

Getting Started and HTML Challenge

Welcome to your first challenge! In this course's challenges, you'll work on completing the HTML Essential Training reference site. You won't be building the site from scratch, rather you'll be completing various portions of the site in varying degrees of completion. The goal is to reinforce and put into practice what you've learned during the seminar and online lessons ([HTML Essential Training chapter 1-6](#)). Please start with setting up the base of your project by following these steps:

Import the base files for the first challenge

1. Download the starting files on Blackboard
 2. Place these files in your repository (in the folder you created during the first seminar)
 3. Open the files in your Code Editor (open folder) and start working on the assignment
 4. When finished with each section of the challenge, commit and sync on GitHub Desktop
 5. Make sure all your code is committed
 6. Push / sync up to GitHub
-

Let's start the challenge!

For each part of the challenge you'll be editing and adding HTML using the starting files found on Blackboard. In each steps we will list the tasks you need to perform, as well as the relevant files you have to edit for each task.

1 Basic HTML

Determine a heading strategy.

1. Applies to: All pages
2. Explore each of the pages (intro.htm, links.htm, next.htm, reference.htm, structure.htm and syntax.htm) and carefully read through the text.
3. Determine which content should be represented as headings, and then which level of headings you should use for each page. Focus on being consistent in how you assign headings for each page.
4. Go through each of the pages, and format headings based on your strategy.

Format paragraphs

1. Applies to: intro.html
2. Determine which content should be represented by paragraphs and then format them appropriately.

Controlling line breaks

1. Applies to: next.html
2. In the section under "learning web design" there are four recommendations. Each recommendation has an initial sentence followed by descriptive text. Using a line break, format these paragraphs so their initial sentence appears on its own line.

Determining emphasis

1. Applies to: next.html
2. In the file next.html examine the four pieces of advice in the Learning Web Design section. Determine how to best emphasize the first line in each paragraph. Think about ways that you could isolate that line or emphasize it in a way that denotes its importance to the rest of the paragraph.
3. In the last advice paragraph ("break things"), display the quotes in italics. In the same paragraph, emphasize the last line "Always challenge yourself" so that it's importance is indicated.
4. In the list of resources below the advice section, make sure all top-level list items appear in bold.

Displaying special characters

1. Applies to: reference.html

2. Scroll down and find the table that displays common named character entities.
3. Preview the page in a browser, and using the table as a reference, add the appropriate entity into the first table cell to provide an example of that character.
4. In the definition for Elements, wrap the word "tags" in quotes using a left and right quote special character.

Inserting Images

1. Applies to: intro.html
2. Examine the existing image at the top of the page. Note the class "flowRight" which is used by CSS to control styling. You'll learn more about this in the chapter on CSS.
3. Scroll down until you locate the paragraph "Shortly After the 1.0 specification..." At the beginning of that paragraph insert an image that:
 - a. uses the alt text "W3C logo"
 - b. passes the value of 200 for width, and 136 for height
 - c. applies a class of "flowLeft"

When finished with each section, don't forget to commit and sync on Github Desktop

2 Structuring Content

Determine a structuring strategy.

1. Applies to: structure.htm
2. Open the structure.htm file and explore the page structure. Write out the document outline, and how structural tags like aside, article, nav, and section are being used.
3. Examine how semantic tags like the header, footer, and main elements are used.

Structure pages

1. Applies to: all pages
2. Using the strategy devised from the structure.htm file, structure each of the remaining files so that the site uses a consistent, logical structure.

When finished with each section, don't forget to commit and sync on Github Desktop

3 Creating Links

Build page navigation

1. Applies to: all pages
2. Using document-relative links, enhance the list at the top of the document so that it links to the appropriate pages in the reference site.

Link to external sites

1. Applies to: links.htm
2. In the footer, locate the first reference to lynda.com
3. Create a link that links that text to the actual lynda.com site.
4. Modify the link so that the site opens in a new tab or window
5. Browse to Twitter and find the profile page for @lynda, copy this URL
6. Back in the footer, add a link to the follow us on Twitter text for lynda.com that links to the @lynda profile

Link to internal sections

1. Applies to: reference.htm, syntax.htm
2. In the reference.htm page, find the text in the first paragraph that refers to the table of named character entities
3. Create a link around that text that jumps to the table at the bottom of the page. Keep in mind that you'll need to modify the table as well to make this work.
4. Open the syntax.htm page.
5. At the bottom of the page, locate the text in the last paragraph that mentions the table of named character entities on the reference page.
6. Create a link around that text that navigates to the reference page and jumps to that specific section.

When finished with each section, don't forget to commit and sync on Github Desktop

4 Creating Lists

Examine site navigation

1. Applies to: all pages
2. Examine the top-level navigation for the site. Currently all links are enclosed in an individual paragraph. Is this semantically sound?
3. If not, organize the links into the appropriate list type

Examine HTML timeline

1. Applies to: intro.htm
2. In the HTML timeline section examine the paragraphs containing relevant dates. Is this the best semantic grouping for this content?
3. If not, organize the timeline into the appropriate list type

Examine doctype grouping

1. Applies to: syntax.htm
2. Find the section on doctypes
3. Examine the three doctypes and their examples. Would a list be the best semantic presentation of this content?
4. If so, organize the links into the appropriate list type

Examine HTML terms

1. Applies to: reference.htm
2. Find the section for HTML terms
3. Examine the terms and their descriptions. Is this the best semantic grouping for the content?
4. If not, organize the links into the appropriate list type

When you finish this challenge, please commit and sync again via Github Desktop to make sure everything is there. You need to submit the repository URL on Blackboard before the deadline (as stated in the Study guide) and we want to see multiple commits.