Step 1: Basic HTML Questions

Step 1: Basic HTML Questions

1. What is HTML? Explain its purpose & Basic Syntax Of HTML ?

Ans: HTML meaning (HyperText Markup Language). It is the standard language we can use to create and structure or build content on the web. HTML provides a way to format text, image, audio, video or multimedia things. That can help show a viewable web page in the browser.

HTML main building structure.

```
<!DOCTYPE html>
<html>
<head>
    <title>My First Webpage</title>
</head>
<body>
    <h1>Hello, World!</h1>
    <p>This is my first webpage created using HTML.</p>
</body>
</html>
```

2. What are HTML tags? How are they structured? Give Us 5 Tags As an Example.

Ans: HTML tags are the building blocks of HTML(HyperText Markup Language) used to structure on the web pages. It is told how to display text, image, audio, and video or other information that needs to be shown or used by a website.

Structure of HTML Tags

- Opening tag: start with a less-than (<) and end with by getter-than (>)
- Closing tag: start with less-than and / ex: (</) and end with (>)
- Content: Place any content text, image and other in the Opening and Closing tags

Example: <p>This is a paragraph.</p>

Common 5 Tags Example:

```
<html> ... </html>
<head> ... </head>
<title>Website title</title>
<body> ... </body>
```

```
<p>This is a sample paragraph.</p>
```

3. **What is the difference between `<head>`, `<body>`, and `<html>` tags? & the role of the `<!DOCTYPE html>` declaration?**

Ans:

1. `<html>` Tag

- Purpose: Defines the root element of the HTML document.
- Structure: It wraps around the entire HTML code, including both `<head>` and `<body>`.
- Placement: The `<html>` tag should be the first tag after the `<!DOCTYPE html>` declaration.

Example:

```
<html>
  <!-- Content goes here -->
</html>
```

2. `<head>` Tag

- Purpose: Contains metadata and information about the document that doesn't appear directly on the webpage.
- Content: Includes elements like `<title>`, `<meta>`, `<link>`, and `<script>`.
  - `<title>`: Sets the title shown on the browser tab.
  - `<meta>`: Provides information like the character set or viewport settings.
  - `<link>`: Links external resources, such as CSS stylesheets.
  - `<script>`: Adds or links JavaScript files.
- Placement: Always placed within the `<html>` tag, above the `<body>` tag.

Example:

```
<head>
  <title>Food village || website</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="style.css">
</head>
```

3. `<body>` Tag

- Purpose: Holds all the content that appears on the web page itself.
- Content: Contains elements like text, images, links, forms, and other visible elements.
- Placement: Directly below the `<head>` tag and within the `<html>` tag.

Example:

```
<body>
  <h1>Welcome to My Webpage</h1>
  <p>This is a paragraph of text on the web page.</p>
</body>
```

Role of the `<!DOCTYPE html>` Declaration

- Purpose: Tells the browser the type and version of HTML being used (in this case, HTML5).
- Effect: Ensures the web page renders in standards-compliant mode rather than quirks mode, which maintains compatibility with older versions of HTML.
- Placement: Always the very first line of an HTML document, above the `<html>` tag.

Example:

```
<!DOCTYPE html>
<html>
  <head>
        <title>Example Document</title>
  </head>
  <body>
        <p>Hello, world!</p>
  </body>
</html>
```

4. **Describe the function of the `<title>` tag. Where does it appear with Example.**

Ans:

Function of the `<title>` Tag

1. Browser Tab: The content of the `<title>` tag appears as the title on the browser tab.
2. Bookmarks/Favorites: When a user bookmarks a page, the title saved is from the `<title>` tag, helping users recognize the page later.

3. SEO (Search Engine Optimization): Search engines use the `<title>` tag to understand the page's main topic, making it a critical element for search ranking.

Placement

The `<title>` tag is placed within the `<head>` section of an HTML document.

Example:

```
<!DOCTYPE html>
<html>
  <head>
       <title>Learning HTML Basics</title>
  </head>
  <body>
    <h1>Welcome to My HTML Tutorial</h1>
    <p>This page teaches the basics of HTML tags.</p>
  </body>
</html>
```

5. **How do you add comments in HTML, and why are they useful?**

Ans:

Syntax for HTML Comments

```
<!-- This is a comment in HTML -->
```

How to Add a Comment in HTML
Place the comment between <!-- and -->. Anything written inside this syntax will be ignored by the browser.

