

Week 4 – Web Design – Sydney Campus



1. Summary of Lecture 3
2. Tutorial Week 4
3. More Practical Examples
4. Attendance & Tutorial Questions - Recognising student participation and engagement specifically identifying those who are most actively involved!
5. Key assessment dates

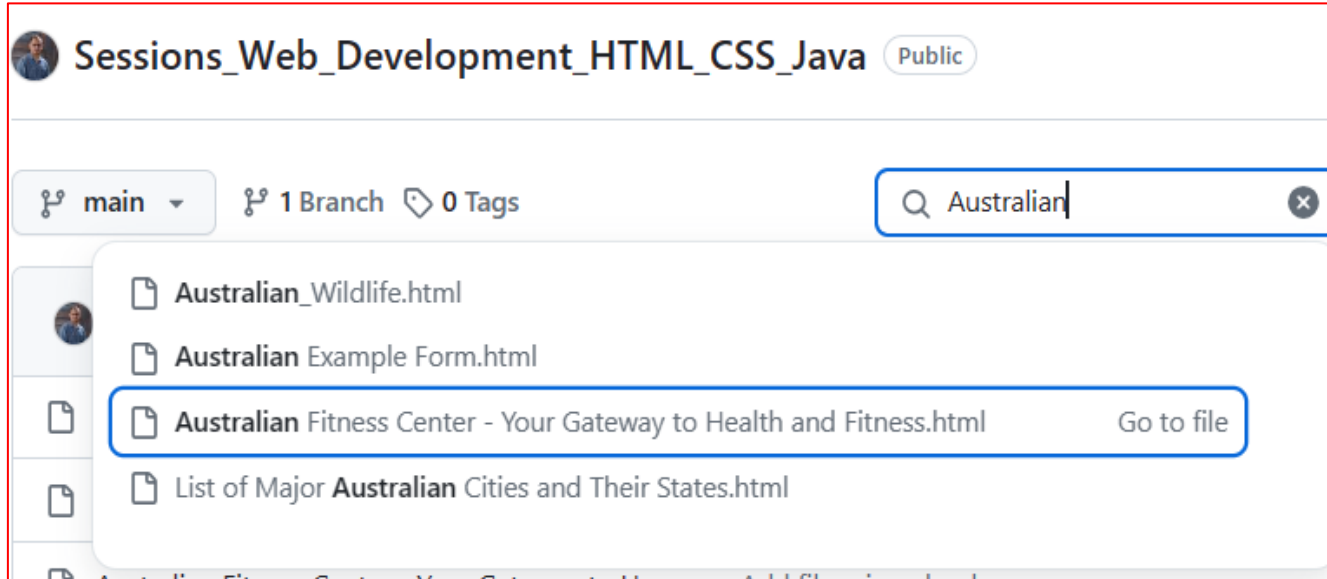
Lecturer/Tutor: Dr. Farshid Keivanian

1. Summary of Lecture 3

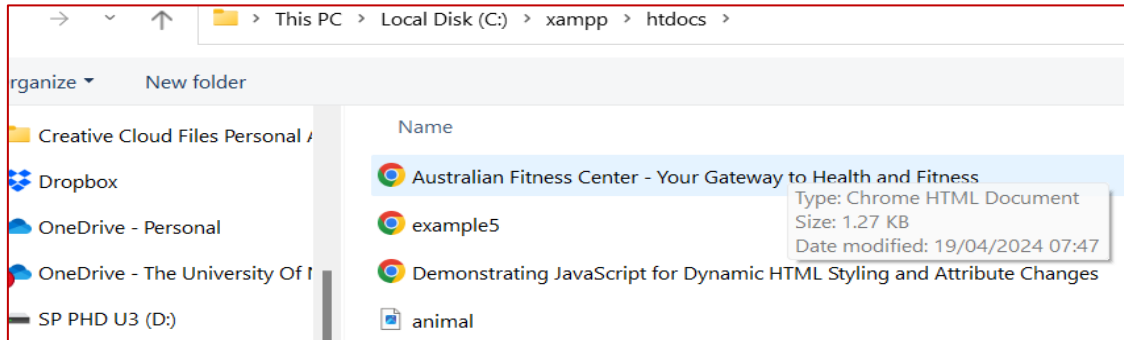
- Lecture 3, titled "Web Design Basics," covered the essential aspects of HTML and webpage design. The key objectives were to manage website files effectively, utilize HTML5 semantic elements, and construct the structural design of a webpage. The lecture also emphasized designing a semantic wireframe, creating and validating webpage templates, and adding unique content tailored to specific needs. It discussed the importance of understanding a website's purpose, its target audience, and ensuring an optimal viewing experience across different devices. Practical examples included creating a basic site map, organizing files within a website, and incorporating static and dynamic content to enhance the user experience.

1. Summary of Lecture 3 - Practical Example: Creating an HTML Page for an Australian Fitness Center

- A simple example of an HTML page for a fitness center based in Australia, following the guidelines from the lecture:



