Syllabus

Course Overview

This course emphasizes fundamental Web development skills. It provides a comprehensive overview of the processes to develop a basic static Web site using a Web development tool. You will design and develop Web pages using commonly accepted visual, architectural, and navigational conventions and well-structured, W3C-compliant HTML5 and Cascading Style Sheets (CSS3) code. The site will include multimedia, forms, and other content.

Course Competencies

(Read Only)

To successfully complete this course, you will be expected to:

1. Write HTML5 code that properly renders a page in a Web browser.
2. Design Web pages using commonly accepted visual, architectural, and navigational conventions.
3. Apply styles to Web pages using CSS3.
4. Communicate effectively.

Course Prerequisites

Pre-requisite or co-requisite: IT3006

Syllabus >> Course Materials

Required

The materials listed below are required to complete the learning activities in this course.

Integrated Materials

Many of your required books are available via the VitalSource Bookshelf link in the courseroom, located in your Course Tools. Registered learners in a Resource Kit program can access these materials using the courseroom link on the Friday before the course start date. Some materials are available only in hard-copy format or by using an access code. For these materials, you will receive an email with further instructions for access. Visit the Course Materials page on Campus for more information.

Book

Carey, P. M. (2018). *Newperspectives HTML5 and CSS3: Comprehensive* (7th ed.). Boston, MA: Cengage Learning. ISBN: 9781305503939.

Suggested

The following materials are recommended to provide you with a better understanding of the topics in this course. These materials are not required to complete the course, but they are aligned to course activities and assessments and are highly recommended for your use.

Optional

The following optional materials are offered to provide you with a better understanding of the topics in this course. These materials are not required to complete the course.

Library

The following optional videos and tutorials are available from Skillsoft via the Capella University Library.

- Calnan, C. (n.d.). Introduction to HTML5 and CSS [Tutorial]. null
- Campbell, J. (2014). HTML5: Browser best practices [Video]. Skillsoft Ireland.
- Campbell, J. (2014). HTML5: How to access form data [Video]. Skillsoft Ireland.
- Campbell, J. (2015). Responsive web design: The CSS3 flexbox [Video]. Skillsoft Ireland.
- Lassoff, M. (2012). HTML 5: Creating a basic document structure [Video]. Skillsoft Ireland.
- Lassoff, M. (2012). HTML5: Form tag [Video]. Skillsoft Ireland.
- Lassoff, M. (2012). HTML5: Playing audio files [Video]. Skillsoft Ireland.
- Lassoff, M. (2012). HTML5: Playing video files [Video]. Skillsoft Ireland.
- Lassoff, M. (2013). CSS3 development: Making images responsive [Video]. Skillsoft Ireland.
- Shen, W. (2017). HTML5, JavaScript, and CSS3: Background images [Video]. Skillsoft Ireland.
- Shen, W. (2017). HTML5, JavaScript, and CSS3: HTML Audio and Video [Video]. Skillsoft Ireland.
- Skillsoft. (n.d.). Creating styles with CSS3 and styling text [Tutorial]. null
- Skillsoft. (n.d.). HTML5 and CSS3 [Tutorial]. null
- Skillsoft. (n.d.). HTML5 APIs and media [Tutorial].
- Skillsoft. (n.d.). HTML5 forms [Tutorial]. null
- Skillsoft. (n.d.). HTML5 Layout with CSS3 [Tutorial]. null

External Resource

Please note that URLs change frequently. While the URLs were current when this course was designed, some may no longer be valid. If you cannot access a specific link, contact your instructor for an alternative URL. Permissions for the following links have been either granted or deemed appropriate for educational use at the time of course publication.

- 2FreeHosting.com. (n.d.). Retrieved from https://www.2freehosting.com/
- Apple. (n.d.). About HTML5 audio and video. Retrieved from https://developer.apple.com/library/content/documentation/AudioVideo/Conceptual/Using_HTML5_Audio_Video/Introduction/Introduction.html
- Brackets. (n.d.). Retrieved from http://brackets.io/
- Codementor. (n.d.). 4 different HTML/CSS layout techniques to create a site. Retrieved from https://www.codementor.io/css/tutorial/4-different-layout-techniques-create-single-page
- CSS Library. (n.d.). Liquid layouts. Retrieved from http://www.dynamicdrive.com/style/layouts/category/C13/
- CSS Zen Garden. (n.d.). Retrieved from http://www.csszengarden.com/
- Freehosting.com. (n.d.). Knowledgebase. Retrieved from https://www.freehosting.com/client/knowledgebase.php
- Freehosting.com. (n.d.). Retrieved from https://www.freehosting.com/
- Google Fonts. (n.d.). Retrieved from https://fonts.google.com/
- HTML Dog. (n.d.). HTML beginner tutorial. Retrieved from http://htmldog.com/guides/html/beginner/
- HTML.com. (n.d.). Learn to code HTML & CSS. Retrieved from http://html.com/
- HTML5Test. (n.d.). Retrieved from https://html5test.com/
- Learn CSS Layout. (n.d.). Retrieved from http://learnlayout.com/
- LePage, P. (2014, January 9). HTML5 video. Retrieved from https://www.html5rocks.com/en/tutorials/video/basics/
- MakeAThumbnail.com. (n.d.). Retrieved from http://www.makeathumbnail.com/
- Mifsud, J. (2011, November 8). An extensive guide to Web form usability. Retrieved from https://www.smashingmagazine.com/2011/11/extensive-guide-web-form-usability/
- Mozilla Developer Network. (n.d.). How to structure an HTML form. Retrieved from https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/How_to_structure_an_HTML_form
- Mozilla Developer Network. (n.d.). Sending form data. Retrieved from http://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Sending_and_retrieving_form_data
- ResizeImage.net. (n.d.). Retrieved from http://resizeimage.net/
- Segal, N. (n.d.). HTML5 audio and video. Retrieved from http://www.htmlgoodies.com/html5/other/html5-audio-and-video.html
- The New Boston. (Producer). (n.d.). *HTML5 tutorial – 1 – Introduction* [Video]. | Transcript Retrieved from https://thenewboston.com/videos.php?cat=43&video=18818/
- TutorialsPoint. (n.d.). HTML tutorial. Retrieved from https://www.tutorialspoint.com/html/
- TutorialsPoint. (n.d.). HTML5 – Audio & video. Retrieved from http://www.tutorialspoint.com/html5/html5_audio_video.htm
- UXMatters. (2010, March 21). Pagination in Web forms | Evaluating the effectiveness of Web forms. Retrieved from http://www.uxmatters.com/mt/archives/2010/03/pagination-in-web-forms-evaluating-the-effectiveness-of-web-forms.php
- UXMatters. (2012, May 7). 7 basic best practices for buttons. Retrieved from http://www.uxmatters.com/mt/archives/2012/05/7-basic-best-practices-for-buttons.php

