \_\_\_\_\_ request modifies data on a server through user input. [Submission Methods]
27. The \_\_\_\_\_ attribute allows one to use a tab key to navigate through form fields. [Accessibility]
28. In a form, a \_\_\_\_\_ element generates a drop-down list, while \_\_\_\_\_ tags determine the available items on that list. [Input Types and Restrictions]
29. The label tag also improves the experience for users utilizing screen \_\_\_\_\_. [The Significance of Labels]
30. The select tag creates a drop-down \_\_\_\_\_ full of options. [Select Tag]
31. The textarea tag allows one to add a field with multiple \_\_\_\_\_ of text to a form. [Textarea Tag]
32. The input tag is generally used instead of a button tag when one is trying to use a button to \_\_\_\_\_ or reset a form. [Button Tag]
33. The output tag allows for displaying a calculation depending upon what one \_\_\_\_\_ into a form. [Output Tag]
34. \_\_\_\_\_ tags help to group option tags and make them more readable. [Option Tag]
35. A datalist tag is used to \_\_\_\_\_ a list of values into a form field. [Datalist Tag]
36. A fieldset tag is used to \_\_\_\_\_ fields in a form. [Fieldset Tag]

# HTML Tables

Tables are used to organize information in rows and columns. Tables used to be used to lay out entire pages (and they still do, sometimes), but with the advancements in Cascading Style Sheets (CSS) for layouts for overall pages, tables are best served for grid-like information.

For creating tables, the table tag creates the table, the tr tag indicates a table row, and the td tag indicates a table cell.

## Purpose

Upon completing this project, you will be able to create a table and use different attributes to style the table.

## Steps for Completion

1. Open the **31a-hours.html** file from your MusicStore3 student folder.
2. Add a table below the text, Hours of operation. The table should have five rows and two columns.
3. Save your changes.
4. Open the **31a-hours2.html** file from your MusicStore2 student folder.
5. Add a 1-point border to the table.
6. Change the first row from being a normal table row to that of a table header.
7. Change the last row to the table to contain just one column instead of two.
8. Save your changes.
9. Preview the webpage in a browser. Your table should look like this:

Days	Hours
M-F	10 a.m. to 7 p.m.
Sat	9 a.m. to 6 p.m.
Sun	11 a.m. to 5 p.m.
Hours vary on holidays	

## Project Details

### Project file

31a-hours.html  
31a-hours2.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 3

**Topic:** Structure Content Through Markup

**Subtopic:** Table Tags; Table Attributes

### Objectives covered

#### 3 Document Structure using HTML

**3.1** Construct and analyze markup to structure content and organize data

**3.1.a** Table tags (table, tr, th, td)

# HTML Headings, Paragraphs, and Line Breaks

Three HTML tags that are used very heavily in the basic content formation of webpages are headings, paragraph (P), and line break (BR) tags. Heading tags have six levels (H1 through H6) with H1 being the largest heading and H6 the smallest. P tags are the equivalent of hard returns in a Word or similar document. BR tags are the equivalent of soft returns in a Word or similar document.

Most HTML tags have both an opening and a closing tag. However, if the content within a tag has a defined ending, a closing tag is not needed. For example, paragraphs have a start and an end. But line breaks are just that, line breaks. Once the line break is set, it is done. Thus, a BR tag does not need a closing tag. A / after the BR indicates the closing portion of it.

## Purpose

Upon completing this project, you will be able to use heading, paragraph, and line break tags within a webpage.

## Steps for Completion

1. Open the **31b-headings.html** file from the MusicStore3 student folder.
2. With the six lines of text for heading examples, add an opening heading tag (H1 through H6) to each line, matching the heading tag with the text in the line.
3. Save your changes.
4. Preview your webpage in a web browser. Your webpage should look like this:
5. Open the **webpagetext.txt** file from the MusicStore2 student folder.
6. Copy the document content.
7. Open the **31b-index.html** file from the MusicStore2 student folder.
8. Paste the copied text in between the two comments on the webpage.
9. Surround the first two blocks of text you pasted with an opening and closing paragraph tag.
10. Add a line break before the text, We look forward to seeing you!
11. Save your changes.
12. Preview your webpage in a web browser.

## Project Details

### Project file

31b-headings.html

31b-index.html

webpagetext.txt

### Estimated completion time

10 minutes

### Video reference

#### Domain 3

**Topic:** Structure Content Through Markup

**Subtopic:** Heading Tags; P Tag; Br Tag

### Objectives covered

#### 3 Document Structure using HTML

**3.1** Construct and analyze markup to structure content and organize data

**3.1.b** H1-h6

**3.1.c** P

**3.1.d** Br

# This is Heading 1

## This is Heading 2

### This is Heading 3

#### This is Heading 4

##### This is Heading 5

This is Heading 6

This is a normal paragraph

# Hr, Div, and Span Tags

Once the basic structure of an HTML document is in place, you will want to organize and separate content further. One way to do this is to add horizontal rules (Hr tags) in between sections of a webpage.

Two other options available are Div and Span tags. A Div tag isolates a block of text and allows for formatting specifically on that block. A Span tag allows for inline formatting around text within a paragraph but does not isolate the text from a paragraph.

## Purpose

Upon completing this project, you will be able to use the Hr, Div, and Span tags in an HTML document.

## Steps for Completion

1. Open the **31e-index.html** file from the MusicStore3 student folder.
2. On the line after the text, We look forward to seeing you!, add a horizontal rule.
3. Surround the three lines of text after the horizontal rule with an opening and closing Div tag.
4. Center align the text inside of the Div tag.
5. Add a Span tag before the text, THE Music Store, below the navigation area.
6. Close the Span tag after the store name.
7. Make the text color for the Span tag dark green.
8. Save your changes.
9. Preview the webpage in a web browser. There should be a line above the footer text and the words, THE Music Store, should be dark green.

## Project Details

### Project file

31e-index.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 3

**Topic:** Structure Content Through Markup

**Subtopic:** Hr Tag; Div Tag; Span Tag

### Objectives covered

#### 3 Document Structure using HTML

**3.1** Construct and analyze markup to structure content and organize data

**3.1.e** Hr

**3.1.f** Div

**3.1.g** Span

# HTML List Tags

An unordered list (Ul tag) is a bulleted list that does not require steps to be done in any specific order, while an ordered list (Ol tag), in most cases, is a numbered list with steps that must be done in a specific order.

Both unordered and ordered lists contain list item tags (Li tags), which serve as bullet points for unordered lists and numbered items for ordered lists.

## Purpose

Upon completing this project, you will be able to create unordered and ordered lists.

## Steps for Completion

1. Open the **31h-index.html** file from the MusicStore3 student folder.
2. Turn the items Guitar, Piano, and Drums into an unordered list.
3. Create a list item for each instrument in the list.
4. Change the bullet created by the unordered list to a square style.
5. Preview the webpage in a browser. You should see a square bulleted list of instruments.
6. Return to your code window.
7. Add an ordered list opening tag before the text, Join our music club, found below the list of instruments you just created.
8. Close the ordered list after the text, Sign up for a course.
9. Make each listed step its own item within the ordered list.
10. Make the ordered list show the steps as capital letters.
11. Save your changes.
12. Preview the webpage in a web browser. Your page should resemble this:

Welcome to <<THE Music Store>>

Here at THE Music Store, we have the finest selection of instruments and accessories. Whether you want to be the proud new owner of a guitar, piano, drums, a ukulele, a trombone, or even a kazoo, you have come to the right place.

In addition to having a wide variety of instruments, we offer a whole host of music courses, to help you learn how to play your new instrument well. After all, we are sure you want to make good noise with your instruments.

And, we have a great selection of music books, sheet music, and other music accessories. We currently offer courses for the following instruments:

- Guitar
- Piano
- Drums

Plus, one Saturday a month we have a free introductory course on an instrument. Signing up is easy. All you need to do is:

- A. Join our music club
- B. Sign up to receive emails on upcoming events
- C. Sign up for a course



We are open seven days a week. Our hours do vary from season to season, so check back with us frequently.  
We look forward to seeing you!

[Home](#) | [Books](#) | [Contact Us](#) | [Spotlight](#) ©2017, THE Music Store 1000 Broadway, Salt Lake City, UT 84103

13. Close all open files, both in your browser and your code editor.

## Project Details

### Project file

31h-Index.html

### Estimated completion time

15 minutes

### Video reference

#### Domain 3

**Topic:** Structure Content Through Markup

**Subtopic:** Ul and Li Tags; Ol and Li Tags

### Objectives covered

#### 3 Document Structure using HTML

**3.1** Construct and analyze markup to structure content and organize data

**3.1.h** Ul

**3.1.i** Ol

**3.1.j** Li

# Semantic Tags

A semantic tag is used to organize a webpage. Semantic tags are a quick reference to let both developers and search engines know where the header, navigation, text sections, articles with self-containing text, aside story, and footer areas are in an HTML document.

Semantic tags do not format content directly. Any formatting on semantic tags is done through Cascading Style Sheets (CSS).

## Purpose

Upon completing this project, you will better understand what semantic tags are used for and where to use them in an HTML document.

## Steps for Completion

1. Open the **32a-spotlight.html** file in the MusicStore3 student folder.
2. Preview the webpage. Make a mental note of what you see.
3. Return to your code editor.
4. Add an opening header tag above the text, This month's instrument:  
The Kazoo.
5. Close the header tag below the text, The instrument to where you make  
the noise.
6. Surround the text, Home | Books | Contact Us, with an opening and closing nav tag.
7. Surround the text in between the header and nav areas you added with an opening and closing section tag.
8. Just after the opening section tag, add an opening article tag.
9. Close the article tag just before the paragraph, More on kazoos.
10. Add an opening and closing aside tag around the paragraph that starts with, Come in and sign up for a free kazoo demonstration.
11. Surround the navigation area you semantically marked earlier plus the text with the copyright and the store address with an opening and closing footer tag.
12. Save your changes.
13. Return to your browser.
14. Refresh the webpage. You should not see any changes as the tags you added in this project are all semantic tags.

## Project Details

### Project file

32a-spotlight.html

### Estimated completion time

15 minutes

### Video reference

#### Domain 3

**Topic:** Semantic HTML Elements

**Subtopic:** Header Tag; Nav Tag;  
Section Tag; Article and Aside Tags,  
Footer Tag

### Objectives covered

**3** Document Structure using HTML

**3.2** Construct and analyze markup  
that uses HTML5 semantic elements

**3.2.a** Semantic tags (header, nav,  
section, article, aside, footer,  
details, summary, figure, caption)

# Details and Summary Tags

Two semantic tags that have some interaction of supported web browsers are the details and summary tags. A details tag, wrapped around an area of text, makes the area expandable and collapsible. A summary tag, contained within a details tag, should be wrapped around the text to show for the expandable and collapsible area.

## Purpose

Upon completing this project, you will better understand how to use the details and summary tags in an HTML document.

## Steps for Completion

1. Open the **32g-books.html** file in the MusicStore3 student folder.
2. Add an opening details tag above the paragraph with the text, Look at all of these great books!
3. Close the details tag after the image tag for the sheet music image.
4. On the line after the opening details tag (and before the next paragraph tag), add the text, **Show or hide these books**
5. Add an opening and closing summary tag around the text you just typed.
6. Save your changes.
7. Preview the webpage in a web browser. You should be able to expand and contract the section for showing and hiding books.

## Project Details

### Project file

32g-books.html

### Estimated completion time

5-10 minutes

### Video reference

#### Domain 3

**Topic:** Semantic HTML Elements

**Subtopic:** Details and Summary Tags

### Objectives covered

#### 3 Document Structure using HTML

**3.2** Construct and analyze markup that uses HTML5 semantic elements

**3.2.a** Semantic tags (header, nav, section, article, aside, footer, details, summary, figure, caption)

# Figure and Caption Tags

A figure tag is used in conjunction with the figcaption tag to mark up an image on a webpage. The figcaption tag allows for adding a caption to an image and for lining up the caption with the image.

The caption tag is used on tables. This tag allows one to place description text for a table in close proximity with a table.

## Purpose

Upon completing this project, you will better understand how to use the figure, figcaption, and caption tags in an HTML document.

## Steps for Completion

1. Open the **32i-spotlight.html** file in the MusicStore3 student folder.
2. Add a figure tag above the image of the Kazoo.
3. Close the figure tag after the image.
4. On the line after the image, add an opening and closing figcaption tag.
5. Within the figcaption tag, add the text, **The Mighty Kazoos**
6. Save your changes.
7. Preview your webpage in a web browser. You should see a caption below the image.
8. Return to your code editor.
9. Open the **32j-hours2.html** file in the MusicStore2 student folder.
10. Add the text, **Current Summer Hours** in between the opening table tag and the first tr tag.
11. Surround the text you just added with an opening and closing caption tag.
12. Surround the text, Hours of operation, with an opening and closing H2 tag.
13. Save your changes.
14. Preview the webpage in a browser. You should see the following:
15. Close all open files in both your browser and your code editor.

**Project Details**

**Project file**  
 32i-spotlight.html  
 32j-hours2.html

**Estimated completion time**  
 15 minutes

**Video reference**

**Domain 3**

**Topic:** Semantic HTML Elements  
**Subtopic:** Figure Tag; Caption Tag

**Objectives covered**

**3 Document Structure using HTML**

**3.2** Construct and analyze markup that uses HTML5 semantic elements

**3.2.a** Semantic tags (header, nav, section, article, aside, footer, details, summary, figure, caption)

## Hours of operation

### Current Summer Hours

Days	Hours
M-F	10 a.m. to 7 p.m.
Sat	9 a.m. to 6 p.m.
Sun	11 a.m. to 5 p.m.
Hours vary on holidays	

# Image Links

Elements, such as images, are not embedded within a webpage. They are linked from an external source to a webpage. With images, the img tag is used to indicate an image on a webpage and the source (src) attribute is used to define the location of the image.

If this image is in the same folder as the page itself, the file name is all that is needed. Otherwise, the path to the file needs to be defined. For example, if the image is on the same website but in an images folder adjacent to the file, the path needs to have the folder name, a backslash, and then the file.

## Purpose

Upon completing this project, you will be able to display an image within an HTML document through linking to it.

## Steps for Completion

1. Open the **33a-books.html** file in the MusicStore3 student folder.
2. Add a link to the image **musicbooks.jpg** on the line below the text, Look at all of these great books.
3. Add the alt text, **Music books**, to the image tag.
4. Set the image width to **920**
5. Set the image height to **690**
6. Save your changes.
7. Preview the webpage in a web browser. An image should display on the page.

## Project Details

### Project file

33a-books.html

### Estimated completion time

5-10 minutes

### Video reference

#### Domain 3

**Topic:** Navigation Elements

**Subtopic:** Image Links

### Objectives covered

**3** Document Structure using HTML

**3.3** Construct and analyze markup that implements navigation

# Hyperlinks

A hyperlink will move the user to another webpage when it is clicked. A hyperlink can be internal or external to a website. A target can control how a hyperlink opens, with the most common example being of a target set to \_blank so that the hyperlink can open in a blank browser tab. Hyperlinks are created using the anchor (a) tag and the hyperlink reference (href) attribute.

## Purpose

Upon completing this project, you will be able to add a hyperlink to an HTML document and set a target for a hyperlink.

## Steps for Completion

1. Open the **33b-index.html** file in the MusicStore3 student folder.
2. Using the text, Spotlight, in the navigation area at the bottom of the page, add a hyperlink to the **33b-spotlight.html** file.
3. Save your changes.
4. Preview the webpage in a web browser.
5. Click the Spotlight link. You should be taken to the 33b-spotlight page.
6. Return to your code editor.
7. Adjust the link target to have the spotlight page open in a new tab.
8. Save your changes.
9. Preview the webpage in a web browser.
10. Test the Spotlight hyperlink. It should open in a new tab.

## Project Details

### Project file

33b-index.html  
33b-spotlight.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 3

**Topic:** Navigation Elements  
**Subtopic:** The A Tag; Targets

### Objectives covered

**3** Document Structure using HTML  
**3.3** Construct and analyze markup that implements navigation  
**3.3.a** Target  
**3.3.b** Anchor (a href)

# Bookmarks

A bookmark is a type of hyperlink that navigates to a specific location on a webpage. This can be a location on the same webpage as the hyperlink or to a specific location on a different page. The location for the bookmark must be defined. This is done using the anchor () tag and the name attribute. Then, the hyperlink uses the # symbol in front of

## Purpose

Upon completing this project, you will be able to add a bookmark to an HTML document. You will also be able to link to the bookmark.

## Steps for Completion

1. Open the **33c-books.html** file in the MusicStore3 student folder.
2. Create a bookmark with the name, **sale**, just above the paragraph starting with the words, Special sale, near the bottom of the page.
3. Near the top of the page, add a hyperlink around the text, This month's sale, with the hyperlink linking to the bookmark you just created.
4. Save your changes.
5. Preview the webpage in a web browser.
6. Select the link on the text, This month's sale. You should be taken to the bottom of the page.