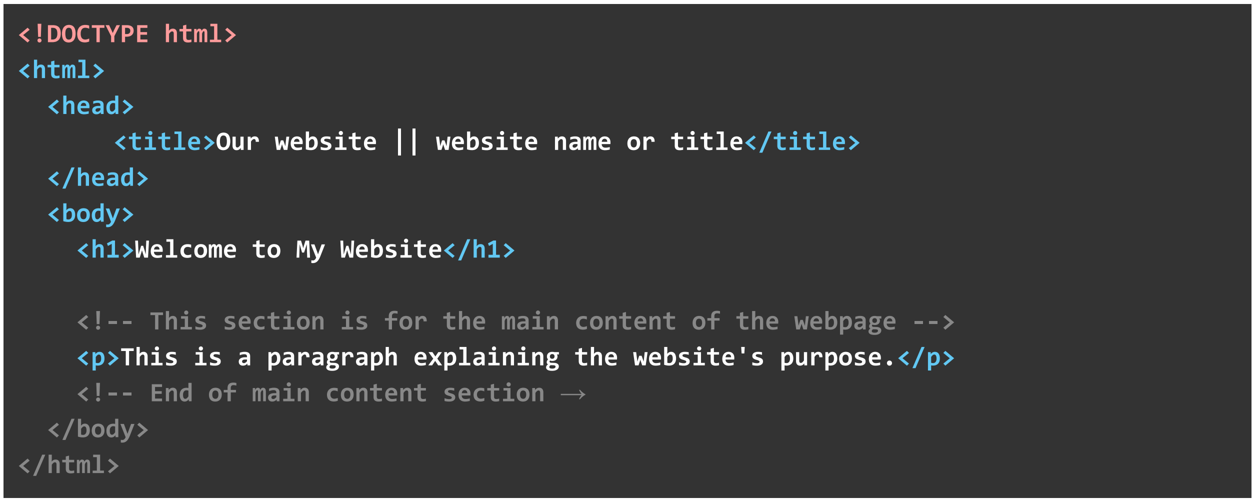
```
<p>This is visible text on the page.</p>
<!-- This text is a comment and will not be displayed on the page -->
```

Why Comments Are Useful

1. Improves Code Readability: Comments make the code easier to understand for you and others who may work on it. They act as notes or explanations for various sections.
2. Provides Context: Adding comments helps explain why certain parts of the code were written, especially if the logic is complex.

3. Facilitates Troubleshooting: Comments can be used to temporarily disable parts of the code for testing and debugging without deleting them.
4. Collaboration: When working on a team, comments help other developers understand the purpose and structure of the code.

Example of Comments in Context:



```
<!DOCTYPE html>
<html>
  <head>
      <title>Our website || website name or title</title>
  </head>
  <body>
    <h1>Welcome to My Website</h1>

    <!-- This section is for the main content of the webpage -->
    <p>This is a paragraph explaining the website's purpose.</p>
    <!-- End of main content section →
  </body>
</html>
```

Step 2: HTML Elements and Attributes

6. **What is the difference between an HTML element and an HTML attribute give us example with proper Explanation.**

Ans:

1. HTML Element

An HTML element is the basic unit of HTML, consisting of a start tag, content, and sometimes an end tag. Elements define the structure and content of a webpage, like headings, paragraphs, images, links, etc.

Structure of an HTML Element:

```
<tagname>Content</tagname>
```

Start Tag: Indicates the beginning of an element (e.g., `<p>`).

Content: The text or other items inside the tags.

End Tag: Closes the element (e.g., `</p>`), though some elements like `<img>` are self-closing.

Example of an HTML Element:

```
<p>This is a paragraph.</p>
```

In this example:

- `<p>` is the element representing a paragraph.
- `This is a paragraph.` It is the content of the element.
- `</p>` is the closing tag.

2. HTML Attribute

An **attribute** is a property that provides additional information about an HTML element, such as style, behavior, or other settings. Attributes are added within the opening tag of an element and consist of a name and value.

