HTML (Hypertext Markup Language)

HTML is a markup language that is used to structure content on the web. HTML is not a programming language but a markup language used to design and structure content on a web page using tags.

Basic HTML Structure:

```
<!DOCTYPE html>

<html>

<head>

    <title>My First Web Page</title>

</head>

<body>
```

- **<!DOCTYPE html>**

  This is a doctype or document type declaration and not an html element. It is an instruction that tells the web client about the version of markup language that the web page has been written in. HTML 5 is the latest version of html that is used in creating and rendering web pages.

- **<html>**

  This is the root element and acts as a container for the entire content of a web page. It also helps in language settings

- **<head>**

This contains important metadata and instructions that aren’t visible to the user but they play a critical role in how the page functions, appears, and interacts with the outside world.

Metadata doesn't display directly on the webpage, but it affects how the page is handled. Examples of metadata include:

- **Title of the page** (`<title>`): Displays in the browser tab or title bar.
- **Character encoding** (`<meta charset="UTF-8">`): Specifies the character encoding for the page (ensures proper display of special characters, like accents or emojis).
- **Description of the page** (`<meta name="description" content="...">`): Often used by search engines to show a snippet of your page in search results.
- **Author** (`<meta name="author" content="...">`): Identifies the creator of the page.
- **Viewport settings**: `<meta name="viewport" content="width=device-width, initial-scale=1.0">` is important for responsive design. It ensures that the page looks good on mobile devices by controlling the layout scaling.

Linking external resources:

- **CSS stylesheets**: `<link rel="stylesheet" href="style.css">` links an external CSS file to style the page.
- **JavaScript files**: `<script src="script.js"></script>` can also be added here (though it's more common to place `<script>` tags at the end of the `<body>` tag for better performance).
- **Favicon**: `<link rel="icon" href="favicon.ico" type="image/x-icon">` links a small icon that appears in the browser tab.
- **Preloading resources**: `<link rel="preload" href="image.jpg" as="image">` can help preload critical resources (like images or fonts), improving the loading speed.

- **`<title>`**

  This sets the title in the web browser tab

- **`<body>`**

  This contains the content of the web page visible to the user. It contains different multiple tags (elements) that are used to style content on a web page. Examples include <h1>, <p>, <ul>, <a> e.t.c

Example Code:

```
<!DOCTYPE html>
<html>
<head>
    <title>My First Web Page</title>
</head>
<body>
    <h1>Welcome to My Web Page</h1>
    <p>This is a paragraph of text.</p>
</body>
</html>
```

HTML Tags, Elements and Attributes:

A **tag** is a markup that is used to define the building block of an element. It marks the beginning and end of an element. Examples of tags include: <h1>, </h1>, <img>, <p>, e.t.c

An **element** is a combination of both the opening and closing tag along with the content. Examples include: <h1>Welcome</h1>, <img src=”image.jpg” alt=”description”>, <a href=”url”>

An **attribute** is additional information inside the opening tag. Examples include href in <a href=”url”>, src and alt in <img src=”image.png” alt=”description”>

Categories of HTML Elements:

Block level elements:

These are elements that take up the full width available (by default) and always start on a new line. They are called block elements because they block any elements that come before or after them causing them to be pushed down. Block elements can contain inline elements and other block elements.

Example tags used for block elements:

- *** *** – Generic container for grouping content.
- ***<p>*** – Represents a paragraph of text.
- ***<h1>, <h2>, <h3>, ... , <h6>*** – Header tags used for headings.
- ***<ul>, *** – Lists (unordered or ordered).
- ***<li>*** – List item (used within `<ul>` or `<ol>`).
- ***<article>*** – Represents a self-contained piece of content.
- ***<section>*** – Used to define sections of a page.
- ***<footer>, <header>, <nav>*** – For structural elements in a page layout.

Inline elements:

These are elements that don’t start on a new line and only take up as much width as necessary. They flow with the content and can thus exist alongside other inline elements on the same line. They can only contain other inline elements.

Examples of tags used for inline elements:

- ****** – Generic inline container, often used for styling parts of text.
- ***<a>*** – Defines a hyperlink.
- ****** – Defines important text (usually rendered as bold).

- ****** – Defines emphasized text (usually rendered as italic).
- ***<img>*** – Embeds an image within the content.
- ***
*** – Represents a line break.
- ***<code>*** – Represents computer code.
- ***<label>*** – Defines a label for an input element in forms.

Common HTML Tags:

1. **Working with Text:**
    - **Headings (`<h1>...<h6>`)**:

      Headings are used to define the structure and hierarchy of text on a webpage. They are represented by h and a digit defining the size and attribute of the text element ranging from 1 to 6 with 1 being the most important and 6 as the least important.

      Examples are: **`<h1>This is a heading</h1>`**

    - **Paragraphs (`<p>`)**:

      This defines a block of text on a web page with similar or related content. Different paragraph tags result in different separated blocks on the web page. It is represented using the `<p> tag.`

      Example: **`<p>This is my paragraph</p>`**

    - **Line Break (`<br>`):**

      This is used to add a line break. It is an **empty tag,** which means it has no closing tag.