## Project Details

### Project file

33c-books.html

### Estimated completion time

5 minutes

### Video reference

#### Domain 3

**Topic:** Navigation Elements

**Subtopic:** Bookmarks

### Objectives covered

#### 3 Document Structure using HTML

**3.3** Construct and analyze markup that implements navigation

**3.3.c** Bookmark

# Relative and Absolute Links

Hyperlinks are relative or absolute depending on where the link is located. A relative link is a hyperlink located within the same website from the HTML document from where the link originates. An absolute link is a hyperlink in which the destination is not in the same website of the webpage from which the link originates. Because of this, to keep users on your webpage you will want to add a target so that the new webpage will open in a new tab.

As the hyperlink you created earlier in this domain was a relative link to a webpage within the same website, this project will focus more on absolute links.

## Purpose

Upon completing this project, you will be able to determine if a link is relative or absolute and use both links in an HTML document.

## Steps for Completion

1. Open the **33d-spotlight.html** file in the MusicStore3 student folder.
2. Add an absolute link to the text: More on kazoos. Use the website: <https://en.wikipedia.org/wiki/kazoo> for the complete absolute link address.
3. Add a target to the open absolute link so the new webpage will open in a new tab.
4. Save your changes.
5. Preview the webpage in a web browser.
6. Click the hyperlink on the text, More on Kazoos. You should see the Wikipedia article appear on its own tab.

## Project Details

### Project file

33d-spotlight.html

### Estimated completion time

5 minutes

### Video reference

#### Domain 3

**Topic:** Navigation Elements

**Subtopic:** Relative vs. Absolute  
Links

### Objectives covered

**3** Document Structure using HTML

**3.3** Construct and analyze markup  
that implements navigation

**3.3.d** Relative vs. absolute links

# Navigating Folder Hierarchies

Relative hyperlinks are often located within folders different from the pages in which they originate. For simple folder hierarchies, the folder name needs to precede the webpage name in a hyperlink if the folder is below the webpage in the hierarchy of the website. However, if the folder is above the webpage in the hierarchy, two dots and a backslash need to be placed in front of the folder name.

## Purpose

Upon completing this project, you will better understand the commands to move up and down a folder path and be able to use the up or down commands on an HTML document.

## Steps for Completion

1. Open the **33e-links.html** file in the TestPages folder within the MusicStore3 student folder.
2. Add a hyperlink around the text, Link to the home page. The hyperlink reference should navigate up a folder path and then over to the index.html page.
3. Save your changes.
4. Preview the webpage in a web browser.
5. Click the link you created in this project. It should navigate you to the home page.
6. Close all open files in both your browser and your editor.

## Project Details

### Project file

33e-links.html

### Estimated completion time

5 minutes

### Video reference

#### Domain 3

**Topic:** Navigation Elements

**Subtopic:** Folder Hierarchy  
Navigation

### Objectives covered

**3** Document Structure using HTML

**3.3** Construct and analyze markup that implements navigation

**3.3.e** Navigating simple folder hierarchies

# Map and Area Tags

One may want to create select areas or hyperlinks within an image on a webpage. One can do so by using the map tag in conjunction with the area tag. The map tag creates an image map that links to an image, while the area tag defines the different areas within the image map. A map tag requires that the image on a webpage includes a selectable area, otherwise no change is applied to the webpage.

## Purpose

Upon completing this project, you will better understand how to use map and area tags in conjunction with image maps.

## Steps for Completion

1. Open the **books2.html** file in the MusicStore3 student folder.
2. There is an empty map tag after the image tag on line 20. What is the name being used for the image's usemap attribute?
  - a. \_\_\_\_\_
3. Preview the webpage in a web browser.
4. Open the **books3.html** file in the MusicStore2 student folder.
5. There is an area tag on line 21, inside the map tag. This area tag currently has four attributes. What are the attributes?
  - a. \_\_\_\_\_
6. Preview the webpage in a web browser.
7. Select the *Electronic and Computer Music* book spine to ensure the selectable area works.
8. Add a target attribute "**\_blank**" within the area tag.
9. Save your changes.
10. Preview the webpage in a web browser.
11. Select the *Electronic and Computer Music* book spine to ensure that the selectable area opens the link in a new tab.

## Project Details

### Project file

books2.html  
books3.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 3

**Topic:** Navigation Elements  
**Subtopic:** Map; Area

### Objectives covered

**3** Document Structure using HTML

**3.3** Construct and analyze markup that implements navigation

**3.3.f** Map

**3.3.g** Area

# Action and Method Attributes

This is the first of several projects in a row that will focus on form building. The first two concepts to cover are action attributes and method attributes. An action attribute references webpages or scripts that process information to a database. The method attribute will determine what type of action the form will take when it is submitted.

There are four types of submission methods:

- GET: A request to retrieve data
- POST: A request to modify data on a server
- PUT: A request to add data to a database
- DELETE: A request to remove information from a database

These attributes are part of the main tag of a form, which is the form tag.

## Purpose

Upon completing this project, you will be able to create a form and add an action attribute and a method attribute to the form.

## Steps for Completion

1. Open the **34a-contactus.html** file in the MusicStore3 student folder.
2. Add an opening and closing form tag on the line below the text: insert form here.
3. Add an action attribute in the open form tag to point the thankyou.html webpage.
4. Add a method attribute that will post information to a server.
5. Save your changes.
6. Which submission method is used to add information to a database?
  - a. \_\_\_\_\_
7. Which submission method adds the words used for submission to the end of a URL?
  - a. \_\_\_\_\_

## Project Details

### Project file

34a-contactus.html

### Estimated completion time

5-10 minutes

### Video reference

#### Domain 3

**Topic:** Form Elements

**Subtopic:** Action and Method Attributes; Submission Methods

### Objectives covered

**3** Document Structure using HTML

**3.4** Construct and analyze markup that uses form elements

**3.4.a** Form attributes

**3.4.b** Action

**3.4.c** Method

**3.4.d** Submission methods

# Accessibility

An input form creates a webpage that information can be added to. Input forms are used for questionnaires, surveys, or applications. The access keys and tab index allow the user to move the cursor between input areas, on a form, using a keyboard rather than a mouse. This can help websites become more accessible-friendly for people who need assistance in navigating a website.

## Purpose

Upon completing this project, you will be able to create an input form using access keys and the tab index to make the HTML document more accessible.

## Steps for Completion

1. Open the **34d-contactus.html** file in the MusicStore3 student folder.
2. To the input tag for the first name, add an accesskey attribute with the letter, f, and a tab index with a setting of **1**
3. To the input tag for the last name, add an accesskey attribute with the letter, l, and a tab index with a setting of **2**
4. To the input tag for the email address, add an accesskey attribute with the letter, e, and a tab index with a setting of **3**
5. Save your changes.
6. Preview the webpage in a browser.
7. Use the Tab key on the keyboard to test navigating through the first three fields on the form.
8. Use the Alt key plus the letters you added as access keys for navigating to specific fields.

## Project Details

### Project file

34d-contactus.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 3

**Topic:** Form Elements

**Subtopic:** Accessibility

### Objectives covered

#### 3 Document Structure using HTML

**3.4** Construct and analyze markup that uses form elements

**3.4.d** Submission methods

# Input Types and Labels

As most information on a form is input by a user, HTML supports several types of input tags, many of which we will see over the next few projects. Some information is supported through an input tag with an attribute such as text, email, check box or a button. Other form fields are created through separate tags, such as the select tag for a drop-down list and button tags, which are used for buttons.

Labels often accompany input tags. Labels help to define text next to an input field. The other benefit of a label is that with a label tag, a user can click a label to activate in a matching input field. For this to work, the for attribute for a label must have the same text as the input field's name attribute. The label tag also improves the experience for users utilizing screen readers, which improves accessibility.

Input fields should also have an id attribute set as those are needed in case any scripts are written to perform dynamic processing on those fields.

## Purpose

Upon completing this project, you will be able to create input tags, input tag attributes, label tags, and label tag attributes.

## Steps for Completion

1. Open the **34e-contactus.html** file in the MusicStore3 student folder.
2. Within the opening and closing form tags, add an input tag with the type set to **text** and the name and id set to **firstname**
3. Add a line break after the input tag.
4. Above the input tag, add an opening and closing label tag with the label's for attribute set to **firstname**
5. In between the opening and closing label tag, add the text, **First Name**
6. Below the existing input tag, add a second opening and closing label tag with the for attribute set to **lastname**
7. In between the opening and closing label tags, add the text, **Last Name**
8. On the next line, add an input tag with the type set to **text** and the name and id set to **lastname**
9. Below the existing input tags, add a third opening and closing label tag with the for attribute set to **email**
10. In between the opening and closing label tags, add the text, **Email Address**
11. On the next line, add an input tag with the type set to **email** and the name and id set to **email**
12. Save your changes.
13. Preview your webpage in a web browser. You should have these fields and labels in your form:

First Name	<input type="text"/>
Last Name	<input type="text"/>
Email Address	<input type="email"/>

## Project Details

### Project file

34e-contactus.html

### Estimated completion time

15 minutes

### Video reference

#### Domain 3

**Topic:** Form Elements

**Subtopic:** Input Types and Restrictions; The Significance of Labels

### Objectives covered

**3** Document Structure using HTML

**3.4** Construct and analyze markup that uses form elements

**3.4.e** Input types and restrictions

**3.4.j** Label

# The Select Tag

A select tag is used to create a drop-down list in a form. Whereas a select tag creates the list, option tags define the items within the drop-down list. With select tags, a value is important to each option within the select tag. A script or database will only recognize the value name. An automatic default can be assigned to an option within the drop-down list to make the form user-friendly. This is done through adding the selected attribute to the default option.

## Purpose

Upon completing this project, you will be able to create a select tag, add options, add values to those options, and assign a default setting to an option.

## Steps for Completion

1. Open the **34f-contactus.html** file in the MusicStore3 student folder.
2. Below the last input tag, add the text: **How did you hear about us?** in a new paragraph tag.
3. Add an opening and closing select tag below the new paragraph.
4. Within the new select tag, add a name attribute with the value of **found** and an id attribute with the value of **found**
5. Between the opening and closing select tag, add three opening and closing option tags on different lines, using the names **TV**, **Web**, and **Other** in between the opening and closing tags, respectively.
6. For each new option, set a value attribute that is equivalent to the text in between the opening and closing tag.
7. Make the **Web** option the default choice within the list of options.
8. Save your changes.
9. Preview the webpage in a browser. You should see the text question, How did you hear about us? with a drop-down list that has Web as the default option.

## Project Details

### Project file

34f-contactus.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 3

**Topic:** Form Elements

**Subtopic:** Select Tag

### Objectives covered

3 Document Structure using HTML

**3.4** Construct and analyze markup that uses form elements

**3.4.f** Select

# Textarea Tag

The input tag, a tag we saw earlier in this domain, allows for a single line of text to be entered in a form. A textarea tag allows for multiple lines of text to be typed into a text box. It is often used to allow people to fill out a comment form, for example.

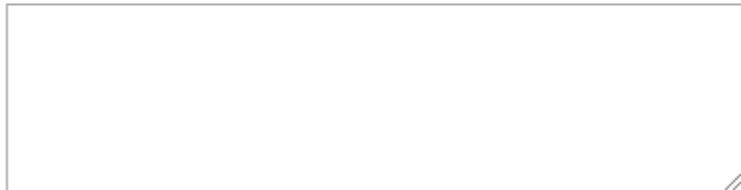
## Purpose

Upon completing this project, you will be able to add a textarea tag to an HTML document.

## Steps for Completion

1. Open the **34g-contactus.html** file in the MusicStore3 student folder.
2. Below the existing select tag, add a paragraph tag with the text, **Add comments below**
3. On the next line, add a textarea tag.
4. To the textarea tag, add attributes to make the text box **6** rows tall and **50** columns wide, with a **255** character maximum input.
5. Save your changes.
6. Preview the webpage in a browser. You should see this text area within the webpage:

**Add comments below**



## Project Details

### Project file

34g-contactus.html

### Estimated completion time

5 minutes

### Video reference

#### Domain 3

**Topic:** Form Elements

**Subtopic:** Textarea Tag

### Objectives covered

#### 3 Document Structure using HTML

**3.4** Construct and analyze markup that uses form elements

**3.4.g** Textarea

# Button Tags

There are two ways to add a button to an HTML document. An input tag with a button attribute allows one to create a button used to submit or reset a form. Most other actions with buttons should be built using the button tag.

## Purpose

Upon completing this project, you will be able to add a button tag and an input tag with a button attribute to an HTML document.

## Steps for Completion

1. Open the **34h-contactus.html** file in the MusicStore3 student folder.
2. Add an opening and closing button tag below the existing textarea tag.
3. Name the button tag, **showimage**
4. Add an empty onclick attribute to the button tag.
5. Between the opening and closing button tag, add the text, **Show image**
6. Below the existing button tag, add an input tag with the type set to **submit**, the name to **submitform**, and the value to **Send it**
7. Save your changes.
8. Preview the webpage in a web browser. You should see two buttons: a button with the text, Show image, and another button with the text, Send it.

## Project Details

### Project file

34h-contactus.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 3

**Topic:** Form Elements

**Subtopic:** Button Tag

### Objectives covered

#### 3 Document Structure using HTML

**3.4** Construct and analyze markup that uses form elements

**3.4.h** Button

# Output Tags

An output tag will take information that is being put into the form and make calculations with the information. The output tag needs a name, which is then referred to in an oninput attribute added to the form tag. This oninput attribute will contain the calculation that is added to the output tag.

## Purpose

Upon completing this project, you will be able to use an output tag in an HTML document.

## Steps for Completion

1. Open the **34i-contactus.html** file in the MusicStore3 student folder.
2. Add an oninput attribute in the form tag, setting the attribute to use the following formula: **fn.value = firstname.value + " " + lastname.value**
3. Add an opening and closing output tag on the line below the button input tag, using the name **fn** for the tag.
4. Save your changes.
5. Preview the webpage in a browser. As you type information into the name fields, the information should show automatically next to the Send it button.

## Project Details

### Project file

34i-contactus.html

### Estimated completion time

5 minutes

### Video reference

#### Domain 3

**Topic:** Form Elements

**Subtopic:** Output Tag

### Objectives covered

#### 3 Document Structure using HTML

**3.4** Construct and analyze markup that uses form elements

**3.4.i** Option

# Option Tag

HTML documents have two ways to create option tags: they can be individual option tags or can be grouped into group option tags. In a previous lesson, individual option tags were added to an input form to add functionality. Group option (optgroup) tags take individual option tags and group them in similar groups to make options easy to read.

## Purpose

Upon completing this project, you will be able to use options and option groups in an HTML document.

## Steps for Completion

1. Open the **34i-contactus.html** file in the MusicStore3 student folder.
2. Insert a blank line above the option value for the TV.
3. On the new blank line, add an opening optgroup tag with the label, **Media**
4. Close the optgroup tag after the option for Other.
5. Surround the options for Google and Bing with an optgroup tag with the label, **Web**
6. Save your changes.
7. Preview your webpage in a web browser.
8. Click the drop-down arrow under the How did you hear about us? line. You should see the following:

Media
TV
Radio
Other
<b>Web</b>
Google
Bing

## Project Details

### Project file

34j-contactus.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 3

**Topic:** Form Elements

**Subtopic:** Option Tag

### Objectives covered

#### 3 Document Structure using HTML

**3.4** Construct and analyze markup that uses form elements

**3.4.i** Option

# Datalist

A datalist tag creates a prepopulated list of possible answers in an input form. This can help users use values you want in a form field without having to display a drop-down list right away, as is the case with a select tag. A datalist typically works in conjunction with an input tag with a list attribute.

## Purpose

Upon completing this project, you will be able to add a datalist tag to an HTML document and populate it with values.

## Steps for Completion

1. Open the **34j-contactus.html** file in the MusicStore3 student folder.
2. Add a paragraph tag on the line before Add comments below, with the text, **What instrument are you most interested in learning?**
3. Add an input tag on the line below the paragraph, with the list attribute set to **instruments** and the name set to **instrument**
4. Add an opening and closing datalist tag on the line below the input tag, with the id set to **instruments**
5. Add four option tags inside the open datalist tag.
6. Set the values of the four option tags to **guitar**, **piano**, **drums**, and **bass**
7. Set the text in between the each opening and closing option tag to **guitar**, **piano**, **drums**, and **bass**, respectively.
8. Save your changes.
9. Preview the webpage in a web browser.
10. Click in the What instrument are you most interested in learning? field.
11. Begin to type the word, **piano**. You should see the piano value appear as a choice.

## Project Details

### Project file

34j-contactus.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 3

**Topic:** Form Elements

**Subtopic:** Datalist Tag

### Objectives covered

**3** Document Structure using HTML

**3.4** Construct and analyze markup that uses form elements

**3.4.j** Label

# Fieldset and Legend Tags

The fieldset tag allows a user to group fields on a form to make the form user-friendly. A fieldset defines a box around a set of fields. A fieldset tag also requires a legend tag, which represents the name of the fieldset.