Structure of an HTML Attribute:

```
<tagname attribute="value">Content</tagname>
```

Attribute Name: Defines the specific property or feature (e.g., `class`, `id`, `href`, `src`).

Attribute Value: Provides the value for that property in quotes (e.g., `href="https://example.com"`).

Example of an HTML Attribute:

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

Key Differences

1. Function:
   - Element: Defines the type of content and structure.
   - Attribute: Adds extra information or functionality to the element.
2. Location:
   - Element: Composed of start and end tags surrounding content.
   - Attribute: Placed inside the opening tag of an element.

Combined Example

```
<img src="image.jpg" alt="A sample image" width="200">
```

Element: `<img>` is an image element.
Attributes: `src`, `alt`, and `width` are attributes for the `<img>` element:

- `src="image.jpg"` specifies the image source.
- `alt="A sample image"` provides alternative text.
- `width="200"` sets the width of the image in pixels.

7. **Explain the global attributes in HTML, such as `class`, `id`, `style`, and `title` With Example.**

Ans: Global attributes in HTML are attributes that can be applied to almost any HTML element, giving added functionality and customization to elements on a webpage. Here are explanations and examples of four key global attributes: `class`, `id`, `style`, **and** `title`.

1. `class` Attribute

- **Purpose**: Used to apply a common name or group to multiple elements. This attribute is helpful when styling multiple elements with CSS or selecting them in JavaScript.

Example:

```
<p class="intro">This is the introduction paragraph.</p>
```

```
<p class="intro">This is another introduction paragraph.</p>
<style>
  .intro {
    font-weight: bold;
    color: blue;
  }
</style>
```

2. `id` Attribute

- **Purpose:** Provides a unique identifier for a specific element on a webpage. Unlike `class`, the `id` value should be unique within the page, which is helpful for applying CSS styles, anchoring links, and JavaScript manipulation.

Example:

```
<h1 id="main-heading">Welcome to My Website</h1>
<style>
  #main-heading {
    color: green;
  }
</style>
```

3. `style` Attribute

- **Purpose:** Allows you to apply inline CSS styles directly to an element. While it's often best to separate CSS from HTML, the `style` attribute can be useful for small, unique styling.

Example:

```
<p style="color: red; font-size: 20px;">This text is styled directly with
the style attribute.</p>
```

4. `title` Attribute

- **Purpose:** Adds a tooltip that displays when a user hovers over the element. It provides extra information, often used for accessibility or additional context.

Example:

<button title="Click to submit the form">Submit</button>

Summary of Usage

- `class`: Groups elements for CSS styling or JavaScript selection.
- `id`: Uniquely identifies an element for specific styling or linking.
- `style`: Directly applies inline CSS styling to an element.
- `title`: Provides additional information that appears on hover as a tooltip

8. **What is the purpose of the `alt` attribute in an `<img>` tag?**

Ans:
The `alt` (alternative text) attribute in an `<img>` tag provides a textual description of the image's content. It serves several important purposes:

1. Accessibility

- Screen Readers: The `alt` text is read aloud by screen readers for visually impaired users, allowing them to understand the purpose of the image.

2. Image Loading Issues

- Fallback Text: If the image fails to load (due to a broken link or slow connection), the `alt` text appears as a placeholder, informing the user what the image represents.

3. SEO (Search Engine Optimization)

- Search Engines: Search engines use `alt` text to better understand the content of an image, which can improve the page's visibility in image search results.

Example

```
<img src="sunset.jpg" alt="A beautiful sunset over the mountains">
```

9. **Describe the `href` attribute and its role in the `<a>` tag?**

Ans:

The `href` (hypertext reference) attribute is a fundamental part of the `<a>` (anchor) tag in HTML, which is used to create hyperlinks. It specifies the destination URL or the location the link points to.

Role of the `href` Attribute in the `<a>` Tag

1. **Link Destination:** The primary purpose of the `href` attribute is to define where the link will take the user when they click on it. This can be a URL, an email address, or a location within the same document.
2. **Types of Links:**
   - External Links: These links direct users to a different website.

