**HTML**

1. What is Document Type Declaration mean?

The Document Type Declaration is used at the beginning of an HTML document to specify the version of HTML being used. It helps the browser understand how to interpret and render the content of the webpage.

1. What is W3C HTML5 validation Service?

The W3C HTML5 Validation Service is a tool provided by the World Wide Web Consortium (W3C) that allows you to check whether your HTML code conforms to the standards defined by HTML5. It helps identify errors and issues in your HTML code.

1. How to use images as Hyperlink with examples?

You can use images as hyperlinks by wrapping the <img> tag within an <a> (anchor) tag. Here's an example:

<a href="https://www.example.com">

<img src="image.jpg" alt="Example Image">

</a>

1. Explain List Tag and Nested list Tag

HTML provides three types of lists: unordered lists (<ul>), ordered lists (<ol>), and definition lists (<dl>). Lists can also be nested within each other to create sub-lists.

Example:

<ul>

<li>Item 1</li>

<li>Item 2

<ul>

<li>Sub-item 1</li>

<li>Sub-item 2</li>

</ul>

</li>

<li>Item 3</li>

</ul>

1. Explain Image tag along with its parameters

The <img> tag is used to display images on a webpage. It has several attributes, including src (source of the image), alt (alternative text for accessibility), width, and height.

Ex:

<img src="image.jpg" alt="Description of the image" width="300" height="200">

1. Explain Table Tag with its members

The <table> tag is used to create a table on a webpage. It can contain several other tags like <tr> (table row), <th> (table header), and <td> (table data).

Ex:

<table>

<tr>

<th>Header 1</th>

<th>Header 2</th>

</tr>

<tr>

<td>Data 1</td>

<td>Data 2</td>

</tr>

</table>

1. What is <Thead>,<Tfooter>,<Tbody>

These tags are used to structure the content of a table. <thead> is used for table headers, <tfoot> for table footers, and <tbody> for the main content of the table.

1. Explain Colspan and Rowspan attributes

The colspan attribute defines how many columns a table cell should span, while rowspan defines how many rows a table cell should span.

1. Explain Form tag with its attributes

The <form> tag is used to create an HTML form for user input.

The <form> element can contain one or more of the following form elements:

* <input>
* <textarea>
* <button>
* <select>
* <option>
* <optgroup>
* <fieldset>
* <label>
* <output>

Attributes

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| accept-charset | *character\_set* | Specifies the character encodings that are to be used for the form submission |
| action | *URL* | Specifies where to send the form-data when a form is submitted |
| autocomplete | on off | Specifies whether a form should have autocomplete on or off |
| method | get post | Specifies the HTTP method to use when sending form-data |
| name | *text* | Specifies the name of a form |
| novalidate | novalidate | Specifies that the form should not be validated when submitted |
| target | \_blank \_self \_parent \_top | Specifies where to display the response that is received after submitting the form |

1. Design a Registration page with Submit and Reset Button

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Registration Page</title>

</head>

<body>

<h1>Registration Form</h1>

<form action="submit\_registration.php" method="post">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required><br><br>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required><br><br>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required><br><br>

<button type="submit">Submit</button>

<button type="reset">Reset</button>

</form>

</body>

</html>

1. What is the use of Internal Linking

Internal linking involves creating links within a webpage that navigate to different sections of the same webpage. This improves navigation and user experience.

1. Explain the importance of id and class attribute in HTML

The id attribute provides a unique identifier for an element, while the class attribute allows multiple elements to be grouped under a common class. They are crucial for styling and scripting purposes.

1. What is the importance of meta tag

The <meta> tag provides metadata about the HTML document, such as character encoding, viewport settings, and author information. It helps browsers and search engines understand the page better.

1. Design and Panel using List tag

<ul class="panel">

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ul>

1. Explain how Page structure elements meaningfully can be designed using header,footer,articles,navigation,figures,section and many more

1.<header>:

The <header> element usually contains the introductory content of the page. This might include the site logo, site title, main navigation, and sometimes a search bar.