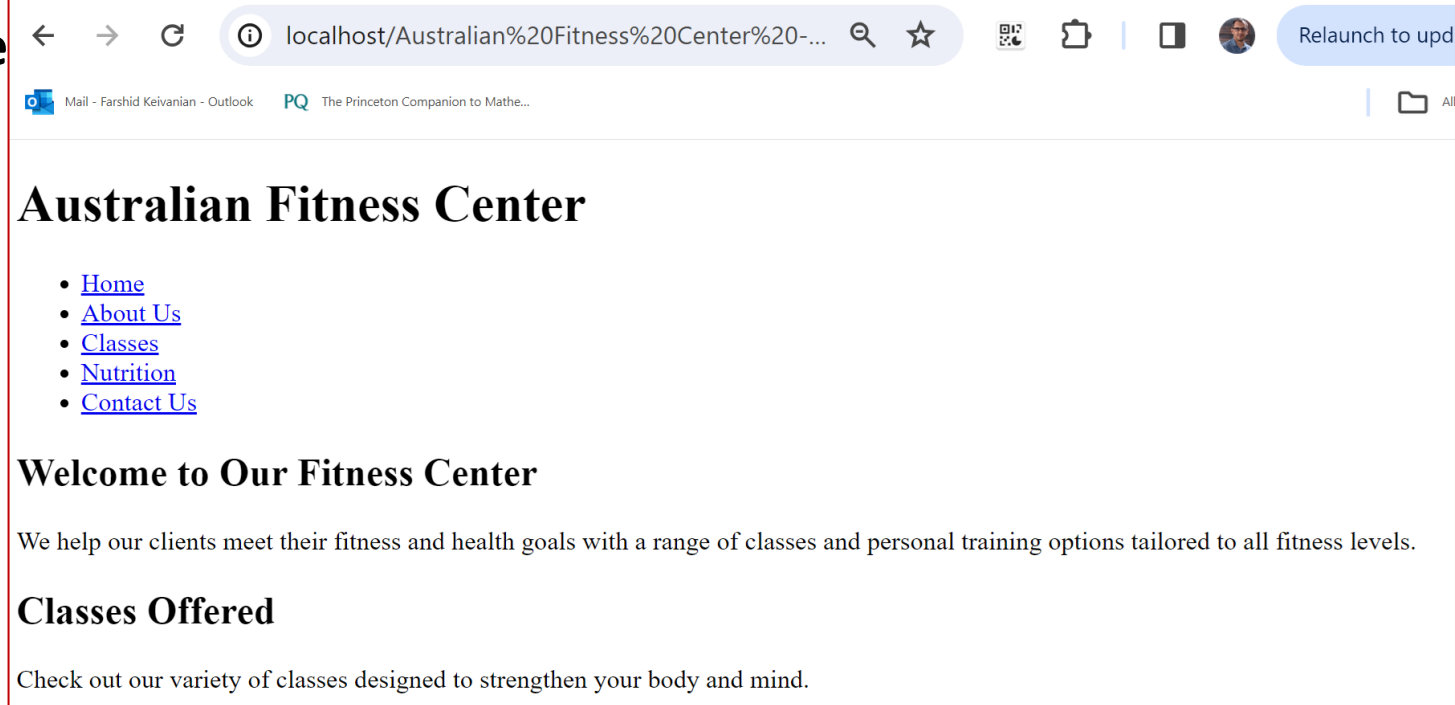
Change it and Save it in C:\\Xampp >> htdocs



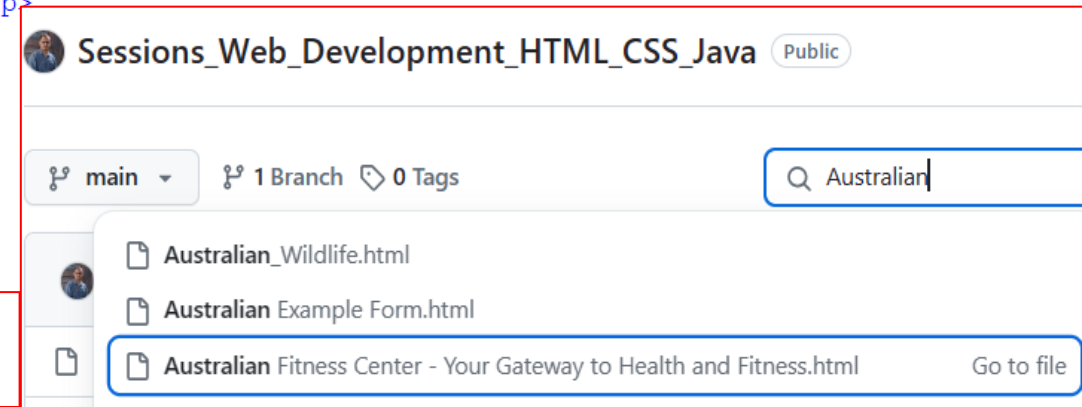
This HTML template uses semantic elements like `<header>`, `<nav>`, `<main>`, and `<footer>` to structure the webpage effectively. Each section is clearly defined, making the content accessible and the page easier to navigate, aligning with the best practices discussed in the lecture. This structure also aids in SEO and ensures compatibility across various devices, catering to the needs of a diverse Australian audience.