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

   - Internal Links: These links point to another page or section within the same website.

```
<a href="about.html">Learn More About Us</a>
```

   - Anchor Links: These links direct users to a specific part of the same page using an ID reference.

```
      <a href="#section1">Go to Section 1</a>
```

3. **Email Links:** The `href` attribute can also be used to create links that open the user's email client with a specified recipient.

```
<a href="mailto:info@example.com">Email Us</a>
```

Example

Here’s a complete example of the `<a>` tag using the `href` attribute:

```
<!DOCTYPE html>
<html>
<head>
      <title>Sample Links</title>
</head>
<body>
    <h1>Welcome to My Website</h1>
    <p>
        Visit our <a href="https://www.example.com">Example Website</a> for more information.
    </p>
    <p>
        For inquiries, <a href="mailto:info@example.com">Email Us</a>.
    </p>
    <p>
        Jump to <a href="#contact">Contact Section</a>.
    </p>

    <h2 id="contact">Contact Section</h2>
    <p>Here you can find our contact details.</p>
</body>
</html>
```

10. **How does the `target` attribute work within an `<a>` tag? What is `_blank`?**

Ans:

The `target` attribute in an `<a>` (anchor) tag specifies where to open the linked document or resource when the link is clicked. This attribute helps control how users navigate between web pages, offering flexibility in user experience.

How the `target` Attribute Works

1. **Default Behavior**: By default, when you click a link, the linked page opens in the same browsing context (i.e., the current window or tab).
2. **Using the `target` Attribute**: You can change this behavior by using the `target` attribute with specific values.

Common Values for the `target` Attribute

- `_self`: (Default) Opens the link in the same frame as it was clicked (same tab or window).

  <a href="https://www.example.com" target="_self">Open in Same Tab</a>

- `_blank`: Opens the link in a new tab or window, depending on the browser settings. This is commonly used for external links to ensure users remain on the original page.

  <a href="https://www.example.com" target="_blank">Open in New Tab</a>

- `_parent`: Opens the link in the parent frame of the current frame. If there is no parent frame, it behaves like `_self`.

  <a href="https://www.example.com" target="_parent">Open in Parent Frame</a>

- `_top`: Opens the link in the full body of the window, effectively breaking out of any frameset. This is useful for links within a framed layout.

  `<a href="https://www.example.com" target="_top">Open in Full Window</a>`

Example of Using the target Attribute

Here's a simple example demonstrating the use of the `target` attribute:

```html
<!DOCTYPE html>
<html>
<head>
    <title>Target Attribute Example</title>
</head>
<body>
    <h1>Links with Target Attribute</h1>
    <p>
        <a href="https://www.example.com" target="_self">Open in Same Tab</a>
    </p>
    <p>
        <a href="https://www.example.com" target="_blank">Open in New Tab</a>
    </p>
    <p>
        <a href="https://www.example.com" target="_parent">Open in Parent Frame</a>
    </p>
    <p>
        <a href="https://www.example.com" target="_top">Open in Full Window</a>
    </p>
</body>
</html>
```

Step 3: Common HTML Tags

11. **How to add favicon in HTML & Why ?**

Ans:

To add a favicon in HTML, you can follow these steps:

1. **Create or Choose a Favicon**: Make sure your favicon image is in `.ico`, `.png`, or another supported format. The standard size is 16x16 pixels or 32x32 pixels.
2. **Upload the Favicon**: Place the favicon file in the root directory of your website or in a designated folder.
3. **Add the `<link>` Tag in the `<head>` Section**: Use the `<link>` element in the `<head>` section of your HTML document to link to the favicon. Here’s how you can do it:

```
<head>
    <link rel="icon" href="favicon.ico" type="image/x-icon">
    <!-- or for PNG format -->
    <link rel="icon" href="favicon.png" type="image/png">
</head>
```

Why Add a Favicon?

- **Branding:** A favicon enhances your website's branding, making it easily recognizable in browser tabs, bookmarks, and history lists.
- **User Experience:** It helps users quickly identify your site among multiple open tabs, improving navigation.
- **Professional Appearance:** Having a favicon gives your website a more polished and professional look, enhancing credibility.

