

1. What is HTML? Explain its purpose in web development

- HTML, or Hypertext Markup Language, is the standard language used to create the structure and content of web pages. It's the backbone of the web, allowing developers to organize and define elements like text, images, links, and multimedia.
- HTML (Hypertext Markup Language) is essential in web development for:
 - Structure: Defines the layout of a webpage using tags and elements.
 - Content: Embeds text, images, and multimedia.
 - Navigation: Creates hyperlinks for easy web navigation.
 - Forms: Collects user input with text fields, buttons, etc.
 - Foundation: Works with CSS for styling and JavaScript for interactivity.

2. What are semantic HTML elements? Provide examples of three semantic tags and explain their significance.

Ans: Semantic HTML elements are HTML5 elements that clearly describe their meaning and content to both the browser and developers. These elements improve accessibility, SEO, and code readability by providing meaningful structure to a webpage

- **<article>**
Significance: Helps search engines and assistive technologies understand distinct content blocks, improving SEO and accessibility.
- **<section>**
Significance: Improves document organization, making it easier to navigate and understand.
- **<nav>**
Significance: Helps screen readers and search engines identify the primary navigation for a website.

3. What is the difference between block-level and inline elements in HTML? Provide examples.

Feature	Block-Level Elements	Inline Elements
Takes Full Width?	yes	No (only as wide as content)
Starts on a New Line?	yes	no
Can Contain Other Block Elements?	yes	no

4. What is the purpose of the tag in HTML? How is it commonly used?

Ans: The <div> tag is a block-level container used to group elements for styling, layout, or scripting. It has no inherent meaning but is essential for organizing webpage content.

Common Uses:

Grouping Content: Wraps multiple elements together.

Applying Styles: Works with CSS for layout and design.

Creating Layouts: Used with Flexbox, Grid, or Bootstrap.

JavaScript Manipulation: Targeted for dynamic changes or event handling.

5. What is the Document Object Model (DOM), and how does it relate to HTML?

What is the DOM?

The Document Object Model (DOM) is a programming interface for web documents. It represents an HTML or XML document as a tree structure, where each element is a node that can be manipulated using JavaScript.

how does it relate to HTML?

The Document Object Model (DOM) represents an HTML document as a tree structure, where each element is a node. It allows JavaScript to access, modify, and interact with HTML dynamically.

6. Explain the difference between <div> and elements in html. Ehen would you use each ?

Feature	<div>	
Type	Block-level element	Inline element
Layout Behavior	Takes up the full width and starts on a new line	Stays within the same line without breaking content

7. What is the purpose of the alt attribute in the tag? Why is it important for accessibility and SEO?

Ans: The alt (alternative text) attribute in the tag provides a text description of an image. It serves as a fallback when an image cannot be displayed and improves accessibility and SEO.

- **Why is it important for accessibility and SEO?**

1. Accessibility (For Visually Impaired Users):
Screen readers read the alt text aloud, helping visually impaired users understand the image content.
2. SEO (Search Engine Optimization):
Search engines use alt text to understand image content, improving a website's ranking in image searches.
Helps in better indexing and relevance for image-based search results.

8. What are HTML attributes, and how do they modify elements?

Ans: HTML **attributes** provide additional information about an element, modifying its behavior, appearance, or functionality. They are placed within the **opening tag** in the format:

attribute="value"

How Attributes Modify Elements:

Define Properties (e.g., href in <a> sets the link destination).

Control Appearance (e.g., style="color:red;" changes text color).

Affect Behavior (e.g., disabled in <button> prevents clicking).

9. Give an example of an HTML tag with attributes.

Ans:

10. . What is the difference between an absolute URL and a relative URL in HTML? Provide an example of each.

Ans: Absolute URL

An absolute URL provides the complete address to a resource, including the domain and protocol:

Click here to view the photo

Relative URL

A relative URL points to a resource relative to the location of the current page:

Click here to view the photo

Q11. Build a simple webpage that displays text as shown in the below image.

This text will be bolded.
This text will be italic.
This text will be underlined
This text will be highlighted
This is normal text This will be super scripted This is normal again
This is normal text This text will be subscripted
Normal Text Small Text
~~This text will be deleted~~

```
<!DOCTYPE html>

<html>
<head>
<title>Text Formatting Example</title>
</head>
<body>

<p><b>This text will be bolded.</b></p>
<p><i>This text will be italic.</i></p>
<p><u>This text will be underlined</u></p>
<p><mark>This text will be highlighted</mark></p>
<p>This is normal text This will be super scripted <sup>This is normal
again</sup></p>
<p>This is normal text This text will be subscripted <sub>This is normal
again</sub></p>
<p>Normal Text <small>Small Text</small></p>
<p><del>This text will be deleted</del></p>

</body>
</html>
```

Output:

This text will be bolded.