- W3C. (n.d.). CSS validation service. Retrieved from https://jigsaw.w3.org/css-validator/
- W3C. (n.d.). Markup validation service. Retrieved from https://validator.w3.org/
- W3C. (n.d.). Retrieved from https://www.w3.org/
- W3Schools.com. (n.d.). CSS introduction. Retrieved from http://www.w3schools.com/css/css_intro.asp
- W3Schools.com. (n.d.). CSS navigation bar. Retrieved from http://www.w3schools.com/css/css_navbar.asp
- W3Schools.com. (n.d.). CSS3 flexbox. Retrieved from http://www.w3schools.com/css/css3_flexbox.asp
- W3Schools.com. (n.d.). HTML audio/video DOM reference. Retrieved from http://www.w3schools.com/tags/ref_av_dom.asp
- W3Schools.com. (n.d.). HTML layouts. Retrieved from http://www.w3schools.com/html/html_layout.asp
- W3Schools.com. (n.d.). HTML YouTube videos. Retrieved from http://www.w3schools.com/html/html_youtube.asp
- W3Schools.com. (n.d.). HTML5 semantic elements. Retrieved from http://www.w3schools.com/HTML/html5_semantic_elements.asp
- W3Schools.com. (n.d.). HTML5 style guide and coding conventions. Retrieved from http://www.w3schools.com/html/html5_syntax.asp
- W3Schools.com. (n.d.). HTML5 tutorial. Retrieved from http://www.w3schools.com/html/
- W3Schools.com. (n.d.). PHP 5 form handling. Retrieved from http://www.w3schools.com/php/php_forms.asp
- W3Schools.com. (n.d.). PHP 5 tutorial. Retrieved from http://www.w3schools.com/php/default.asp
- Weakly, R. (2003, December 30). Liquid layouts the easy way. Retrieved from http://maxdesign.com.au/articles/liquid/
- Williams, P. (n.d.). Using <div> tags for layout. Retrieved from http://www.beginnersguidetohtml.com/guides/css/layout/

Projects

Project >> Building a Web Site

Project Overview

Your course project is to build a Web site for an organization of your choosing. It is designed to allow you to develop and display your coding skills and knowledge of HTML5 and CSS3. It is also designed to guide you in developing the foundations for a simple Web site. Each step builds on a prior step so it is crucial to stay on task. In the course assignments, you will develop various key components such as HTML semantics, forms, video, audio, CSS layout, CSS effects, template pages, and image galleries. Your Web site is built during a succession of five assignments with topics that include:

- Building home and template pages using HTML5.
- Creating and adding an external CSS file and site navigation.
- Page layout using CSS3.
- Adding Media.
- Adding Forms.

After each assignment, you will publish your site to a Web host.

Note: This course requires to you to use a text editor to complete your work. There are many free open source options on the Internet from which you may choose. One such tool is *Brackets*. It can be freely downloaded at Brackets Web site.

Expectations

The code that you use to create your work should conform to the following expectations:

- Pages should render properly in at least two of the following browsers: Chrome, Firefox, Edge, or Safari.
- Code should use all of the tags specified in the directions.
- Code should be verified to be error free, well documented with comments, and constructed using industry best practices and conventions.

Unit 1 >> Web Page Basics and HTML5

Introduction

Since the inception of the World Wide Web, HTML has been the basic building block for all Web pages. As the Internet grew more advanced and its users began to ask more from the technology in terms of both flexibility and design, markup languages required more advanced features. Users wanted Web sites that were easy to use and that looked good on a variety of devices. Additionally, they wanted to be able to share and reuse data across Web sites.

Although HTML has been utilized to code Web sites from the inception of the Internet in the 1990s, versions of HTML have required updates over the years to remain current with the standards and the new demands and opportunities associated with today's multimedia elements.

HTML5 and CSS3 represent the building blocks of the most modern, up-to-date Web sites. In this course, we use these tools to build Web sites that are usable and communicate effectively with users.

In your first assignment, you will start building your Web site by creating a homepage using some of the key elements that define a Web page. This includes the root element (HTML), may include document metadata elements (head, title, base, link, meta, et cetera), section elements (body, article, section, nav, aside, header, footer, address, et cetera), grouping content elements (p, hr, ol, ul, li, div, et cetera), embedded content elements (img, video, and audio) and text-level elements (a, em, strong, small, cite, et cetera). Do not worry if this sounds like a lot to you as we will only be introducing some of the elements at a time.

Once you have built your homepage, you will have a better idea of the consistent elements that you will need across all of your pages. Typically, Web pages have some aspects that will be the same from page to page so that a viewer does not need to relearn how to navigate a Web site. These components typically include the header (consists of the logo and organization name), main navigation (consists of the main pages or categories of the page), main content area (where the bulk of the information about the page is located), and the footer (consists of secondary navigation, address, copyright information, et cetera). The main content area is generally what will be changing from page to page while the rest of the other areas will most likely be the same.

This is where templates become invaluable. By simply duplicating the template file, you can quickly create all of your pages leaving only content to add to finish off each of the pages. As you are creating your homepage, you will want to be cognizant of the fact that you will be using the homepage to then create your template file. You will do this by removing the main content area and leaving just the various page elements that should be displayed on each of the other pages.

Learning Activities

u01s1 - Studies

Readings

Use your *New Perspectives HTML5 and CSS3* text to read the following:

- Tutorial 1, "Getting Started With HTML5," pages 1–70.

Walkthroughs and Exemplars

You may view the following walkthroughs and exemplars to help you understand the concepts addressed in this unit:

- Setting Up and Using Brackets.io Walkthrough.
- Setting up and Using Brackets Exemplar [ZIP].
- Create a Simple Web Page Using HTML Walkthrough.
- Create a Simple Web Page Using HTML Exemplar [ZIP].
- Best Practices of Writing HTML Codes Walkthrough.
- Best Practices of Writing HTML Codes Exemplar [ZIP].
- How to Test an HTML File in Web Browser Walkthrough.
- How to Test an HTML File in Web Browser Exemplar [ZIP].
- Validating HTML Documents Walkthrough.
- Validating HTML Documents Exemplar [ZIP].
- Importance of Comments in HTML Walkthrough.
- Importance of Comments in HTML Exemplar [ZIP].

Optional Readings

You may choose to complete the following Internet resources:

World Wide Consortium

Review the following Web site in preparation for the first discussion in this unit:

- W3C. (n.d.). Retrieved from https://www.w3.org/
  - World Wide Consortium (W3C) is an international community that develops open standards to ensure the long-term growth of the Internet.

HTML Basics

Review the following resources related to HTML:

- The New Boston. (Producer). (n.d.). *HTML5 tutorial – 1 – Introduction* [Video]. Retrieved from https://thenewboston.com/videos.php?cat=43&video=18818/
- HTML Dog. (n.d.). HTML beginner tutorial. Retrieved from http://htmldog.com/guides/html/beginner/
  - This page provides HTML tutorial assuming that you have no previous knowledge of HTML or CSS.
- W3Schools.com. (n.d.). HTML5 tutorial. Retrieved from http://www.w3schools.com/html/
  - This page provides basic HTML5 tutorials.
- W3Schools.com. (n.d.). HTML5 semantic elements. Retrieved from http://www.w3schools.com/HTML/html5_semantic_elements.asp
  - This page describes semantic elements in HTML5 to aid in the understanding of tags.
- TutorialsPoint. (n.d.). HTML tutorial. Retrieved from https://www.tutorialspoint.com/html/
  - This tutorial is designed for the aspiring Web designers and developers with a need to understand the HTML in enough detail along with its simple overview and practical examples.
- HTML.com. (n.d.). Learn to code HTML & CSS. Retrieved from http://html.com/
  - This page offers basic HTML tutorials.