1. Summary of Lecture 3 – Practical Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Australian Fitness Center</title>
</head>
<body>
  <!-- Header section with navigation links -->
  <header>
    <h1>Australian Fitness Center</h1>
    <nav>
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="about.html">About Us</a></li>
        <li><a href="classes.html">Classes</a></li>
        <li><a href="nutrition.html">Nutrition</a></li>
        <li><a href="contact.html">Contact Us</a></li>
      </ul>
    </nav>
  </header>
  <!-- Main content area -->
  <main>
    <section>
      <h2>Welcome to Our Fitness Center</h2>
      <p>We help our clients meet their fitness and health goals with a range of classes and personal training options tailored to all fitness levels.</p>
    </section>
    <section>
      <h2>Classes Offered</h2>
      <p>Check out our variety of classes designed to strengthen your body and mind.</p>
    </section>
  </main>
  <!-- Footer section with contact information -->
  <footer>
    <p>Contact us at info@australianfitness.com or visit us in Sydney, NSW.</p>
  </footer>
</body>
</html>
```



The code is available online for my students!



2. Tutorial Week 4

The purpose of the assignment as detailed in the provided documents is to practice and understand the application of semantic HTML5 elements within the structure of a web page, and to apply CSS styling techniques to enhance the presentation of the content. This assignment ties into the overall learning goals of web design in several ways:

1. **Understanding Semantic HTML:** By creating a template with semantic wireframe elements like **header**, **nav**, **main**, and **footer**, you learn the importance of semantic HTML in web design. Semantic HTML5 elements convey meaning about the type of content they contain, not just its appearance, which is crucial for accessibility and search engine optimization (SEO).
2. **Practical Application:** The tasks require you to apply the concepts learned in lectures by creating a webpage template and styling it using CSS. This solidifies your theoretical knowledge through practical application.

2. Tutorial Week 4

3. **Research and Documentation:** The assignment asks you to research various HTML elements and document your findings as comments in the code, promoting the skill of self-learning and the importance of code documentation for future reference.
4. **Webpage Validation:** By validating the webpage using the W3C validator, you ensure that the HTML and CSS code adheres to web standards, which is important for cross-browser compatibility and long-term maintainability of the code.
5. **Styling with CSS:** The project requires the use of external, embedded, and inline CSS to style the webpage, giving you hands-on experience with different methods of applying styles, and understanding the advantages and best use-cases for each method.

2. Tutorial Week 4

6. **CSS Layout Techniques:** You will practice CSS layout techniques such as using the float property and the margin property to align elements, which are essential skills for creating visually appealing and well-structured web pages.
7. **Problem Solving and Debugging:**

In summary, this assignment is designed to develop a comprehensive set of web design skills, from understanding the structure and semantics of HTML to applying styles and layouts with CSS. These are fundamental skills for any web designer or developer and are essential for creating efficient, accessible, and maintainable web pages

2. Tutorial Week 4

- Use the **apply02.docx** document and the **Weeks 4-5 Lab.pdf** as guides through each task. For HTML and CSS syntax, you can refer to Notepad++ or web resources such as <https://w3schools.com/>
- Given the instructions and details from the **Weeks 4-5 Lab** PDF and the apply02.docx, let's write the HTML code needed to create a web page template with semantic HTML5 elements and then style it with CSS.

2. Tutorial Week 4

Step 1) HTML Code for Template Creation (template02.html)

Here's the basic structure with semantic HTML5 elements and HTML comments to indicate the type of content each section should contain:

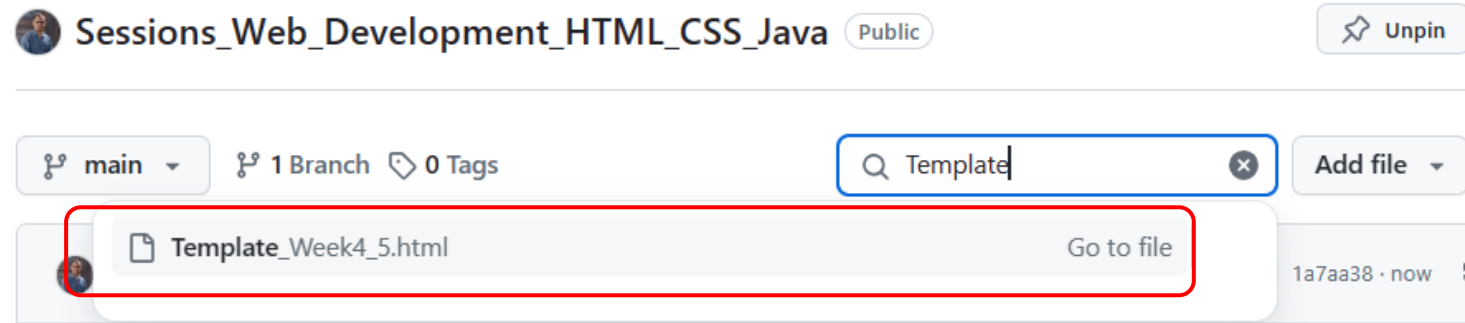
- Please always change, showcase your coding skills, then save it as **html extension** in file in **C:\xampp\htdocs**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title><!-- The title element briefly describes the topic of the document --></title>
</head>
<body>
  <!-- The header element represents a container for introductory content or a set of navigational links -->
  <header>
    <!-- Navigation links can go here -->
  </header>

  <!-- The nav element represents a section of a page whose purpose is to provide navigation links -->
  <nav>
    <!-- Menu items or the table of contents can go here -->
  </nav>

  <!-- The main element represents the dominant content of the <body> of a document -->
  <main>
    <!-- Primary content, like articles or page-specific content, goes here -->
  </main>

  <!-- The footer element represents a footer for its nearest ancestor sectioning content or sectioning root element -->
  <footer>
    <!-- Footer content, like contact info or page-specific links, goes here -->
  </footer>
</body>
</html>
```



2. Tutorial Week 4 – Understanding the Main Document

Project 1: Creating a Template

with a Semantic Wireframe

1. **HTML Tags Entry:** Use an HTML editor to enter HTML tags for creating a webpage template with semantic HTML5 structural tags.

- You will need to input HTML tags as specified in the document and use indents to make the code readable.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title><!-- The title element briefly describes the topic of the document --></title>
</head>
<body>
  <!-- The header element represents a container for introductory content or a set of navigational links -->
  <header>
    <!-- Navigation links can go here -->
  </header>

  <!-- The nav element represents a section of a page whose purpose is to provide navigation links -->
  <nav>
    <!-- Menu items or the table of contents can go here -->
  </nav>

  <!-- The main element represents the dominant content of the <body> of a document -->
  <main>
    <!-- Primary content, like articles or page-specific content, goes here -->
  </main>