12. **What is formatting in HTML?**

Ans:

Formatting in HTML refers to the use of various HTML tags and attributes to control the presentation and layout of text and elements on a webpage. It involves applying styles, organizing content, and enhancing readability to create visually appealing and well-structured documents. Common formatting elements include headings, paragraphs, lists, bold or italic text, and block quotes, which help define the hierarchy and importance of the content. Additionally, CSS (Cascading Style Sheets) can be used alongside HTML to provide more advanced styling options, allowing for greater control over colors, fonts, spacing, and overall design.

13. **What Is Inline and block elements in html5 Give Us The Proper Explain With Example?**

Ans:

In HTML5, elements are categorized into two main types: **inline elements** and **block elements**. Understanding the difference between these two types is essential for effective web layout and design.

Inline Elements

Definition: Inline elements only take up as much width as necessary and do not start on a new line. They allow other elements to sit beside them on the same line. Inline elements are typically used for smaller pieces of content within a block-level element.

Examples:

- `<span>`: A generic container for text.
- `<a>`: Defines a hyperlink.
- `<strong>`: Indicates strong importance (usually bold).
- `<em>`: Indicates emphasized text (usually italicized).

Usage Example:

```
<p>This is a <strong>bold</strong> statement  within a paragraph.</p>
```

In this example, the `<strong>` element is inline, so it does not break the flow of the paragraph.

Block Elements

Definition: Block elements take up the full width available, effectively starting on a new line. They create a "block" of content and are typically used for larger pieces of content or structural elements on a webpage.

Examples:

- `<div>`: A generic container for grouping elements.
- `<h1>`, `<h2>`, `<h3>`, etc.: Headings of different levels.
- `<p>`: Represents a paragraph of text.
- `<ul>`, `<ol>`, `<li>`: Used for lists.

Usage Example:

```
<div>
    <h1>This is a heading</h1>
    <p>This is a paragraph of text.</p>
</div>
```

In this example, both `<h1>` and `<p>` are block elements, meaning they each start on a new line and occupy the full width of their container.

Summary

- **Inline Elements**: Do not start on a new line, take only the necessary width, and are typically used for small pieces of content.
- **Block Elements**: Start on a new line, take up the full width of their container, and are used for larger sections of content. Understanding these differences helps in structuring HTML documents and applying CSS effectively.

14. **What is a marquee?**

Ans:

A marquee is an HTML element used to create scrolling text or images that move horizontally or vertically across a webpage. It was commonly used in early web design for dynamic, eye-catching effects but is now outdated and discouraged in modern web development due to accessibility issues and limited browser support.

15. **Explain the difference between `<p>`, `<span>`, and `<div>` tags. When would you use each?**

Ans:

The `<p>`, `<span>`, and `<div>` tags are all used to organize and structure content in HTML, but they each serve different purposes:

1. **`<p>` Tag:**
   - Represents a paragraph of text and adds some default spacing above and below the text.
   - Use it when you want to define blocks of text, like paragraphs in articles or sections.

```
<p>This is a paragraph of text.</p>
```

2. **`<span>` Tag:**
   - An inline tag, meaning it doesn't break the flow of text. It's used for styling or applying effects to small parts within text, like words or phrases.
   - Use it when you want to style or target a specific portion of text within a paragraph or line.

```
<p>This is a <span style="color:blue;">blue word</span> within a
paragraph.</p>
```

3. **`<div>` Tag:**

   A block-level element, meaning it takes up the full width of its container. It's used as a generic container for grouping elements together, especially for layout purposes.

   Use it when you need to group multiple elements or create larger sections, like dividing a webpage into headers, footers, or content sections.

```
<div class="content-section">
    <h2>Section Title</h2>
    <p>Content goes here...</p>
</div>
```

summary:

- Use `<p>` for paragraphs of text,

- Use `<span>` for inline styling within text,
- Use `<div>` for grouping and layout purposes across sections of content.

16. **Describe the `<ul>`, `<ol>`, and `<li>` tags for creating lists.**

Ans: The `<ul>`, `<ol>`, and `<li>` tags are used in HTML to create lists.