This text will be italic.

This text will be underlined

This text will be highlighted

This is normal text This will be super scripted ^{This is normal again}

This is normal text This text will be subscripted _{This is normal again}

Normal Text Small Text

~~This text will be deleted~~

Q12. Build a simple webpage that helps users navigate different web development-related websites. Note: On clicking the hyperlink the web pages should open in a new tab. Below is a reference image.

Navigate Me:

Take me to [PW Skills](#) to buy a course.

Take me to [MDN docs](#) to know more about Web Development.

Take me to [PW Skills Lab](#) to practice live coding.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Navigation Links</title>
  <style>
    span{
```

```

        color:blue;

    }
</style>
</head>
<body>

    <h1>Navigate Me:</h1>

    <p>
        <a >Take me to <span href="#">PW Skills </span> to buy a course.</a>
    </p>

    <p>
        <a >Take me to <span href="#">MDN docs </span> to know more about Web
Development.</a>
    </p>

    <p>
        <a >Take me to <span href="#">PW Skills Lab </span> to practice live
coding.</a>
    </p>

</body>
</html>

```

Output:

Navigate Me:

Take me to [PW Skills](#) to buy a course.

Take me to [MDN docs](#) to know more about Web Development.

Take me to [PW Skills Lab](#) to practice live coding.

Q13. Build a simple blog web page with 3 pages home, web development, and web design. Each page must contain hyperlinks to other pages in the top, a heading of the page topic and a paragraph of information. For the home page you can add some information about yourself.

Ans:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Web Design</title>
  <style>
    /* ... (same styles as index.html) ... */
  </style>
</head>
<body>
  <nav>
    <a href="index.html">Home</a>
    <a href="webdev.html">Web Development</a>
    <a href="webdesign.html">Web Design</a>
  </nav>

  <h1>Web Design</h1>

  <p>Web design is the process of planning and creating the visual appearance and functionality of a website. It focuses on user experience (UX) and user interface (UI) design. A good web design should be visually appealing, easy to navigate, and accessible to all users. Key elements of web design include layout, typography, color scheme, imagery, and content. Web designers often use tools like Figma or Adobe XD to create mockups and prototypes before the website is developed. Effective web design plays a crucial role in attracting and retaining website visitors.</p>

</body>
</html>
```

Ans:

Web Design

Web design is the process of planning and creating the visual appearance and functionality of a website. It focuses on user experience (UX) and user interface (UI) design. A good web design should be visually appealing, easy to navigate, and accessible to all users. Key elements of web design include layout, typography, color scheme, imagery, and content. Web designers often use tools like Figma or Adobe XD to create mockups and prototypes before the website is developed. Effective web design plays a crucial role in attracting and retaining website visitors.

Q14. Create an ordered list of HTML tags. Each list item must include the tag name and some information about the tag.

```
Ans: <!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>HTML Tag List</title>
  <style>
    body {
      font-family: sans-serif;
    }
    ol {
      list-style-type: decimal; /* Use decimal numbering */
      padding-left: 20px; /* Add left padding for indentation */
    }
    li {
      margin-bottom: 10px; /* Add spacing between list items */
    }
    strong { /* Make tag names bold */
      font-weight: bold;
    }
    code { /* Style code snippets */
      font-family: monospace;
      background-color: #f0f0f0; /* Light gray background */
      padding: 2px 4px;
      border-radius: 3px;
    }
  </style>
  <ol>
    <li><h1>HTML</h1>: The root element of an HTML document. It defines the document's language and provides a container for all other elements. It is required to be present in every HTML document.
    <li><h2>Head</h2>: A container for meta-information about the document, such as the title, character set, and viewport settings. It is used to control the document's appearance and behavior.
    <li><h2>Body</h2>: The main content area of the document. It contains all the visible elements, such as text, images, and links.
    <li><h2>Footer</h2>: A container for information at the bottom of the page, such as the page number, copyright notice, and contact information.
  </ol>
</body>
</html>
```



```

    }
  </style>
</head>
<body>

  <h1>Ordered List of HTML Tags</h1>

  <ol>
    <li>
      <strong>&lt;html&gt;</strong>: The root element of an HTML document.
      It tells the browser that this is an HTML document.
      Example: <code>&lt;html lang="en"&gt; ... &lt;/html&gt;</code>
    </li>
    <li>
      <strong>&lt;head&gt;</strong>: Contains meta-information about the
      HTML document, such as character set, title, and links to stylesheets.
      Example: <code>&lt;head&gt; &lt;title&gt;My Page&lt;/title&gt;
      &lt;/head&gt;</code>
    </li>
    <li>
      <strong>&lt;title&gt;</strong>: Specifies the title of the HTML
      document, which is shown in the browser's title bar or tab.
      Example: <code>&lt;title&gt;My Website&lt;/title&gt;</code>
    </li>
    <li>
      <strong>&lt;body&gt;</strong>: Contains the visible content of the
      HTML document. This is where all the content you want to display goes.
      Example: <code>&lt;body&gt; &lt;h1&gt;Welcome&lt;/h1&gt;
      &lt;/body&gt;</code>
    </li>
    <li>
      <strong>&lt;h1&gt;</strong> to <strong>&lt;h6&gt;</strong>: Define
      headings of different levels. <code>&lt;h1&gt;</code> is the highest (most
      important) level, and <code>&lt;h6&gt;</code> is the lowest.
      Example: <code>&lt;h1&gt;Main Heading&lt;/h1&gt;</code>,
      <code>&lt;h2&gt;Subheading&lt;/h2&gt;</code>
    </li>
    <li>
      <strong>&lt;p&gt;</strong>: Defines a paragraph. Used for blocks of
      text.
      Example: <code>&lt;p&gt;This is a paragraph of text.&lt;/p&gt;</code>
    </li>
    <li>
      <strong>&lt;a&gt;</strong>: Defines a hyperlink, used to link to
      other web pages or resources.