2. <footer>:

The <footer> element typically contains information that goes at the bottom of the page, such as copyright notices, contact details, social media links, and site maps.

3. <nav>:

The <nav> element is used to define a section of navigation links that help users navigate around the website.

4. <article>:

The <article> element represents a standalone piece of content that can be distributed or linked separately. It's used for blog posts, news articles, and similar content.

5. <section>:

The <section> element is used to define a section of content that is thematically related. It can help divide content into meaningful chunks.

6. <figure>:

The <figure> element is used to encapsulate media content, usually an image, along with its caption using the <figcaption> element.

* These elements help create a well-structured and semantic webpage, making it more accessible to users with disabilities and improving search engine optimization (SEO).
* They also make it easier to style and maintain your webpage, as the purpose of each section is clearly defined.
* By using these elements, you can enhance the user experience and make your webpage more understandable for both human readers and software agents.

**CSS**

1. Explain with example Internal (Embedded), Inline, External css .

* Internal CSS: CSS defined within the <style> tag in the HTML document itself.
* Inline CSS: CSS defined directly within the HTML element's style attribute.
* External CSS: CSS defined in a separate .css file and linked to the HTML document using the <link> tag.

Ex:

<!DOCTYPE html>

<html>

<head>

<style>

/\* Internal CSS \*/

h1 {

color: blue;

}

</style>

<link rel="stylesheet" href="styles.css"> <!-- External CSS -->

</head>

<body>

<h1 style="font-size: 24px;">Welcome!</h1> <!-- Inline CSS -->

</body>

</html>

1. Style the page in such a way that text should be on left and Image should be on right

<!DOCTYPE html>

<html>

<head>

<style>

.container {

display: flex;

align-items: center;

}

.text {

flex: 1;

}

.image {

flex: 1;

text-align: right;

}

</style>

</head>

<body>

<div class="container">

<div class="text">

<p>This is some text on the left.</p>

</div>

<div class="image">

<img src="image.jpg" alt="Image" width="200">

</div>

</div>

</body>

</html>

1. What is relative positioning and absolute positioning

* Relative Positioning: Elements are positioned relative to their normal position in the document flow. You can use top, bottom, left, and right properties to adjust the position.
* Absolute Positioning: Elements are positioned relative to the nearest positioned ancestor (or the body if no positioned ancestor is found). It's taken out of the normal document flow.

1. Explain Box Model

The CSS Box Model is a fundamental concept that describes how elements are rendered on a web page. It defines the properties and dimensions of the rectangular "boxes" that contain an element's content, padding, border, and margin. Understanding the box model is essential for designing and laying out web pages effectively.

The box model consists of the following components:

Content:

The innermost part of the box that contains the actual content of the element, such as text, images, or other HTML elements.

Padding:

The space between the content and the element's border. Padding adds space inside the box and can be used to create spacing between the content and the border.

Border:

The border surrounds the content and padding. It can have properties like width, style, and color. Borders separate the content from the margin and provide a visual distinction between the element and its surroundings.

Margin:

The space outside the element's border. Margins are used to create spacing between elements and control their positioning in relation to other elements on the page.

1. Explain the importance of Float attribute

The float attribute in CSS was originally introduced to enable text to wrap around images in a fluid manner. While its primary use case has evolved over time due to the emergence of more advanced layout techniques like Flexbox and Grid, the float attribute still holds some significance in modern web development. Here are some reasons why the float attribute is important:

1.Text Wrapping Around Images:

The original purpose of the float attribute was to allow text to wrap around images, creating visually appealing layouts. This was especially useful for creating magazine-style designs where images and text needed to be closely integrated.

2.Creating Columns:

The float attribute can be used to create multi-column layouts without using advanced layout techniques. It was one of the early methods used for dividing content into columns before CSS Grid and Flexbox were introduced.