## Purpose

Upon completing this project, you will be able to use a fieldset tag and legend tag in an HTML document.

## Steps for Completion

1. Open the **34j-contactus.html** file in the MusicStore3 student folder.
2. Add an opening fieldset tag before the first label (the one with the name of **firstname**).
3. Close the fieldset tag after the third input tag (the one with the name of **email**).
4. Just after the open fieldset tag, add an opening and closing legend tag with the text, **Personal**, in between.
5. Add another fieldset tag on the line before the question, **How did you hear about us?** Close the fieldset tag on the line after the last closed datalist tag with the question, **What instrument are you most interested in learning?**
6. Add a legend tag with the text, **Questions**, between the opening and closing tag.
7. Save your changes.
8. Preview the webpage in a web browser. Your completed form should look like this:

### Contact Us

Interested in more information? Fill out the form below and submit it to us. We look forward to hearing from you!

Personal

First Name

Last Name

Email Address

Questions

How did you hear about us?

What instrument are you most interested in learning?

Add comments below

Show image  Send it

9. Save and close all your open files, both in the browser and your code editor.

## Project Details

### Project file

34j-contactus.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 3

**Topic:** Form Elements

**Subtopic:** Fieldset Tag

### Objectives covered

#### 3 Document Structure using HTML

**3.4** Construct and analyze markup that uses form elements

**3.4.j** Label

# HTML and CSS

---

Domain 4

Presented By  
**LK LearnKey®**

# Fill-in-the-Blanks

**Instructions:** While watching Domain 4, fill in the missing words according to the information presented by the instructor. [References are found in the brackets.]

## Image, SVG, and Canvas Elements

1. As users add images onto webpages, they should know the image's \_\_\_\_\_ so they can scale it accordingly. [Add Images]
2. Use either a table or \_\_\_\_\_ when lining up images in rows or columns. [Image Layouts]
3. When adding a hyperlink to an image, make sure the img tag is surrounded by the \_\_\_\_\_ tag and not the other way around. [Image Links]

## Video and Audio Formats

4. In order to have a play button on a video in HTML5, the attribute and value \_\_\_\_\_ needs to be added to the video tag. [Video and Source Tags]
5. Audio tags should have the \_\_\_\_\_ attribute added to allow a user to control when the audio plays. [Audio and Source Tags]
6. When using .vtt files, the two most common formats are \_\_\_\_\_ and subtitles. [Track Tag]
7. An iframe is essentially a frame inside of a webpage that holds content and allows you to put one \_\_\_\_\_ file inside of another. [iFrame Implementations]

# Add Images

When linking an image to a webpage, make sure the image dimensions fit the webpage. The quality of the image is affected by changing the dimensions of an image, so choose images that will still maintain a high-quality image.

## Purpose

Upon completing this project, you will be able to link an image in an HTML document and change the image dimensions to fit the webpage.

## Steps for Completion

1. Open the **41a-instruments.html** file in the MusicStore4 student folder.
2. Add an image tag on the line above the text, Mandolin.
3. Set the image source to be the **mandolin.jpg** file from the images folder.
4. Set the width of the image to **460**
5. Set the height of the image to **345**
6. Set the alternative text of the image to **mandolin**
7. Above the paragraph with the word, Trombone, add an image tag with the source set to the **trombone.jpg** file from the images folder, the width set to 460, the height set to 345, and the alternative text set to **trombone**
8. Save your changes.
9. Preview the webpage. You should see two pictures: one of a mandolin and one of a trombone.

## Project Details

### Project file

41a-instruments.html

mandolin.jpg

trombone.jpg

### Estimated completion time

10 minutes

### Video reference

#### Domain 4

**Topic:** Image, SVG, and Canvas Elements

**Subtopic:** Add Images

### Objectives covered

4 Multimedia Presentation using HTML

**4.1** Construct and analyze markup that displays images

**4.1.a** Img and picture elements and their attributes

# Image Layouts and Links

For a consistent image layout in a webpage, there are two main ways to accomplish this. Either the images should be placed in a table, or one can use CSS styles to lay out the images. As CSS is covered later in this course, this project will show how to use a table to align images to text.

Hyperlinks can also be added to images. A common use for this is to link an image to a larger version of it, which technically is a different file.

## Purpose

Upon completing this project, you will see how to use a table to align images within a webpage. You will also be able to link an image to another image.

## Steps for Completion

1. Open the **41a-instruments2.html** file from your MusicStore4 student folder.
2. Look for the code containing the two images.
3. Above the images, but below the Featured Instruments heading, add a table with two rows and two cells in each row.
4. Move the two image tags to the two table cells in the first row of the table.
5. Move the two paragraphs, Mandolin and Trombone, to the two cells in the second row of the table without including the paragraph tags.
6. Delete the two empty paragraph tags.
7. Add a hyperlink to the mandolin image tag that links to the image `mandolincloseup.jpg` in the images folder.
8. Save your changes.
9. Preview the webpage in a browser. You should see the mandolin on the left and the trombone on the right with the picture titles below each.
10. Click the mandolin image. You should see a close-up image of the mandolin.

## Project Details

### Project file

41a-instruments2.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 4

**Topic:** Image, SVG, and Canvas

Elements

**Subtopic:** Image Layouts; Image Links

### Objectives covered

**4** Multimedia Presentation using HTML

**4.1** Construct and analyze markup that displays images

**4.1.a** Img and picture elements and their attributes

# Video and Source Tags

When linking a video to a webpage you will want to use multiple sources with multiple video formats as not all browsers support all video formats. When adding a video, it is good practice to add controls so a user can control the play settings of the video.

## Purpose

Upon completing this project, you will be able to add a video tag, video controls, and multiple play sources to an HTML document.

## Steps for Completion

1. Open the **43a-media.html** file in the MusicStore4 student folder.
2. Add an opening and closing video tag below the paragraph, See this cool close-up of a piano player. Set the video height to **480**
3. Add the controls attribute to the opening video tag.
4. Inside the video tag, add a source tag for the video file, **Happy.mp4**, located in the Videos folder, with the type **video/mp4**
5. On the next line, add a source for the video file, **Happy.mov** located in the Videos folder, with the type **video/quicktime**
6. Save your changes.
7. Preview the webpage in a web browser. You should see a video player with controls.
8. Play the video.

## Project Details

### Project file

43a-media.html

Happy.mp4

Happy.mov

### Estimated completion time

10 minutes

### Video reference

#### Domain 4

**Topic:** Video and Audio Formats

**Subtopic:** Video and Source Tags

### Objectives covered

**4** Multimedia Presentation using HTML

**4.2** Construct and analyze markup  
that plays video and audio

**4.2.a** Video

**4.2.d** Source

# Audio and Track Tags

Audio tags are added to an HTML file in a very similar fashion as that of video tags. An audio tag can have multiple source tags and should have audio controls for the user to control audio play settings.

The track tag works with VTT (web video techs track standard) files. Track tags and VTT files are not supported by all web browsers. The most common VTT formats that are used are captions and subtitles. The language to use will need to be specified in the track tag. The track tags can have multiple files with multiple languages if needed.

## Purpose

Upon completing this project, you will be able to add an audio tag with controls to an HTML document. You will also be able to add a track tag using subtitles to a video tag.

## Steps for Completion

1. Open the **43b-media.html** file in the MusicStore4 student folder.
2. Add an opening and closing audio tag below the paragraph that starts with, Don't know what a metronome sounds like?
3. Add the controls attribute to the opening audio tag.
4. Inside the audio tag, add a source attribute for the audio file, **metronome\_audio.mp3**, located in the Audio folder, with the type **audio/mp3**
5. Below the two source tags within the video tag, add a track tag with the source set to **subtitles.vtt** from the Videos folder, the kind to **subtitles**, the source language to **en**, and the label to **English**
6. Save your changes.
7. Preview the webpage in a web browser. You should see an audio control with controls.
8. Play the audio.

## Project Details

### Project file

43b-media.html  
metronome\_audio.mp3  
subtitles.vtt

### Estimated completion time

10 minutes

### Video reference

#### Domain 4

**Topic:** Video and Audio Formats

**Subtopic:** Audio and Source Tags;  
Track Tag

### Objectives covered

**4** Multimedia Presentation using HTML

**4.2** Construct and analyze markup  
that plays video and audio

**4.2.b** Audio

**4.2.c** Track

# Simple Iframe Implementations

An iframe is a frame that holds content inside a webpage. The iframe can hold a source from another webpage that can easily be changed with the click of a button or other user interaction. An iframe will let you put one HTML file within another HTML file.

## Purpose

Upon completing this project, you will better understand what an iframe does, as well as how to apply it to a webpage.

## Steps for Completion

1. Open the **43e-media.html** file in the MusicStore4 student folder.
2. Add an opening and closing iframe tag on the line below the paragraph, Here is an example of an iframe.
3. Set the iframe width to **300**, the height to **300**, the id to **vtt**, and the source to **vtt.html**
4. Save your changes.
5. Preview the webpage in a web browser. You should see the following at the bottom of the page:

Here is an example of an iframe

### WEBVTT FILE

00:00:00.000 --> 00:00:06.000

There is no audio here.

6. Save and close all open files, both in the browser and in your code editor.

## Project Details

### Project file

43e-media.html  
vtt.html

### Estimated completion time

5 minutes

### Video reference

#### Domain 4

**Topic:** Video and Audio Formats  
**Subtopic:** iFrame Implementations

### Objectives covered

4 Multimedia Presentation using HTML

**4.2** Construct and analyze markup that plays video and audio

**4.2.e** Iframe

# HTML and CSS

---

Domain 5

Presented By  
**LK LearnKey®**

# Fill-in-the-Blanks

**Instructions:** While watching Domain 5, fill in the missing words according to the information presented by the instructor. [References are found in the brackets.]

## Position Text with CSS

1. The best way to take control over layouts on a webpage is through a combination of \_\_\_\_\_ tags and ids and then to use styles on those ids. [Height and Width]
2. In using CSS to control your overall page layouts, the max-width property takes precedence over the \_\_\_\_\_ property. [Max-Width Used with an Element]
3. \_\_\_\_\_, none, right, and left are valid values used in setting the float property. [Float Positioning]
4. Relative positioning can be used to shift elements in relation to their normal \_\_\_\_\_ location. [Relative Positioning]
5. Absolute positioning changes the \_\_\_\_\_ of elements below the element being positioned. [Absolute Positioning]
6. Users can access the \_\_\_\_\_ tools to see the default position for elements on a webpage. [Static]
7. Fixed positioning is like \_\_\_\_\_ positioning in that it removes an element from the natural flow of the page. [Fixed]
8. Setting the overflow property helps control what happens when extra content breaks out of an element's \_\_\_\_\_ area. [Overflow Settings]
9. The alignment for a block element's inner content can be controlled with the \_\_\_\_\_ property. [Alignment]
10. \_\_\_\_\_ elements render the full width of a page and will always render on their own lines. [Display]
11. Inline is the default setting of the \_\_\_\_\_ property, which defines a given element's rendering box. [Inline vs. Block]
12. To make an object invisible, set its visibility property to \_\_\_\_\_. [Visibility]
13. Using shorthand to set margins sets them in the following order: \_\_\_\_\_, \_\_\_\_\_, bottom, and \_\_\_\_\_. [Box Model Margins]
14. \_\_\_\_\_ control gaps between objects while padding controls the gaps between the edges of an object and text inside of it. [Box Model Padding]

## Format Text with CSS

15. A font-family property should have at least one \_\_\_\_\_ font. [Font-Family]
16. In HTML, colors are often assigned using \_\_\_\_\_ values instead of names. [Color]
17. The font-style property can be used to italicize text, but the property will not make text \_\_\_\_\_. [Font-Style]
18. The font-weight property controls the levelness of \_\_\_\_\_ on text. [Font-Weight]
19. The default color for a hyperlink that has been visited is \_\_\_\_\_. [Link Colors]
20. The font-variant property can change text to be in the \_\_\_\_\_ style, meaning lowercase letters are in an uppercase style with a slightly reduced size. [Font Variant]

21. The \_\_\_\_\_ property can move text to the bottom of an element. [Text-Alignment]
22. The text-decoration property determines whether an element is \_\_\_\_\_. [Text-Decoration]
23. The \_\_\_\_\_ property controls the first line text indent in a paragraph. [Indentation]
24. The line-height property can be set using several different units including \_\_\_\_\_, centimeters, percentages, and others. [Line Height]
25. Word-wrap is a CSS property that tells a browser how to display long words that over-extend their \_\_\_\_\_ element. [Word-wrap]
26. Letter spacing sets the space between \_\_\_\_\_ in a word. [Letter-Spacing, Word-Spacing]

### Backgrounds and Borders

27. For setting colors for borders, one can use a hexadecimal color or a \_\_\_\_\_ color. [Border-Color]
28. A border style that is not supported in a browser will make a border \_\_\_\_\_. [Border-Style]
29. If a border width is not set, a \_\_\_\_\_ will determine the border width. [Border-Width]
30. Backgrounds are typically set using either a color or an \_\_\_\_\_. [Backgrounds]
31. The more \_\_\_\_\_ a layout, the less fluid the layout is across multiple devices. [Divs]
32. \_\_\_\_\_ colors have better support across browsers than do named colors. [Working with Colors]

### Responsive Layouts

33. The \_\_\_\_\_ measurement is equivalent to the font size being used. [Units of Measure]
34. To optimize pages for a variety of devices, one can use a \_\_\_\_\_ tag with a viewport setting to instruct the browser on how to adjust the page's scaling and dimensions. [Viewport]
35. A media query can change style sheets according to screen \_\_\_\_\_. [Media Query]
36. Using percentages for measurements helps one to avoid having to \_\_\_\_\_ left to right to view a webpage. [Percentages and Pixels]
37. A framework can be created or \_\_\_\_\_ from a third party. [Frameworks and Templates]
38. The max-width property can help fix fluid measurements on screens with a very large \_\_\_\_\_. [Max-Width Property]
39. With CSS Grids, developers can take far greater control over how their content displays across \_\_\_\_\_. [Grids]

# Height and Max-Width Used with an Element

A div tag with an id that is styled will help you have more control of the webpage layout you are creating. The max-width property helps the webpage layout to stay the size of the average screen size of your users, regardless of the set screen width in the CSS style sheet.

## Purpose

Upon completing this project, you will be able to use height and width in webpage layouts and set a max-width property for a webpage.

## Steps for Completion

1. Open the **51a-positioningexample.html** file in the MusicStore5 student folder.
2. Add the width of **250px** and a height of **250px** to the blue, yellow, and coral rules within the inline style sheet.
3. Save your changes.
4. Preview the webpage in a browser. The three colored elements should be stacked one on top of the other.
5. Scroll down the page. Notice that the text for the coral element does not fit inside the element.
6. Return to the code editor.
7. Adjust the coral rule to a width and max-width of **750px** so that the text will be contained in the coral box.
8. Save your changes.
9. Preview the webpage in a browser. You should see all the text within blue, yellow, and coral boxes.

## Project Details

### Project file

51a-positioningexample.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 5

**Topic:** Position Text with CSS

**Subtopic:** Height and Width; Max-Width Used with an Element

### Objectives covered

#### 5 Webpage Styling using CSS

**5.1** Construct and analyze styles that position content

**5.1.a** Positioning, including float, relative, absolute, static, fixed, max-width, overflow, height, width, align

# Float Positioning

When an element is floated, it is moved to a specific area on the screen so other elements below it can move up. Elements can be float positioned side by side if the screen width will allow the change, or they can float with one side or both sides of the element cleared out using the clear property within CSS.

## Purpose

Upon completing this project, you will be able to use the float property to style a webpage with elements that are side by side or alone on the webpage.

## Steps for Completion

1. Open the **51af-positioningexample.html** file in the MusicStore5 student folder.
2. In the style sheet, add a property/value pair to the blue rule set that sets the float property to **left**
3. In the yellow rule, add a property/value pair that sets the float property to **left**
4. Save your changes.
5. Preview the webpage in a web browser. Notice that the blue element now floats to the left of the yellow element and the yellow element floats to the left of the coral element, but the text is spilling out of the coral element.
6. Return to the code editor.
7. In the yellow rule, add a property/value pair that sets the clear property to **both**
8. Save your changes.
9. Preview the webpage in a browser. You should see the blue and coral elements side by side at the top, with the yellow element on the lower left part of the page, alone.

## Project Details

### Project file

51af-positioningexample.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 5

**Topic:** Position Text with CSS

**Subtopic:** Float Positioning

### Objectives covered

#### 5 Webpage Styling using CSS

**5.1** Construct and analyze styles that position content

**5.1.a** Positioning, including float, relative, absolute, static, fixed, max-width, overflow, height, width, align

# Relative and Absolute Positioning

Relative and absolute positioning give more control over where elements are placed on a webpage than the float property. Relative positioning of an element positions it offset relative to where the normal position was and is not an exact set location. Relative positioning does not otherwise change the overall flow of elements.