```

```

        Example: <a href="https://www.example.com">Link to
Example</a>
    </li>
    <li>
        <strong><img>: Embeds an image in the HTML document.
        Example: 
    </li>
    <li>
        <strong><ul>: Defines an unordered (bulleted) list.
        Example: <ul> <li>Item 1</li> <li>Item
2</li> </ul>
    </li>
    <li>
        <strong><ol>: Defines an ordered (numbered) list.
        Example: <ol> <li>First item</li>
<li>Second item</li> </ol>
    </li>
    <li>
        <strong><li>: Defines a list item within
<ul> or <ol>.
        Example: <li>A list item</li>
    </li>
    <li>
        <strong><div>: Defines a division or section in an
HTML document. Often used as a container for other elements and styled with CSS.
        Example: <div style="color:blue;">This is a blue
div</div>
    </li>
    <li>
        <strong><span>: Defines an inline element, often used
to style a small portion of text within a paragraph.
        Example: <p>This is some <span
style="color:red;">red</span> text.</p>
    </li>
</ol>

</body>
</html>

```

Output:

Ordered List of HTML Tags

1. **<html>**: The root element of an HTML document. It tells the browser that this is an HTML document. Example: `<html lang="en"> ... </html>`
2. **<head>**: Contains meta-information about the HTML document, such as character set, title, and links to stylesheets. Example: `<head> <title>My Page</title> </head>`
3. **<title>**: Specifies the title of the HTML document, which is shown in the browser's title bar or tab. Example: `<title>My Website</title>`
4. **<body>**: Contains the visible content of the HTML document. This is where all the content you want to display goes. Example: `<body> <h1>Welcome</h1> </body>`
5. **<h1> to <h6>**: Define headings of different levels. `<h1>` is the highest (most important) level, and `<h6>` is the lowest. Example: `<h1>Main Heading</h1>`, `<h2>Subheading</h2>`
6. **<p>**: Defines a paragraph. Used for blocks of text. Example: `<p>This is a paragraph of text.</p>`
7. **<a>**: Defines a hyperlink, used to link to other web pages or resources. Example: `Link to Example`
8. ****: Embeds an image in the HTML document. Example: ``
9. ****: Defines an unordered (bulleted) list. Example: ` Item 1 Item 2 `
10. ****: Defines an ordered (numbered) list. Example: ` First item Second item `
11. ****: Defines a list item within `` or ``. Example: `A list item`
12. **<div>**: Defines a division or section in an HTML document. Often used as a container for other elements and styled with CSS. Example: `<div style="color:blue;">This is a blue div</div>`
13. ****: Defines an inline element, often used to style a small portion of text within a paragraph. Example: `<p>This is some red text.</p>`

Q15. create a description list of full stack web development tech stack, using the tag. Each term should be a tech stack name and each description should be a brief explanation of what the tech stack is used for.

