HTML4 & HTML5

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

Basic structure of HTML document

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Output:

My First Heading

My first paragraph.

Tags: headings, paragraphs, links, images, tables, forms

1. Heading

```
<!DOCTYPE html>
<html>
<head>
```

```
<title>Page Title</title>
</head>
<body>
<h1>This is Heading 1</h1>
<h2>This is Heading 2</h2>
<h3>This is Heading 3</h3>
<h4>This is Heading 4</h4>
<h5>This is Heading 5</h5>
<h6>This is Heading 6</h6>

</body>
</html>
```

Explanation:
Headings are used to define titles and subtitles.

- `<h1>` is the largest and most important.
- `<h6>` is the smallest.

2. Paragraphs

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<p>This is a paragraph of text.</p>
</body>
</html>
```

Explanation:
The `<p>` tag defines a paragraph.
It's used to write normal sentences or information on the webpage.

3. Links

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
```

```
</head>
<body>
<a href="https://www.example.com">Visit Example</a>
</body>
</html>
```

Explanation:

- The `<a>` tag creates a hyperlink.
- `href` is the attribute that holds the link URL.
  When users click the text "Visit Google", it opens the Google website.

4. Images

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<img src="flower.jpg" alt="A beautiful flower" width="300" height="200">
</body>
</html>
```

Explanation:

- `<img>` is used to show an image.
- `src` = source of the image file
- `alt` = alternative text (shows if image fails to load)
- `width` and `height` = size of the image in pixels

5. Table

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<table border="1">
  <tr>
    <th>Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Vaishnavi</td>
    <td>24</td>
  </tr>
  <tr>
    <td>John</td>
    <td>25</td>
  </tr>
</table>
</body>
</html>
```

Explanation:

- `<table>` starts the table
- `<tr>` is a table row
- `<th>` is a table heading (bold and centered)
- `<td>` is a table data cell
  The table displays rows and columns.

6. Form

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<form action="submit.php" method="post">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name"><br><br>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email"><br><br>

  <input type="submit" value="Submit">
</form>
</body>
</html>
```

Explanation:

- `<form>` creates a form where users can enter and submit data

- `action="submit.php"` is the file that receives the data
- `method="post"` sends the data securely
- `<input type="text">` for name
- `<input type="email">` for email
- `<input type="submit">` to send the form

Semantic elements in HTML5

What are Semantic Elements?

Semantic elements clearly describe their meaning in both the browser and to developers.

For example, instead of using a `<div>` for everything, we can use tags like `<header>`, `<nav>`, `<article>`, etc., to clearly define the purpose of that section.

Semantic Elements in HTML5

Tag	**Meaning**
<header>	Defines the top/header section of a page or section
<nav>	Contains navigation links (menu, navbar)
<main>	Main content of the document (unique content)
<section>	Defines a section in the document
<article>	Independent, self-contained content (e.g. blog post, news)
<aside>	Sidebar or extra content (ads, related links)
<footer>	Bottom/footer of a page or section
<figure>	Used to group media (image, chart, etc.)
<figcaption>	Caption/description for the <figure>
<mark>	Highlights text (like a highlighter)
<time>	Represents a specific time/date
<summary>	Summary for <details> tag (expandable content)

<details>	Used for toggleable content (like FAQ answers)

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Semantic Elements Example</title>
</head>
<body>
  <header>
    <h1>My Website</h1>
    <nav>
      <a href="#">Home</a> | <a href="#">About</a> | <a href="#">Contact</a>
    </nav>
  </header>
  <main>
    <section>
      <h2>About Me</h2>
      <p>This is a section about me.</p>
    </section>
    <article>
      <h2>Blog Post</h2>
      <p>This is an independent article or blog post.</p>
    </article>
    <aside>
      <h3>Related Links</h3>
      <ul>
        <li><a href="#">Portfolio</a></li>
        <li><a href="#">Projects</a></li>
      </ul>
    </aside>
  </main>
  <footer>
    <p>&copy; 2025 Vaishnavi Kuchan</p>
  </footer>
</body>
</html>
```

Multimedia support: audio, video

Audio Tag

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<audio controls>
  <source src="music.mp3" type="audio/mpeg">
  <source src="music.ogg" type="audio/ogg">
  Your browser does not support the audio element.
</audio>
</body>
</html>
```

Explanation:

- `<audio>`: Used to embed audio files.
- `controls`: Displays audio player (play, pause, volume).
- `<source>`: Allows different file formats for browser compatibility.
- Fallback text: Shows if the browser doesn't support audio.

Video Tag

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<video width="640" height="360" controls>
  <source src="video.mp4" type="video/mp4">
  <source src="video.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
</body>
</html>
```

Explanation:

- `<video>`: Used to embed video content.
- `controls`: Shows default video controls (play, pause, volume, full screen).
- `width` and `height`: Set video size.
- `<source>`: Allows different video formats.
- Fallback text: Shown if the browser doesn't support the video tag.

