

DMIT 1530: Assignment 1

Introduction

This Assignment is an individual exercise worth 20% of your overall mark.

For this Assignment, you will build a two-page website with a responsive layout. The topic is a fictitious architectural firm. It will assess the following concepts:

1. How to structure markup for flexbox layouts.
2. CSS properties for the flex container (the parent element).
3. CSS properties for flex items (the direct child elements).
4. Relative Units
5. Advanced CSS Selectors
6. Fluid Web Design and Maximum-Width Containers
7. Media Queries and Responsive Web Design

Instructions

HTML

Before adding any styles, you must mark-up all of your content. Begin by filling in the meta information in the `<head>` of each HTML document; remember to give a unique `<title>` and description for each page.

Then, start building your `<body>` by pasting all of the content from the `**website-content` markdown file into your HTML files, semantically marking them up, and adding appropriate sectioning elements.

Navigation & Footer

The top-level navigation and footer will be identical on every page of your website.

All of the links referencing other pages on the website should be fully functional; however, links to external resources (mail clients, maps, telephone numbers, etc.) may use placeholder characters (#).

Validation

Your HTML must **validate without any errors or warning**; any submissions that do not validate will receive the maximum deduction possible.

Media Queries

As you work through the media queries for each page, be mindful of the specificity and scope of your selectors.

You should have a minimum of two media queries, for a total of three 'views' or layouts per page. You will **need to choose** which breakpoints (i.e. which minimum widths) achieve the look, feel, and functionality of the website in your provided screenshots.

When defining padding and margins, do not fixate on a pixel-perfect measurement. Instead, focus on making sure that the layout is fluid and looks good at any viewport width. Do this by utilising relative units and flexbox properties such as `justify-content` and `align-items`.

A Note on Your Maximum-Width Layouts

At your largest breakpoint or view, your layout should switch to a maximum-width centred layout.

Even when you switch to a maximum-width layout, your background colours and background images should span the entire width of the viewport.

A Hint About Differentiating Page Layouts: If you find that you are having difficulty differentiating between each page with a single stylesheet, try adding a unique class to a high-level element and using this class in your selector statements.

Submission

Push the assignment files to the GitHub Classroom assignment before the due date.

Late assignments will not be accepted.

Marking Guide

This assignment is worth 20% of your final mark.

Marking Guide for Assignment 1		
Task	Description	Value
Flexible Images	All inline images are flexible and adjust with a changing container size. Inline images cannot be larger than their parent container.	2
Relative Font Sizing	All fonts are sized using relative units. Fonts are sized to match the original composition.	2
CSS & Media Queries	CSS is well organised with a top-down structure and follows a mobile-first development. Advanced CSS selectors are used in appropriate ways to reduce bloated code. Two media queries are used to adjust the layout at appropriate screen sizes.	5
Fluid Layout to Fixed-Width Centred Container	The defined widths of the content and elements in the layout are in fluid measurements (% or vw). The content grows or shrinks along with the viewport. The content does not overflow the width of the screen size at any point. On smaller and medium viewports, the content stretches edge-to-edge, then for larger viewports the content becomes centred with an overall fixed width.	5
Multi-Page Layout - Responsive & Styling	Page 1 - Matches the look and layout of the screenshots provided with flexbox techniques, padding, and margins. Page 2 - Matches the look and layout of the screenshots provided with flexbox techniques, padding, and margins.	10
Deductions	Possible deductions include not following best practice techniques, validation errors, not using proper formatting or semantic tags, inappropriate CSS selectors, or spelling mistakes.	-5
Total		24