Ans: `<!DOCTYPE html>`

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Full Stack Web Development Tech Stack</title>
  <style>
    body {
      font-family: sans-serif;
      margin: 20px;
    }
    dl { /* Description list container */
      margin-bottom: 20px;
    }
    dt { /* Term (tech stack name) */
      font-weight: bold;
      margin-top: 10px; /* Add spacing above each term */
    }
    dd { /* Description */
      margin-left: 20px; /* Indent descriptions */
      margin-bottom: 10px; /* Space between descriptions */
    }
  </style>
```

```
</head>
<body>

  <h1>Full Stack Web Development Tech Stack</h1>

  <dl>
    <dt>MEAN (MongoDB, Express.js, AngularJS/React, Node.js)</dt>
    <dd>A JavaScript-based stack for building modern web applications.
MongoDB is a NoSQL database, Express.js is a web framework, AngularJS/React is a
front-end framework/library, and Node.js is a server-side JavaScript runtime
environment.</dd>

    <dt>MERN (MongoDB, Express.js, React, Node.js)</dt>
    <dd>Similar to MEAN, but specifically uses React for the front-end. A
popular choice for building dynamic and interactive user interfaces.</dd>

    <dt>LAMP (Linux, Apache, MySQL, PHP)</dt>
    <dd>A classic and widely used stack. Linux is the operating system,
Apache is the web server, MySQL is the database, and PHP is the server-side
scripting language.</dd>

    <dt>Ruby on Rails</dt>
    <dd>A powerful framework for web development using the Ruby
language. Known for its convention-over-configuration approach, which speeds up
development.</dd>

    <dt>Python/Django</dt>
    <dd>A combination of the Python programming language and the Django web
framework. Python's versatility and Django's robust features make it suitable
for various web projects.</dd>

    <dt>Java/Spring</dt>
    <dd>Java, a robust and versatile programming language, is often paired
with the Spring framework for building enterprise-level web applications. Spring
provides tools for dependency injection and managing complex application
components.</dd>

    <dt>.NET (C#)</dt>
    <dd>Microsoft's web development platform. C# is a powerful language, and
ASP.NET provides a framework for building web applications. Often used with
Microsoft SQL Server as the database.</dd>

    <dt>Serverless (AWS Lambda, Google Cloud Functions, Azure Functions)</dt>
    <dd>A cloud-based approach where server management is handled by the
cloud provider. Functions are triggered by events, and you only pay for the
```

```

compute time you consume. Can be combined with various front-end
technologies.</dd>
</dl>

</body>
</html>

```

Output:

Full Stack Web Development Tech Stack

MEAN (MongoDB, Express.js, AngularJS/React, Node.js)

A JavaScript-based stack for building modern web applications. MongoDB is a NoSQL database, Express.js is a web framework, AngularJS/React is a front-end framework/library, and Node.js is a server-side JavaScript runtime environment.

MERN (MongoDB, Express.js, React, Node.js)

Similar to MEAN, but specifically uses React for the front-end. A popular choice for building dynamic and interactive user interfaces.

LAMP (Linux, Apache, MySQL, PHP)

A classic and widely used stack. Linux is the operating system, Apache is the web server, MySQL is the database, and PHP is the server-side scripting language.

Ruby on Rails

A powerful framework for web development using the Ruby language. Known for its convention-over-configuration approach, which speeds up development.

Python/Django

A combination of the Python programming language and the Django web framework. Python's versatility and Django's robust features make it suitable for various web projects.

Java/Spring

Java, a robust and versatile programming language, is often paired with the Spring framework for building enterprise-level web applications. Spring provides tools for dependency injection and managing complex application components.

.NET (C#)

Microsoft's web development platform. C# is a powerful language, and ASP.NET provides a framework for building web applications. Often used with Microsoft SQL Server as the database.

Serverless (AWS Lambda, Google Cloud Functions, Azure Functions)

A cloud-based approach where server management is handled by the cloud provider. Functions are triggered by events, and you only pay for the compute time you consume. Can be combined with various front-end technologies.

Q16. Create an ordered list of the full stack web development tech stack HTML, CSS, and JS. For each tech stack, create a table that lists the tech stack name, its primary use cases, and some key features or benefits. Below is a reference image.