3.Historical Compatibility:

Many older web pages and templates still use the float attribute for layout. Understanding how it works is valuable when working with legacy code.

4.Responsive Design in Legacy Context:

In certain scenarios, the float attribute can be used for responsive design in older projects that don't use more modern layout methods. However, it's worth noting that Flexbox and Grid provide more efficient and predictable ways to create responsive layouts.

5.Compatibility with Older Browsers:

While modern layout methods are widely supported, understanding the float attribute can be useful when working with older browsers that may not fully support newer layout techniques.

1. Explain Media Types

Media types in CSS allow you to specify different styles for different types of devices or media conditions. By using media queries and media types, you can create responsive designs that adapt to various screen sizes, orientations, and other presentation characteristics. Media types are part of the broader concept of media queries.

Media types are specified using the @media rule followed by the media type you want to target. Here are some common media types:

Screen (screen):

This is the default media type and is used for devices with screens, such as computer monitors, tablets, and smartphones.

Print (print):

This media type targets styles specifically for printing. You can define how the content should be formatted when printed on paper.

Speech (speech):

This media type is used for screen readers and other speech-based applications. It's used to provide styles for audio-based rendering.

All (all):

The all media type is used when you want to apply styles regardless of the output device or media type.

Braille (braille):

This media type is used for braille tactile feedback devices.

Projection (projection):

This media type targets projected presentations, such as slideshows using projectors.

1. Design a drop down menu

<!DOCTYPE html>

<html>

<head>

    <style>

        .dropdown {

            position: relative;

            display: inline-block;

        }

        .dropdown-content {

            display: none;

            position: absolute;

            background-color: #f9f9f9;

            min-width: 100px;

            box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

            z-index: 1;

        }

        .dropdown:hover .dropdown-content {

            display: block;

        }

    </style>

</head>

<body>

    <div class="dropdown">

        <button>Menu</button>

        <div class="dropdown-content">

            <a href="#">Item 1</a><br>

            <a href="#">Item 2</a><br>

            <a href="#">Item 3</a><br>

        </div>

    </div>

</body>

</html>

1. Make the use of Text shadow

<!DOCTYPE html>

<html>

<head>

    <style>

        h1 {

            text-shadow: 2px 2px 4px red;

        }

    </style>

</head>

<body>

    <h1>Text with Shadow</h1>

</body>

</html>

1. How rounded corners can be given to the rectangle

You can use the border-radius property in CSS to give rounded corners to rectangular elements, such as <div> elements, images, buttons, and more. The border-radius property allows you to control the degree of curvature for each corner individually.

Here's an example of how you can apply rounded corners to a <div> element:

<!DOCTYPE html>

    <html>

    <head>

        <style>

            .rounded {

                width: 150px;

                height: 100px;

                background-color: lightblue;

                border-radius: 20px;

            }

        </style>

    </head>

    <body>

        <div class="rounded">Rounded Rectangle</div>

    </body>

    </html>

1. Explain Color and in what why it can be represented

Color in web design is a crucial aspect of visual communication. It allows you to create appealing and meaningful designs that attract users' attention and convey emotions or messages. In CSS, colors can be represented in several ways:

1.Color names: predefined colors

2.Hex notation: Combination of 6 char, starts with ‘#’

3.RGB (Red,Green,Blue)

4.RGBA: A is alpha ie. Transpernacy of color

5.HSL: hue in degrres, saturation and lightness in %

6.HSLA

**Javascript**

1.Explain DOM Model

2.Write a program of Multiplication

<!DOCTYPE html>

<html>

<head>

    <title>Multiplication Program</title>

</head>

<body>

    <script>

        // Prompt the user for two numbers

        var num1 = parseFloat(prompt("Enter the first number:"));

        var num2 = parseFloat(prompt("Enter the second number:"));

        // Calculate the multiplication result

        var result = num1 \* num2;

        // Display the result

        document.write("The result of " + num1 + " \* " + num2 + " is: " + result);

    </script>

</body>

</html>