  <!-- The footer element represents a footer for its nearest ancestor sectioning content or sectioning root element -->
  <footer>
    <!-- Footer content, like contact info or page-specific links, goes here -->
  </footer>
</body>
</html>
```

2. Tutorial Week 4 – Understanding the Main Document

2. Title Element Research:

Utilise a search engine to understand the <'title'> element and document findings as HTML comments. You need to research what a title element is and its importance for a web page.

3. Applying Research:

Save the webpage addresses used for research in the apply02.docx document.

- This involves documenting sources of your research for future reference or validation.

2. Tutorial Week 4 – Understanding the Main Document

4. Semantic HTML Research: You will need to answer specific questions about semantic HTML5 elements (**header, nav, main, footer**) using a search engine.

- This will require you to understand what each tag represents and its role in the structure of a webpage.

5. HTML Commenting: Replace comments with researched answers in the **template02.html** document.

- You will replace placeholder comments with the appropriate explanations as HTML comments.

6. Validation with W3C: Validate the **template02.html** document using W3C validator and correct any issues.

- This step ensures that the HTML code adheres to current web standards.

Project 2: Styling a Webpage

1. CSS Styling: Create styles for the semantic wireframe sections (**header, nav, main, footer**)

and a **div** element.

- You will create an external stylesheet, link it to the HTML document, and apply styles as instructed.

2. Centering Content: Use CSS to center content within a **div** wrapper.

- A specific **div** with an id of **outerwrapper** will be styled to center the page content.

3. Font Family Styling: Apply sans-serif and serif fonts to different sections of the webpage.

- This requires understanding which font styles are appropriate for various parts of a webpage.

Project 2: Styling a Webpage

4. Image Floating: Float an image to the left and clear the float where necessary.

1. You will learn how to manipulate text flow around images.

5. Margins, Borders, and Professional Touch: Add margins and borders to the page elements to make the page look professional.

1. This teaches the aesthetic spacing and bordering of page elements.

6. Embedded and Inline Styles: Use different methods of CSS application (external, embedded, and inline) and understand their use cases.

1. You will apply CSS directly within the HTML document for specific elements.

7. Code Validation and Submission: Validate the code again using W3C and submit the files as per instructor's format.

2. Tutorial Week 4 – Understanding the Code - Main Document

1. Document Type and Meta Information:

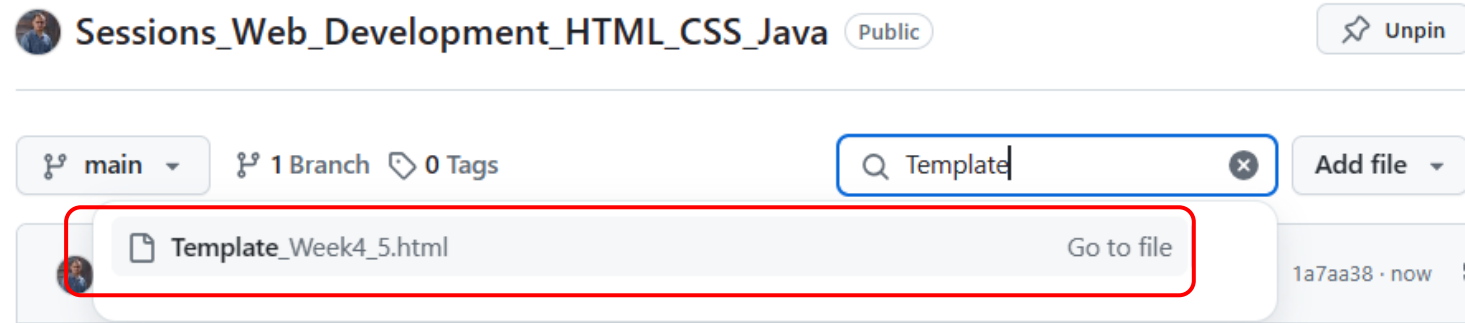
- **<!DOCTYPE html>**: Declares the document type and version of HTML (HTML5).
- **<meta charset="UTF-8">**: Specifies the character encoding for the webpage (UTF-8).
- **<meta name="viewport" content="width=device-width, initial-scale=1.0">**: Ensures the page is optimized for responsive design.
- **<title>**: The title of the document, which appears in the browser tab; it should describe the topic or contents of the document.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title><!-- The title element briefly describes the topic of the document --></title>
</head>
<body>
  <!-- The header element represents a container for introductory content or a set of navigational links -->
  <header>
    <!-- Navigation links can go here -->
  </header>

  <!-- The nav element represents a section of a page whose purpose is to provide navigation links -->
  <nav>
    <!-- Menu items or the table of contents can go here -->
  </nav>

  <!-- The main element represents the dominant content of the <body> of a document -->
  <main>
    <!-- Primary content, like articles or page-specific content, goes here -->
  </main>

  <!-- The footer element represents a footer for its nearest ancestor sectioning content or sectioning root element -->
  <footer>
    <!-- Footer content, like contact info or page-specific links, goes here -->
  </footer>
</body>
</html>
```



2. Tutorial Week 4 – Understanding the Code - Main Document

2. Structural Elements:

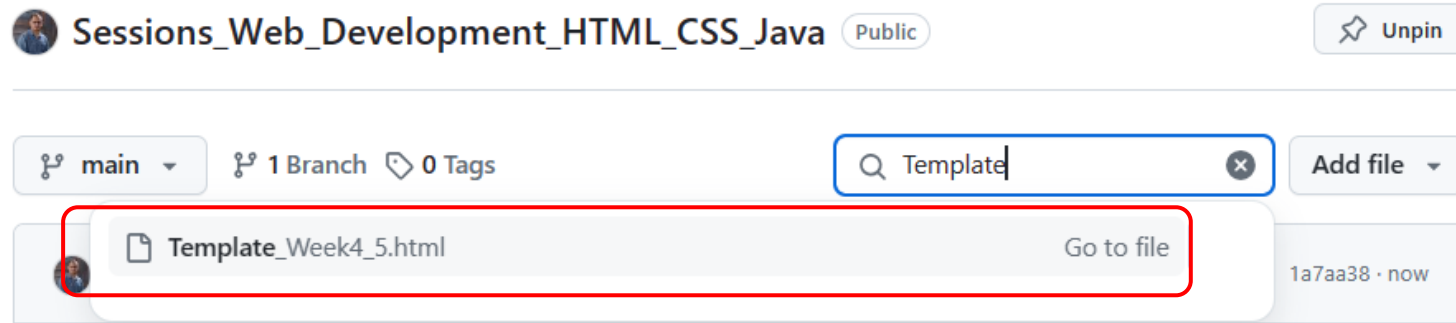
- **<header>**: Typically contains introductory content or navigation links. It's the first thing users see in the "body" of the document.
- **<nav>**: Dedicated to navigation links, helping users navigate the site or app.
- **<main>**: The core content of the webpage, distinct from sidebars, footers, and navigation bars. It's the primary focus of the document.
- **<footer>**: Contains auxiliary information like copyrights, links to privacy policies, or contact info.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title><!-- The title element briefly describes the topic of the document --></title>
</head>
<body>
  <!-- The header element represents a container for introductory content or a set of navigational links -->
  <header>
    <!-- Navigation links can go here -->
  </header>