```

Ans: <!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Full Stack Web Development Tech Stack</title>

```

```

<style>
  body {
    font-family: sans-serif;
    margin: 20px;
  }
  h1 {
    text-align: center; /* Center the main heading */
  }
  ol {
    list-style-type: decimal;
    padding-left: 20px;
    margin-bottom: 20px; /* Space below the ordered list */
  }
  table {
    width: 80%; /* Make tables responsive */
    margin: 20px auto; /* Center tables */
    border-collapse: collapse;
  }
  th, td {
    border: 1px solid #ddd;
    padding: 8px;
    text-align: left;
  }
  th {
    background-color: #f0f0f0;
    font-weight: bold;
  }
  caption {
    font-weight: bold;
    margin-bottom: 10px;
  }
</style>
</head>
<body>

  <h1>Full Stack Web Development Tech Stack</h1>

  <ol>
    <li>
      <h2>HTML (HyperText Markup Language)</h2>
      <table>
        <caption>HTML</caption>
        <tr>
          <th>Tech Stack</th>
          <th>Primary Use Cases</th>

```

```

        <th>Key Features/Benefits</th>
    </tr>
    <tr>
        <td>HTML</td>
        <td>Structuring the content of web pages.</td>
        <td>Provides semantic meaning to content, forms the
foundation of all web pages, easy to learn.</td>
    </tr>
</table>
</li>

<li>
    <h2>CSS (Cascading Style Sheets)</h2>
    <table>
        <caption>CSS</caption>
        <tr>
            <th>Tech Stack</th>
            <th>Primary Use Cases</th>
            <th>Key Features/Benefits</th>
        </tr>
        <tr>
            <td>CSS</td>
            <td>Styling the visual presentation of web pages (layout,
colors, fonts, etc.).</td>
            <td>Separation of content from presentation, allows for
responsive design, improves user experience.</td>
        </tr>
    </table>
</li>

<li>
    <h2>JavaScript (JS)</h2>
    <table>
        <caption>JavaScript</caption>
        <tr>
            <th>Tech Stack</th>
            <th>Primary Use Cases</th>
            <th>Key Features/Benefits</th>
        </tr>
        <tr>
            <td>JavaScript</td>
            <td>Adding interactivity and dynamic behavior to web pages
(animations, form validation, handling user events, making API calls).</td>
            <td>Client-side scripting, enhances user engagement, enables
creation of web applications.</td>
        </tr>
    </table>
</li>

```

```
        </tr>
      </table>
    </li>
  </ol>

</body>
</html>
```

Output:

Full Stack Web Development Tech Stack

1. HTML (HyperText Markup Language)

HTML

Tech Stack	Primary Use Cases	Key Features/Benefits
HTML	Structuring the content of web pages.	Provides semantic meaning to content, forms the foundation of all web pages, easy to learn.

2. CSS (Cascading Style Sheets)

CSS

Tech Stack	Primary Use Cases	Key Features/Benefits
CSS	Styling the visual presentation of web pages (layout, colors, fonts, etc.).	Separation of content from presentation, allows for responsive design, improves user experience.

3. JavaScript (JS)

JavaScript

Tech Stack	Primary Use Cases	Key Features/Benefits
JavaScript	Adding interactivity and dynamic behavior to web pages (animations, form validation, handling user events, making API calls).	Client-side scripting, enhances user engagement, enables creation of web applications.

Q17. Build a complex nested list structure representing a multi-level table of contents. Use unordered lists () and list items (li) with inline-block styling to create a structured layout. Apply formatting tags to enhance the presentation of list items.

Table of Contents

- [Part 1: Introduction](#)
- [Part 2: Getting Started](#)
 - [2.1 Installing the Software](#)
 - [2.2 Creating a New Project](#)
 - [2.2.1 Project Templates](#)
 - [2.2.2 Customizing Settings](#)
 - [2.3 Exploring the Interface](#)
 - [2.3.1 Toolbar Features](#)
 - [2.3.2 Panel Layout](#)
 - [2.3.2.1 Docking Panels](#)
 - [2.3.2.2 Tabbed Interface](#)
- [Part 3: Advanced Topics](#)
 - [3.1 Working with Plugins](#)
 - [3.1.1 Installing Plugins](#)
 - [3.1.2 Plugin Configuration](#)
 - [3.2 Customizing the UI](#)
 - [3.2.1 Changing Themes](#)
 - [3.2.2 Configuring Shortcuts](#)
 - [3.3 Optimizing Performance](#)
 - [3.3.1 Caching Strategies](#)
 - [3.3.2 Resource Minification](#)
- [Part 4: Conclusion](#)

Ans: `<!DOCTYPE html>`

`<html lang="en">`

`<head>`

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Table of Contents</title>
<style>
  body {
    font-family: sans-serif;
    margin: 20px;
  }
  .toc-title {
    font-size: 1.5em;
    font-weight: bold;
    margin-bottom: 10px;
  }
  .toc-list {
    list-style-type: none;
    padding-left: 0;
  }
  .toc-item {
    margin-bottom: 5px;
  }
  .toc-item a {
    text-decoration: none;
    color: #333;
  }
  .toc-item ul { /* Style for nested lists */
    list-style-type: circle; /* Or 'square', 'disc', etc. */
    padding-left: 20px;
    margin-top: 5px;
  }
  .toc-item-non-link {
    font-weight: bold;
  }
</style>
</head>
<body>

  <div class="toc-title">Table of Contents</div>

  <ul class="toc-list">
    <li class="toc-item-non-link">Part 1: Introduction</li>
    <li class="toc-item-non-link">Part 2: Getting Started</li>
    <ul class="toc-list">
      <li class="toc-item"><a href="#installing">2.1
Installing the Software</a></li>
      <li class="toc-item">2.2 Creating a New Project</li>
    </ul>
  </ul>