Browser and Hosting

Review the following resources related to browser and hosting:

- HTML5Test. (n.d.). Retrieved from https://html5test.com/
  - This is a browser testing service to see how well your browser scores in HTML5 compatibility. **Important**: Please make sure you are using the latest versions of your browsers.
- Freehosting.com. (n.d.). Knowledgebase. Retrieved from https://www.freehosting.com/client/knowledgebase.php
  - This page has links to articles to help with uploading files to Freehosting.com servers.

Best Practices

Review the following resource for best practices in HTML coding:

- W3Schools.com. (n.d.). HTML5 style guide and coding conventions. Retrieved from http://www.w3schools.com/html/html5_syntax.asp

Hosting and Validation Services

Review the following resources related to hosting and validation:

- 2FreeHosting.com. (n.d.). Retrieved from https://www.2freehosting.com/
- Freehosting.com. (n.d.). Retrieved from https://www.freehosting.com/
- W3C. (n.d.). Markup validation service. Retrieved from https://validator.w3.org/

Text Editor

Review the following free open source text editor, which you can use to complete the assignments in this course:

- Brackets. (n.d.). Retrieved from http://brackets.io/

Optional Skillsoft Resources

- Calnan, C. (n.d.). Introduction to HTML5 and CSS [Tutorial].
- Lassoff, M. (2012). HTML 5: Creating a basic document structure [Video]. Skillsoft Ireland.

u01s1 - Learning Components

- Validate and test HTML code.
- Examine the basic structure of HTML code and how to properly include comments.
- Study examples of developer comments.
- Examine principles of semantic layout and how page elements are interrelated.
- Identify types and functions of page elements.
- Examine the behaviors and functions of HTML tags.
- View working examples of proper code used to add a Web page element.

u01a1 - Creating the Home and Template Pages

Overview

In this assignment, you will start building your Web site for your fictional organization by creating a homepage using HTML5 and some of the key elements that define a Web page. You are required to use either a simple text editor to write your code, or an enhanced text editor such as *Brackets*.

> **Note:** Microsoft Word is not a good tool for developing code because it is a document processor and not a text editor as it adds other formatting to the background code.

Once you have built your homepage, you will have a better idea of the consistent elements that you will need across all of your pages. Typically, Web pages have some aspects that will be the same from page to page so that a user does not need to relearn how to navigate a Web site. These components typically include the header (consists of the logo and organization name), main navigation (consists of the main pages or categories of the page), main content area (where the bulk of the information about the page is located), and the footer (consists of secondary navigation, address, copyright information, et cetera).

The creation and reuse of these common elements is where templates become invaluable. By simply duplicating the template file, you can quickly create all of your pages leaving only content to add to finish off each of the pages. As you are creating your homepage, you will want to be cognizant of the fact that you will be using it to create your template file.

Once completed, you will publish your pages with a Web host of your choosing.

> **Note**: The Web host that you choose must use PHP.

Expectations

The code that you use to create your page (and during this course in general) should conform to the following expectations:

- Pages should render properly in at least two of the following browsers: Chrome, Firefox, Edge, and Safari.
- Code should use all of the tags specified in the directions.
- Code should be verified to be error free, well documented with comments, and appropriately constructed.

> **Hint**: Consider creating a mock-up or wireframe that depicts the precise layout of your homepage. It will act as a useful guide as you develop your site.

> **Note:** This course requires to you to use a text editor to complete your work. There are many free open source options on the Internet from which you may choose. One such tool is *Brackets*. It can be freely downloaded from the Brackets Web site.

Directions

Using a text editor, create an HTML5-compliant homepage with a file name of index.htm. The page should include the following page sections:

- Header: Include the organization name and logo.
- Navigation: Include links to all site pages (including the homepage).
- Content: This section will later be customized for each of the site pages. For the homepage, it should, at a minimum, consist of information that introduces the organization.
- Footer: Include secondary navigation, copyright notice, or other text of your choosing.

Be sure to:

- Include the following tags in the code: title, article, section, head, header, html, body, footer, nav, and doctype.
- Make sure to include developer comments for each page section to explain or describe the coding.
- Include a comment block at the top of each of your Web pages with your name, the date, the file name, and a short blurb about what the page will consist of.

Save a copy of the homepage and name it template.html. Remove the content that you placed in the "Content" section. This file will serve as a template to be used later to create the other four pages of your site.

Make sure to do the following:

- Submit error-free HTML5 code.
- Create a Web page that demonstrates the accurate use of specified HTML5 tags.
- Create Web page elements that accurately reflect design requirements.
- Write developer comments to describe and explain code.

Once completed, view your pages in your selected Web browsers to see if the content renders appropriately and consistently within each. Next, ensure that your HTML code is HTML5 compliant. You may use the Markup Validation Service from W3C for testing this. **Important**: Take a screen capture of your validation results and save it for submission.

> **Note**:Errors from the validation should be fixed. However, warnings do not need to be fixed but should be reviewed. Fix the issues from the top down as one error can cascade into 20 other errors.

> **Hint**: If you are using Brackets as your enhanced text editor, there is a live preview feature that will allow you to see the results of your code as you enter it in. This will help you to quickly identify if you may be missing a closing tag or an attribute.

Submission Requirements

Submit your work in the courseroom using a single zip file containing the following:

- index.html.

- template.html.
- image_and_url.doc. Include a Word document with:
  - URLs for each of your pages.
  - Names of the browsers you used to view the pages.
  - Pasted image from your validation screen capture.
- Any ancillary files such as graphic or photos you may have decided to include.

Upload your Web site files to a hosting service of your choice. Two free sites to consider are Freehosting.com and 2FreeHosting.com.

> **Note**: If you are uncertain on how to upload your files to a Web host, you can find assistance in the resources given in this unit's studies.

Course Resources
Markup Validation Service
2FreeHosting.com
Brackets
Freehosting.com

u01d1 - HTML Editors

Traditional Web page development used to be limited to the use of simple text editor such Notepad. Technological advances introduced other software tools (such as Microsoft Expression Web) to make developing Web pages easier. The W3C calls the first set of tools *text editors* or *HTML editors* while the second set of tools are called *visual* or *WYSIWYG editors*. The W3C Web site also lists pros and cons for using each set of tools.

For this discussion, you are encouraged to visit the W3C Web site; find standards, browsers, and authoring tools; and research the use of these two sets of tools. You also need to search the Internet and find free HTML editors to answer the following points:

- List the pros and cons of developing Web pages using simple text editors versus visual editors.
- Select an HTML editor that you like and explain your reasons for selecting it.
- Discuss the tool you are currently using to build your Web page. (Make sure you are using an HTML editor.)

Response Guidelines

Comment on the posts of at least two other learners. Offer insights, solutions, examples, or opinions that add depth and value to the conversation.

Note regarding discussions in this course: The content topic should determine the length of your post, however, a minimum of 150 words is recommended. Refer to the Discussion Participation Scoring Guide for posting expectations. **Make your initial posts by Wednesday** to allow time sufficient for peers to respond. The expectation within the course discussions is to respond to at least two posts by the end of the unit, but it is highly recommended that you extend the dialog further. Responding over multiple days will help stimulate a lively discussion.

Course Resources
Undergraduate Discussion Participation Scoring Guide

W3C

u01d2 - Challenges to Creating Your First Page

Note: The second discussion in each unit of this course is dedicated to community sharing of challenges and solutions you have encountered while completing your weekly assignment. You are encouraged to post and respond early in the week to share experiences that you are having, and to help others with theirs. This is how coders solve tough issues. They rely on one another.