      Example: **`<p>This is <br> now on a new line </p>`**

    - **Preformatted Text (`<pre>`):**

This defines preformatted text. Text in the **pre** element is displayed in a fixed-width font, and the text preserves both the spaces and line breaks. The text appears exactly how it is written in the html code.

Example: **<pre>This text will be preserved as </pre>**

- **Horizontal Rule (<hr>):**

This is used to add a horizontal line that emphasizes a change in topic. It is an **empty tag,** which means it has no closing tag.

Example:
```
<p>This is the content </p>
<hr>
<p>This is the other content</p>
```

2. **Working with Characters:**

- **Bold Face ():**

This is used to make text **bold** without marking it as important.

Example: ** This is the content **

- **Italic ():**

This is used to italicise text (text that is set off from the normal prose in a document)

Example: ** This is the content **

- **Underline (<u>):**

This is used to underline text.

Example: ** <u>This is the content </u> **

- **Small (<small>):**

This is used to make text smaller. It is basically used for copyrights or side comments.

Example: `<p><small>This is smaller text</small></p>`

- **Superscript (`<sup>`) and Subscript (`<sub>`):**

  These are used to superscript and subscript a piece of text.

  Example:
  ```
  <p>This is <sup>top</sup> text</p>
  <p>This is <sub>below</sub> text</p>
  ```

- **Deleted (`<del>`) and Inserted Text (`<ins>`):**

  The **del tag** is used to add a strikethrough line in a text. For any other text to be added after the **del** tag, we use the **ins tag** to note the inserted text.

  Example:
  ```
  <p>This is
  <del>deleted</del><ins>text</ins></p>
  ```

3. **Working with Lists:**

A list is a collection of elements that are usually related. Lists can be ordered or unordered.

1. **Ordered Lists (`<ol>`):**

This is a list where the order and arrangement of elements in the list is of priority and importance. By default, the browser will automatically number the elements in the list. Elements in the list are represented using list items tag **(`<li>`).**

Example:

```
1.  <ol>
        <li>First step</li>
        <li>Second step</li>
        <li>Third step</li>
```

```
    </ol>
```

```
Output:
  1. First step
  2. Second step
  3. Third step
```

The style of numbering can be changed using the `type` attribute on the ordered list. Ordered lists can be numbered using any of the following:

- **`type="1"`**: Numeric (default, 1, 2, 3…)
- **`type="A"`**: Uppercase letters (A, B, C…)
- **`type="a"`**: Lowercase letters (a, b, c…)
- **`type="I"`**: Uppercase Roman numerals (I, II, III…)
- **`type="i"`**: Lowercase Roman numerals (i, ii, iii…)

Example:

```
2.  <ol type=”A”>
        <li>First step</li>
        <li>Second step</li>
        <li>Third step</li>
    </ol>
```

```
Output:
      A. First step
      B. Second step
      C. Third step
```

2. Unordered Lists (`<ul>`):

This is a type of list where the order and arrangement of elements in the list isn't of very much significance. In the browser, elements are represented by bullet points by default. Elements in the list are represented using list items tag **(`<li>`).**

Example:

```
1.  <ul>
        <li>First step</li>
```

```
            <li>Second step</li>
            <li>Third step</li>
        </ul>
```

Output:

- **First step**
- **Second step**
- **Third step**

The bullet style can be changed using the **`type`** attribute. The customization of the bullet style can be of the following types:

- **`circle`:** Empty circle bullet
- **`disc`:** Filled circle bullet (default)
- **`square`:** Square bullet

Example:

```
2.  <ul type=”square">
        <li>Milk</li>
        <li>Bread</li>
        <li>Eggs</li>
    </ul>
```

Output:

- **Milk**
- **Bread**
- **Eggs**

4. Working with Images:

- **Images**: **`(<img>)`**

  This is used to add an image to a web page. It has the following attributes:

  `src` - used to define the source of the image (local or remote)

  `alt` - used to provide a text (alternative text) description to the image.

`width and height -` They define the dimensions of the image.

`title -` This provides text that acts as a tooltip when a user hovers over the image.

`align -` This allows alignment of the image within a page

`vspace -` This adds vertical space i.e on the top and bottom of the image.

`hspace -` This adds horizontal space i.e on the left and right of the image.

Example: **`<img src="images/photo.jpg" alt="A scenic view of mountains" width="600" height="400" />`**

5. Working with Links:

- **Hyperlinks/Links `(<a>)`** :

  A hyperlink is a clickable element that allows a user to navigate between different web pages or websites. It is represented using the anchor `<a> tag` with the `href attribute.`

  The **`target`** attribute specifies where the web page is to be displayed. E.g

  - `_self` - Default. Opens the document in the same window/tab as it was clicked
  - `_blank` - Opens the document in a new window or tab
  - `_parent` - Opens the document in the parent frame
  - `_top` - Opens the document in the full body of the window

  Example: **`<a href="https://my.com">`** redirects to my.com website/web page

  The color of a link is determined by some attributes available in the body tag. These attributes are:

alink - Sets color of an active link.

vlink - Sets color of a visited link.