1. **`<ul>` (Unordered List)**: This tag creates a list where items are marked with bullet points by default. It's useful for lists where order doesn't matter.

```html
<ul>
    <li>Apples</li>
    <li>Oranges</li>
    <li>Bananas</li>
</ul>
```

2. **`<ol>` (Ordered List):** This tag creates a numbered list, making it ideal for steps or items that follow a specific order.

```html
<ol>
    <li>Preheat the oven</li>
    <li>Mix ingredients</li>
    <li>Bake for 20 minutes</li>
</ol>
```

3. **`<li>` (List Item):** The `<li>` tag is used within both `<ul>` and `<ol>` to define each list item. Each `<li>` tag represents a single item in the list, whether it's bulleted or numbered.

Summary :

`<ul>` and `<ol>` define the type of list (unordered or ordered), and `<li>` defines each individual item within that list.

17. **Describe how `<input>` works and some of its types, such as `text`, `radio`, and `checkbox`..**

Ans:

The `<input>` element in HTML is used to create interactive fields where users can enter data. It has different types that define the kind of input required:

1. **Text**: `<input type="text">` - Allows users to enter single-line text input.
2. **Radio**: `<input type="radio">` - Lets users select one option from a set of predefined choices; only one option can be selected at a time.
3. **Checkbox**: `<input type="checkbox">` - Enables users to select one or more options from a list of choices; multiple selections are possible.

Each type serves a specific purpose, helping to gather user information effectively.

18. **What are `<button>`, `<textarea>`, and `<select>` tags, and how do they function?**

Ans:
`<button>`: Creates a clickable button, often used to submit forms or trigger JavaScript functions.
`<textarea>`: Provides a multi-line text input area, useful for collecting larger amounts of text.
`<select>`: Creates a dropdown menu with multiple `<option>` elements, allowing users to select from predefined choices.

19. **How to link CSS to HTML?**

Ans:
To link CSS to HTML, add this line inside the `<head>` section of your HTML file:

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

20. **What is the `<label>` tag, and how is it associated with form inputs?**

Ans:

The `<label>` tag in HTML is used to define a label for form elements like `<input>`, `<textarea>`, and `<select>`. It's associated with a form input by using the `for` attribute, which matches the `id` of the input element, or by wrapping the input element directly. This improves accessibility and makes the label clickable, focusing on the associated input.

Step 4: Advanced HTML Concepts

21. **What is the `<iframe>` tag, and what are some common attributes used with it?**

Ans:

The `<iframe>` tag in HTML is used to embed another HTML page within the current page. It’s commonly used for displaying embedded content like videos, maps, or web pages.

Common attributes include:

- **src:** URL of the page to display.
- **width and height:** Size of the iframe.
- **frameborder:** Controls border visibility (deprecated in HTML5).
- **allow:** Permissions for content (e.g., autoplay, fullscreen).
- **sandbox:** Adds extra security restrictions on the content.
- **allowfullscreen:** Enables full-screen mode for the content.

22. **Describe how the `<audio>` and `<video>` tags work, including common attributes like `controls`, `autoplay`, and `loop`.**

Ans:

The `<audio>` and `<video>` tags in HTML are used to embed audio and video files in a webpage. Both tags allow for media playback directly in the browser without additional plugins.

- `controls`: Adds play, pause, volume, and other basic controls.
- `autoplay`: Automatically starts playing the media when the page loads (not always recommended, as it can disrupt user experience).
- `loop`: Replays the media continuously once it finishes.

Example usage:

```
<audio controls autoplay loop>
  <source src="audio.mp3" type="audio/mpeg">
```

```
</audio>

<video controls autoplay loop>
  <source src="video.mp4" type="video/mp4">
</video>
```

23. **Explain the `<picture>` tag and its usage in responsive design**

Ans:

The `<picture>` tag in HTML is used to provide multiple versions of an image for different screen sizes or device capabilities, enabling responsive design. It works with `<source>` elements to specify different image files and conditions, like screen width, pixel density, or format support. Browsers evaluate each `<source>` and load the best-fit image, improving performance and flexibility. A fallback `<img>` tag is used inside `<picture>` to ensure compatibility with browsers that don't support it.