Absolute positioning of an element will position it exactly in the location placed in the code. Absolute positioning will change the flow of elements below it if they are not also absolutely positioned. Absolute positioning will give you the most control of elements in

## Purpose

Upon completing this project, you will be able to use relative and absolute positioning of elements in a webpage.

## Steps for Completion

1. Open the **51ar-positioningexample.html** file in the MusicStore5 student folder.
2. In the internal style sheet, comment out the float property/value combination in the blue rule set.
3. Comment out both the float and clear property/value combinations in the yellow rule set.
4. In the yellow rule set, add the position property and set the value to **relative**
5. In the yellow rule set, set the top property to **100px** and the left property to **100px**
6. Comment out the max-width property/value combination in the coral rule set.
7. In the coral rule set, add the position property and set the value to **relative**
8. In the coral rule set, add the top property and set the value to **100px**
9. Save your changes.
10. Preview the webpage in a web browser. The yellow element should start 100 pixels below and 100 pixels to the right of the blue element and the coral element should start 100 pixels below the yellow element.
11. Return to your code editor.
12. In the internal style sheet, change the value of the position property in the yellow rule set to **absolute**
13. Save your changes.
14. Preview the webpage in a web browser. The yellow element should overlap part of the blue element.

## Project Details

### Project file

51ar-positioningexample.html

### Estimated completion time

15 minutes

### Video reference

#### Domain 5

**Topic:** Position Text with CSS

**Subtopic:** Relative Positioning; Absolute Positioning

### Objectives covered

#### 5 Webpage Styling using CSS

**5.1** Construct and analyze styles that position content

**5.1.a** Positioning, including float, relative, absolute, static, fixed, max-width, overflow, height, width, align

# Static and Fixed

The natural flow of a webpage is dependent upon the designated position of the elements on the webpage. Users can accept the default positioning, which is usually static, or change it, depending on how they want their elements laid out.

When an element is positioned as absolute, and does not specify top, right, bottom, or left, it is positioned at the location it would have been had it been static. Fixed positioning is like absolute positioning in that it removes the element from the natural flow of the page. The difference is that absolute positioning positions an element based on its parent element, whereas an element that has a position of fixed is positioned absolutely in relation to the browser window.

## Purpose

Upon completing this project, you will better understand how to position elements on a webpage using static, absolute, and fixed positioning.

## Steps for Completion

1. Open the **positioningelement1.html** file in the MusicStore5 student folder.
2. View the webpage in a browser.
3. Open the Developer Tools and view the default position for the webpage elements.
4. What is the default position of the webpage elements?
  - a. \_\_\_\_\_
5. Navigate back to the **positioningelement1.html** file.
6. Add a new attribute on line 24, setting the position to **absolute**
7. View the webpage in a browser.
8. Navigate back to the **positioningelement1.html** file.
9. Change the position from absolute to **fixed**
10. Add an html height of **2000px** so that the code looks like this:

```

20      background-color: lightcoral;
21      border: 1px solid black;
22      width: 750px;
23      height: 250px;
24      position: fixed;
25  }
26  html {
27      height: 2000px;
28  }
```

11. View the webpage in a browser.

## Project Details

### Project file

positioningelement1.html

### Estimated completion time

5-10 minutes

### Video reference

#### Domain 5

**Topic:** Position Text with CSS

**Subtopic:** Static; Fixed

### Objectives covered

#### 5 Webpage Styling using CSS

**5.1** Construct and analyze styles that position content

**5.1.a** Positioning, including float, relative, absolute, static, fixed, max-width, overflow, height, width, align

# Overflow and Alignment Settings

Overflow settings keep the text from overflowing an element. The main use of the overflow property is to tell text to scroll if it otherwise does not fit within an element. It is good practice to set overflow settings to maintain the control of element designs in different browsers.

Another property worth setting for text in CSS is the text-align property. The text-align property can be used in any tag or class that uses text. It is best practice to use CSS for text alignment.

## Purpose

Upon completing this project, you will be able to add alignment to an element, and a scroll bar, using overflow settings, to a webpage element.

## Steps for Completion

1. Open the **51ao-positioningexample.html** file in the MusicStore5 student folder.
2. Change the coral element width to **250px**
3. Add an overflow property to the coral element with the value set to **scroll**
4. Save your changes.
5. Preview the webpage in a browser. You should see a scroll bar on the side of the coral element that allows you to see all the text within the coral box.
6. Return to your code editor.
7. Add a text-align property to the blue rule set with the value set to **left**
8. Add a text-align property to the yellow rule set with the value set to **center**
9. Add a text-align property to the coral rule set with the value set to **justify**
10. Save your changes.
11. Preview the webpage in a browser. You should have left-aligned text in the blue box, center-aligned text in the yellow box, and justified text in the coral box.

## Project Details

### Project file

51ao-positioningexample.html

### Estimated completion time

5-10 minutes

### Video reference

#### Domain 5

**Topic:** Position Text with CSS

**Subtopic:** Overflow Settings; Alignment

### Objectives covered

#### 5 Webpage Styling using CSS

**5.1** Construct and analyze styles that position content

**5.1.a** Positioning, including float, relative, absolute, static, fixed, max-width, overflow, height, width, align

# Display

The display property can take many different values, including the h2, span, and aside tags. Each tag includes its own properties and can change the positioning of elements on a webpage.

When it comes to displays, users should also understand the difference between block and inline elements. Block elements render the full width of a page and will always render on their own lines. Inline elements take up the necessary space and can position themselves alongside other elements.

## Purpose

Upon completing this project, you will better understand display elements.

## Steps for Completion

1. Open the **display.html** file in the MusicStore5 student folder.
2. View the webpage in a browser.
3. Label the elements on the page as block or inline.
  - a. \_\_\_\_\_ h2 tag
  - b. \_\_\_\_\_ Span tag #1
  - c. \_\_\_\_\_ Span tag #2
  - d. \_\_\_\_\_ Span tag #3
  - e. \_\_\_\_\_ Aside tag

## Project Details

### Project file

display.html

### Estimated completion time

5-10 minutes

### Video reference

#### Domain 5

**Topic:** Position Text with CSS

**Subtopic:** Display

### Objectives covered

#### 5 Webpage Styling using CSS

**5.1** Construct and analyze styles that position content

**5.1.b** Display

# Inline vs. Block

CSS style sheets will default elements to blocks, which places one element after another on a webpage. The display property can be set to move elements to be in line with each other on a webpage.

## Purpose

Upon completing this project, you will better understand the difference between inline and block layouts and be able to use them in a CSS style sheet.

## Steps for Completion

1. Open the **51b-main.css** file in the MusicStore5 student folder.
2. Remove the comment from the background-color property in the figcaption rule set.
3. Inside the figcaption rule set, add the display property and set the value to **inline**
4. Inside the sidenote rule set, add the display property and set the value to **inline**
5. Save your changes.
6. Open the **51b-spotlight.html** file in the MusicStore5 student folder.
7. Preview the webpage in a web browser. Notice that with the inline display, the two highlighted pieces of text no longer take up the width of the page.

## Project Details

### Project file

51b-main.css

51b-spotlight.html

### Estimated completion time

5-10 minutes

### Video reference

#### Domain 5

**Topic:** Position Text with CSS

**Subtopic:** Inline vs. Block

### Objectives covered

5 Webpage Styling using CSS

5.1 Construct and analyze styles that position content

5.1.c Inline vs. block

# Visibility

Elements can be set not to be visible on a webpage and then dynamically set to display, usually on either a click or a mouse hover by a user over a visible element along with the getElementById method on the object to set the visibility property to visible. This is not the same as the details and summary HTML tags, but the visible element can be

## Purpose

Upon completing this project, you will be able to start with an element not set to visible and then use an event to make an element visible.

## Steps for Completion

1. Open the **51c-positioningexample.html** file in the MusicStore5 student folder.
2. In the internal style sheet, add the visibility property, with a value of **hidden**, to the coral rule set.
3. Save your changes.
4. Preview the webpage in a web browser. The coral element should be invisible.
5. Return to your code editor.
6. In the div tag with the id of yellow, found in the body of the HTML document, add a paragraph tag with the text, **Click to see the coral box**
7. In the opening paragraph tag, make an **onclick** event using JavaScript that will set the visibility property of the div tag with the coral id to **visible**
8. Save your changes.
9. Preview the webpage in a web browser.
10. Click the text, Click to see the coral box. The coral box should appear.

## Project Details

### Project file

51c-positioningexample.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 5

**Topic:** Position Text with CSS

**Subtopic:** Visibility

### Objectives covered

#### 5 Webpage Styling using CSS

**5.1** Construct and analyze styles that position content

**5.1.d** Visibility

# Box Model Margins and Padding

Margins control the distance between objects on a webpage, while padding controls the distance between edges of an object and the text inside the object. Margins and padding can be set for all sides in one property/value set or a separate property/value set can be used for each side. When setting the code for all sides at once, the measurements will move in a clockwise position starting with the top measurement.

## Purpose

Upon completing this project, you will better understand the difference between margins and padding, and you will be able to add margins and padding to objects in a webpage.

## Steps for Completion

1. Open the **51d-positioningexample.html** file in the MusicStore5 student folder.
2. In the blue rule set, remove the comment around the float property/value combination.
3. In the yellow rule set, adjust the comment block so that the float property/value combination is not commented out and the clear, position, top, and left property/ value combinations are commented out.
4. In the yellow rule set, add a margin property and set the values to **0px**, **10px**, **0px**, and **5px** for the top, right, bottom, and left margins, respectively.
5. In the blue rule set, add a padding property with the value set to **.5em**
6. In the yellow rule set, add a property that will set the top side padding to **.5em**
7. Save your changes.
8. Preview the webpage in a browser. There should be space between the box edge and the text on all sides of the blue box and between the top line and the text in the yellow box. There should also be a margin in between the two boxes.
9. Close all your open files in the browser and the code editor, saving changes if prompted.

## Project Details

### Project file

51d-positioningexample.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 5

**Topic:** Position Text with CSS

**Subtopic:** Box Model Margins; Box Model Padding

### Objectives covered

#### 5 Webpage Styling using CSS

**5.1** Construct and analyze styles that position content

**5.1.e** Box model, margins, and padding

# Font Families in CSS

The font properties can have type, size, color, style, and weight all adjusted in a CSS style sheet. For font families, it is a best practice to use multiple fonts and then a font family default so that users who do not have the first font specified can still see the text with a font you choose, not the default font on their browsers.

## Purpose

Upon completing this project, you will be able to apply font families to a CSS style sheet.

## Steps for Completion

1. Open the **52a-main.css** file in the MusicStore5 student folder.
2. On the line above the figcaption rule set, add a rule set for all six headings (h1 through h6).
3. Within the new rule set, add a font family property and set the values to **helvetica**, **arial**, and **sans-serif**
4. Save your changes.
5. Open the **52a-books.html** file in the MusicStore5 student folder.
6. Surround the text, Welcome to THE Music Store, with an h1 tag.
7. Save your changes.
8. Preview the webpage in a web browser. You should see the heading with a font of helvetica, arial, or a similar sans-serif font.

## Project Details

### Project file

52a-main.css

52a-books.html

### Estimated completion time

5-10 minutes

### Video reference

#### Domain 5

**Topic:** Format Text with CSS

**Subtopic:** Font-Family

### Objectives covered

**5** Webpage Styling using CSS

**5.2** Construct and analyze styles that format text

**5.2.a** Font-family

# Font Colors in CSS

Font colors can and should be controlled through CSS. There are two basic ways to assign values to colors. First, color names can be used. Some browsers support more color names than others, however, so hexadecimal values are a better choice for color values, as hexadecimal values are supported in all browsers.

## Purpose

Upon completing this project, you will be able to apply font colors within a CSS style sheet.

## Steps for Completion

1. Open the **52b-main.css** file in the MusicStore5 student folder.
2. Within the figcaption rule set, add a color property with a value of **#032f0d**
3. Save your changes.
4. Open the **52b-spotlight.html** file in the MusicStore5 student folder.
5. Preview the webpage in a web browser. You should see the text, The Mighty Kazoos, in a dark green color.

## Project Details

### Project file

52b-main.css

52b-spotlight.html

### Estimated completion time

5 minutes

### Video reference

#### Domain 5

**Topic:** Format Text with CSS

**Subtopic:** Color

### Objectives covered

**5** Webpage Styling using CSS

**5.2** Construct and analyze styles that format text

**5.2.b** Color

# Font Styles and Weights in CSS

In CSS, font styles control the shape of a font from the standpoint of whether the font is italic. Font weight controls the boldness of a font, which can be indicated either with a value of bold or a value from 0 to 1000. A bold font will be more consistent with the bold value than with numbers across different browsers. Many browsers will

only recognize 400 and 700 as measurements of boldness, so use other measurements sparingly.

## Purpose

Upon completing this project, you will be able to apply font styles and font weights to a CSS style sheet.

## Steps for Completion

1. Open the **52c-main.css** file in the MusicStore5 student folder.
2. In the sale rule set, add a font style property with the value set to **italic**.
3. In the sale rule set, add a font weight property with the value set to **700**.
4. Save your changes.
5. Open the **52c-books.html** file in the MusicStore5 student folder.
6. Preview the webpage in a web browser. The text, Special Sale, should be bold and italic.
7. Return to the **52c-main.css** file in the code editor.
8. Change the value of the font weight property from 700 to **bold**.
9. Save your changes.
10. Preview the **52c-books.html** page in a web browser again. The text, Special Sale, should have the same or similar boldness as before.

## Project Details

### Project file

52c-main.css

52c-books.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 5

**Topic:** Format Text with CSS

**Subtopic:** Font-Style; Font-Weight

### Objectives covered

5 Webpage Styling using CSS

**5.2** Construct and analyze styles that format text

**5.2.c** Font-style

**5.2.d** Font-size

**5.2.e** Font-weight

# Link Colors in CSS

The color property changes the font color of text within a CSS rule set. This property can also be used in pseudo classes. By default, unvisited links are blue, visited links are purple, and active links are red. These colors can be changed, but in doing so you will want to make sure that link colors are consistent throughout your site.

## Purpose

Upon completing this project, you will be able to apply colors to link-based pseudo classes in a CSS style sheet.

## Steps for Completion

1. Open the **52e-main.css** file in the MusicStore5 student folder.
2. In the existing rule set for the a tag, add a color property with the value set to **darkgreen**
3. Below the existing rule set, add a rule set for the a tag and the active pseudo class.
4. Inside the new rule set, add a color property with the value set to **#63f60a**
5. Below the existing rule set, add a rule set for the a tag and the visited pseudo class.
6. Inside the new rule set, add a color property with the value set to **#808080**
7. Save your changes.
8. Open the **52e-spotlight.html** file in the MusicStore5 student folder.
9. Preview the webpage in a web browser. Any unvisited link should be dark green and any visited link should be gray.

## Project Details

### Project file

52e-main.css

52e-spotlight.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 5

**Topic:** Format Text with CSS

**Subtopic:** Link Colors

### Objectives covered

5 Webpage Styling using CSS

5.2 Construct and analyze styles that format text

5.2.g Link colors

# Font Variant

The font-variant property can change text to be in the Small Caps style, meaning lowercase letters are in an uppercase style with a slightly reduced size. This style is most commonly used on headings, but it can be used anywhere.

## Purpose

Upon completing this project, you will be able to use the font-variant property.

## Steps for Completion

1. Open the **headings1.html** file in the MusicStore5 student folder.
2. View the webpage in a browser.
3. Navigate to the linked **main.css** style sheet.
4. On line 10, add a font-variant property with a value of **small-caps**
5. Save your changes.
6. Refresh the webpage in the browser.

## Project Details

### Project file

headings1.html  
main.css

### Estimated completion time

5 minutes

### Video reference

#### Domain 5

**Topic:** Format Text with CSS  
**Subtopic:** Font Variant

### Objectives covered

**5** Webpage Styling using CSS  
**5.2** Construct and analyze styles that format text  
**5.2.f** Font-variant

# Text Alignment and Decoration

Text alignment has been covered in previous lessons. In this project, we will be covering text alignment in tables. Aligning text in a table is important to the presentation of the table as it can make a table stand out and make it easier to read. Often, the vertical align property is used to position text vertically within a cell.

The text-decoration property will control if text is underlined or not underlined. There are several types of underline styles, but they may not be supported in multiple browsers. When adding any type of styling it is good practice to check if the style is supported in different browsers.

## Purpose

Upon completing this project, you will be able to apply text alignment to a table using CSS. You will also be able to apply the text decoration property within CSS.

## Steps for Completion

1. Open the **52f-main.css** file in the MusicStore5 student folder.
2. At the bottom of the style sheet, add a class named **vbot** that only applies to table cells.
3. Within the new class, add a vertical align property with the value of **bottom**
4. Within the new class, add a height property with a value of **30px**
5. Within the `a:hover` class, change the value of the text decoration property to **double**
6. Save your changes.
7. Open the **52f-hours.html** file in the MusicStore5 student folder.
8. Preview the webpage in a web browser. The text should be vertically aligned in the bottom of the last cell of the table.
9. Hover your mouse over the Home link. If your browser supports double underlines, the link should show a double underline.