Coding your very first Web page can have a lot of challenges, such as using an HTML editor for the first time, testing a page in a browser, uploading files to a Web host, or knowing how to order page elements, et cetera.

Discuss one or more of the challenges or experiences that you encountered when trying to complete this unit's assignment. How did you try to address it or solve it?

Response Guidelines

Comment on the posts of at least two other learners. Share approaches or techniques that you might suggest to help the learner to address his or her challenge.

Course Resources

Undergraduate Discussion Participation Scoring Guide

Unit 2 » Fundamentals of CSS

Introduction

Users of Web sites often form mental images about what a Web page should look like. In this assignment, you will learn about the importance of navigation in a well-integrated Web design. Relative links and anchor links are essential components of informational architecture that allow users to find sought-after information quickly and efficiently when they come to a site.

Absolute links are absolutely essential to connect all the sites with the framework of the modern Internet. The World Wide Web would be pretty much useless without linkages to shared information and valuable content; therefore, today's users depend on reliable and efficient navigational links.

We will look at CSS that represent a design approach allowing for the separation of content from formatting. The use of CSS is a centralized means of controlling formatting such as page color and text font characteristics. CSS can similarly be applied to layout. CSS is seen as contributing to making the Internet increasingly efficient. Rules written for the style sheet are arranged in a hierarchy of design rules and choices that cascade through either a single page (embedded or inline style sheets) or an entire site (via an external style sheet).

You will be focusing on creating relative links in this unit's assignment to link all of the pages in your Web site. As all of your HTML files will be in the same folder, your links will only need to include the file name. Using CSS, you will be able to use a simple list of links and format it to look like a navigation bar.

Using an external style sheet saves a lot of work as it can control the layout and look and feel of an entire site from one place. If any design changes need to be made, it can simply be done in a single place rather than from each of the files. You may have noticed that your site is fairly plain but through the use of applying CSS, you will begin to bring your Web site to life. What is important as you start your development is to plan ahead and think about the color and font schemes that you plan to use. There are resources that can assist you in selecting a color palette for your site. However, a good guide or starting point might be the colors that you use in the logo that you create for your chosen organization.

Using CSS, you can control every single aspect of your Web page in terms of color, font, size, margin, padding, and position. All HTML elements can be considered as boxes. When it comes to the layout and design, you will hear the term *box model* being used. Essentially, there is a box that wraps around each HTML element that we use in our pages. They consist of the content as the most inner item, then padding, border, and finally margin on the most outer part of the box.

Content is the is the area in which we can have text, images, and other components that are displayed to the user. Padding is the area that is around the content and is transparent to the user; border is a box that goes around the padding and content; and margin is the area that is outside of the border and is transparent. Using the box model, we are able to add borders around our elements and spacing between the content, border, and other elements.

It is important to note that when you set the width and height of an element, you just set the width and height of the content area. To determine the full size of an element, we also have to add in the padding, border, and margin sizes as well.

Learning Activities

u02s1 - Studies

Readings

Use your *New Perspectives HTML5 and CSS3* text to read the following:

- Tutorial 2, "Getting Started With CSS," pages 83–157.

Walkthroughs and Exemplars

You may view the following walkthroughs and exemplars to help you understand the concepts addressed in this unit:

- Formatting Text and Images With CSS Walkthrough.
- Formatting Text and Images With CSS Exemplar [ZIP].
- Inline, Embedded, and External Style Sheets Walkthrough.
- Inline, Embedded, and External Style Sheets Exemplar [ZIP].
- Classes and IDs in CSS Walkthrough.
- Classes and IDs in CSS Exemplar [ZIP].

Optional Readings

You may choose to review the following Internet resources:

- W3Schools.com. (n.d.). CSS introduction. Retrieved from http://www.w3schools.com/css/css_intro.asp
  - This page provides an introduction to CSS to explain how it works. Along the left navigation, you can follow the tutorial path on specific CSS topics.
- W3Schools.com. (n.d.). CSS navigation bar. Retrieved from http://www.w3schools.com/css/css_navbar.asp
  - This page offers a demonstration of using CSS to format a navigation bar. In particular, pay attention to the use of a list of links with HTML.
- Learn CSS Layout. (n.d.). Retrieved from http://learnlayout.com/
  - This site provides the CSS fundamentals to use CSS for layouts. This is a tutorial that is highly recommended to review.
- Google Fonts. (n.d.). Retrieved from https://fonts.google.com/
- Freehosting.com. (n.d.). Retrieved from https://www.freehosting.com/
- 2FreeHosting.com. (n.d.). Retrieved from https://www.2freehosting.com/
- W3C. (n.d.). CSS validation service. Retrieved from https://jigsaw.w3.org/css-validator/

Optional Skillsoft Resources

- Campbell, J. (2014). HTML5: Browser best practices [Video]. Skillsoft Ireland.
- Skillsoft. (n.d.). HTML5 and CSS3 [Tutorial].
- Skillsoft. (n.d.).Creating styles with CSS3 and styling text [Tutorial].

u02s1 - Learning Components

- Understand the naming conventions for file structure.
- Examine examples of working code.
- Validate and test code.
- Examine examples of external CSS integration with a Web page.
- Identify the types of properties that are controlled using CSS.
- Examine HTML rules, attributes, and elements.
- Study how CSS and HTML5 work together to create page elements.

u02a1 - Externally Linked CSS and Site Navigation

Overview

Imagine that you have to change the font size for a headline in your Web site. No problem, right? Just go in and change the font size property on that page. But, what if you had a site with a hundred such titles that you had to change one by one? It would be both time consuming and inefficient. That is where CSS come in. They allow developers and designers to make wholesale changes to a Web site by changing code in only one file. In fact, with HTML5, formatting must all be done using CSS.

In this assignment, you will create your site navigation and a single external CSS file that controls text elements on each of your Web pages.

Directions

Complete the following:

- Create and save an externally linked, CSS3-compliant style sheet to control all your Web pages (see page list below). It should define properties (such as position, margin, padding, font family, color, size, et cetera) for each of the following:
  - Headlines.
  - Header.
  - Body text.
  - Background (image or color).
  - Navigation (use an unordered list to format the list items for navigation).
  - Footer (location/font size and color).
  - Anchor links (current, hover, visited).
- Update the code in your homepage and template to link to the new style sheet.
- Use your page template to create the other four pages for the site and title each appropriately.
  - About Us.
  - Services.
  - Gallery.
  - Contact Us.
- Create functional page navigation using CSS. Use textual links using either a horizontal navigation across the top or vertical navigation down the left side.
- Define text and background properties for all page elements using CSS.
- Make sure to enter context appropriate text in the content sections for each of the pages to demonstrate that the formatting and design are consistent across the site.

Make sure you do the following:

- Submit error-free HTML5 and CSS code.
- Create functioning site navigation using CSS.
- Create a functioning external CSS file linked to a Web site.
- Use CSS to effectively manipulate properties of Web page elements.
- Write developer comments to describe and explain code.

Notes

Make a note of the following as you complete this assignment:

- Use the same CSS formatting that references a single external style sheet for all pages.
- Use Web fonts such as those available on Google Fonts Web site.
- Use <div> tags only when there is no better element for the job to describe the content that it is containing. Use the semantic layout elements such as header, article, section, nav, and footer instead.

> **Hint:** Consider building your entire layout interface first. It can help you plan your development of your pages to figure out where you want your content to go.