Example:

```
<picture>
  <source srcset="image-large.jpg" media="(min-width: 800px)">
  <source srcset="image-small.jpg" media="(max-width: 799px)">
  <img src="image-default.jpg" alt="Description">
</picture>
```

24. **Design a basic form that includes text input, radio buttons, a checkbox, and a submit button.**

Ans:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Basic Form</title>
</head>
<body>
    <form action="/submit" method="post">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name"><br><br>

        <label>Gender:</label><br>
        <input type="radio" id="male" name="gender" value="male">
        <label for="male">Male</label><br>
        <input type="radio" id="female" name="gender" value="female">
        <label for="female">Female</label><br><br>

        <input type="checkbox" id="agree" name="agree">
        <label for="agree">I agree to the terms and
conditions</label><br><br>

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

25. **Embed a responsive YouTube video using the `<iframe>` tag.**

Ans:

```
<iframe width="100%" height="315"
src="https://www.youtube.com/embed/VIDEO_ID" frameborder="0"
allow="accelerometer; autoplay; encrypted-media; gyroscope;
picture-in-picture" allowfullscreen></iframe>
```

26. **Make a collapsible/expandable section using `<details>` and `<summary>` tags.**

Ans:

```
<details>
```

```
  <summary>Click to expand</summary>
  <p>This is the content that can be shown or hidden.</p>
</details>
```

27. **Create a simple navigation bar using `<nav>`, `<ul>`, and `<li>` tags with three links.**

Ans:

```
<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>
```

28. **Create a complex table with merged cells (using `colspan` and `rowspan`) and include a `<caption>`, `<thead>`, `<tfoot>`, and `<tbody>`. Add descriptive `<th>` elements for accessibility and structure.**

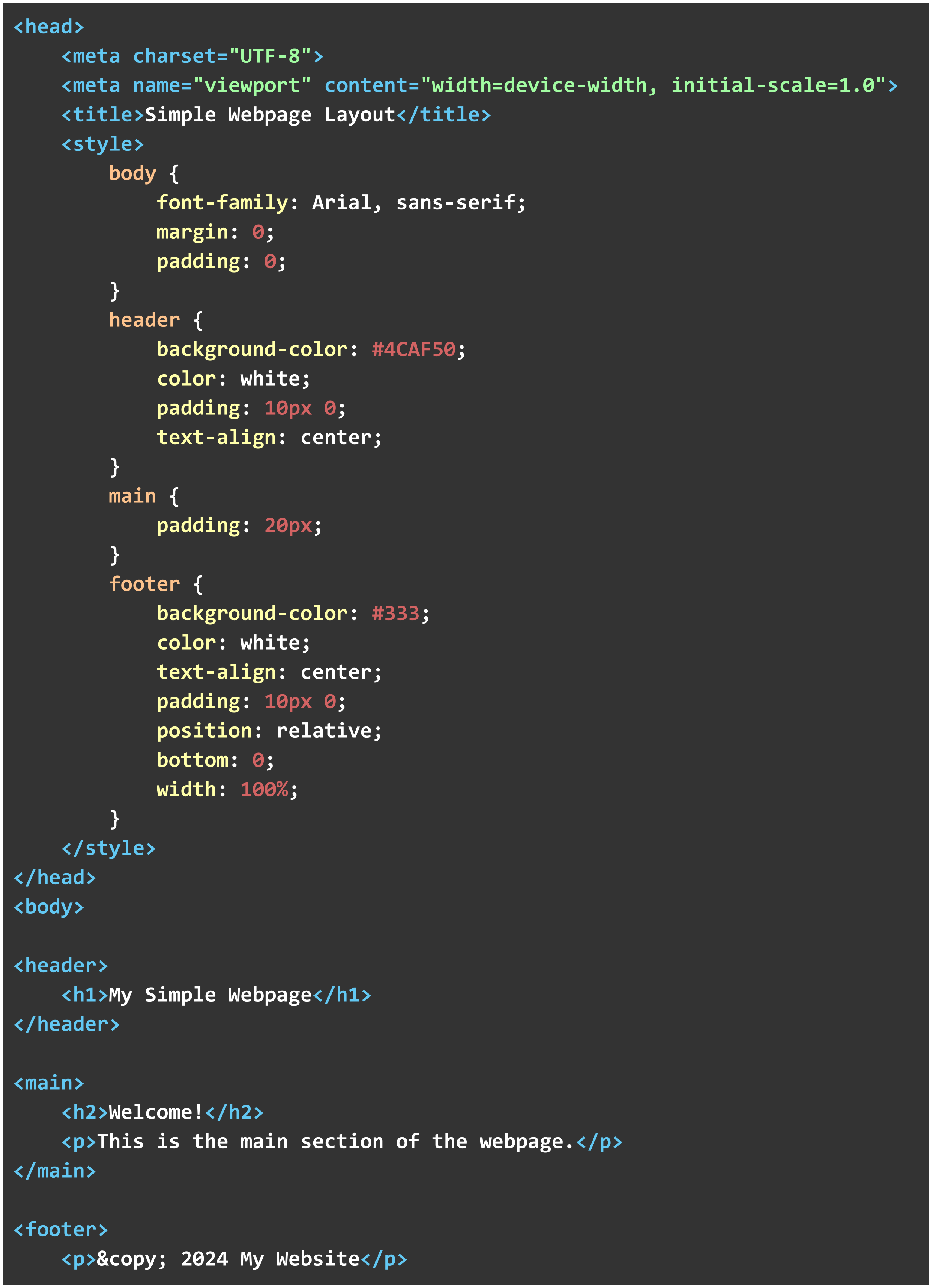
Ans:

```
<table>
    <caption>Monthly Sales Report</caption>
    <thead>
        <tr>
            <th rowspan="2">Product</th>
            <th colspan="3">Sales Data</th>
            <th rowspan="2">Total Revenue</th>
        </tr>
        <tr>
            <th>January</th>
            <th>February</th>
            <th>March</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <th scope="row">Product A</th>
            <td>100</td>
            <td>150</td>
            <td>200</td>
            <td>$4500</td>
        </tr>
        <tr>
            <th scope="row">Product B</th>
            <td>80</td>
            <td>120</td>
            <td>160</td>
            <td>$3600</td>
        </tr>
        <tr>
            <th scope="row">Product C</th>
            <td colspan="3">Total</td>
            <td>$8100</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <th>Total</th>
            <td>280</td>
            <td>390</td>
            <td>360</td>
            <td>$12600</td>
        </tr>
    </tfoot>
</table>
```