  <!-- The nav element represents a section of a page whose purpose is to provide navigation links -->
  <nav>
    <!-- Menu items or the table of contents can go here -->
  </nav>

  <!-- The main element represents the dominant content of the <body> of a document -->
  <main>
    <!-- Primary content, like articles or page-specific content, goes here -->
  </main>

  <!-- The footer element represents a footer for its nearest ancestor sectioning content or sectioning root element -->
  <footer>
    <!-- Footer content, like contact info or page-specific links, goes here -->
  </footer>
</body>
</html>
```



Step 2) CSS Styling for the Template (styles.css) - Please always change, showcase your coding skills, then save it as **Cascade Style Sheets File (CSS)** extension in file in **C:\xampp\htdocs**

```
/* Add a CSS comment with your name and the current date */
/* Your Name - Date */

body {
    color: #000000;
    background-color: #D9F1C1;
    font-family: "Times New Roman", serif;
}

#outerwrapper {
    width: 80%;
    margin: 0 auto; /* This will center the div on the page */
}

header, nav, footer {
    font-family: Calibri, Arial, sans-serif;
}

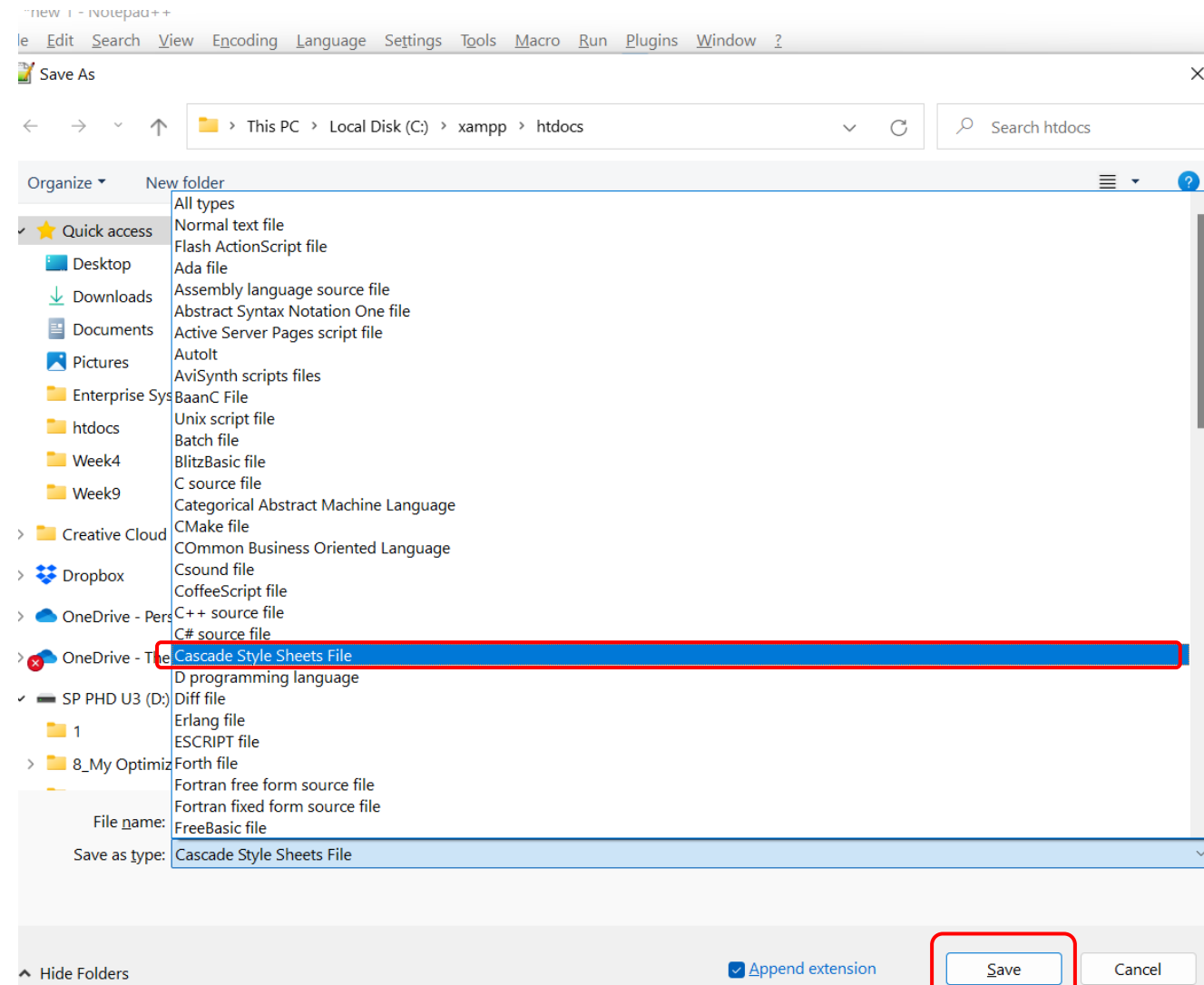
#main {
    font-family: Garamond, "Times New Roman", serif;
    background-color: #F0F9E6;
    padding: 10px;
}

img {
    float: left;
    margin-right: 2%; /* Creates space between the image and the text */
}

nav {
    clear: both; /* Ensures that the nav element clears the floated image */
}

footer {
    font-size: 0.75em;
    text-align: center; /* Centers the text within the footer */
}

/* Additional styles can be added as needed */
```



Explanation of the CSS code:

1. Comment with name and date: This part of the code is a and the current date. Comments in CSS are ignored by browsers and serve only for documentation purposes.