## Project Details

### Project file

52f-main.css

52f-hours.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 5

**Topic:** Format Text with CSS

**Subtopic:** Text-Alignment; Text-Decoration

### Objectives covered

**5** Webpage Styling using CSS

**5.2** Construct and analyze styles that format text

**5.2.h** Text formatting, text alignment, text decoration, indentation

# Indentation

The text indent property controls the first line text indent in a multi-lined paragraph. To indent more than the first line in a paragraph, use the padding property.

## Purpose

Upon completing this project, you will be able to use the text indent property to control the indentation of the first line in a paragraph.

## Steps for Completion

1. Open the **52fi-main.css** file in the MusicStore5 student folder.
2. Below the sale class, add a class named **indent**
3. Within the new class, add a text indent property with a value of **5em**
4. Open the **52fi-index.html** file in the MusicStore5 student folder.
5. Down the page, wrap an opening and closing paragraph tag around the text: **We are open seven days a week...We look forward to seeing you!**
6. Add the indent class to the open paragraph tag.
7. Save your changes.
8. Preview the webpage in a browser. The first sentence in the last paragraph on the webpage will be indented.
9. Go back to the **52fi-main.css** file in your code editor and change the property you set earlier from the left indent property to the padding property.
10. Return to the **52fi-index.html** webpage in the code editor.
11. Preview the webpage in a web browser. The entire paragraph to which you added the left indent class should be indented.
12. Close all open files in your browser and code editor, saving changes if prompted.

## Project Details

### Project file

52fi-main.css

52fi-index.html

### Estimated completion time

10-15 minutes

### Video reference

#### Domain 5

**Topic:** Format Text with CSS

**Subtopic:** Indentation

### Objectives covered

5 Webpage Styling using CSS

**5.2** Construct and analyze styles that format text

**5.2.h** Text formatting, text alignment, text decoration, indentation

# Line Height and Word-Wrap

One can use the line-height property to add space between separate lines of text within the same element. The line-height property sets the height of individual lines within an element. The line-height property can be set using several different units, including pixels, centimeters, percentages, and more.

Word-wrap is a CSS property that tells the browser how to display long words that overextend their parent element.

## Purpose

Upon completing this project, you will better understand the uses of the line-height and word-wrap properties to display elements on a webpage.

## Steps for Completion

1. Open the **positioningexample2.html** file in the MusicStore5 student folder.
2. View the webpage in a browser.
3. On line 18, add the line-height property with the units set to **20px**
4. View the webpage in the browser and refresh the page.
5. Open the **wordwrap.html** file.
6. View the webpage in a browser.
7. Describe the differences between the CSS word-wrap properties of normal and break-word.

a. \_\_\_\_\_

---

---

## Project Details

### Project file

positioningexample2.html  
wordwrap.html

### Estimated completion time

5-10 minutes

### Video reference

#### Domain 5

**Topic:** Format Text with CSS

**Subtopic:** Line Height; Word-wrap

### Objectives covered

5 Webpage Styling using CSS

**5.2** Construct and analyze styles that format text

**5.2.i** Line-height

**5.2.j** Word-wrap

# Letter-Spacing, Word-Spacing

Letter spacing sets to the space between characters in a word. Word spacing sets the space between words.

## Purpose

Upon completing this project, you will better understand how to alter letter and word spacing on a webpage.

## Steps for Completion

1. Open the **spacing.html** file in the MusicStore5 student folder.
2. View the webpage in a browser.
3. In the HTML file, change the letter spacing on line 19.
4. Change the word spacing on line 23.
5. Save your changes.
6. View the webpage in a browser.

## Project Details

### Project file

spacing.html

### Estimated completion time

5-10 minutes

### Video reference

#### Domain 5

**Topic:** Format Text with CSS

**Subtopic:** Letter-Spacing; Word-Spacing

### Objectives covered

**5** Webpage Styling using CSS

**5.2** Construct and analyze styles that format text

**5.2.k** Letter-spacing

# Border Color and Style

The border color property sets a color for a border. The border color only works if a border style is set, as a border style must be set for a border to be displayed. The more complex the border style, the more important it is to test the style in multiple browsers to make sure the border style is supported properly.

## Purpose

Upon completing this project, you will be able to set border colors and styles in a stylesheet.

## Steps for Completion

1. Open the **53a-main.css** file in the MusicStore5 student folder.
2. In the sidenote rule set, set a border color property with a value of **navy**
3. In the sidenote rule set, set a border style property with a value of **solid**
4. Save your changes.
5. Open the **53a-spotlight.html** file in the MusicStore5 student folder.
6. Preview the webpage in a web browser. The sidenote text should show on the right side of the screen with a solid navy border around the text.
7. Return to your code editor.
8. Return to the **53a-main.css** file.
9. In the sidenote rule set, change the value of the border style property to **dashed**
10. Save your changes.
11. Return to the **53a-spotlight.html** file.
12. Preview the webpage in a web browser. The border should now be dashed.

## Project Details

### Project file

53a-main.css  
53a-spotlight.html

### Estimated completion time

10-15 minutes

### Video reference

#### Domain 5

**Topic:** Backgrounds and Borders  
**Subtopic:** Border-Color; Border-Style

### Objectives covered

**5** Webpage Styling using CSS  
**5.3** Construct and analyze styles that format backgrounds and borders  
**5.3.a** Border-color  
**5.3.b** Border-style

# Border Width

The border width property is typically set using pixels as a measurement. A border width can be set for all sides of an object, or individual widths can be set for each side of an object. Furthermore, shorthand can be used as a border property can have a width, style, and color all set at once.

## Purpose

Upon completing this project, you will be able to set border widths for an entire object and for individual sides of an object.

## Steps for Completion

1. Open the **53c-main.css** file in the MusicStore5 student folder.
2. In the sidenote rule set, add a border width property with a value of **1px**
3. Save your changes.
4. Open the **53c-spotlight.html** file in the MusicStore5 student folder.
5. Preview the webpage in a web browser. You should see a one-pixel border around the sidenote text.
6. Return to the **53c-main.css** file in the code editor.
7. In the sidenote rule set, set the border width property value to **0px**
8. In the sidenote rule set, add a property/value combination that will set the left border width to **2px**
9. In the sidenote rule set, add a property/value combination that will set the right border width to **2px**
10. Save your changes.
11. Preview the **53c-spotlight** webpage in a web browser. You should see a two-pixel dashed border on the left and right sides of the sidenote text.
12. Return to the **53c-main.css** file in the code editor.
13. Comment out the border color, border style, border width, border left width, and border right width properties in the sidenote rule set.
14. In the sidenote rule set, add a border property to set the width to **2px**, the style to **solid**, and the color to **navy**
15. Save your changes.
16. Preview the **53c-spotlight** webpage in a web browser. You should see a two-pixel solid navy border around the sidenote text.

## Project Details

### Project file

53c-main.css

53c-spotlight.html

### Estimated completion time

15 minutes

### Video reference

#### Domain 5

**Topic:** Backgrounds and Borders

**Subtopic:** Border-Width

### Objectives covered

#### 5 Webpage Styling using CSS

**5.3** Construct and analyze styles that format backgrounds and borders

**5.3.c** Border-width

# Backgrounds

There are two main types of backgrounds that can be used with webpages: colors and images. In a previous lesson we applied the color property to the body tag, which creates a background color. In this project, we will use an image as a background. To use an image, a pointer to a URL is needed.

An additional property that is often used with an image is the background repeat property. A background image can be set to not repeat (no-repeat), to repeat across the top of a page (repeat-x), or to repeat vertically (repeat-y). The default repeat is to tile the background image through the page.

## Purpose

Upon completing this project, you will be able to apply an image to the webpage background and have the image repeat itself on the page.

## Steps for Completion

1. Open the **53d-positioningexample.html** file in the MusicStore5 student folder.
2. At the bottom of the internal style sheet, add a class named **backimage**
3. In the new backimage class, add a background image property with its value set to the URL of the **kazoo-transparent.jpg** file in the images folder.
4. In the backimage class, add a background repeat property and set its value to have the background image repeat vertically on the page.
5. Apply the backimage class to the body tag.
6. Save your changes.
7. Preview the webpage in a web browser. You should see a background picture of a kazoo running down the left side of the page.

## Project Details

### Project file

53d-positioningexample.html  
kazoo-transparent.jpg

### Estimated completion time

5-10 minutes

### Video reference

#### Domain 5

**Topic:** Backgrounds and Borders  
**Subtopic:** Backgrounds

### Objectives covered

**5** Webpage Styling using CSS  
**5.3** Construct and analyze styles that format backgrounds and borders  
**5.3.d** Background properties

# Divs and Colors

A div tag can be used to control the entire width of a webpage. The more fixed your layout is, the less fluid your webpage will be across different devices. However, this may be necessary if your website is viewed on a lot of smaller devices.

Colors have been covered throughout this course. Remember that hexadecimal colors are the safest colors to use as they have the largest browser support. Hexadecimal colors closer to #000000 are darker and hexadecimal colors closer to #FFFFFF are lighter. The first pair of values indicate the amount of red in the color, the second pair, green, and the third pair, blue. Colors with matching red, green, and blue pairs, such as #333333 and #0F0F0F are shades of gray.

## Purpose

Upon completing this project, you will be able to use a div tag to set the width of a webpage. You will also have a further understanding of how colors are used on a webpage.

## Steps for Completion

1. Open the **53e-main.css** file in the MusicStore5 student folder.
2. Before the a tag rule set, add an id-based rule set named **pagewidth**
3. In the new pagewidth rule set, add a width property with the value set to **500px**
4. In the sidenote rule set, change the background color property value to **#333333**
5. In the sidenote rule set, add a color property with the value set to the same color as the body background color.
6. Save your changes.
7. Open the **53e-spotlight.html** file in the MusicStore5 student folder.
8. Wrap a div tag around the entire body of the webpage (not the body tag itself, but everything inside the body tag).
9. Set the id of the div tag to **pagewidth**
10. Save your changes.
11. Preview the webpage in a web browser. The contents of the page should have a width of 500 pixels, which will bring the side note closer to the body of the text. In addition, the background color should appear to mask through the sidenote text as the text color is now the same as the background.
12. Close all your open files in both the browser and the code editor, saving changes where prompted.

## Project Details

### Project file

53e-main.css  
53e-spotlight.html

### Estimated completion time

10-15 minutes

### Video reference

#### Domain 5

**Topic:** Backgrounds and Borders  
**Subtopic:** Divs; Working with Colors

### Objectives covered

**5** Webpage Styling using CSS  
**5.3** Construct and analyze styles that format backgrounds and borders  
**5.3.d** Background properties  
**5.3.e** Colors

# Units of Measure

There are several different units of measure that can be used for sizing. Many of these have already been seen in this course. Here is a recap of units of measure and their uses:

- em: A relative measurement equivalent to the font size being used. For example, 3em = 3 x the size of the current font.
- ex: A measurement relative to the height of the letter x
- px: Technically, a measurement that is 1/96" of an inch but can vary based on screen
- pt: A measurement that is 1/72 of an inch
- cm: A centimeter
- in: An inch
- mm: A millimeter

## Purpose

Upon completing this project, you will better understand how to set measurements for one or more elements on a webpage.

## Steps for Completion

1. Open the **54a-main.css** file in the MusicStore5 student folder.
2. Below the rule set for the six headings, add a rule set for the h1 heading.
3. In the h1 heading rule set, add a font size property with the value set to **18pt**
4. Save your changes.
5. Open the **54-index.html** file in the MusicStore5 student folder.
6. Preview the webpage in a web browser. The heading, Welcome to the Music Store, should have a size of 18 points.

## Project Details

### Project file

54a-main.css

54a-index.html

### Estimated completion time

5 minutes

### Video reference

#### Domain 5

**Topic:** Responsive Layouts

**Subtopic:** Units of Measure

### Objectives covered

5 Webpage Styling using CSS

**5.4** Construct and analyze styles that create a simple responsive layout

**5.4.a** Units of measurement  
(percentages, pixels, em, vw, vh)

# Viewport and Media Query Settings

Responsive web design allows websites to respond to different devices. The viewport property allows a website to work with devices of different sizes. The viewport property is not designed to be used on webpages with large images. The image scale will be independent of the webpage scale, and this will make the webpage not scale correctly for a webpage with large images.

A media query will check for the device screen width and will change the styles of the webpage to fit the device screen width. A style sheet can have multiple style layouts with the media query determining which of the styles to use.

## Purpose

Upon completing this project, you will better understand when to use a viewport or media query and be able to use them to create a responsive web design.

## Steps for Completion

1. Open the **54b-spotlight.html** file in the MusicStore5 student folder.
2. In the head tag, add a meta tag that will use the viewport setting.
3. In the meta tag with the viewport name, set the content to have the width adjust to the device width and set the initial scale to **1.0**
4. Save your changes.
5. Preview the webpage in a web browser. As the scale was set to 1.0, the page should have a normal look and feel to it.
6. Open the **54b-main.css** file in the MusicStore5 student folder.
7. Add a media query for screens with a max-width of **500px**
8. Within the new media query, add a body rule set.
9. Within the body rule set, add a background color property with a value of **white**
10. Save your changes.
11. Return to the 54b-spotlight.html webpage.
12. Preview the webpage in a web browser.
13. Narrow the browser width to a setting under 500 pixels. At the 500 pixel mark, the background should turn white.

## Project Details

### Project file

54b-spotlight.html  
54b-main.css

### Estimated completion time

15 minutes

### Video reference

#### Domain 5

**Topic:** Responsive Layouts  
**Subtopic:** Viewport; Media Query

### Objectives covered

#### 5 Webpage Styling using CSS

**5.4** Construct and analyze styles that create a simple responsive layout

**5.4.b** Viewport and media query

**5.4.d** Working with breakpoints

# Percentages and Pixels

It is important that measurements of elements on a webpage do not exceed the screen width of a device. Percentages are used to create fluidity with the element and device screen size. A fixed pixel measurement is needed at times to control the measurements of an element, like a fixed image. Small tables usually work well with a fluid layout using percentages.

## Purpose

Upon completing this project, you will better understand when to use percentages or pixels for units of measurement on a webpage and how to set a max width for these measurements.

## Steps for Completion

1. Open the **54c-hours.html** file in the MusicStore5 student folder.
2. Add an internal style sheet to the head tag, on the line below the link tag.
3. In the internal style sheet, add a rule set for the table tag.
4. Within the table rule set, add a width property with the value set to **400px**
5. Save your changes.
6. Preview the webpage in a web browser.
7. Adjust the size of the web browser. Notice that the table width stays the same despite the size of the page.
8. Return to your code editor.
9. Change the value of the width property from 400px to **50%**
10. Save your changes.
11. Preview the webpage in a browser.
12. Adjust the size of the web browser. The table width will adjust in size to 50% of the browser size.

## Project Details

### Project file

54c-hours.html

### Estimated completion time

15 minutes

### Video reference

#### Domain 5

**Topic:** Responsive Layouts

**Subtopic:** Percentages and Pixels

### Objectives covered

#### 5 Webpage Styling using CSS

**5.4** Construct and analyze styles that create a simple responsive layout

**5.4.a** Units of measurement  
(percentages, pixels, em, vw, vh)

# Frameworks, Templates, and Max Width

A CSS framework is a template that is created or purchased and simplifies building a website. When purchasing a template from the web, be aware of any licensing requirements or restrictions.

Not related to frameworks, but in the final two topics for this course (along with frameworks and templates) is the max width property in CSS. This property can prevent a design from looking awkward, especially for a large screen.

## Purpose

Upon completing this project, you will better understand how to explore CSS frameworks and set a max-width for an element.

## Steps for Completion

1. In a web browser, spend a few minutes looking at CSS frameworks, such as the ones at [csszengarden.com](http://csszengarden.com)
2. When you are done viewing frameworks, open the **54e-hours.html** file in the MusicStore5 student folder.
3. In the table rule set within the internal style sheet, add a max width property with a value of **400px**
4. Save your changes.
5. Preview the webpage in a web browser.
6. Resize the browser back and forth. The table width should never exceed 400 pixels.
7. Close all open files in both your browser and your code editor, saving changes where prompted.

## Project Details

### Project file

54e-hours.html

### Estimated completion time

10 minutes

### Video reference

#### Domain 5

**Topic:** Responsive Layouts

**Subtopic:** Frameworks and Templates; Max-Width Property

### Objectives covered

#### 5 Webpage Styling using CSS

**5.4** Construct and analyze styles that create a simple responsive layout

**5.4.c** Frameworks and templates

**5.4.e** Grids

# Grids

When it comes to designing modern, responsive web pages, one of the most powerful tools CSS has to offer is CSS Grids. In the past, developers relied heavily on tables or floats for positioning, which were not the easiest tools to use for the job. CSS Grids allows developers to take far greater control over how their content displays across devices. The grid system is a series of columns and rows organized within a parent container, or wrapper.