```

```

        <ul class="toc-list"> <li class="toc-item"><a
href="#templates">2.2.1 Project Templates</a></li>
        <li class="toc-item"><a href="#settings">2.2.2 Customizing
Settings</a></li>
        </ul>
        <li class="toc-item"><a href="#interface">2.3 Exploring the
Interface</a></li>
        <ul class="toc-list"> <li class="toc-item"><a href="#toolbar">2.3.1
Toolbar Features</a></li>
        <li class="toc-item">2.3.2 Panel Layout</li>
        <ul class="toc-list"> <li class="toc-item"><a
href="#docking">2.3.2.1 Docking Panels</a></li>
        <li class="toc-item"><a href="#tabbed">2.3.2.2 Tabbed
Interface</a></li>
        </ul>
        </ul>
        </ul>
        <li class="toc-item-non-link">Part 3: Advanced Topics</li>
        <ul class="toc-list"> <li class="toc-item">3.1 Working with Plugins</li>
        <ul class="toc-list"> <li class="toc-item"><a href="#installing-
plugins">3.1.1 Installing Plugins</a></li>
        <li class="toc-item"><a href="#configuring-plugins">3.1.2 Plugin
Configuration</a></li>
        </ul>
        <li class="toc-item">3.2 Customizing the UI</li>
        <ul class="toc-list"> <li class="toc-item"><a href="#themes">3.2.1
Changing Themes</a></li>
        <li class="toc-item"><a href="#shortcuts">3.2.2 Configuring
Shortcuts</a></li>
        </ul>
        <li class="toc-item">3.3 Optimizing Performance</li>
        <ul class="toc-list"> <li class="toc-item"><a href="#caching">3.3.1
Caching Strategies</a></li>
        <li class="toc-item"><a href="#minification">3.3.2 Resource
Minification</a></li>
        </ul>
        </ul>
        <li class="toc-item-non-link">Part 4: Conclusion</li>
    </ul>

</body>
</html>

```

Q18. Create a table to display a conference schedule. Each row corresponds to a time slot, and each column corresponds to a room. Some time slots might have multiple sessions running simultaneously in different rooms. Utilize rowspan and colspan attributes as necessary to accommodate this complex schedule. (use table attribute "cellpadding" to give extra padding in each table cell).

Output should look like this:

Conference Schedule				
Time	Room 1	Room 2	Room 3	Room 4
9:00 AM - 10:00 AM	Keynote	Session A	Session B	Session C
		Session D	Session E	
	10:30 AM - 11:30 AM	Session F		
12:00 PM - 1:00 PM	Lunch Break			
1:00 PM - 2:00 PM	Session G	Session H	Session I	Session J
	Session K		Session L	Session M

```
Code: <!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Conference Schedule</title>
  <style>
    body {
      font-family: sans-serif;
    }
    table {
      width: 80%;
      margin: 20px auto;
      border-collapse: collapse;
      border: 1px solid #ddd; /* Add a border to the table */
      table-layout: fixed; /* Ensure even column widths */
    }
    th, td {
      border: 1px solid #ddd;
      padding: 10px; /* Add padding to cells */
      text-align: center; /* Center the content */
      vertical-align: middle; /* Center content vertically */
    }
  </style>
</head>
<body>
  <table>
    <tr>
      <th data-cs="5" data-kind="parent">Conference Schedule</th>
      <th data-kind="ghost"></th>
      <th data-kind="ghost"></th>
      <th data-kind="ghost"></th>
      <th data-kind="ghost"></th>
    </tr>
    <tr>
      <th>Time</th>
      <th>Room 1</th>
      <th>Room 2</th>
      <th>Room 3</th>
      <th>Room 4</th>
    </tr>
    <tr>
      <td data-kind="parent" data-rs="3">9:00 AM - 10:00 AM</td>
      <td data-kind="parent" data-rs="2">Keynote</td>
      <td>Session A</td>
      <td>Session B</td>
      <td data-kind="parent" data-rs="3">Session C</td>
    </tr>
    <tr>
      <td data-kind="ghost"></td>
      <td data-kind="ghost"></td>
      <td>Session D</td>
      <td>Session E</td>
      <td data-kind="ghost"></td>
    </tr>
    <tr>
      <td data-kind="ghost"></td>
      <td>10:30 AM - 11:30 AM</td>
      <td data-cs="2" data-kind="parent">Session F</td>
      <td data-kind="ghost"></td>
      <td data-kind="ghost"></td>
    </tr>
    <tr>
      <td>12:00 PM - 1:00 PM</td>
      <td data-cs="4" data-kind="parent">Lunch Break</td>
      <td data-kind="ghost"></td>
      <td data-kind="ghost"></td>
      <td data-kind="ghost"></td>
    </tr>
    <tr>
      <td data-kind="parent" data-rs="2">1:00 PM - 2:00 PM</td>
      <td>Session G</td>
      <td data-kind="parent" data-rs="2">Session H</td>
      <td>Session I</td>
      <td>Session J</td>
    </tr>
    <tr>
      <td data-kind="ghost"></td>
      <td>Session K</td>
      <td data-kind="ghost"></td>
      <td>Session L</td>
      <td>Session M</td>
    </tr>
  </table>
</body>
</html>
```

```

    th {
      background-color: #f0f0f0;
      font-weight: bold;
    }
    caption {
      font-size: 1.2em;
      font-weight: bold;
      margin-bottom: 10px;
    }
  }
</style>
</head>
<body>

  <table>
    <caption>Conference Schedule</caption>
    <thead>
      <tr>
        <th>Time</th>
        <th>Room 1</th>
        <th>Room 2</th>
        <th>Room 3</th>
        <th>Room 4</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td rowspan="2">9:00 AM - 10:00 AM</td>
        <td rowspan="2">Keynote</td>
        <td>Session A</td>
        <td>Session B</td>
        <td rowspan="2">Session C</td>
      </tr>
      <tr>
        <td>Session D</td>
        <td>Session E</td>
      </tr>
      <tr>
        <td>10:30 AM - 11:30 AM</td>
        <td colspan="2">Session F</td>
        <td/>
        <td/>
      </tr>
      <tr>
        <td>12:00 PM - 1:00 PM</td>
        <td colspan="4">Lunch Break</td>
      </tr>
    </tbody>
  </table>

```

```

    </tr>
    <tr>
        <td rowspan="2">1:00 PM - 2:00 PM</td>
        <td>Session G</td>
        <td rowspan="2">Session H</td>
        <td>Session I</td>
        <td>Session J</td>
    </tr>
    <tr>
        <td>Session K</td>
        <td>Session L</td>
        <td>Session M</td>
    </tr>
</tbody>
</table>

</body>
</html>

```

Output:

Conference Schedule

Time	Room 1	Room 2	Room 3	Room 4
9:00 AM - 10:00 AM	Keynote	Session A	Session B	Session C
		Session D	Session E	
10:30 AM - 11:30 AM	Session F			
12:00 PM - 1:00 PM	Lunch Break			
1:00 PM - 2:00 PM	Session G	Session H	Session I	Session J
	Session K		Session L	Session M

19. Create an HTML document that properly incorporates semantic elements like <header>, <article>, <section> or <nav> to improve SEO and document structure.

Ans: `<!DOCTYPE html>`

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Semantic HTML Example</title>
  <meta name="description" content="Example of a webpage using semantic HTML
elements for improved SEO and structure.">
  <meta name="keywords" content="semantic html, seo, web development,
accessibility">
  <style>
    body {
      font-family: sans-serif;
      margin: 20px;
      line-height: 1.6;
    }
    header {
      background-color: #f0f0f0;
      padding: 20px;
      text-align: center;
      margin-bottom: 20px;
    }
    nav ul {
      list-style: none;
      padding: 0;
      margin: 0;
      text-align: center; /* Center the navigation links */
      margin-bottom: 20px;
    }
    nav li {
      display: inline;
      margin: 0 10px;
    }
    article {
      border: 1px solid #ddd;
      padding: 20px;
```

```

        margin-bottom: 20px;
    }
    section {
        margin-bottom: 20px;
    }
    aside {
        background-color: #f8f8f8;
        padding: 10px;
        border: 1px solid #eee;
        float: right; /* Place the aside on the right */
        width: 30%; /* Adjust width as needed */
        margin-left: 20px; /* Add some space to the left */
    }
    footer {
        background-color: #333;
        color: white;
        text-align: center;
        padding: 10px;
        position: relative; /* Needed for sticky footer */
        bottom: 0; /* Stick to the bottom */
        width: 100%;
    }
}
</style>
</head>
<body>