```
/* Your Name – Date */
```

```
/* Add a CSS comment with your name and the current date */
/* Your Name – Date */

body {
    color: #000000;
    background-color: #D9F1C1;
    font-family: "Times New Roman", serif;
}

#outerwrapper {
    width: 80%;
    margin: 0 auto; /* This will center the div on the page */
}

header, nav, footer {
    font-family: Calibri, Arial, sans-serif;
}

#main {
    font-family: Garamond, "Times New Roman", serif;
    background-color: #F0F9E6;
    padding: 10px;
}

img {
    float: left;
    margin-right: 2%; /* Creates space between the image and the text */
}

nav {
    clear: both; /* Ensures that the nav element clears the floated image */
}

footer {
    font-size: 0.75em;
    text-align: center; /* Centers the text within the footer */
}

/* Additional styles can be added as needed */
```

2. Body Styling:

- **color: #000000;** This sets the text color of the entire document to black.
- **background-color: #D9F1C1;** The background color of the entire page is set to a light green.
- **font-family: "Times New Roman", serif;** The default font for the page is set to Times New Roman with a fallback to any generic serif font if Times New Roman is not available.

```
/* Add a CSS comment with your name and the current date */
/* Your Name - Date */

body {
    color: #000000;
    background-color: #D9F1C1;
    font-family: "Times New Roman", serif;
}

#outerwrapper {
    width: 80%;
    margin: 0 auto; /* This will center the div on the page */
}

header, nav, footer {
    font-family: Calibri, Arial, sans-serif;
}

#main {
    font-family: Garamond, "Times New Roman", serif;
    background-color: #F0F9E6;
    padding: 10px;
}

img {
    float: left;
    margin-right: 2%; /* Creates space between the image and the text */
}

nav {
    clear: both; /* Ensures that the nav element clears the floated image */
}

footer {
    font-size: 0.75em;
    text-align: center; /* Centers the text within the footer */
}

/* Additional styles can be added as needed */
```

3. Outer Wrapper:

- **width: 80%;**: The width of the **#outerwrapper** element is set to 80% of its parent element, typically the **body**.
- **margin: 0 auto;**: This centers the **#outerwrapper** element horizontally within its parent element.

4. Header, Nav, Footer Styling:

- **font-family: Calibri, Arial, sans-serif;**: These elements are styled with a sans-serif font, preferring Calibri or Arial.

```
/* Add a CSS comment with your name and the current date */
/* Your Name - Date */

body {
    color: #000000;
    background-color: #D9F1C1;
    font-family: "Times New Roman", serif;
}

#outerwrapper {
    width: 80%;
    margin: 0 auto; /* This will center the div on the page */
}

header, nav, footer {
    font-family: Calibri, Arial, sans-serif;
}

#main {
    font-family: Garamond, "Times New Roman", serif;
    background-color: #F0F9E6;
    padding: 10px;
}

img {
    float: left;
    margin-right: 2%; /* Creates space between the image and the text */
}

nav {
    clear: both; /* Ensures that the nav element clears the floated image */
}

footer {
    font-size: 0.75em;
    text-align: center; /* Centers the text within the footer */
}

/* Additional styles can be added as needed */
```

5. Main Section Styling:

- **font-family: Garamond, "Times New Roman", serif;** The text within the **#main** section uses Garamond, with Times New Roman as a fallback.
- **background-color: #F0F9E6;** A lighter shade of green is set as the background color.
- **padding: 10px;** Adds padding around the content inside the **#main** section, increasing its readability.

6. Image Styling:

- **float: left;** Images will float to the left, allowing text to wrap around on the right.
- **margin-right: 2%;** Adds a right margin to the images, creating space between them and any adjacent content.

```
/* Add a CSS comment with your name and the current date */
/* Your Name - Date */

body {
    color: #000000;
    background-color: #D9F1C1;
    font-family: "Times New Roman", serif;
}

#outerwrapper {
    width: 80%;
    margin: 0 auto; /* This will center the div on the page */
}

header, nav, footer {
    font-family: Calibri, Arial, sans-serif;
}

#main {
    font-family: Garamond, "Times New Roman", serif;
    background-color: #F0F9E6;
    padding: 10px;
}

img {
    float: left;
    margin-right: 2%; /* Creates space between the image and the text */
}

nav {
    clear: both; /* Ensures that the nav element clears the floated image */
}

footer {
    font-size: 0.75em;
    text-align: center; /* Centers the text within the footer */
}

/* Additional styles can be added as needed */
```

7. Navigation Clearing:

- **clear: both;**: Ensures that the **nav** element does not wrap around any floated elements above it, effectively clearing previous floats.

8. Footer Styling:

- **font-size: 0.75em;**: The font size in the footer is smaller than the rest of the text.
- **text-align: center;**: Centers the text within the footer.

```
/* Add a CSS comment with your name and the current date */
/* Your Name - Date */

body {
    color: #000000;
    background-color: #D9F1C1;
    font-family: "Times New Roman", serif;
}

#outerwrapper {
    width: 80%;
    margin: 0 auto; /* This will center the div on the page */
}

header, nav, footer {
    font-family: Calibri, Arial, sans-serif;
}

#main {
    font-family: Garamond, "Times New Roman", serif;
    background-color: #F0F9E6;
    padding: 10px;
}

img {
    float: left;
    margin-right: 2%; /* Creates space between the image and the text */
}

nav {
    clear: both; /* Ensures that the nav element clears the floated image */
}

footer {
    font-size: 0.75em;
    text-align: center; /* Centers the text within the footer */
}

/* Additional styles can be added as needed */
```

Step 3) Linking the CSS to HTML

In the <head> of the HTML document, add the link to the external CSS file:

