

Task 1: HTML and Basic Tags

Objective:

Build a comprehensive HTML webpage for your personal profile page that demonstrates the use of various HTML elements such as headings, paragraphs, links, images, lists, a navigation bar, buttons, icons, and tables. Additionally, incorporate HTML attributes to modify these elements' behavior and presentation.

Pre-requisites:

- Basic understanding of HTML elements and attributes
- Familiarity with a code editor like Visual Studio Code

Concepts Covered:

- HTML Document Structure
- Headings and Paragraphs
- Attributes
- Links
- Images
- Buttons
- Lists
- Navigation Bar
- Icons
- Tables

Concepts:

1. HTML Document Structure:

The basic structure of an HTML document includes the `<!DOCTYPE html>` declaration, `<html>`, `<head>`, and `<body>` tags.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document Title</title>
</head>
<body>
  <!-- Content goes here -->
</body>
</html>
```

2. Headings and Paragraphs:

Headings (`<h1>` to `<h6>`) define different levels of headings, and `<p>` tags are used for paragraphs.

```
<h1>Main Heading</h1>
<h2>Subheading</h2>
<p>This is a paragraph.</p>
```

3. Attributes:

Attributes provide additional information about HTML elements. Common attributes include `id`, `class`, and `style`.

```
<p id="para1" class="text" style="color: blue;">Styled paragraph</p>
```

4. Links:

The `<a>` tag creates hyperlinks. Use attributes like `href` for the URL, `target` to specify where to open the link, and `title` for additional information.

```
<a href="https://example.com" target="_blank" title="Visit Example">Visit
Example</a>
```

5. Images:

The `` tag embeds images. Use `src` for the image source and `alt` for alternative text.

```

```

6. Buttons:

The `<button>` tag creates clickable buttons. Use the `onclick` attribute to define a function to be called when the button is clicked.

```
<button onclick="alert('Button clicked!')">Click Me</button>
```

7. Lists:

`` and `` tags create unordered and ordered lists, respectively. `` defines list items.

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
<ol>
  <li>First Item</li>
  <li>Second Item</li>
</ol>
```

8. Navigation Bar:

A navigation bar can be constructed using a list of links.

```
<nav>
  <ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#about">About</a></li>
    <li><a href="#contact">Contact</a></li>
  </ul>
</nav>
```

9. Icons:

Icon libraries like Font Awesome can be used to include icons in your webpage.

```
<i class="fa fa-home"></i>
```

10. Tables:

The `<table>` tag creates a table, with `<tr>` for rows, `<th>` for headers, and `<td>` for cells. Use attributes like `colspan` and `rowspan` for spanning columns and rows.

```
<table border="1">
  <tr>
    <th>Header 1</th>
    <th>Header 2</th>
  </tr>
  <tr>
    <td>Data 1</td>
    <td>Data 2</td>
  </tr>
</table>
```

Setup:

1. Install Visual Studio Code (VS Code):

Download and install VS Code from [Visual Studio Code](#).

2. Web Browsers:

Use Google Chrome or Mozilla Firefox for viewing your webpage and utilizing their developer tools for debugging.

Tasks:

1. HTML Document Structure :

- Create an `index.html` file and write the `<!DOCTYPE html>` declaration.
- Define the structure with `html`, `head`, and `body` tags.

- In the `head` section, add a title and relevant meta tags for character set and viewport settings.

2. Headings, Paragraphs, and Attributes :

- Insert a variety of headings using `<h1>` through `<h6>` tags to understand their semantic importance.
- Write multiple paragraphs using the `<p>` tag, experimenting with text content.
- Practice adding attributes such as `id`, `class`, and `style` to these elements for identification and inline styling.

3. Links, Images, and Buttons :

- Use the `<a>` tag to create internal and external links. Explore attributes like `href`, `target`, and `title`.
- Embed images using the `` tag, ensuring proper use of `src` and `alt` attributes for accessibility.
- Introduce buttons with the `<button>` tag and implement a simple click event using the `onclick` attribute.

4. Lists, Navigation Bar, and Icons :

- Organize content with ordered and unordered lists, using the ``, ``, and `` tags.
- Construct a basic navigation bar by creating a list of links styled to be horizontal and interactive.
- Integrate icons into your webpage using an icon library like Font Awesome, exploring the `<i>` tag and class attribute for icon selection.

5. Tables and Table Attributes :


- Create a table with headers and multiple rows using the `<table>`, `<tr>`, `<th>`, and `<td>` tags.
- Utilize attributes like `colspan`, `rowspan`, and `border` to control the layout and design of the table.
- Experiment with adding a caption for the table and styling the table headers and cells using inline CSS.

Instructions:

1. Write the required code in `index.html`.
2. Open the file in your web browser to ensure the code displays correctly.
3. Use the browser's developer tools to debug and inspect the elements.


Resources:

o



HTML: HyperText Markup Language | MDN

HTML (HyperText Markup Language) is the most basic building block of the Web. It de...


<https://developer.mozilla.org/en-US/docs/Web/HTML>

o

Get started with
Visual Studio Code

Documentation for Visual Studio Code

Find out how to set-up and get the most from Visual Studio Code. Optimized for building a...



<https://code.visualstudio.com/docs>

Videos:

o



GitHub Instructions:

1. Open in Visual Studio Code:

After clicking on the "Open in Visual Studio Code" button from the GitHub Classroom confirmation page, VSCode will open the repository directly. If prompted, select "Open" or "Allow" to open the repository in VSCode.

2. Open the Terminal in VSCode:

In VSCode, open a terminal by selecting Terminal > New Terminal from the top menu.

3. Complete the Task:

In VSCode, write your solution in the `index.html` file.

4. **Run and Test Your

Code:**

Open your `index.html` file in a web browser to ensure it works correctly. Use the following command:

```
open index.html
```

5. Commit Your Changes:

In the VSCode terminal, add your changes to git:

```
git add index.html
```

Commit your changes with a meaningful message:

```
git commit -m "Completed task 1"
```

6. Push Your Changes to Your Repository:

Push your changes to your forked repository:

```
git push origin main
```

7. Create a Pull Request:

Go to your repository on GitHub.

Click on the "Pull Requests" tab.

Click the "New Pull Request" button.

Ensure the base repository is the original template repository and the base branch is `main`.

Ensure the head repository is your forked repository and the compare branch is `main`.

Click "Create Pull Request".

Add a title and description for your pull request and submit it.

Summary of Commands:

```
# Open in Visual Studio Code

# Open terminal in VSCode

# Complete the task by editing index.html

# Navigate to the directory containing index.html
cd path/to/your/index.html

# Run your code
open index.html

# Add, commit, and push your changes
git add index.html
git commit -m "Completed task 1"
git push origin main

# Create a pull request on GitHub
```