Once completed, view your pages in your selected Web browsers to see if the content renders appropriately and consistently within each. Next, ensure that your HTML code is HTML5 compliant and your CSS code is valid. You may use the Markup Validation Service for testing the former and CSS Validation Service for testing the latter, both from W3C. **Important**: Take a screen capture of each of your validation results and save it for submission.

> **Note**: Errors from the validation should be fixed. However, warnings do not need to be fixed but should be reviewed. Fix the issues from the top down as one error can cascade into 20 other errors.

Submission Requirements

Submit your work in the courseroom using a single zip file containing the following:

- index.html.
- template.html.
- image_and_url.doc. Include a Word document with:
  - The URLs for each of your pages.
  - The names of the browsers you used to view the pages.
  - Pasted images from your validation screen capture.
- Any ancillary files such as graphic or photos you may have decided to include.

Upload your Web site files to a hosting service of your choice. Two free sites to consider are Freehosting.com and 2FreeHosting.com.

Course Resources

- Freehosting.com
- 2FreeHosting.com
- Markup Validation Service
- CSS Validation Service
- Google Fonts

u02d1 - Web Site Navigation

A Web site's navigation is one of the most important aspects of a Web page as users need to be able to use your site and find what they are looking for. Users need to be able to find the information they are looking for as quickly as possible.

In your post, address the following:

- What are some of the best practices when it comes to navigating a Web site?
- Should a Web site deviate from the standard horizontal navigation across the top or vertical navigation down the left side?
- How does the use of the HTML5 <nav> tag differ from previous versions of HTML when creating navigation?

Response Guidelines

Comment on the posts of at least two other learners. Offer insights, solutions, examples, or opinions that add depth and value to the conversation.

Course Resources
Undergraduate Discussion Participation Scoring Guide

u02d1 - Learning Components

- Examine examples of navigation and lists and their associated code.

u02d2 - Challenges to Creating Navigation or External CSS

Creating navigation and external CSS can have their own unique challenges, from alignment of images to making sure your Web page is communicating with your style sheet.

Discuss one or more of the challenges or experiences that you encountered when trying to complete this unit's assignment. How did you try to address it or solve it? What questions do you still have?

Response Guidelines

Comment on the posts of at least two other learners. Share approaches or techniques that you might suggest to help the learner to address his or her challenge.

Course Resources
Undergraduate Discussion Participation Scoring Guide

Unit 3 >> Page Layout and CSS

Introduction

So far, we have been developing our Web pages using HTML5 and then stylizing the content using CSS3. The last step is to focus on positioning our HTML elements on our Web page how we would like them to be. However, it is important that we consider all of our users as we design our layouts. Using a fixed width layout in which all of the elements and containers have a specific width can be too limiting.

Liquid layouts are a series of rules that we can apply to rules on the page until we change the Web page size. They also come into effect when our visitors may use different mobile devices, tablets, laptops, and desktops to visit our site. We cannot predict what our visitors will be navigating our site with, so we cannot simply design for one optimal page size for all users.

As there are so many varying browser widths that can be displayed, if we used a fixed width that is too large, some visitors will have to scroll horizontally on the page just to read content. On the other hand, if we set the fixed width to be too small, visitors on larger browsers will only have their content displayed on a small section of the page. To get around this, we will apply liquid layouts using CSS on our page.

The liquid layout will help expand or contract all of the elements of the page to the entire width that the browser window is sized to. To do this, all of the containers on the page have to have their widths defined using percentages rather than fixed widths (px). By giving our visitors a bit more control of the layout of the page, it will make our user interface to be more flexible.

CSS3 has a new layout mode that uses this liquid layout model that you can also use although it is a bit more complex to apply. It may be an option that you will want to explore further.

In your content areas, typically we want to use <div> tags to add in content to arrange your layout. The <div> tags are block-level elements so by default, they will take up 100 percent of the available width within the container that it is located in. Using CSS, we will be able to control how they are stacked together in terms of columns and rows.

So far, we have been developing our Web pages using HTML5 and then stylizing the content using CSS3. The last step is to focus on positioning our HTML elements on our Web page how we would like them to be. However, it is important that we consider all of our users as we design our layouts. Using a fixed width layout in which all of the elements and containers have a specific width can be too limiting.

Learning Activities

u03s1 - Studies

Readings

Use your *New Perspectives HTML5 and CSS3* text to read the following:

- Tutorial 4, "Graphic Design With CSS," pages 258–309.
- Tutorial 6, "Working With Tables and Columns," pages 433–474.

Walkthroughs and Exemplars

You may view the following walkthroughs and exemplars to help you understand the concepts addressed in this unit:

- Adding a Data Table Using CSS Walkthrough.
- Adding a Data Table Using CSS Exemplar [ZIP].
- Formatting the Page Layout Using CSS Walkthrough.
- Formatting the Page Layout Using CSS Exemplar [ZIP].
- Page Layouts Walkthrough.
- Page Layouts Exemplar [ZIP].

Optional Readings

You may choose to complete the following Internet resources:

Layouts - Basic and Liquid

Review the following resources related to layouts:

- CSS Library. (n.d.). Liquid layouts. Retrieved from http://www.dynamicdrive.com/style/layouts/category/C13/
  - This page provides various examples of liquid layouts and the code for each of the examples.
- Weakly, R. (2003, December 30). Liquid layouts the easy way. Retrieved from http://maxdesign.com.au/articles/liquid/

- This article provides explanations and descriptions of how liquid layouts as well as fixed-width and em-driven layouts are used.

- W3Schools.com. (n.d.). HTML layouts. Retrieved from http://www.w3schools.com/html/html_layout.asp
  - This page explains HTML layout elements along with semantic elements. It also explains different layout techniques with their pros and cons.
- Williams, P. (n.d.). Using <div> tags for layout. Retrieved from http://www.beginnersguidetohtml.com/guides/css/layout/
  - This page explains using <div> tags for layout with the control through absolute and relative positions.
- Codementor. (n.d.). 4 different HTML/CSS layout techniques to create a site. Retrieved from https://www.codementor.io/css/tutorial/4-different-layout-techniques-create-single-page
  - This tutorial provides different techniques to design a single Web page.

CSS

Review the following resources related to CSS:

- W3Schools.com. (n.d.). CSS3 flexbox. Retrieved from http://www.w3schools.com/css/css3_flexbox.asp
  - Flexible boxes, or flexbox, is a new layout mode in CSS3. Use of flexbox ensures that elements behave predictably when the page layout must accommodate different screen sizes and different display devices.
- CSS Zen Garden. (n.d.). Retrieved from http://www.csszengarden.com/
  - This Web site is unique as all of the pages have the exactly same HTML code but they each have their own unique CSS files. View the source of the CSS for each to find inspiration for your own designs.

Optional Skillsoft Resources

- Campbell, J. (2015). Responsive web design: The CSS3 flexbox [Video]. Skillsoft Ireland.
- Skillsoft. (n.d.).HTML5 Layout with CSS3 [Tutorial].

u03s1 - Learning Components

- Examine how CSS controls page layout.
- Understand the box model, including margins, padding, border, et cetera.
- Examine various methods used for positioning content.
- Examine CSS table formatting of table rows, columns, and cells.
- Examine the basic structure of HTML code and how to properly include comments.
- Examine how to manipulate various semantic layouts.
- Understand the properties of a table.

u03a1 - Page Layout With CSS

Overview

You have created your Web pages, navigation, and text that use CSS. Now it is time to concentrate on how CSS can control page layout to ensure consistency across all pages. It is also time to customize the layout of each page to appropriately accommodate content using CSS.