```
<link rel="stylesheet" href="Style_Week4_5.css">
```

Remember to validate your HTML and CSS using the W3C validators to ensure that there are no errors in your code. Also, you may need to replace placeholders with actual content and URLs based on the project requirements.

To link the external CSS file to the HTML document, we'll need to place the link tag inside the <head> section of your HTML code. Assuming your CSS file is named "Style_Week4_5.css" and it's located in the same directory as our HTML file, here is how we would modify the HTML

HTML linked to the external CSS

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title><!-- The title element briefly describes the topic of the document --></title>
  <!-- Link to external CSS file -->
  <link rel="stylesheet" href="Style_Week4_5.css">
</head>
<body>
  <!-- The header element represents a container for introductory content or a set of navigational links -->
  <header>
    <!-- Navigation links can go here -->
  </header>

  <!-- The nav element represents a section of a page whose purpose is to provide navigation links -->
  <nav>
    <!-- Menu items or the table of contents can go here -->
  </nav>

  <!-- The main element represents the dominant content of the <body> of a document -->
  <main>
    <!-- Primary content, like articles or page-specific content, goes here -->
  </main>

  <!-- The footer element represents a footer for its nearest ancestor sectioning content or sectioning root element -->
  <footer>
    <!-- Footer content, like contact info or page-specific links, goes here -->
  </footer>
</body>
</html>
```

- Ensure that the path in the **href** attribute correctly points to the location of our CSS file. If the CSS file is in a subfolder, for instance, a folder named "css", the **href** would be **css/Style_Week4_5.css**. If it's in the same directory, just using **Style_Week4_5.css** as shown above is sufficient.
- Remember that the name of the CSS file in the **href** attribute must match exactly the name of the CSS file we've created, including letter case, as some servers are case-sensitive.

Choose Title for the HTML)

- To choose the best **title** for the HTML document, consider the content and purpose of the website. The **<title>** tag is crucial because it provides context for what the page is about, and it's the text that appears in the browser tab, bookmarks, and search results. Here's how to think about a good title:
 1. **Descriptive:** It should clearly indicate the content of the page.
 2. **Concise:** Keep it short but meaningful.
 3. **Unique:** Each page on your website should have a unique title to help distinguish it in search results and bookmarks.
 4. **Keyword-Rich:** Include keywords that are relevant to the content of your page, which can help with SEO.

Choose Title for the HTML)

For instance:

- If it's a homepage for a business, you might use: "Brand Name - Leading Solutions in Industry".
- If it's an informational page about a specific topic, a title like "Complete Guide to Topic - Brand Name" could be appropriate.

Let's fill in the <title> tag with a placeholder based on a hypothetical topic. For example, if your site is about sustainable living, the title could be:

```
<title>Sustainable Living Tips and Practices | GreenGuide</title>
```


You can replace "Sustainable Living Tips and Practices | GreenGuide" with whatever fits your page content and purpose best!

By following these steps and employing the mentioned documents, you should gain a solid understanding of both the practical and theoretical aspects of creating and styling a semantic wireframe for a webpage.

- **Creating A Web Saver Using Xampp or Aptana Studio**

Install Xampp as the webserver on your systems and test it.


<https://www.apachefriends.org/de/download.html>

 **XAMPP for Windows** 8.0.30, 8.1.25 & 8.2.12

Version		Checksum			Size
8.0.30 / PHP 8.0.30	What's Included?	md5	sha1	Download (64 bit)	144 Mb
8.1.25 / PHP 8.1.25	What's Included?	md5	sha1	Download (64 bit)	148 Mb
8.2.12 / PHP 8.2.12	What's Included?	md5	sha1	Download (64 bit)	149 Mb

[Requirements](#) [More Downloads »](#)

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

 **XAMPP for Linux** 8.0.30, 8.1.25 & 8.2.12

Version		Checksum			Size
8.0.30 / PHP 8.0.30	What's Included?	md5	sha1	Download (64 bit)	151 Mb
8.1.25 / PHP 8.1.25	What's Included?	md5	sha1	Download (64 bit)	153 Mb
8.2.12 / PHP 8.2.12	What's Included?	md5	sha1	Download (64 bit)	151 Mb

[Requirements](#) [More Downloads »](#)

 **XAMPP for OS X** 8.0.28, 8.1.17 & 8.2.4

Version		Checksum			Size
8.0.28 / PHP 8.0.28	What's Included?	md5	sha1	Download (64 bit)	150 Mb
8.1.17 / PHP 8.1.17	What's Included?	md5	sha1	Download (64 bit)	151 Mb
8.2.4 / PHP 8.2.4	What's Included?	md5	sha1	Download (64 bit)	150 Mb

[Requirements](#) [More Downloads »](#)

A Native installer installs MariaDB, PHP, Perl, etc. directly onto your macOS system. It supports intel (x64) or Apple M1 (arm64) CPUs.