6. Working with tabular Data:

- **Tables: (<table>):**

  This is used to display data in rows and columns. The table is created using the following elements:

  - **<table>**: Defines the table container.
  - **<tr>** (Table Row): Represents a row in the table.
  - **<th>** (Table Header): Defines a table header cell. By default, text in a <th> is bold and centered.
  - **<td>** (Table Data): Defines a standard table cell.
  - **<caption>:** Defines a title for the table
  - **<rowspan>:** Allows a cell to span across multiple rows
  - **<colspan>:** Allows a cell to span across multiple columns
  - **<thead>, <tbody>, and <tfoot>:** Used to group the table into header, body and footer.

  Example:

```
<table>
     <tr>
          <th>Header 1</th>
          <th>Header 2</th>
          <th>Header 3</th>
     </tr>
     <tr>
          <td>Data 1</td>
```

```
            <td>Data 2</td>
            <td>Data 3</td>
        </tr>
        <tr>
            <td>Data 4</td>
            <td>Data 5</td>
            <td>Data 6</td>
        </tr>
    </table>
```

The output:

Header 1	Header 2	Header 3
Data 1	Data 2	Data 3
Data 4	Data 5	Data 6

Tables can be created using different attributes which include:

- **`border`**: defines whether the table has a border.
- **`cellspacing`**: defines the space between the cells.
- **`cellpadding`**: defines the space between the content of a cell and its border.

7. Working with Groups of information:

- **Divs and Spans**: **`<div>` (block-level container), `<span>` (inline container)**

These provide containers where other elements can be grouped. The div tag is used to group large elements while the span tag is used to group smaller inline elements.

Example:

1. `<div class="container"> <h1>Welcome to My Website</h1> <p>This is a section of the page.</p> <p>More content goes here...</p> </div>`

2. `<p>The <span style="color: red;">quick</span> brown fox jumps over the lazy dog.</p>`

8. Working with User Input:

- **Forms**:

  A form is a collection of different elements that enables a user to enter information to a web page. The different elements are all added under the `<form>` tag and they include:

  - **`<form>`**: The main container for form elements.
  - Input elements:
    - **`<input>`** (types: text, password, checkbox, radio, date, etc.)
    - **`<textarea>`**
    - **`<select>`**, **`<option>`**
  - Buttons: **`<button>`**, **`<input type="submit">`**

  Example:

```
<form action="/submit" method="POST">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
        <label for="message">Message:</label>
        <textarea id="message" name="message" required></textarea>
        <button type="submit">Submit</button>
    </form>
```

Attributes for Forms:

- **`action`** (where form data is sent)
- **`method`** (http method used when submitting data, GET, POST e.t.c)
- **`name`** (used to identify form data)

Trial Questions:

1. Write HTML codes to come up with the outlook below:

   i. $a^2 + b^2 = c^2$

   ii. $\log_b a = \log_c a / \log_c b$

   iii. This text contains $^{\text{sensitive}}$ information which may be $_{\text{deciphered}}$ by ***NASA***.

2. Write simple html codes to produce the results below:

i. My First List

1. **Fruits**
   - **i. Berries:**
     - **Apples**
     - **Tomatoes**
   - **ii. Drupes**
     - **Mango**
     - **Avocado**
2. **Vegetables**
   - **a. Leafy**
     - **Kale**
     - **Spinach**
   - **b. Non-leafy**
     - **Cabbage**
3. **Meats**
   - **White meats**
     - **A. Chicken**
     - **B. Tuna**
   - **Red Meats**
     - **I. Beef**
     - **II. Mutton**

---

ii. My Second List

A. Fruits

1. **Berries:**
   - **a. Apples**
   - **b. Tomatoes**
2. **Drupes**
   - **i. Mango**
   - **ii. Avocado**

B. Vegetables

- **Leafy**
  - **i. Kale**
  - **ii. Spinach**
- **Non-leafy**
  - **a. Cabbage**

C. Meats

- **White meats**

iii. My First Table

Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
SCS101	SCS104	SCS105		
		SCS104		SCS105
	SCS101			
SCS101			SCS102	SCS102

iv. My Second Table

Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
SCS101	SCS104	SCS105		
		SCS104		SCS105
	SCS101			
SCS101			SCS102	SCS102

v. My Third Table

Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
SCS101	SCS104	SCS105	**i. Eat**	
		SCS104	**ii. Sleep**	SCS105
	SCS101			
SCS101			SCS102	SCS102

vi. My Forth Table

Monday	**Tuesday**		**Wednesday**		**Thursday**	**Friday**
SCS101	SCS104	**BREAK**	SCS105	**LUNCH**	**SCS102**	
SCS111			SCS104			SCS105