## Purpose

Upon completing this project, you will better understand how to organize webpage content using grids.

## Steps for Completion

1. Open the **grids.html file** in the MusicStore5 student folder.
2. View the webpage in a browser.
3. In the HTML file, change the grid-template-columns property to **33% 33% 33%**
4. View the webpage in a browser and refresh the page.
5. Save your changes.

## Project Details

### Project file

grids.html

### Estimated completion time

5-10 minutes

### Video reference

#### Domain 5

**Topic:** Responsive Layouts

**Subtopic:** Grids

### Objectives covered

#### 5 Webpage Styling using CSS

**5.4** Construct and analyze styles that create a simple responsive layout

**5.4.e** Grids

# HTML and CSS

---

Domain 6

Presented By  
**LK LearnKey®**

# Fill-in-the-Blanks

**Instructions:** While watching Domain 6, fill in the missing words according to the information presented by the instructor. [References are found in the brackets.]

## Construct HTML and CSS

1. Code that does not need to be typed from scratch can help prevent \_\_\_\_\_ in code. [Reusing Rules and Rule Sets]
2. CSS comments start with a \_\_\_\_\_ and end with a \_\_\_\_\_. [CSS Comments]
3. In CSS, the more \_\_\_\_\_ the style, the more likely you will have problems when testing across different browsers. [Multiple Browser Tests]
4. Always take control of the fonts displayed on your website to prevent the users' \_\_\_\_\_ fonts from being used. [Web Safe Fonts]
5. As \_\_\_\_\_ grow in size and complexity, it becomes increasingly more important to test for cross-platform usability. [Cross-Platform Usability]
6. A clear separation between structure and styling makes code easier to edit and \_\_\_\_\_. [Separation of Structure (HTML) and Style (CSS)]

## Accessibility Principles and Evaluate Content Accessibility

7. The \_\_\_\_\_ Guidelines (WCAG) help define a baseline for accessible and adaptable web design. [Text Alternatives]
8. Low color contrast can make text and images difficult to see while high contrast can cause \_\_\_\_\_ and readability issues. [Color Contrast and Usage]
9. Avoid using a font that is too busy as it can be difficult to read at regular and \_\_\_\_\_ font-sizes. [Legibility of Typography]
10. Proper use of the \_\_\_\_\_ attribute and a logical tab order scheme is necessary for users who rely on navigation through that means. [Tab Order]
11. With the \_\_\_\_\_ feature, websites will not typically have text resizing issues because the browser is zooming in on the browser window as a whole. [Text Resizing]
12. Text Hierarchy refers to the use of \_\_\_\_\_ tags to help organize webpages in the most logical way. [Text Hierarchy]

## Structural Integrity of HTML and CSS Markup

13. A simple syntax error in \_\_\_\_\_ will cause the remainder of the file to fail. [Syntax Errors]
14. One of the best ways to guard against tag mismatch is by using proper spacing and \_\_\_\_\_ in your HTML. [Tag Mismatch]
15. CSS Specificity is a \_\_\_\_\_ that dictates the precedence of styles at a more granular level than just stylesheet precedence. [Cascading Issues]

# Reusing Rules and Rule Sets

Code that can be reused will save time and limit errors. To reuse code is simply just a matter of copying and pasting the code where it is needed. Of course, you will want to make sure that if you intend to make slight adjustments to code that you are reusing you do so in the correct spots.

## Purpose

Upon completing this project, you will be able to reuse code from one CSS rule set to another.

## Steps for Completion

1. Open the **23a-main.css** file in the Domain 6 Student folder.
2. Create a rule set for the figcaption tag.
3. Fill the rule set for the figcaption tag with the color property and value from the paragraph tag and the font weight and background color from the sale class.
4. Save your changes.
5. Open the **23a-spotlight.html** file.
6. Preview the webpage in a browser. The caption below the image should have a bold, navy font and have an Aqua line highlight.

## Project Details

### Project file

23a-main.css

23a-spotlight.html

### Estimated completion time

5 minutes

### Video reference

#### Domain 6

**Topic:** Construct HTML and CSS

**Subtopic:** Reusing Rules and Rule Sets

### Objectives covered

#### 6 Accessibility, Readability, and Testing

**6.1** Construct well-formed HTML and CSS markup that conforms to industry best practices

**6.1.a** Reusing rules and rule sets

# CSS Comments

Comments are used to explain code and are best practice when multiple people are collaborating on a CSS style sheet. Code or text within a comment will not show on a webpage. Whereas HTML comments start with a <!-- and end with a -->, CSS comments start with a /\* and end with a \*/.

Comments can also be used over existing code to force a style sheet to ignore that code. This can be useful when testing rule sets and property/value combinations on webpages.

## Purpose

Upon completing this project, you will be able to add comments to a CSS style sheet.

## Steps for Completion

1. Open the **23b-main.css** file in the Domain 6 Student folder.
2. Add a comment above the figcaption that says, **The background color extends across the page, so, we have commented it out for now**
3. Within the figcaption rule set, comment the background color property/value combination out so that it will be ignored in the code.
4. Save your changes.
5. Open the **23b-spotlight.html** file.
6. Preview the webpage in a browser. There should be no Aqua line highlighting on the caption below the image.

## Project Details

### Project file

23b-main.css

23b-spotlight.html

### Estimated completion time

5 minutes

### Video reference

#### Domain 6

**Topic:** Construct HTML and CSS

**Subtopic:** CSS Comments

### Objectives covered

#### 6 Accessibility, Readability, and Testing

**6.1** Construct well-formed HTML and CSS markup that conforms to industry best practices

**6.1.b** Commenting

# Multiple Browser Tests and Web Safe Fonts

A webpage style should work on multiple browsers and media types. It is important to use simple styles in CSS so that the styles used are viewable in different browsers. It is good practice to test your webpages in multiple browsers before publishing the webpages, a concept known as cross-platform usability. The more complex the styles, the more likely it is the styles will not work fully.

When working with the font-family property in style sheets, you will want to use multiple fonts as not every user has every font you may want to display. You will also want to indicate the font family being used so that a user's default font for a specific family can be used if needed. This will at least ensure that the font displayed is like the font you intended for that style. This is the concept of using Web Safe fonts, fonts that will properly display in a web browser.

## Purpose

Upon completing this project, you will be able to test webpage styles in multiple browsers. You will also know how to set up a CSS rule set to use Web Safe fonts.

## Steps for Completion

1. Open the **23c-spotlight.html** file in the Domain 6 Student folder.
2. Test the webpage in as many web browsers as you have available.
3. Were there any browsers that the styles in the webpage varied?
  - a. \_\_\_\_\_.
4. Open the **23c-main.css** file.
5. Add a font-family setting to the paragraph tag to set the font to Arial, then Segoe UI, and then the default sans-serif font.
6. Save your changes.
7. Preview the **23c-spotlight.html** page in a web browser. The paragraphs should display as Arial, Segoe UI, or a similar sans-serif font.
8. Save and close all open files, both in your web browser and in your code editor.

## Project Details

### Project file

23c-spotlight.html  
23c-main.css

### Estimated completion time

10 minutes

### Video reference

#### Domain 6

**Topic:** Construct HTML and CSS

**Subtopic:** Multiple Browser Tests; Web Safe Fonts; Cross-Platform Usability

### Objectives covered

#### 6 Accessibility, Readability, and Testing

**6.1** Construct well-formed HTML and CSS markup that conforms to industry best practices

**6.1.c** Web-safe fonts

**6.1.d** Cross-platform usability

# Separation of Structure and Style

Another industry best practice is the separation of structure and style, specifically the separation between HTML and CSS code. A clear separation between structure and styling makes the code easier to edit and debug. Developers create that separation with the use of their styling.

Styling a webpage in the most efficient way can be a difficult task. Users should have a plan, considering exactly where they are adding each style and what effect that will have on the separation of structure and style.

## Purpose

Upon completing this project, you will better understand how to separate structure and style.

## Steps for Completion

1. Use \_\_\_\_\_ as much as possible to keep consistency within a website.
2. While styling an individual page, use an \_\_\_\_\_.
3. \_\_\_\_\_ should be used very sparingly. This method of styling is difficult to reuse across elements and webpages.

## Project Details

### Project file

N/A

### Estimated completion time

5 minutes

### Video reference

#### Domain 6

**Topic:** Construct HTML and CSS

**Subtopic:** Separation of Structure (HTML) and Style (CSS)

### Objectives covered

6 Accessibility, Readability, and Testing

**6.1** Construct well-formed HTML and CSS markup that conforms to industry best practices

**6.1.e** Separation of structure (HTML) and style (CSS)

# Text Alternatives

Developers should design webpages with accessibility in mind. There are several factors to consider, and many can be found in the Web Content Accessibility Guidelines (WCAG). These guidelines help define a baseline for accessible and adaptable web design.

Text alternatives are assigned as alternate text, an HTML element attribute. Developers should avoid inputting alt text that is too long or overly descriptive, as well as alt text that is too short and does not give a user enough context.

## Purpose

Upon completing this project, you will better understand text alternatives within the scope of accessibility.

## Steps for Completion

1. Describe three main reasons for using the alt attribute.

a. \_\_\_\_\_  
\_\_\_\_\_.

b. \_\_\_\_\_  
\_\_\_\_\_.

c. \_\_\_\_\_  
\_\_\_\_\_.

## Project Details

### Project file

N/A

### Estimated completion time

5 minutes

### Video reference

#### Domain 6

**Topic:** Accessibility Principles and Evaluate Content Accessibility

**Subtopic:** Text Alternatives

### Objectives covered

**6** Accessibility, Readability, and Testing

**6.2** Apply accessibility principles and evaluate content accessibility

**6.2.a** Text alternatives

# Color Contrast and Typography

When designing webpages, it is important to be aware of color usage and typography. Not only do these factors affect the aesthetics of a webpage, but the accessibility as well.

Keep in mind that users access webpages with differing screen sizes, screen resolutions, room lighting, and eyesight. For these reasons, it is important to have an established baseline for color contrast best practices. The Web Content Accessibility Guidelines (WCAG) lay out very specific criteria in sections 1.4.3 and 1.4.6 regarding color contrast.

When it comes to typography, there are many different fonts to use, ranging from Courier New/Terminal to Comic-Sans MS, and everything in-between. Some CSS properties to consider when styling text include default font size, font type, font-weight bold, font-style italic, letter-spacing and word-spacing, and line-height.

## Purpose

Upon completing this project, you will better understand the importance of color contrast and font usage on webpages.

## Steps for Completion

1. What negative effects could occur if color contrast is too high?

a. \_\_\_\_\_

2. What negative effects could occur if color contrast is too low?

a. \_\_\_\_\_

3. Avoid using a font that is too busy as it can be difficult to read at regular and reduced font \_\_\_\_\_.

4. To make websites \_\_\_\_\_ and accessible, it is always better to rely on fewer fonts, with each font having a specific purpose in the design.

5. Always rely on the \_\_\_\_\_ property, and multiple values, to ensure that there are backup fonts available to the browser.

## Project Details

### Project file

N/A

### Estimated completion time

5-10 minutes

### Video reference

#### Domain 6

**Topic:** Accessibility Principles and Evaluate Content Accessibility

**Subtopic:** Color Contrast and Usage; Legibility of Typography

### Objectives covered

**6** Accessibility, Readability, and Testing

**6.2** Apply accessibility principles and evaluate content accessibility

**6.2.b** Color contrast and usage

**6.2.c** Legibility of typography

# Tab Order and Text Resizing

Developers can use the tabindex attribute to input tags in their HTML code. When creating a form, the tab order naturally flows from one input field to the next. However, with other page styles such as landing pages, shopping cart pages, or just longer pages in general, it is important to pay attention to the tabindex attribute throughout the code, ensuring that users who rely on the Tab Index to navigate the form can do so in a logical and simple way.

Generally, users resize text in their browsers using one of two methods. The first, is by adjusting the zoom setting in the browser and the second is by changing the appearance settings. With the zoom feature, websites do not usually have text resizing issues because the browser is zooming in on the browser window, not resizing elements in the HTML. The relationship between the elements should stay consistent.

When creating a webpage, one should avoid horizontal scrolling. A webpage that tends to scroll vertically will lend itself better to responsive design and appearance on mobile devices.

## Purpose

Upon completing this project, you will better understand tab order and you will be able to resize text on a webpage.

## Steps for Completion

1. Applying proper tab order will mean different things on different webpages, but the proper use of the tabindex attribute and a logical tab order scheme is necessary for users who rely on \_\_\_\_\_ through that means.
2. Open the **resizing.html** file in the Domain 6 Student folder.
3. View the webpage in a browser.
4. Zoom into the page using the Zoom option in the browser's settings. Enlarge to any preferred size.
5. Reset the size back to 100%.
6. Resize the webpage text to the largest font option using the Appearance options within the browser's settings.

## Project Details

### Project file

resizing.html

### Estimated completion time

5 minutes

### Video reference

#### Domain 6

**Topic:** Accessibility Principles and

Evaluate Content Accessibility

**Subtopic:** Tab Order; Text Resizing

### Objectives covered

#### 6 Accessibility, Readability, and Testing

**6.2** Apply accessibility principles and evaluate content accessibility

**6.2.d** Tab order

**6.2.e** Text resizing

# Text Hierarchy

Text hierarchy refers to the use of HTML tags, semantic or not, to help organize webpages in the most logical way. One way to organize elements is with the heading tag. When numbered, heading tags can be used to structure sections on a webpage. Section tags and other semantic tags promote webpage organization as well. Proper use of all organizational HTML tags will improve a webpage's accessibility.

## Purpose

Upon completing this project, you will better understand how text on a webpage can be structured hierarchically with heading tags.

## Steps for Completion

1. Open the **hierarchy.html** file in the Domain 6 Student folder.
2. Which tag is used to distinguish the main title, LearnKey OnlineExpert? Why?

a. \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_.

3. If you were to list another course adjacent to HTML & CSS, which tag would you use? Why?

a. \_\_\_\_\_  
\_\_\_\_\_.

4. How many times is the h3 tag listed?

a. \_\_\_\_\_.

## Project Details

### Project file

hierarchy.html

### Estimated completion time

5 minutes

### Video reference

#### Domain 6

**Topic:** Accessibility Principles and Evaluate Content Accessibility

**Subtopic:** Text Hierarchy

### Objectives covered

**6** Accessibility, Readability, and Testing

**6.2** Apply accessibility principles and evaluate content accessibility

**6.2.f** Text hierarchy

# Syntax Errors

Even the most experienced programmers encounter syntax errors. No matter one's experience level, syntax errors are part of anyone's programming endeavors. Syntax errors can include missing punctuation, incorrect letter-casing, invalid opening and closing tags or brackets, or incorrect spelling.

Syntax errors in HTML are very forgiving because of how browsers parse the code. Browsers are prepared to run despite syntax errors. When you run HTML with errors, the page will run, though it might not appear as intended. A simple syntax error in CSS will cause the remainder of a file to fail. This is more consistent with how other programming languages react to syntax errors, so it is vitally important to be clear and accurate with your CSS.

## Purpose

Upon completing this project, you will better understand syntax errors in CSS and HTML.

## Steps for Completion

1. Open the **main.css** file in the Domain 6 Student folder.
2. On line 3, set the background color property to background-colors: by adding an **s**
  - a. A squiggly line should appear, indicating that the property is not a known CSS property name.
3. Open the **index.html** file.
4. View the page in a browser.
5. Navigate to the **main.css** file.
6. On line 1, delete the opening bracket.
7. Save the changes.
8. View the webpage in a browser and refresh the page.
  - a. The page should have lost its background color and should look significantly different from the original.

## Project Details

### Project file

index.html  
main.css

### Estimated completion time

5-10 minutes

### Video reference

#### Domain 6

**Topic:** Structural Integrity of HTML and CSS Markup

**Subtopic:** Syntax Errors

### Objectives covered

**6** Accessibility, Readability, and Testing

**6.3** Evaluate the structural integrity of HTML and CSS markup

**6.3.a** Syntax errors

# Tag Mismatch

Once HTML becomes more complex, developers may encounter tag mismatch. Tag mismatch describes a situation where opening and closing tags do not match up properly. It can also refer to a tag that is not opened and closed within another single tag. This can be difficult to troubleshoot because the browser will sometimes just work around the error and render the webpage.

## Purpose

Upon completing this project, you will be able to identify tag mismatch and identify ways to avoid it.

## Steps for Completion

1. Open the **hours-complete.html** file in the Domain 6 Student folder.
2. Scan the code for matching opening and closing tags.
3. Are there any mismatching tags in the file?
  - a. \_\_\_\_\_.
4. Open the **hours-mismatch.html** file.
5. Scan the code for matching opening and closing tags.
6. One of the best ways to guard against tag mismatch is by using proper \_\_\_\_\_ and \_\_\_\_\_ in your HTML code.