Setup



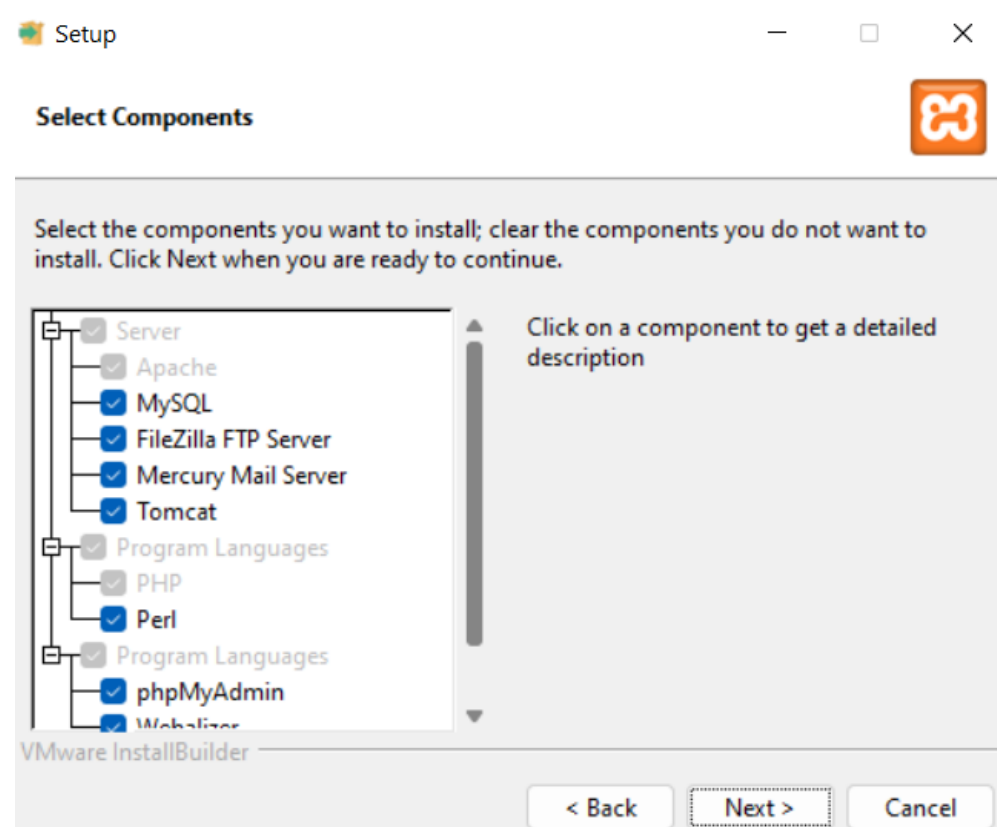
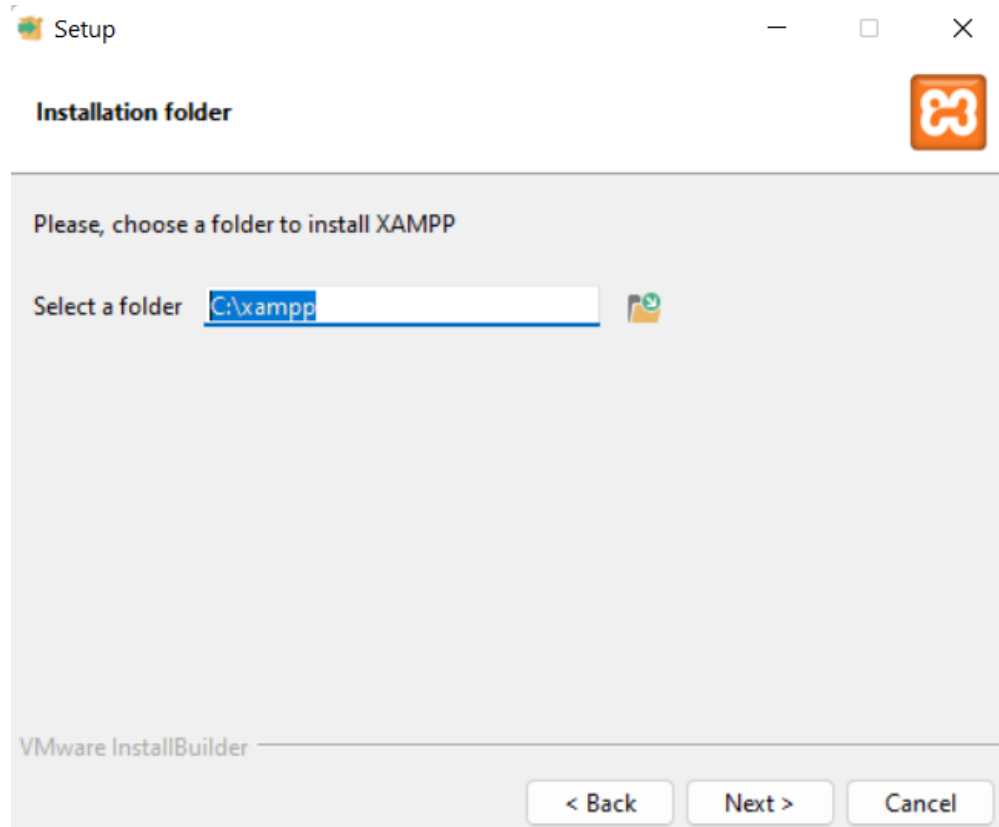
Setup - XAMPP

Welcome to the XAMPP Setup Wizard.

< Back

Next >

Cancel



Setup



Completing the XAMPP Setup Wizard


Setup has finished installing XAMPP on your computer.

☒ Do you want to start the Control Panel now?

< Back

Finish

Cancel



XAMPP Control Panel v3.3.0

Modules

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	11404 35064	80, 443	<div>Stop</div> <div>Admin</div> <div>Config</div> <div>Logs</div>
<input type="checkbox"/>	MySQL			<div>Start</div> <div>Admin</div> <div>Config</div> <div>Logs</div>
<input type="checkbox"/>	FileZilla			<div>Start</div> <div>Admin</div> <div>Config</div> <div>Logs</div>
<input type="checkbox"/>	Mercury			<div>Start</div> <div>Admin</div> <div>Config</div> <div>Logs</div>
<input type="checkbox"/>	Tomcat			<div>Start</div> <div>Admin</div> <div>Config</div> <div>Logs</div>

Config

Netstat

Shell

Explorer


Services

Help

Quit

```

23:13:02 [main] XAMPP Installation Directory: "c:\xampp\"
23:13:02 [main] Checking for prerequisites
23:13:04 [main] All prerequisites found
23:13:04 [main] Initializing Modules
23:13:04 [main] Starting Check-Timer
23:13:04 [main] Control Panel Ready
23:13:54 [Apache] Attempting to start Apache app...
23:13:54 [Apache] Status change detected: running
    
```


A top-down view of several wrapped gifts on a dark wood-grain surface. The gifts are wrapped in various patterns: red with white hearts, red with white snowflakes, brown with red plaid, and brown with red geometric patterns. Most are tied with red ribbons, while one is tied with green and white plaid, and another with white twine. In the bottom left, a pair of hands is shown unwrapping a gift with a red ribbon. The text "Thank you, Happy A Learning Day!" is overlaid in white, centered on the image.

Thank you,
Happy A Learning Day!