29. **Create a simple webpage layout with a header, main section, and footer using only HTML and semantic tags.**

Ans:

```
<!DOCTYPE html>
<html lang="en">
```



```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simple Webpage Layout</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }
        header {
            background-color: #4CAF50;
            color: white;
            padding: 10px 0;
            text-align: center;
        }
        main {
            padding: 20px;
        }
        footer {
            background-color: #333;
            color: white;
            text-align: center;
            padding: 10px 0;
            position: relative;
            bottom: 0;
            width: 100%;
        }
    </style>
</head>
<body>

<header>
    <h1>My Simple Webpage</h1>
</header>

<main>
    <h2>Welcome!</h2>
    <p>This is the main section of the webpage.</p>
</main>

<footer>
    <p>&copy; 2024 My Website</p>
```

```
</footer>

</body>
</html>
```

30. **Write Your favourite 10 HTML Tags & Explain Us Why That’s Your favourite tags.**

Ans:

`<div>`: My favorite for structuring content. It allows me to create distinct sections on a webpage, making layout management easier.

`<span>`: I love using this for inline styling. It helps me apply styles to specific parts of text without breaking the flow.

`<header>`: This tag is great for defining the top section of a webpage. It enhances semantic meaning and improves accessibility.

`<footer>`: Just like the header, the footer defines the bottom section. It’s perfect for placing copyright info, links, and additional context.

`<nav>`: Essential for navigation menus, it organizes links in a semantically meaningful way, improving user experience and SEO.

`<article>`: I appreciate this tag for encapsulating self-contained content. It’s ideal for blog posts or news articles, promoting better readability.

`<section>`: This tag allows me to group related content, making the structure of the document clearer and more organized.

`<img>`: I love how this tag lets me embed images easily. Visuals enhance engagement and help convey messages more effectively.

`<form>`: Essential for user interaction, I enjoy using this tag to create forms for data collection, such as user feedback or subscriptions.

`<button>`: I find this tag versatile for interactive elements. It allows me to create buttons for actions, improving the overall interactivity of a webpage.