## Project Details

### Project file

hours-complete.html  
hours-mismatch.html

### Estimated completion time

5 minutes

### Video reference

#### Domain 6

**Topic:** Structural Integrity of HTML and CSS Markup

**Subtopic:** Tag Mismatch

### Objectives covered

**6** Accessibility, Readability, and Testing

**6.3** Evaluate the structural integrity of

HTML and CSS markup

**6.3.b** Tag mismatch

# Cascading Issues

When there are multiple CSS rules for the same element, one can experience cascading issues. It is not uncommon to have the same property set in two different places for the same element. While sometimes it can be convenient to layer CSS rule sets in this way, it will eventually cause issues. The simplest way to deal with these issues is to use the Developer Tools along with the knowledge of stylesheet precedence.

## Purpose

Upon completing this project, you will better understand cascading issues and how to avoid them.

## Steps for Completion

1. CSS specificity is a hierarchy that dictates the precedence of styles at an even more granular level than \_\_\_\_\_ precedence.
2. CSS has an important rule that can be added after a CSS property/value declaration, like so:

textbox1 { background-color: red !important; }

- a. This keyword will cause the style to \_\_\_\_\_ any styles declared before it.
3. The important keyword should be used with caution. What will happen if this keyword is overused?
    - a. \_\_\_\_\_

## Project Details

### Project file

N/A

### Estimated completion time

5 minutes

### Video reference

#### Domain 6

**Topic:** Structural Integrity of HTML and CSS Markup

**Subtopic:** Cascading Issues

### Objectives covered

#### 6 Accessibility, Readability, and Testing

**6.3** Evaluate the structural integrity of HTML and CSS markup

**6.3.c** Cascading issues

# HTML and CSS

---

## Appendix

Presented By  
**LK LearnKey®**

# Glossary

Term	Definition
<b>A Tag</b>	An anchor (a) tag is used to implement a hyperlink for a user. This means that the user is anchoring himself to another page.
<b>Absolute Link</b>	A link which contains the complete address to another link.
<b>Absolute Positioning</b>	A type of positioning in CSS which places items in an exact location on a webpage.
<b>Action Attribute</b>	An attribute used within a form tag to navigate users to the next page or script once the form is submitted.
<b>Article Tag</b>	A tag which is used to mark off self-containing text that can make up a story, blog entry, or an article.
<b>Aside Tag</b>	A tag which is often used to mark off a side story.
<b>Audio Tag</b>	A tag which can have multiple sources and should have the controls attribute added to allow a user to control when the audio plays.
<b>Background</b>	The color or image used as the backdrop of an entire webpage.
<b>Body Tag</b>	A tag which contains everything that displays on the webpage. Most of the tags used in CSS are placed inside of a body tag.
<b>Border-Color Property</b>	A CSS property which is used to alter the color of a border either by name or using hexadecimal colors.
<b>Border-Style Property</b>	An element used in CSS to control the overall look of a border.
<b>Border-Width Property</b>	A property which is used in CSS to control the width of a border.
<b>Br Tag</b>	A tag which is used to create a line break.
<b>Button Tag</b>	A tag which can be used on a form to add a button option.
<b>Caption Tag</b>	A tag which is used on tables to allow a table description to be placed in close proximity to a table.
<b>Comment</b>	A tool used to put placeholders in for text that will need to be added later to a webpage. It can also be used to explain parts of a webpage that do not display in a browser but will appear in a text editor. It will help any collaborative efforts to build a website.
<b>CSS</b>	Cascading Style Sheets (CSS) is a type of style sheet language which is used in web design to describe the presentation of a document that is written in a markup language, such as HTML5. It is the code used to format the text in a webpage.
<b>CSS Framework</b>	A template which is created by the web designer, downloaded for free, or purchased from a third-party which has built in CSS the designer can use to ease in the development process of a
<b>Datalist Tag</b>	A tag which is used to create a pre-populated list within a form.
<b>Delete</b>	A common submission method which requests to remove information from a database.
<b>Details Tag</b>	The summary tag surrounds the line explaining what is in the details section in a details tag.
<b>Div Tag</b>	A tag which is used in CSS to control alignment, backgrounds, margins, padding, and text flow for webpage elements. It can also be used to control the width of an entire page.
<b>Em</b>	A relative measurement which is equivalent to the font size being used. For example, 3em = 3 x the size of the current font.
<b>Ex</b>	A measurement type which is relative to the height of the letter x.
<b>Fieldset Tag</b>	A tag which creates fieldsets that allow you to group fields on forms to make them easier to read.
<b>Figure Tag</b>	A tag which is used to mark up an image in a webpage.
<b>Float Element</b>	An element which is used to move one item next to another on a webpage.
<b>Font-Family Property</b>	A property which is used in CSS to ensure there are backup fonts and web safe fonts available in the event a user does not have access to the web designers designated fonts.

Term	Definition
<b>Font-Style Property</b>	A font property which is used in CSS to apply style to fonts, such as italics.
<b>Font-Weight Property</b>	A font property which controls the bold level of text on a webpage.
<b>Get</b>	A common submission method which requests to retrieve something, such as information from another webpage, an image, or a search engine.
<b>Head Tag</b>	A tag type which contains information that describes the webpage but does not show on the webpage.
<b>Heading Tags</b>	HTML has six levels of headings, with tags of h1 through h6. H1 is the largest heading while h6 is the smallest.
<b>Hexadecimal Colors</b>	A color format which uses a pound symbol and six hexadecimal numbers, that is, six numbers 0 through 9 or A through F. 0 represents black, while F represents white. The six colors are in three pairs, which signify the amounts of red, green, and blue used in the color.
<b>Hr Tag</b>	A horizontal rule (hr) tag is a tag that creates a paragraph-level break within text on a web document.
<b>Html Tag</b>	A tag which contains data that makes up a webpage.
<b>HTML5</b>	Hypertext Markup Language 5 (HTML5) is a type of markup language used to build and present content on the Internet.
<b>iFrame</b>	A content-holding frame inside of a webpage. It can be altered using programming.
<b>Img Tag</b>	An image (img) tag defines an image in an HTML document, which requires the src and alt attributes.
<b>JavaScript</b>	The dynamic aspect of webpage building where scripts are used to manipulate objects based on an event.
<b>Li Tag</b>	A list item (li) tag goes around every item listed in either an ordered or unordered list.
<b>Link Tag</b>	A tag which allows web designers to link to external files, such as style sheets and scripts.
<b>Margins</b>	The space between the top, bottom, or side of a webpage's content and its edge.
<b>Max-Width Property</b>	A property which is used to control layouts on webpages using CSS. It can ensure that the total width of a webpage does not exceed the screen size of the user.
<b>Media Query</b>	A fairly granular method used to control how a webpage displays across different types of devices. It checks for screen width and changes styles according to that screen width.
<b>Meta Tag</b>	A type of tag used to embed metadata in CSS.
<b>Metadata</b>	A tool used in CSS to describe items and elements and their usage.
<b>Method Attribute</b>	An attribute which is used within a form tag to tell the form what type of action to take once it is submitted.
<b>MP3</b>	A compressed sound file which allows for digital storage and transmission.
<b>MP4</b>	A digital multimedia container often used to store video and audio data.
<b>MPEG</b>	Moving Picture Experts Group (MPEG), also known as the H.264 format, is the most common format for output from video editing software.
<b>Nav Tag</b>	The nav tag is used to help spot, in HTML code, the area on the page containing navigation links on a webpage.
<b>Noscript Tag</b>	A tag which notifies the user if their browser does not support the programming language that is currently in use.
<b>Ogg</b>	An open container format which is designed to provide for efficient streaming of digital multimedia. It is a royalty-free video format.
<b>Ol Tag</b>	An ordered list (ol) tag is a numbered list.
<b>Output Tag</b>	A tag which allows you to take what is in a form and run an instant calculation depending upon what is being input in a form.
<b>Overflow Property</b>	A property which is used to ensure that text will fit on the screen regardless of the device.
<b>P Tag</b>	A tag used to create paragraphs.

Term	Definition
<b>Percentage Measurement</b>	A type of measurement used within CSS. It can be applied to elements on a webpage where it will allow the element to fluctuate its size based on the screen size the user has.
<b>Pixel Measurement</b>	A static form of measurement in CSS. It allows the web designer to assign a specific measurement to a webpage element.
<b>Post</b>	A common submission method which requests to take submitted data and modify data on the server as a result of user input.
<b>Pt</b>	A measurement which is 1/72 of an inch.
<b>Put</b>	A common submission method which requests to take data submitted and insert it into a database.
<b>Px</b>	A measurement which is normally 1/96 of an inch but can vary based on screen sizes.
<b>Relative Link</b>	Hyperlinks used to link pages within a website.
<b>Relative Positioning</b>	A type of positioning in CSS which places items relative to where they normally are in relation to how things normally flow on a webpage.
<b>Responsive Web Design</b>	A concept which refers to designing webpages which can be viewed on different devices, such as desktops, laptops, tablets, and smartphones.
<b>SVG</b>	Scalable Vector Graphics (SVG) is a vector image format for two-dimensional graphics with the ability to support interactivity and animation.
<b>Script Tag</b>	A tag which is used to hold programming scripts for programming languages.
<b>Section Tag</b>	A tag which marks off sections on a webpage.
<b>Select Tag</b>	A form-based tag which creates a drop-down list full of options.
<b>Semantic Tag</b>	A tag which is used to help organize a webpage and can be used to determine how a webpage should be rendered.
<b>Span Tag</b>	A tag which is used to format a word or phrase.
<b>Summary Tag</b>	A semantic tag which can hide the text inside of a details tag. The summary tag surrounds the line explaining what is in the details section.
<b>Tag</b>	A command within an HTML document that specifies the formatting of the document or a portion of that document.
<b>Text-Align Property</b>	An alignment property which is used to align text on a webpage.
<b>Textarea Tag</b>	A tag used to create text boxes within a form so that users can type responses.
<b>Text-Decoration Property</b>	An element which is used in CSS to apply underlines to text on a webpage.
<b>Text-Indent Property</b>	A text property which is used in CSS to indent the first line of text in a multi-line paragraph.
<b>Track Tag</b>	A tag which works in conjunction with vtt files as part of the Web Video Text Tracks (WebVTT) standard.
<b>UI Tag</b>	An unordered list (ul) tag adds a bulleted list with no specific hierarchy to a webpage.
<b>Video Tag</b>	A tag which adds a video to a webpage.
<b>Visibility Property</b>	A property which is used to make an object invisible using CSS.
<b>WebM</b>	A video file format which works well on low-power devices, such as netbooks and tablets.

# Objectives

## HTML and CSS Objectives

Domain 1 <b>HTML Fundamentals</b>	Domain 2 <b>CSS Fundamentals</b>	Domain 3 <b>Document Structure using HTML</b>
1.1 Construct markup that uses metadata elements 1.1.a Script 1.1.b Noscript 1.1.c Style 1.1.d Link 1.1.e Meta tags, including encoding, keywords, viewport, description, and translate	2.1 Analyze and implement inline styles, internal (embedded) style sheets, and external style sheets 2.1.a When to use inline styles, internal (embedded) style sheets, or external style sheets 2.1.b Precedence when using a combination of inline styles and style sheets	3.1 Construct and analyze markup to structure content and organize data 3.1.a Table tags (table, tr, th, td) 3.1.b H1-h6 3.1.c P 3.1.d Br 3.1.e Hr 3.1.f Div 3.1.g Span 3.1.h Ul 3.1.i Ol 3.1.j Li
1.2 Construct well-formed page markup 1.2.a DOCTYPE declaration 1.2.b Html 1.2.c Head 1.2.d Body 1.2.e Proper syntax 1.2.f Closing tags and commonly used symbols	2.2 Construct and analyze rule sets 2.2.a Valid syntax for the CSS rule set 2.2.b Selectors, including class, id, elements, and pseudo-class	3.2 Construct and analyze markup that uses HTML5 semantic elements 3.2.a Semantic tags (header, nav, section, article, aside, footer, details, summary, figure, caption)
		3.3 Construct and analyze markup that implements navigation 3.3.a Target 3.3.b Anchor (a href) 3.3.c Bookmark 3.3.d Relative vs. absolute links 3.3.e Navigating simple folder hierarchies 3.3.f Map 3.3.g Area
		3.4 Construct and analyze markup that uses form elements 3.4.a Form attributes 3.4.b Action 3.4.c Method 3.4.d Submission methods 3.4.e Input types and restrictions 3.4.f Select 3.4.g Textarea 3.4.h Button 3.4.i Option 3.4.j Label

## HTML and CSS Objectives

Domain 4 <b>Multimedia Presentation using HTML</b>	Domain 5 <b>Webpage Styling using CSS</b>	Domain 6 <b>Accessibility, Readability, and Testing</b>
4.1 Construct and analyze markup that displays images 4.1.a Img and picture elements and their attributes	5.1 Construct and analyze styles that position content 5.1.a Positioning, including float, relative, absolute, static, fixed, max-width, overflow, height, width, align 5.1.b Display 5.1.c Inline vs. block 5.1.d Visibility 5.1.e Box model, margins and padding	6.1 Construct well-formed HTML and CSS markup that conforms to industry best practices 6.1.a Reusing rules and rule sets 6.1.b Commenting 6.1.c Web-safe fonts 6.1.d Cross-platform usability 6.1.e Separation of structure (HTML) and style (CSS)
4.2 Construct and analyze markup that plays video and audio 4.2.a Video 4.2.b Audio 4.2.c Track 4.2.d Source 4.2.e Iframe	5.2 Construct and analyze styles that format text 5.2.a Font-family 5.2.b Color 5.2.c Font-style 5.2.d Font-size 5.2.e Font-weight 5.2.f Font-variant 5.2.g Link colors 5.2.h Text formatting, text alignment, text decoration, indentation 5.2.i Line-height 5.2.j Word-wrap 5.2.k Letter-spacing	6.2 Apply accessibility principles and evaluate content accessibility 6.2.a Text alternatives 6.2.b Color contrast and usage 6.2.c Legibility of typography 6.2.d Tab order 6.2.e Text resizing 6.2.f Text hierarchy
	5.3 Construct and analyze styles that format backgrounds and borders 5.3.a Border-color 5.3.b Border-style 5.3.c Border-width 5.3.d Background properties 5.3.e Colors	6.3 Evaluate the structural integrity of HTML and CSS markup 6.3.a Syntax errors 6.3.b Tag mismatch 6.3.c Cascading issues
	5.4 Construct and analyze styles that create a simple responsive layout 5.4.a Units of measurement (percentages, pixels, em, vw, vh) 5.4.b Viewport and media query 5.4.c Frameworks and templates 5.4.d Working with breakpoints 5.4.e Grids	

# HTML and CSS Lesson Plan

Approximately 32 hours of videos, labs, and projects.