In this assignment, you will create a liquid layout for your site and add the final text copy to your pages while ensuring the same level of consistency across all of the pages through CSS. Liquid layouts are designed to resize the page components as the page width changes. Regardless of the browser width or device width, the user will not have to do any horizontal scrolling as all of the page components will be resized to fit the elements within the page width. This liquid layout would apply to all pages and all sections of the page including the header, navigation, content, and footer areas.

Directions

Complete the following:

- Amend the *content* section of each of the following pages using CSS as described below:
    - About Us: Create a CSS-formatted table with a header, rows, columns, and cells, and populate it with content.
    - Services: Add at least two dynamic-width columns.
    - Contact Us: Create multiple rows.
    - Homepage: Create both rows and columns.
- Use CSS to format how the HTML document is rendered in the Web browser based on liquid layouts such as the header, navigation, and footer. Each of these sections should resize based on the browser width.

Make sure you do the following:

- Create a table that properly displays data.
- Submit code that is error free, well documented, and appropriately constructed.
- Use CSS to define how HTML elements are displayed on a Web page per specifications.
- Create custom layouts for individual page content areas.
- Write developer comments to describe and explain code.

Once completed, view your pages in your selected Web browsers to see if the content renders appropriately and consistently within each. Next, ensure that your HTML code is HTML5 compliant and your CSS code is valid. You may use the Markup Validation Service for testing the former and CSS Validation Service for testing the latter, both from W3C. **Important**: Take a screen capture of each of your validation results and save it for submission.

> **Note**: Errors from the validation should be fixed. However, warnings do not need to be fixed but should be reviewed. Fix the issues from the top down as one error can cascade into 20 other errors.

Submission Requirements

Submit your work in the courseroom using a single Zip file containing the following:

- All page files.
- Image_and_url.doc. Include a Word document with:
    - The URLs for each of your pages.
    - The names of the browsers you used to view the pages.
    - Pasted image from your validation screen capture.
- Any ancillary files such as graphic or photos you may have decided to include.

Upload your Web site files to your Web host.

Course Resources
CSS Validation Service
Markup Validation Service

u03d1 - CSS and Web Layout Styles

Some Web sites use the same HTML code but different CSS layouts for each of the pages. You may view the CSS Zen Garden Web site to understand this.

Consider the following before you address this discussion:

- Which design inspired you?

- How would you apply some of those design elements to your own Web site?
- What layout features could you apply to your own site?

Modern Web pages use a variety of layout styles such as:

- Static.
- Liquid.
- Adaptive.
- Responsive.

Choose a Web site that uses one of these layouts. In your post:

- Evaluate if this layout style is the most effective and why. What considerations did you have in your evaluation?
- Describe the CSS changes that would need to be made to alter the layout style of your chosen site to another of the above layouts. For example: A static layout being changed to a liquid layout would require all fixed widths to become dynamic widths using <em> or % values.

  Note: Make sure to share the URL of the chosen Web site, if you chose one other than the CSS Zen Garden.

Response Guidelines

Comment on the posts of at least two other learners. Do you agree with the evaluation of the learner? Why or why not? Are there other considerations that he or she missed?

Course Resources
Undergraduate Discussion Participation Scoring Guide
CSS Zen Garden

u03d1 - Learning Components

- Understand the box model, including margins, padding, border, et cetera.
- Examine various methods used for positioning content.
- Examine how to manipulate various semantic layouts.

u03d2 - Page Layout Challenges

CSS offers powerful tools for defining page layout. However, it can be one of the most frustrating challenges for new Web developer. Understanding of the box model is imperative, as are other page alignment properties.

Discuss one or more of the challenges or experiences that you encountered when trying to complete this unit's assignment. How did you try to address it or solve it? What questions do you still have?

Response Guidelines

Comment on the posts of at least two other learners. Share approaches or techniques that you might suggest to help the learner to address his or her challenge.

Course Resources
Undergraduate Discussion Participation Scoring Guide

Introduction

Multimedia in the context of the Internet focuses around audio, music, videos, and animations. These multimedia components being added helps us to be able to attract and keep the attention of our viewers. By adding in multimedia, it helps an organization present its branding in a different format. Not all viewers like to read content and when you have the ability to present your content through media, it can be much more effective.

There are a lot of different formats for each of these that are available. In the past, audio and video had to be handled differently by each of the common browsers. However, with HTML5, a lot of the multimedia options have been streamlined for consistency between browsers. This also means that there are limited supported formats with HTML5. In particular, HTML5 only supports MP4, WebM, and Ogg for video and MP3, WAV, and Ogg for audio.

Prior to HTML5, videos that needed to be played within a browser had to use different plugins like Flash. However, with HTML5, the <video> element allows you to embed videos in a standard way for modern browsers. There is a *controls* attribute that adds different video controls to enable pausing and playing the video. In addition, we have the *width* and *height* attributes to set the size of the video.

Similar to the <video> element, we are able to use the <audio> element with the same attributes to add in audio. There are many other attributes that you can add to customize your video and audio such as being able to preload the media, automatically play it when the page loads up, allowing the media to loop, et cetera.

Another option when it comes to incorporating videos is to use a third-party site such as YouTube. Video files can be quite large and takes up quite a bit of bandwidth when being streamed to users. By using third-party sites such as YouTube, you can avoid having to host the video. However, when you use a third-party host, you have to ensure your video follows the rules and regulations set by the host.

Image galleries are common in many Web sites. They frequently will have a number of thumbnail images in which the user can click to view the full-sized image. It is important to note that this type of gallery is only a precursor for more interactive image galleries.

Learning Activities

u04s1 - Studies

Readings

Use your *New Perspectives HTML5 and CSS3* text to read the following:

- Tutorial 8, "Enhancing Your Website With Multimedia," pages 586–621.

Walkthroughs and Exemplars

You may view the following walkthroughs and exemplars to help you understand the concepts addressed in this unit:

- Adding and Formatting Images Walkthrough.
- Adding and Formatting Images Exemplar [ZIP].
- Inserting Video in a Web Page Walkthrough.
- Inserting Video in a Web Page Exemplar [ZIP].
- Inserting Audio in a Web Page Walkthrough.
- Inserting Audio in a Web Page Exemplar [ZIP].
- Embedding a Video From YouTube Walkthrough.
- Embedding a Video From YouTube Exemplar [ZIP].

Optional Readings

You may choose to complete the following Internet resources:

HTML Audio and Video

Review the following resources related to HTML audio and video:

- W3Schools.com. (n.d.). HTML audio/video DOM reference. Retrieved from http://www.w3schools.com/tags/ref_av_dom.asp
    - This is a reference page for the methods, properties, and events for the <audio> and <video> elements.
- TutorialsPoint. (n.d.). HTML5 – Audio & video. Retrieved from http://www.tutorialspoint.com/html5/html5_audio_video.htm
    - This tutorial is about how to add audio and video into a site using HTML5.
- Segal, N. (n.d.). HTML5 audio and video. Retrieved from http://www.htmlgoodies.com/html5/other/html5-audio-and-video.html
    - This article offers visual elements of how to add audio and video.
- LePage, P. (2014, January 9). HTML5 video. Retrieved from https://www.html5rocks.com/en/tutorials/video/basics/
    - This article offers a walk-through of how to add video into a site with multiple source files, media fragments, captions, and subtitles.
- Apple. (n.d.). About HTML5 audio and video. Retrieved from https://developer.apple.com/library/content/documentation/AudioVideo/Conceptual/Using_HTML5_Audio_Video/Introduction/Introduction.html
    - This guide offers an introduction to explain how to use HTML5 audio and video.