    <header>
        <h1>My Awesome Website</h1>
        <p>Welcome to my website! This is a demonstration of semantic HTML.</p>
    </header>

    <nav>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">About</a></li>
            <li><a href="#">Services</a></li>
            <li><a href="#">Contact</a></li>
        </ul>
    </nav>

    <article>
        <h2>Main Article Title</h2>
        <section>
            <h3>Section 1</h3>

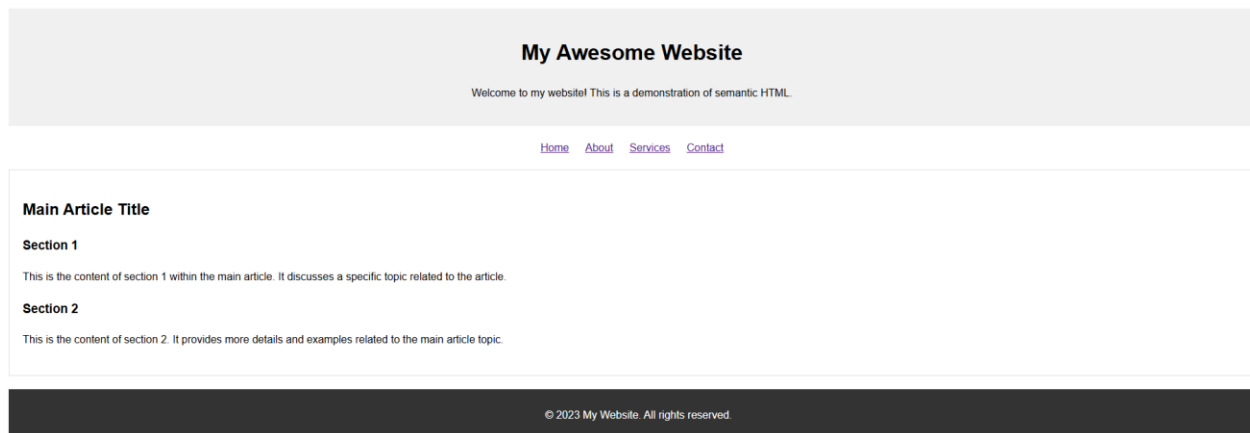
```



```
        <p>This is the content of section 1 within the main article.  It
discusses a specific topic related to the article.</p>
    </section>
    <section>
        <h3>Section 2</h3>
        <p>This is the content of section 2. It provides more details and
examples related to the main article topic.</p>
    </section>
</article>
<footer>
    <p>&copy; 2023 My Website. All rights reserved.</p>
</footer>

</body>
</html>
```

Ans:



Q20. Create an HTML document with appropriate <title> and <meta> tags for SEO optimization. Ensure the title is descriptive and the meta description is concise.

Ans: `<!DOCTYPE html>`

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Unlock Your Coding Potential: Learn Full Stack Web Development</title>
  <meta name="description" content="Become a full-stack web developer with our comprehensive program. Master front-end and back-end technologies, build real-world projects, and launch your tech career. Enroll today!">
  <meta name="keywords" content="full stack web development, web development course, coding bootcamp, learn to code, front-end development, back-end development, HTML, CSS, JavaScript, React, Node.js, Python, career change, tech skills">
  <meta name="author" content="Your Name or Your Company">
  <meta name="robots" content="index, follow">

  <meta property="og:title" content="Unlock Your Coding Potential: Learn Full Stack Web Development">
  <meta property="og:description" content="Become a full-stack web developer with our comprehensive program. Master front-end and back-end technologies, build real-world projects, and launch your tech career. Enroll today!">
  <meta property="og:url" content="https://www.yourwebsite.com/full-stack-program"> <meta property="og:image" content="https://www.yourwebsite.com/images/full-stack-image.jpg"> <meta property="og:type" content="website">

  <meta name="twitter:card" content="summary_large_image">
  <meta name="twitter:title" content="Unlock Your Coding Potential: Learn Full Stack Web Development">
  <meta name="twitter:description" content="Become a full-stack web developer with our comprehensive program. Master front-end and back-end technologies, build real-world projects, and launch your tech career. Enroll today!">
  <meta name="twitter:image" content="https://www.yourwebsite.com/images/full-stack-image.jpg"> </head>
<body>
  </body>
</html>
```

