

TASK 1 - Create a Simple HTML Page (Source Code is available in github)

Create a basic HTML page with the following elements. Feel free to choose the content for each element:

Title: <title>

- define the title of the document

Meta tags: <meta>

- define metadata about an html document
- Metadata - data (information) about data.
- Always go inside the <head> element
- Specify character set, page description, keywords, author of the document and viewport settings
- Used by browsers, search engines

Headings: Use <h1>, <h2>, and <h3> for different levels of headings

Paragraphs: <p>

Lists: Create an ordered or unordered list (or) with at least 3 items

Links: <a> (Include both internal and external links)

Images: (Make sure to properly use the alt attribute)

Table: <table> (Create a table with at least 2 rows and 2 columns, using <tr> and <td>)

Multimedia: Include an audio (<audio>) and a video (<video>) element with controls enabled

Embed Content: Use <iframe> or <embed> to embed external content such as a YouTube video or a map

Comments: Use HTML comments (<!-- -->) to explain the purpose of each element

Task 2: Inspect the Source Code of a Website

Visit any website of your choice and inspect its HTML source code. Identify and explain the following elements:

Metadata: Locate the `<meta>` tags and explain their purpose in the page.

Headings and Semantic Elements: Examine the use of headings (`<h1>`, `<h2>`, etc.) and other semantic elements (e.g., `<header>`, `<footer>`, `<section>`).

Multimedia: Look for multimedia elements such as `<video>`, `<audio>`, or embedded content (e.g., `<iframe>`).

Security-Relevant Tags:

Identify any security-related tags such as Content Security Policy (CSP) or sandboxed iframes, and explain how they help secure the page.

CORS (Cross-Origin Resource Sharing): Look for CORS headers and explain their role in securing resources.

SRI (Subresource Integrity): Check if SRI attributes are used to ensure the integrity of external resources and explain their purpose.

Deliverable:

Submit a summary of your findings, including the metadata, headings, multimedia elements, and any security-related tags identified on the website.

Website: <https://pytorch.org/>

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
```

- **`<meta charset="UTF-8">`** - specifies the character encoding for HTML documents. UTF-8 is widely used and supports a wide range of characters (including special characters, symbols).
- **`<meta name="viewport" content="width=device-width, initial-scale=1.0">`** - ensures that the layout of the page adapts to any width of the device's screen and sets the initial zoom level to 1.
- **`<meta http-equiv="X-UA-Compatible" content="ie=edge">`** - ensures that older versions of Internet Explorer display your page correctly.

```
<footer class="site-footer">
  <div class="container footer-container">

    <div class="newsletter" id="newsletter">
```

- **`<footer>`** - represents the footer of a document usually containing copyright information, contact links and other footer content. In our case, they have mentioned about their organization's policy and their copyrights in footer section.

```
<div class="container">
  <h1><br /
```

- **<h1>** - used to define hierarchy and structure of content on a webpage that ranges from <h1> (the most important heading) to <h6> (the least important heading).

```
<noscript><iframe src="https://www.googletagmanager.com/ns.html?id=GTM-T8XT4PS"
height="0" width="0" style="display:none;visibility:hidden"></iframe></noscript>
```

- **<iframe>** - used to embed another HTML document within the current document. In our case, they have included www.googletagmanager.com page

Security-Relevant Tags: As far as I searched, I didn't find any of the security relevant tags in the website mentioned above. But below are some of the security relevant tags that are widely used.

- **Content Security Policy (CSP):** Prevents XSS and other code injection attacks.
- **HSTS (HTTP Strict Transport Security):** Ensures the use of HTTPS.
- **X-Content-Type-Options:** Prevents MIME sniffing attacks.
- **X-Frame-Options:** Prevents clickjacking by blocking iframe embedding.
- **Referrer-Policy:** Controls how much referrer information is shared.
- **Permissions-Policy:** Controls the use of features like camera, microphone, etc.
- **Secure Cookies:** Ensures cookies are sent securely and not accessible by JavaScript.
- **Cache-Control:** Prevents caching of sensitive data.
- **CORS:** Restricts cross-origin resource sharing.

Task 3: Inspect the Source Code of Your Page

Inspect the page you created in Task 1 using your browser's developer tools. Analyze the structure and content of the page and make sure the HTML elements are correctly implemented.

Deliverable:

Submit a report with your observations on the HTML structure and CSS usage. Include any suggestions for improvements or corrections you could make.

Observations: -

- The document starts with `<!DOCTYPE html>` which ensures HTML5 compliance. Followed by `<meta>` tags that define how the data must be in HTML document.
- Then comes the style section where various style formatting are defined for different kinds of sections.
- After comes the main body part wherein it is divided into various sections that includes various elements starting from heading, audio, video, image, iframe, table tag and so on.

Suggestions for improvement or corrections:-

- In terms of security perspective, the code can be improved by including `rel="noopener noreferrer"` along with `target="_blank"` in the external links. That ensures that clicking the link opens the target URL in a new tab without exposing the original page's URL or allowing the new page to manipulate the original page.

Task 4: Apply Styles Using CSS

In a separate external CSS file, apply the following styles to the webpage you created in Task 1: Change the font color of the <h1> heading.

Set the background color of the page.

Set the font size of the paragraph <p> to 16px (or as desired).

Style the list (or) by adding bullets and setting the list items to have a margin of 10px.

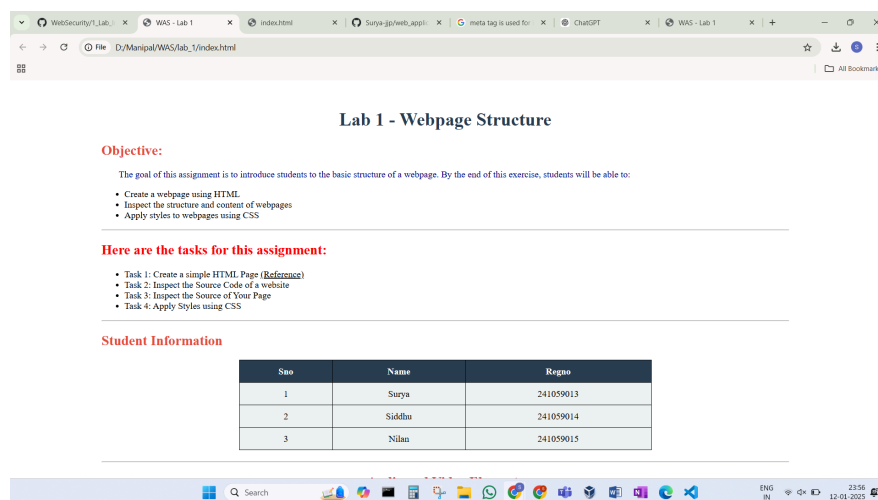
Add a border and padding to a <div> container element.

You may also apply additional styles of your choice to enhance the page.

Deliverable:

Submit the styled webpage, including the external CSS file with the required changes applied.

Before Styling -



After Styling -