YouTube Integration

Review the following resource related to YouTube integration:

- W3Schools.com. (n.d.). HTML YouTube videos. Retrieved from http://www.w3schools.com/html/html_youtube.asp
    - This is a simple site that explains how to embed YouTube videos into a Web page.

Thumbnail Creation

Review the following resources related to Thumbnail creation:

- MakeAThumbnail.com. (n.d.). Retrieved from http://www.makeathumbnail.com/
    - This is a simple online tool to generate consistent-sized thumbnails.
- ResizeImage.net. (n.d.). Retrieved from http://resizeimage.net/
    - This is an online tool that allows you to create thumbnails with more control.

Optional Skillsoft Resources

- Lassoff, M. (2013). CSS3 development: Making images responsive [Video]. Skillsoft Ireland.
- Shen, W. (2017). HTML5, JavaScript, and CSS3: Background images [Video]. Skillsoft Ireland.
- Shen, W. (2017). HTML5, JavaScript, and CSS3: HTML Audio and Video [Video]. Skillsoft Ireland.
- Lassoff, M. (2012). HTML5: Playing audio files [Video]. Skillsoft Ireland.
- Lassoff, M. (2012). HTML5: Playing video files [Video]. Skillsoft Ireland.
- Skillsoft. (n.d.).HTML5 APIs and media [Tutorial].

u04s1 - Learning Components

- Understand how to create an image gallery table using HTML or CSS, or both.
- Understand how the HTML5 <video> element is used on a Web page.
- Examine examples of image galleries in a Web page and how they are coded.
- Examine the basic structure of HTML code and how to properly include comments.
- Understand how the HTML5 <audio> element is used on a Web page.

u04a1 - Adding Media

Overview

Adding media to your site is almost considered as the *price of admission* for a modern Web site. Identifying or creating contextually compelling media content greatly enhances a Web site.

In this assignment, you will add video, audio, and an image gallery to your Web site. You can create the images, video, and audio on your own, or you can use them from an existing source such as YouTube. Make sure to only use content that does not infringe on copyrights and include proper credits where applicable.

> **Note**: Content used in the creation of your Web pages should be original (created by you) or content that is not restricted by copyright and royalty issues.

Directions

Add the following media to your site. Content should be appropriate to the look, feel, and context of your site.

- **Image Gallery** (add to the Gallery page): Create an image gallery with at least three images. The images should be displayed as thumbnails that open a larger image in a pop-up window when clicked. All of the images on your site should use the *alt* attribute to describe the image.
- **Video**: Can be taken from YouTube or other sources or made by you. Be sure to include player controls.
- **Audio**: Use .mp3, .wav, or .ogg files. Be sure to include player controls.

Make sure to do the following:

- Add a functioning video to a Web page.
- Add a functioning audio track to a Web page.
- Create a functioning image gallery according to specification.
- Write developer comments to describe and explain code.

Notes

Make a note of the following as you complete this assignment:

- Images should ideally be created using JPEG, GIF, and PNG-8 file formats. Use the online image resizing tools to resize your image to the intended display size. You want to avoid the browser scale your image as it can cause color depth and resolution problems.
- Audio and video file sizes can get quite large, so be sure your file sizes are less than 10 MB to be able to quickly upload and download them. Your hosting server may limit your file sizes as well, so be sure to check with them if there are any limitations.

Hints

A few hints to successfully complete your assignment:

- If you are looking for resources around audio and video, make sure that you are looking specifically for HTML5 resources. The extensive plugins and browser workarounds are no longer a concern. All video and audio can be used through the <video> and <audio> elements.
- It is sometimes useful to put video, audio, and graphics files in a single directory with a name that is descriptive such as *media*.

Once completed, view your pages in your selected Web browsers to see if the content renders appropriately and consistently within each. Next, ensure that your HTML code is HTML5 compliant and your CSS code is valid. You may use the Markup Validation Service for testing the former and CSS Validation Service for testing the latter, both from W3C. **Important**: Take a screen capture of each of your validation results and save it for submission.

> **Note**: Errors from the validation should be fixed. However, warnings do not need to be fixed but should be reviewed. Fix the issues from the top down as one error can cascade into 20 other errors.

Submission Requirements

Submit your work in the courseroom using a single zip file containing the following:

- All page files.
- All media files.
- image_and_url.doc. Include a Word document with:
    - The URLs for each of your pages.
    - Each of your media files.
    - The names of the browsers you used to view the pages.
    - Pasted images from your validation screen capture.
- Any ancillary files such as graphic or photos you may have decided to include.

Upload your Web site files your Web host.

Course Resources

Markup Validation Service

CSS Validation Service

u04d1 - Building a Photo Gallery

There are many types of photo or image galleries available on the Internet. Fundamentally, they are all designed to display graphics in an interesting, innovative, or expedient way.

In your post:

- Discuss some of your primary design considerations for your photo gallery.
- Discuss how you are building your gallery. Consider topics like the following:
    - Thumbnail creation.
    - Mechanism to display your full-size images.
    - Graphic optimization.
    - How you used CSS to control your gallery.
    - HTML5 tags or elements that you used.
    - How you control layout.

Response Guidelines

Comment on the posts of at least two other learners. Offer insights, solutions, examples, or opinions that add depth and value to the conversation.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u04d1 - Learning Components

- Understand how to create an image gallery table using HTML or CSS, or both.
- Examine examples of image galleries in a Web page and how they are coded.

u04d2 - Challenges to Adding Media to a Web Site

Adding media to a Web site can be a great way to enhance the user experience. The evolution to HTML5 has greatly simplified and standardized the process. Still there can be challenges.

Discuss one or more of the challenges or experiences that you encountered when trying to complete this unit's assignment. How did you try to address it or solve it? What questions do you still have?

Response Guidelines

Comment on the post of at least two other learners. Share approaches or techniques that you might suggest to help the learner to address his or her challenge.

Course Resources

Undergraduate Discussion Participation Scoring Guide

Unit 5 >> Forms

Introduction

HTML forms allow our viewers to interact with our organization. They provide a means to be able to send data from them to us. There are many different types of forms with a varying number of input options. A simple search bar is one example of a form whereas a user registration or contact form can get more complex.

The form is made up of multiple widgets with options such as text fields, select boxes, check boxes, or radio buttons. These widgets should be paired with a specific label that describes what value or data should be entered into that field. All of the input fields should be wrapped into a single <form> element. In this element, we have two attributes that should be used to send the data to a Web server for processing. These would be the *action* attribute and *method* attribute. The *action* attribute defines the URL where the form's data should be sent to. The *method* attribute defines which HTTP method should be used to send the data (get or post).

As you design your form, it can be helpful to think about the user experience. Ideally, we want to keep our user forms simple and focused with only key fields that are required. In addition, your submit button should be created. A click on that button will send the form's data to the *action* attribute that is part of the <form> element.

Each input field in your form will need to have an *id* and *name* attribute that should remain consistent. This will be used behind the scenes when your data is being passed to the server. Using the get method, the form data is appended to the URL to the *action* attribute using the name/value pairs from your form input fields. Using the post method, it sends the data in the body of the HTTP request, so you will not see the values being passed to the server.