## Domain 1 Lesson Plan

Domain 1 - HTML Fundamentals [approximately 2.5 hours of videos, labs, and projects]				
Lesson	Lesson Topic and Subtopics	Objectives	Exercise Labs	Workbook Projects and Files
<b>Pre-Assessment</b> Assessment time - 00:10:00	HTML Fundamentals: Pre-Assessment			
<b>Lesson 1</b> Video time - 00:06:17 Exercise Lab time - 00:04:00 Workbook time - 00:15:00	<b>Markup with Metadata</b> <b>Part 1</b> How to Study for This Exam Script Tag Noscript Tag	1.1.a Script 1.1.b Noscript	Tags	Script and Noscript Tags – pg. 8 11a-scripts.html
<b>Lesson 2</b> Video time - 00:05:49 Exercise Lab time - 00:04:00 Workbook time - 00:15:00	<b>Markup with Metadata</b> <b>Part 2</b> Link Tag Meta Tags	1.1.c Style 1.1.d Link 1.1.e Meta tags, including encoding, keywords, viewport, description, and translate	Using Meta Tags	Link and Meta Tags – pg. 9 11c-scripts.html
<b>Lesson 3</b> Video time - 00:11:05 Exercise Lab time - 00:04:00 Workbook time - 00:20:00	<b>Well-Formed Markup</b> DOCTYPE Declaration and Html Tag Head Tag Body Tag Closing Tags Entities Comments Domain 1 Exam Tips	1.2.a DOCTYPE declaration 1.2.b Html 1.2.c Head 1.2.d Body 1.2.e Proper syntax 1.2.f Closing tags and commonly used symbols	Body Tag Usage	HTML Fundamentals – pg. 10 N/A
<b>Post-Assessment</b> Assessment time - 00:40:00	HTML Fundamentals: Post-Assessment			

# Domain 2 Lesson Plan

Domain 2 - CSS Fundamentals [approximately 2.5 hours of videos, labs, and projects]				
Lesson	Lesson Topic and Subtopics	Objectives	Exercise Labs	Workbook Projects and Files
<b>Pre-Assessment</b> Assessment time - 00:10:00	CSS Fundamentals: Pre-Assessment			
<b>Lesson 1</b> Video time - 00:05:48 Exercise Lab time - 00:00:00 Workbook time - 00:30:00	<b>Define Style Sheets</b> Inline, Internal, and External Styles Style Sheet Precedence	2.1.a When to use inline styles, internal (embedded) style sheets, or external style sheets 2.1.b Precedence when using a combination of inline styles and style sheets	N/A	Creating Style Sheets – pg. 13 21a-index.html Style Sheet Precedence – pg. 14 21b-main.css; 21b-books.html; 21b-index.html
<b>Lesson 2</b> Video time - 00:09:00 Exercise Lab time - 00:08:00 Workbook time - 00:40:00	<b>CSS Rule Sets</b> Valid Syntax Apply Classes Apply IDs Apply Pseudo Classes Domain 2 Exam Tips	2.2.a Valid syntax for the CSS rule set 2.2.b Selectors, including class, id, elements, and pseudo-class	Class Selector Pseudo Class	Valid Syntax for Rule Sets – pg. 15 22a-main.css; 22a-index.html Apply Classes – pg. 16 22b-main.css; 22b-books.html Apply IDs – pg. 17 22b2-main.css; 22b2-spotlight.html Apply Pseudo Classes – pg. 18 22b3-main.css; 22b3-books.html
<b>Post-Assessment</b> Assessment time - 00:30:00	CSS Fundamentals: Post-Assessment			

# Domain 3 Lesson Plan

Domain 3 - Document Structure using HTML [approximately 10.5 hours of videos, labs, and projects]				
Lesson	Lesson Topic and Subtopics	Objectives	Exercise Labs	Workbook Projects and Files
<b>Pre-Assessment</b> Assessment time - 01:00:00	Document Structure using HTML: Pre-Assessment			
<b>Lesson 1</b> Video time - 00:08:53 Exercise Lab time - 00:08:00 Workbook time - 00:30:00	<b>Structure Content Through Markup Part 1</b> Table Tags Table Attributes Heading Tags P Tag Br Tag	3.1.a Table tags (table, tr, th, td) 3.1.b H1-h6 3.1.c P 3.1.d Br	Table Rows Table Attributes	HTML Tables – pg. 22 31a-hours.html; 31a-hours2.html HTML Headings, Paragraphs, and Line Breaks – pg. 23 31b-headings.html; 31b-index.html; webpagetext.txt
<b>Lesson 2</b> Video time - 00:09:36 Exercise Lab time - 00:04:00 Workbook time - 00:35:00	<b>Structure Content Through Markup Part 2</b> Hr Tag Div Tag Span Tag Ul and Li Tags Ol and Li Tags	3.1.e Hr 3.1.f Div 3.1.g Span 3.1.h Ul 3.1.i Ol 3.1.j Li	Using the Div Tag	Hr, Div, and Span Tags – pg. 24 31e-index.html HTML List Tags – pg. 25 31h-Index.html
<b>Lesson 3</b> Video time - 00:10:03 Exercise Lab time - 00:12:00 Workbook time - 00:50:00	<b>Semantic HTML Elements</b> Header Tag Nav Tag Section Tag Article and Aside Tags Footer Tag Details and Summary Tags Figure Tag Caption Tag	3.2.a Semantic tags (header, nav, section, article, aside, footer, details, summary, figure, caption)	Header Tag Determining Asides Determining Footers	Semantic Tags – pg. 26 32a-spotlight.html Details and Summary Tags – pg. 27 32g-books.html Figure and Caption Tags – pg. 28 32i-spotlight.html; 32j-hours2.html
<b>Lesson 4</b> Video time - 00:08:09 Exercise Lab time - 00:04:00 Workbook time - 00:35:00	<b>Navigation Elements Part 1</b> Image Links The A Tag Targets Bookmarks	3.3.a Target 3.3.b Anchor (a href) 3.3.c Bookmark	Missing Image	Image Links – pg. 29 33a-books.html Hyperlinks – pg. 30 33b-index.html; 33b-spotlight.html Bookmarks – pg. 31 33d-books.html
<b>Lesson 5</b> Video time - 00:07:07 Exercise Lab time - 00:16:00 Workbook time - 00:30:00	<b>Navigation</b> Elements Part 1 Relative vs. Absolute Links Folder Hierarchy Navigation Map Area	3.3.d Relative vs. absolute links 3.3.e Navigating simple folder hierarchies 3.3.f Map 3.3.g Area	Relative Link Navigating Folders Name Attribute Image Map Attributes	Relative and Absolute Links – pg. 32 33e-spotlight.html Navigating Folder Hierarchies – pg. 33 33f-links.html Map and Area Tags – pg. 34 books2.html; books3.html

**Domain 3 - Document Structure using HTML [approximately 10.5 hours of videos, labs, and projects]**

Lesson	Lesson Topic and Subtopics	Objectives	Exercise Labs	Workbook Projects and Files
<b>Lesson 6</b> Video time - 00:07:10 Exercise Lab time - 00:00:00 Workbook time - 00:45:00	<b>Form Elements Part 1</b> Action and Method Attributes Submission Methods Methods Accessibility Input Types and Restrictions	3.4.a Form attributes 3.4.b Action 3.4.c Method 3.4.d Submission methods 3.4.e Input types and restrictions	N/A	Action and Method Attributes – pg. 35 34a-contactus.html Accessibility – pg. 36 34d-contactus.html Input Types and Labels – pg. 37 34e-contactus.html
<b>Lesson 7</b> Video time - 00:09:14 Exercise Lab time - 00:08:00 Workbook time - 00:45:00	<b>Form Elements Part 2</b> The Significance of Labels Select Tag Textarea Tag Button Tag Output Tag	3.4.f Select 3.4.g Textarea 3.4.h Button 3.4.j Label	Select Tag Elements Textarea Tag	The Select Tag – pg. 38 34f-contactus.html Textarea Tag – pg. 39 34g-contactus.html Button Tags – pg. 40 34h-contactus.html Output Tags – pg. 41 34i-contactus.html
<b>Lesson 8</b> Video time - 00:08:03 Exercise Lab time - 00:00:00 Workbook time - 00:40:00	<b>Form Elements Part 3</b> Option Tag Datalist Tag Fieldset Tag Domain 3 Test Tips	3.4.i Option	N/A	Option Tag – pg. 42 34j-contactus.html Datalist – pg. 43 34k-contactus.html Fieldset and Legend Tags – pg. 44 34l-contactus.html
<b>Post-Assessment</b> Assessment time - 02:00:00	Document Structure using HTML: Post-Assessment			

# Domain 4 Lesson Plan

Domain 4 - Multimedia Presentation using HTML [approximately 2 hours of videos, labs, and projects]				
Lesson	Lesson Topic and Subtopics	Objectives	Exercise Labs	Workbook Projects and Files
<b>Pre-Assessment</b> Assessment time - 00:10:00	Multimedia Presentation using HTML: Pre-Assessment			
<b>Lesson 1</b> Video time - 00:05:12 Exercise Lab time - 00:04:00 Workbook time - 00:25:00	<b>Image, SVG, and Canvas Elements</b> Add Images Image Layouts Image Links	4.1.a Img and picture elements and their attributes	Understanding Image Elements	Add Images – pg. 47 41a-instruments.html; mandolin.jpg; trombone.jpg Image Layouts and Links – pg. 48 41a-instruments2.html
<b>Lesson 2</b> Video time - 00:08:33 Exercise Lab time - 00:00:00 Workbook time - 30:00:00	<b>Video and Audio Formats</b> Video and Source Tags Audio and Source Tags Track Tag iFrame Implementations Domain 4 Exam Tips	4.2.a Video 4.2.b Audio 4.2.c Track 4.2.d Source 4.2.e Iframe	N/A	Video and Source Tags – pg. 49 43a-media.html; Happy.mp4; Happy.mov Audio and Track Tags – pg. 50 43b-media.html; metronome_audio.mp3; subtitles.vtt Simple Iframe Implementations – pg. 51 43e-media.html; vtt.html
<b>Post-Assessment</b> Assessment time - 00:40:00	Multimedia Presentation using HTML: Post-Assessment			

# Domain 5 Lesson Plan

Domain 5 - Webpage Styling using CSS [approximately 10.5 hours of videos, labs, and projects]				
Lesson	Lesson Topic and Subtopics	Objectives	Exercise Labs	Workbook Projects and Files
<b>Pre-Assessment</b> Assessment time - 00:40:00	Webpage Styling using CSS: Pre-Assessment			
<b>Lesson 1</b> Video time - 00:09:58 Exercise Lab time - 00:04:00 Workbook time - 00:40:00	<b>Position Text with CSS</b> <b>Part 1</b> Height and Width Max-Width Used with an Element Float Positioning Relative Positioning Absolute Positioning	5.1.a Positioning, including float, relative, absolute, static, fixed, max-width, overflow, height, width, align	Positioning	Height and Max-Width Used with an Element – pg. 55 51a-positioningexample.html Float Positioning – pg. 56 51af-positioningexample.html Relative and Absolute Positioning – pg. 57 51ar-positioningexample.html
<b>Lesson 2</b> Video time - 00:10:13 Exercise Lab time - 00:04:00 Workbook time - 00:35:00	<b>Position Text with CSS</b> <b>Part 2</b> Static Fixed Overflow Settings Alignment Display Inline vs. Block	5.1.a Positioning, including float, relative, absolute, static, fixed, max-width, overflow, height, width, align 5.1.b Display 5.1.c Inline vs. block	Element Position	Static and Fixed – pg. 58 positioningelement1.html Overflow and Alignment Settings – pg. 59 51ao-positioningexample.html Display – pg. 60 display.html Inline vs. Block – pg. 61 51b-main.css; 51b-spotlight.html
<b>Lesson 3</b> Video time - 00:08:39 Exercise Lab time - 00:00:00 Workbook time - 00:25:00	<b>Position Text with CSS</b> <b>Part 3</b> Visibility Box Model Margins Box Model Padding	5.1.d Visibility 5.1.e Box model, margins and padding	N/A	Visibility – pg. 62 51c-positioningexample.html Box Model Margins and Padding – pg. 63 51d-positioningexample.html
<b>Lesson 4</b> Video time - 00:08:44 Exercise Lab time - 00:04:00 Workbook time - 00:30:00	<b>Format Text with CSS</b> <b>Part 1</b> Font-Family Color Font-Style Font-Weight	5.2.a Font-family 5.2.b Color 5.2.c Font-style 5.2.d Font-size 5.2.e Font-weight	Hexadecimal Colors	Font Families in CSS – pg. 64 52a-main.css; 52a-books.html Font Colors in CSS – pg. 65 52b-main.css; 52b-spotlight.html Font Styles and Weights in CSS – pg. 66 52c-main.css; 52c-books.html
<b>Lesson 5</b> Video time - 00:11:10 Exercise Lab time - 00:08:00 Workbook time - 00:40:00	<b>Format Text with CSS</b> <b>Part 2</b> Link Colors Font Variant Text-Alignment Text-Decoration Indentation	5.2.f Font-variant 5.2.g Link colors 5.2.h Text formatting, text alignment, text decoration, indentation	Font Variants Hyperlink Underline	Link Colors in CSS – pg. 67 52e-main.css; 52e-spotlight.html Font Variant – pg. 68 headings1.html; main.css Text Alignment and Decoration – pg. 69 52f-main.css; 52f-hours.html Indentation – pg. 70 52fi-main.css; 52fi-index.html
<b>Lesson 6</b> Video time - 00:03:41 Exercise Lab time - 00:00:00 Workbook time - 00:25:00	<b>Format Text with CSS</b> <b>Part 3</b> Line Height Word-wrap Letter-Spacing, Word-Spacing	5.2.i Line-height 5.2.j Word-wrap 5.2.k Letter-spacing	N/A	Line Height and Word-Wrap – pg. 71 positioningexample2.html; wordwrap.html Letter-Spacing, Word-Spacing – pg. 72 spacing.html

## Domain 5 - Webpage Styling using CSS [approximately 10.5 hours of videos, labs, and projects]

Lesson	Lesson Topic and Subtopics	Objectives	Exercise Labs	Workbook Projects and Files
<b>Lesson 7</b> Video time - 00:06:54 Exercise Lab time - 00:00:00 Workbook time - 00:35:00	<b>Backgrounds and Borders Part 1</b> Border-Color Border-Style Border-Width	5.3.a Border-color 5.3.b Border-style 5.3.c Border-width	N/A	Border Color and Style – pg. 73 53a-main.css; 53a-spotlight.html Border Width – pg. 74 53c-main.css; 53c-spotlight.html
<b>Lesson 8</b> Video time - 00:07:25 Exercise Lab time - 00:04:00 Workbook time - 00:25:00	<b>Backgrounds and Borders Part 2</b> Backgrounds Divs Working with Colors	5.3.d Background properties 5.3.e Colors	Text Color	Backgrounds – pg. 75 53d-positioningexample.html; kazoo-transparent.jpg Divs and Colors – pg. 76 53e-main.css; 53e-spotlight.html
<b>Lesson 9</b> Video time - 00:07:07 Exercise Lab time - 00:04:00 Workbook time - 00:40:00	<b>Responsive Layouts Part 1</b> Units of Measure Viewport Media Query Percentages and Pixels	5.4.a Units of measurement (percentages, pixels, em, vw, vh) 5.4.b Viewport and media query 5.4.d Working with breakpoints	Relative Measurement	Units of Measure – pg. 77 54a-main.css; 54a-index.html Viewport and Media Query Settings – pg. 78 54b-spotlight.html; 54b-main.css Percentages and Pixels – pg. 79 54c-hours.html
<b>Lesson 10</b> Video time - 00:06:13 Exercise Lab time - 00:04:00 Workbook time - 00:25:00	<b>Responsive Layouts Part 2</b> Frameworks and Templates Max-Width Property Grids Domain 5 Exam Tips	5.4.c Frameworks and templates 5.4.e Grids	Grid Columns	Frameworks, Templates, and Max Width – pg. 80 54e-hours.html Grids – pg. 81 grids.html
<b>Post-Assessment</b> Assessment time - 02:00:00	Webpage Styling using CSS: Post-Assessment			

# Domain 6 Lesson Plan

Domain 6 - Accessibility, Readability, and Testing [approximately 4 hours of videos, labs, and projects]				
Lesson	Lesson Topic and Subtopics	Objectives	Exercise Labs	Workbook Projects and Files
<b>Pre-Assessment</b> Assessment time - 00:20:00	Accessibility, Readability, and Testing: Pre-Assessment			
<b>Lesson 1</b> Video time - 00:10:38 Exercise Lab time - 00:08:00 Workbook time - 00:30:00	<b>Construct HTML and CSS</b> Ensuring Accessibility Reusing Rules and Rule Sets CSS Comments Multiple Browser Tests Web Safe Fonts Cross-Platform Usability Separation of Structure (HTML) and Style (CSS)	6.1.a Reusing rules and rule sets 6.1.b Commenting 6.1.c Web-safe fonts 6.1.d Cross-platform usability 6.1.e Separation of structure (HTML) and style (CSS)	Comments Testing Browsers	Reusing Rules and Rule Sets – pg. 84 23a-main.css; 23a-spotlight.html CSS Comments – pg. 85 23b-main.css; 23b-spotlight.html Multiple Browser Tests and Web Safe Fonts – pg. 86 23c-spotlight.html; 23c-main.css Separation of Structure and Style – pg. 87 N/A
<b>Lesson 2</b> Video time - 00:06:02 Exercise Lab time - 00:00:00 Workbook time - 00:20:00	<b>Accessibility Principles and Evaluate Content Accessibility Part 1</b> Text Alternatives Color Contrast and Usage Legibility of Typography	6.2.a Text alternatives 6.2.b Color contrast and usage 6.2.c Legibility of typography	N/A	Text Alternatives – pg. 88 N/A Color Contrast and Typography – pg. #89 N/A
<b>Lesson 3</b> Video time - 00:06:16 Exercise Lab time - 00:08:00 Workbook time - 00:15:00	<b>Accessibility Principles and Evaluate Content Accessibility Part 2</b> Tab Order Text Resizing Text Hierarchy	6.2.d Tab order 6.2.e Text resizing 6.2.f Text hierarchy	Change Font Size Text Hierarchy	Tab Order and Text Resizing – pg. 90 resizing.html Text Hierarchy – pg. 91 hierarchy.html
<b>Lesson 4</b> Video time - 00:07:18 Exercise Lab time - 00:04:00 Workbook time - 00:25:00	<b>Structural Integrity of HTML and CSS Markup</b> Syntax Errors Tag Mismatch Cascading Issues Final Exam Tips Final Thoughts	6.3.a Syntax errors 6.3.b Tag mismatch 6.3.c Cascading issues	CSS Error	Syntax Errors – pg. 92 index.html; main.css Tag Mismatch – pg. 93 hours-complete.html; hours-mismatch.html Cascading Issues – pg. 94 N/A
<b>Post-Assessment</b> Assessment time - 01:00:00	Accessibility, Readability, and Testing: Post-Assessment			