HTML Integration + Basic CSS & Bootstrap

1. CSS Selectors

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
  <style>
    /* Element Selector */
    h1 {
      color: blue;
    }

    /* Class Selector */
    .highlight {
      background-color: yellow;
    }
    /* ID Selector */
    #main-text {
      font-size: 20px;
    }
  </style>
</head>
<body>
 <h1>Hello Vaishnavi!</h1>
  <p id="main-text">This is the main paragraph.</p>
  <p class="highlight">This is highlighted text.</p>
</body>
</html>
```

Explanation:

- `h1` → targets all `<h1>` elements
- `.highlight` → targets class `highlight`
- `#main-text` → targets specific element with id `main-text`

2. CSS Box Model

Every HTML element is a box made of:

1. Content
2. Padding
3. Border
4. Margin

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .box {
      width: 300px;
      padding: 20px;
      border: 5px solid black;
      margin: 30px;
      background-color: lightblue;
    }
  </style>
</head>
<body>
  <div class="box">This is a box model example.</div>
</body>
</html>
```

Explanation:

- `padding`: space **inside** the box (between content & border)
- `border`: the outline of the box
- `margin`: space **outside** the box (between boxes)

3. CSS Layout: Flexbox

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .flex-container {
      display: flex;
      gap: 20px;
      background-color: #f0f0f0;
      padding: 20px;
    }

    .flex-box {
      background-color: lightgreen;
      padding: 20px;
      width: 100px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-box">Box 1</div>
    <div class="flex-box">Box 2</div>
    <div class="flex-box">Box 3</div>
  </div>
</body>
</html>
```

Explanation:

- `display: flex;` creates a **flex container**
- Children become flexible boxes in a **row by default**
- `gap`: adds space between boxes

4. CSS Layout: Grid

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .grid-container {
      display: grid;
      grid-template-columns: 1fr 1fr;
      gap: 20px;
      background-color: #e3e3e3;
      padding: 20px;
    }
    .grid-item {
      background-color: #ffa07a;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="grid-container">
    <div class="grid-item">Item 1</div>
    <div class="grid-item">Item 2</div>
    <div class="grid-item">Item 3</div>
    <div class="grid-item">Item 4</div>
  </div>
</body>
</html>
```

Explanation:

- `display: grid;` enables CSS Grid Layout
- `grid-template-columns: 1fr 1fr;` = two equal columns
- `gap`: spacing between items

Bootstrap: grid system, components, responsive design

1. Bootstrap Grid System

Bootstrap uses a **12-column layout system** to build responsive layouts.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Grid</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
  <div class="container mt-4">
    <div class="row">
      <div class="col-md-6 bg-primary text-white p-3">
        Column 1 (6 of 12 columns)
      </div>
      <div class="col-md-6 bg-success text-white p-3">
        Column 2 (6 of 12 columns)
      </div>
    </div>
  </div>
</body>
</html>
```

Explanation:

- `container`: centers and adds padding
- `row`: creates a horizontal row
- `col-md-6`: each takes 6 columns (50%) on medium+ screens

2. Bootstrap Components

a) Button

```
<button class="btn btn-primary">Click Me</button>
```

b) Alert

```
<div class="alert alert-warning" role="alert">
  This is a warning alert!
</div>
```

c) Card

```
<div class="card" style="width: 18rem;">
  <img src="https://via.placeholder.com/150" class="card-img-top" alt="...">
  <div class="card-body">
    <h5 class="card-title">Card Title</h5>
    <p class="card-text">This is a card with image and text.</p>
    <a href="#" class="btn btn-primary">Read More</a>
  </div>
</div>
```

d) Navbar

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">MySite</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ms-auto">
        <li class="nav-item"><a class="nav-link active" href="#">Home</a></li>
        <li class="nav-item"><a class="nav-link" href="#">About</a></li>
        <li class="nav-item"><a class="nav-link" href="#">Contact</a></li>
      </ul>
    </div>
  </div>
</nav>
```

3. Responsive Design with Bootstrap

Bootstrap uses breakpoints like:

- `col-sm-*` (small: ≥576px)
- `col-md-*` (medium: ≥768px)
- `col-lg-*` (large: ≥992px)
- `col-xl-*` (extra large: ≥1200px)

Responsive Columns Example

```
<div class="container mt-4">
  <div class="row">
    <div class="col-12 col-md-4 bg-info p-3">1 of 3</div>
    <div class="col-12 col-md-4 bg-warning p-3">2 of 3</div>
    <div class="col-12 col-md-4 bg-danger p-3">3 of 3</div>
  </div>

</div>
```

Explanation:

- On mobile: Each `col-12` takes full width (stacked)
- On desktop: Each `col-md-4` takes 1/3 width (side by side)

Integrating HTML + CSS + Bootstrap in a website

Project: Responsive personal portfolio or landing page