Using CSS, you can stylize your form so that it is easier to read and aesthetically pleasing. Try to consider organizing your form so that the labels and input fields align vertically. This will make it easier for your viewers to scan your form and input the data as needed.

Rather than simply having your form do nothing, you will be using your hosting server and using a simple PHP page for the *action* attribute. You will need to edit the PHP page to reflect the id or names that you have used in your form along with the input field types. This will simply be a simple data entry page and content being echoed back to you but it should demonstrate how it works. Note though that this PHP page does have to execute on the hosting server itself and will not run directly from your local machine.

Learning Activities

u05s1 - Studies

Readings

Use your *New Perspectives HTML5 and CSS3* text to read the following:

- Tutorial 7, "Designing a Web Form," pages 499–545.

Multimedia

You may view the following walkthroughs to help you understand the concepts addressed in this unit:

- Creating a Form With Labels Walkthrough.
- Setting up XAMPP and Testing PHP Pages Walkthrough.
- Format a Form on the Page Walkthrough.

Optional Readings

You may choose to complete the following Internet resources:

Form Design Guidelines

Review the following resources related to form design:

- Mifsud, J. (2011, November 8). An extensive guide to Web form usability. Retrieved from https://www.smashingmagazine.com/2011/11/extensive-guide-web-form-usability/
  - This article offers detailed guidelines to follow when it comes to Web form usability.
- UXMatters. (2012, May 7). 7 basic best practices for buttons. Retrieved from http://www.uxmatters.com/mt/archives/2012/05/7-basic-best-practices-for-buttons.php
  - This article offers tips about using buttons in HTML.
- UXMatters. (2010, March 21). Pagination in Web forms | Evaluating the effectiveness of Web forms. Retrieved from http://www.uxmatters.com/mt/archives/2010/03/pagination-in-web-forms-evaluating-the-effectiveness-of-web-forms.php
  - This article focuses on Web form design and evaluation considerations.
- Mozilla Developer Network. (n.d.). How to structure an HTML form. Retrieved from https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/How_to_structure_an_HTML_form
  - This article explains how to use all of the HTML form elements and goes through the process to ensure that the forms are accessible.

HTML Form Submission

Review the following resource related to HTML form submission:

- Mozilla Developer Network. (n.d.). Sending form data. Retrieved from http://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Sending_and_retrieving_form_data
  - This article explains where the data goes when an HTML form is submitted.

PHP

Review the following resource to to better understand why PHP needs the server:

- W3Schools.com. (n.d.). PHP 5 form handling. Retrieved from http://www.w3schools.com/php/php_forms.asp
  - This page explains the process of form handling using PHP.
- W3Schools.com. (n.d.). PHP 5 tutorial. Retrieved from http://www.w3schools.com/php/default.asp
  - This page offers an overview of what PHP is and how it is used.

Optional Skillsoft Resources

- Skillsoft. (n.d.).HTML5 forms [Tutorial].

- Lassoff, M. (2012). HTML5: Form tag [Video]. Skillsoft Ireland.
- Campbell, J. (2014). HTML5: How to access form data [Video]. Skillsoft Ireland.

u05s1 - Learning Components

- Examine the various methods to lay out form elements on a Web page.
- Understand the purposes of each type of input field in correlation to its data.
- Examine the importance uniqueness of name and id for each form element.
- Understand how to apply CSS to format the form elements on the Web page.
- Validate and test code.

u05a1 - Creating a User Form

Overview

You have added interactivity to your Web site with media. Another form of common user interaction is with forms. In this assignment, you will create a form to be used in one of your Web pages that will allow users to share or request information. Examples of types of forms to consider are:

- Request for information.
- Volunteer sign-up.
- Event registration.
- Newsletter sign-up.

Directions

Create a user-friendly and professionally designed input form on the *Contact Us* page. The form should be contextually relevant for the site's purpose and intuitive for the user. At a minimum, the form must include the following functioning tags:

- Fieldset.
- Legend.
- Label.
- Textfield.
- Textarea.
- Radio buttons or check boxes.
- Select box.
- Submit button. (See the Form Submission details below.)

Make sure to do the following:

- Create a Web form that functions per specifications.
- Submit error-free HTML5 and CSS code.
- Create a user-friendly layout for a Web form.
- Write developer comments to describe and explain code.

Form Submission

When a user form is submitted, proper UX design dictates that there is some sort of a response that lets you know your submission has been successful. This is often done through server-side scripting using PHP, which is beyond the scope of this course. Because it is an important component to any form, we have included such a page called response_page.php for your use. To enable this function, do the following:

- Add the following code to your form in the form element:
  - *<form action="response_page.php" method="post">*
- Download the response_page.php file (given in the resources) to your computer, and then upload it to your site (also include it in your zip file submission). **Note**: Make sure the path to the file is correct. The PHP file should be placed in the same folder as your contact page.

This is often an area where mistakes are made.

> **Hint**:In HTML5, you can add the required attribute to input fields that specifies that the input field is required. Browsers that support this feature will not allow a form submission until that field has been filled.

Once completed, view your pages in your selected Web browsers to see if the content renders appropriately and consistently within each. Next, ensure that your HTML code is HTML5 compliant and your CSS code is valid. You may use the Markup Validation Service for testing the former and CSS Validation Service for testing the latter, both from W3C. **Important**: Take a screen capture of each of your validation results and save it for submission.

> **Note**: Errors from the validation should be fixed. However, warnings do not need to be fixed but should be reviewed. Fix the issues from the top down as one error can cascade into 20 other errors.

Submission Requirements

Submit your work in the courseroom using a single zip file containing the following:

- All page files including the response_page.php file.
- image_and_url.doc. Include a Word document with:
  - The URLs for each of your pages.
  - The names of the browsers you used to view the pages.
  - Pasted images from your validation screen capture.
- Any ancillary files such as graphic or photos you may have decided to include.

Upload your Web site files to your Web host.

Course Resources
response_page.php
Markup Validation Service
CSS Validation Service

u05d1 - PHP and Form Submission

Forms allow users to be able to interact with Web sites and to submit content. However, not all forms are created equally. We have all experienced forms that were too long that you gave up filing it out or perhaps ones that had fields that you did not know what should be placed into them. Consider Web usability and functionality as you discuss *any two* of the following:

- How should the fields and labels be organized on a site or a form, and in what order?
- How should we determine what type of input field each input should be? What input type would reduce the chances of errors?
- What role does CSS play in creating an effective form?
- How is Web usability applied to Web forms? Why is it so important that Web forms provide the user with the optimal user experience?

Response Guidelines

Comment on the posts of at least two other learners. Offer insights, solutions, examples, or opinions that add depth and value to the conversation.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u05d1 - Learning Components

- Understand the purposes of each type of input field in correlation to its data.
- Define the layout of form elements including field, fieldset, label, input fields, and buttons.

u05d2 - Challenges to Creating Forms

Forms allow users to interact with Web sites in important ways. Forms can be complex or simple, but creating them to work and display properly can be tricky.

Discuss one or more of the challenges or experiences that you encountered when trying to complete this unit's assignment. How did you try to address it or solve it? What questions do you still have?

Response Guidelines

Comment on the posts of at least two other learners. Share approaches or techniques that you might suggest to help the learner to address his or her challenge.

Course Resources

Undergraduate Discussion Participation Scoring Guide