

## WDD 131

Home

W1

W2

W3

W4

W5

W6

W7

# W01 Assignment: Home Page

## Overview

This assignment allows you to demonstrate your prerequisite knowledge by applying HTML and CSS in the design and development of a home page. In addition, you will need to apply learning activity concepts - including JavaScript - to provide some dynamic information to the home page.

## Task

Design and develop the following home page as your **wdd131** course landing page. Use your own content and basic styling.


### WDD 131: Dynamic Web Fundamentals - Rubia M. Francesco

[LinkedIn](#)[Canvas](#)[Scripture](#)[Final Project](#)

#### About Me

My name is Rubia and I go by Ruby. I was born in South Africa and live with my family in Madagascar. I am currently working as an administrative legal assistant at a law firm. My children are my world and I love spending time with them. I love to travel and I love to learn new things.

#### Toloria, Madagascar



Official Flag of Madagascar

Madagascar is the 4th largest island in the world and is located in the Indian Ocean. It is home to 5% of the world's plant and animal species, of which more than 80% are endemic to Madagascar. The island is best known for its lemurs, baobab trees, and vanilla.

#### Web Dev Resources

- [Dev.to Community](#)
- [Smashing Magazine](#)
- [CSS Tricks](#)
- [W3 Schools](#)
- [Udemy](#)
- [Image Placeholder](#)

©2025 🌸 Rubia Magdalena Francesco 🌸 Madagascar  
Last Modification: 01/09/2025 12:13:33

Example Home Page

# Instructions


## Step 1: File and Folder Setup

1. In VS Code, open the **wdd131** local repository folder.
2. Create a new page in this root folder named "**index.html**".

Why is this file named "index.html"?

► Check Your Understanding

3. Create new folders (directories) named "**styles**", "**scripts**", and "**images**".  
These sub-folders, on the repository root directory, will contain the relevant, asset files.

Note that 'directory' and 'folder' essentially have the same meaning. Directory is the more accurate term for file systems while 'folder'  refers to the widely accepted graphical metaphor because of its association with physical files and organization.

4. Add a CSS file named "**base.css**" to the "styles" directory. This is your main CSS file.
5. Add a JS file named "**getdates.js**" to the "scripts" directory.

## Step 2: HTML

1. In the **index.html** document, include the:
  - ✓ required **document type** definition
  - ✓ **html** tag with language attribute
  - ✓ **head** tag
  - ✓ **body** tag
2. In the **<head>**, include the following elements with appropriate attributes and content:

Refer to the [course development standards](https://byui-cse.github.io/wdd131-ww-course/week01/assignment-01.html) if you need to review any of these **<head>** elements.

✓ **meta charset**

- ✓ **meta viewport**
- ✓ **title**
- ✓ **meta description**
- ✓ **meta author**

3. Set the **title** content to "**WDD 131 - Dynamic Web Fundamentals - *Your Name***".

4. Phrase the **meta description** that it at least contains the following content:

- ✓ "WDD 131 - Dynamic Web Fundamentals"
- ✓ Your full name
- ✓ Keyword summary of page content

5. In the **<body>**, create a layout using a **header**, a **main**, and a **footer** element.

6. The **<header>** element contains:

- ✓ the page heading as shown in an **<h1>** tag
- ✓ a **<nav>** tag with these four menu links:

- Your **LinkedIn** profile page.

If you do not have a free LinkedIn account profile, consider making one or just provide a general link to LinkedIn.

- **Canvas** using <https://byupw.instructure.com/>.
- Link to a favorite **Scripture** using <https://www.churchofjesuschrist.org/study/scriptures>

#### Example

<https://www.churchofjesuschrist.org/study/scriptures/bofm/1-ne/3?lang=eng&id=p7#p7>

- **Final Project** folder link (this will be a placeholder for now).

7. The **<main>** element contains the following:

- ✓ Two **section** tags each with the class attribute named "**card**". These sections should include a heading, images, and content as shown in the example figure above.

In this class, all images must be optimized and must be local (no external, absolute references to images).

8. The **<footer>** has two paragraphs **p**:

✓ The first paragraph contains the following:

- The copyright symbol and current year where the year will be dynamically populated using JavaScript code. Provide a blank **<span>** tag to do this.

#### Example

```
<span id="currentyear"></span>
```

- Your name.
- Your state or country.

✓ The second paragraph has an id of "**lastModified**" and will be populated with JavaScript code.


## Step 3: CSS

Use the external **base.css** file to layout and style the page as shown in the example screenshot above.

1. Use your color schema and typography choices.

You are responsible to practice good design principles of alignment, color contrast, proximity, repetition, and usability in all of your work.

2. Use the [Google Fonts API](#) to select one or two fonts to use on the page.

If you need help using Google Fonts, watch this demonstration video  [Google Fonts API](#)

3. Use **CSS Flex** on the navigation **nav**.

Demonstration Video:  [CSS Flex Navigation Menu](#)  
CodePen Source:  [CSS Flex Menu](#)

4. The navigation must employ a well designed **hover** affect. See the Codepen above for an example.
5. The **main** element has a limited width and is centered on the screen horizontally.
6. Layout the main column cards using **CSS Grid**.

## Step 4: JavaScript

1. Reference the "**getdates.js**" JavaScript file by using a `<script>` reference in the **head** of the HTML file and using the attribute **defer**.

Why is the [defer](#) boolean attribute important?  
How else can we successful reference and employ JavaScript?

2. In **getdates.js**, write JavaScript statements that dynamically output the following:
  - ✓ the copyright year (the current year) in the footer's first paragraph, and

Note this  [CodePen](#) summary of using the **Date** object in different ways.

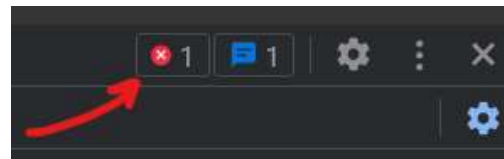
- ✓ the date the document was last modified in the second paragraph.

Use the [lastModified](#) property of the document object to get this date/time dynamically.  
You do **not** need to alter the **document.lastModified** native format.

## Step 5: Testing

Demonstration Video:  [Testing and auditing your own work](#)

1. Continuously check your work by rendering the page locally using Live Server in VS Code.
2. Use the browser's **DevTools** to check for JavaScript runtime errors in the console or click the red, error icon in the upper right corner of DevTools.



"DevTools" is an abbreviation for "Developer Tools." It refers to a set of tools or utilities provided by web browsers to help developers debug, profile, and analyze web pages during the development process. The tools are typically

accessed by pressing the F12 function key or selecting the menu option for the browser's developer tools.

3. Use DevTools [CSS Overview](#) to check your color contrast.
4. Generate the DevTools **Lighthouse** report and run diagnostics for **Accessibility**, **Best Practices**, and **SEO** in the Desktop view.

The **DevTools Lighthouse** tool serves as a comprehensive auditing tool integrated into the browser. In this class, you will focus on accessibility, best practices, SEO (Search Engine Optimization), and page weight measured by the total transfer bytes when the page is hard reloaded from an empty cache. For more information, see the [Lighthouse documentation](#).

It is best to test your page in a **Private** or **Incognito** browser window.

5. Hard reload the page using **Empty Cache and Hard Reload** in **DevTools** with the **Network** tab open to view the total transferred bytes at the bottom of the tab. Check that the page is 500 kB or less. Here are some common things to look for if the page is too large:
  - Make sure that all images are optimized.
  - Do not use third party libraries that bloat your page.

---

## Submission and Audit

1. Commit your local repository and push or upload your work to your GitHub Pages enabled wdd 131 repository on GitHub.
2. Use this ☒ [audit tool](#) to self-check your work for some of the required HTML elements and CSS content. This audit tool is also used by the assessment team.
3. **Share** your work by posting your URL in the course's **Microsoft Teams weekly forum** and comment on your peers' work by providing constructive feedback.
4. Return to Canvas and submit your GitHub Pages enabled URL for wdd131, e.g.,

**`https://your-github-username.github.io/wdd131`**

Note that you do not need to include **index.html** in your reference because

index.html is the default file retrieved with a folder request.

---

Copyright © Brigham Young University-Idaho | All rights reserved