

MODULE - 1

HTML

1. Introduction:

What is HTML and its types?

HTML (HyperText Markup Language) is the standard markup language for creating web pages. It defines the structure and layout of content on a web page using a system of tags and attributes. There are different versions of HTML, including HTML4, XHTML, HTML5, and the upcoming HTML6.

Why it is used?

HTML is used to create the structure of web pages by defining elements such as headings, paragraphs, lists, links, and more. It provides a way to organize and present content on the web, making it accessible and navigable for users.

HTML Structure

HTML (HyperText Markup Language) is the standard markup language used to create the structure and content of web pages. It consists of a series of elements, which are enclosed in tags, and these elements define the structure and content of a web page.

HTML Structure Components:

1. Document Type Declaration (<!DOCTYPE html>):

The document type declaration (<!DOCTYPE html>) is used to specify the HTML version and document type being used. It ensures that the web browser renders the HTML document correctly.

Example:

```
<!DOCTYPE html>
```

2. HTML Element (<html>):

The **<html>** element is the root element of an HTML document and wraps all the other elements. It defines the entire HTML document.

Example:

```
<!DOCTYPE html>
<html>
  <!-- Other HTML elements go here -->
</html>
```

3. Head Element (<head>):

The **<head>** element contains meta-information about the HTML document, such as the title, character set, stylesheets, scripts, and other metadata.

Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Web Page</title>
    <!-- Other metadata elements go here -->
  </head>
</html>
```

4. Body Element (<body>):

The **<body>** element contains the main content of the HTML document, including text, images, links, forms, and other elements that are visible to the user.

Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Web Page</title>
</head>
<body>
  <h1>Hello, World!</h1>
  <p>This is my first web page.</p>
</body>
</html>
```

HTML Structure Best Practices:

- **Valid HTML:** Ensure that your HTML document follows the rules and guidelines specified by the HTML standard.
- **Semantic HTML:** Use semantic HTML elements (e.g., **<header>**, **<nav>**, **<section>**, **<article>**, **<footer>**) to provide meaning and structure to your content.
- **Proper Nesting:** Nest HTML elements properly to maintain a logical hierarchy and improve readability.
- **Indentation:** Use indentation to visually represent the hierarchical structure of HTML elements, making the code easier to understand and maintain.
- **Comments:** Use comments (**<!-- -->**) to add explanatory notes within your HTML code, especially for complex or nested structures.

HTML Elements

HTML elements are the building blocks of web pages. They define the structure and content of a web page by encapsulating different types of content, such as text, images, links, forms, and multimedia. HTML elements are represented by tags, which are enclosed in angle brackets (< and >). Here's an overview of HTML elements along with examples:

1. Basic Structure:

HTML elements consist of an opening tag, content, and a closing tag. The opening tag identifies the beginning of the element, and the closing tag identifies the end of the element. The content of the element is placed between the opening and closing tags.

```
<tagname>Content goes here</tagname>
```

2. Examples of HTML Elements:

a. Paragraphs (<p>):

The **<p>** element defines a paragraph of text.

Example:

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

b. Headings (<h1> to <h6>):

Headings define the titles or subtitles of sections within a document. There are six levels of headings, ranging from **<h1>** (most important) to **<h6>** (least important).

Example:

```
<h1>Main Heading</h1>  
<h2>Subheading</h2>
```

c. Links (<a>):

The **<a>** element creates hyperlinks to other web pages, files, or locations within the same page.

Example:

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

d. Images ():

The **** element embeds images into web pages.

Example:

```

```

e. Lists (, ,):

Lists are used to group related items together. There are two types of lists: unordered lists (<**ul**>) and ordered lists (<**ol**>). Each item in a list is defined using the <**li**> element.

Example (Unordered List):

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
```

Example (Ordered List):

```
<ol>
  <li>First Item</li>
  <li>Second Item</li>
</ol>
```

f. Divisions (<div>):

The **<div>** element is a generic container used to group and style other HTML elements.

Example:

```
<div>
  <p>This is inside a div element.</p>
```

</div>

3. Self-Closing Tags:

Some HTML elements do not require closing tags and are self-closing. These elements are written with a single tag.

Example (Line Break):

```
<p>This is the first line.<br>This is the second line.</p>
```

HTML Attributes

HTML attributes provide additional information about HTML elements and are used to modify their behavior or appearance. Attributes are defined within the opening tag of an element and are written as name/value pairs, separated by an equals sign (=). Here's an overview of HTML attributes along with examples:

1. Common Attributes:

a. **id** Attribute:

The **id** attribute specifies a unique identifier for an HTML element, which can be used to target the element with CSS or JavaScript.

Example:

```
<div id="main-content">  
  <!-- Content goes here -->  
</div>
```

b. **class** Attribute:

The **class** attribute specifies one or more class names for an HTML element, which can be used to apply styles or target multiple elements with CSS or JavaScript.

Example:

```
<p class="highlighted">This is a highlighted paragraph.</p>
```

c. **style** Attribute:

The **style** attribute specifies inline CSS styles for an HTML element, allowing you to apply specific styles directly to the element.

Example:

```
<p style="color: blue; font-size: 16px;">This is styled text.</p>
```

2. Attributes for Links:

a. **href** Attribute:

The **href** attribute specifies the URL (web address) to which the link points.

Example:

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

b. **target** Attribute:

The **target** attribute specifies where to open the linked document. It can be set to **_blank** to open the link in a new tab or window.

Example:

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

3. Attributes for Images:

a. **src** Attribute:

The **src** attribute specifies the URL of the image to be displayed.

Example:

```

```

b. **alt** Attribute:

The **alt** attribute provides alternative text for an image, which is displayed if the image cannot be loaded or for accessibility purposes.

Example:

```

```

HTML Headings

HTML headings are used to define the titles or subtitles of sections within a web page. Headings provide structure and hierarchy to the content, making it easier for users and search engines to understand the organization of the page. HTML provides six levels of headings, ranging from **<h1>** (most important) to **<h6>** (least important). Here's an overview of HTML headings along with examples:

1. Heading Levels:

a. **<h1>** to **<h6>**:

HTML provides six levels of headings, where **<h1>** is the highest level (main heading) and **<h6>** is the lowest level (subheading). Headings should be used to represent the structure of the content, with **<h1>** typically used for the main title of the page and subsequent headings used to denote subsections.

Example:


```
<h1>Main Heading</h1>
<h2>Subheading Level 2</h2>
<h3>Subheading Level 3</h3>
```

HTML Paragraphs

HTML paragraphs are used to group blocks of text content together within a web page. They provide structure and organization to textual content, making it easier for users to read and understand. Paragraphs are defined using the **<p>** element in HTML. Here's an overview of HTML paragraphs along with examples:

1. <p> Element:

The **<p>** element is used to define a paragraph of text content. It represents a block-level element, which means it starts on a new line and takes up the full width available.

2. Syntax:

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

3. Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Paragraphs Example</title>
</head>
<body>
  <p>This is the first paragraph of text.</p>
  <p>This is the second paragraph of text.</p>
</body>
</html>
```

4. Attributes:

The `<p>` element does not have any specific attributes. However, it can be styled using CSS or have other attributes added for specific functionality.

Adding Styles and its Types

Adding styles to HTML elements enhances the appearance and layout of web pages. CSS (Cascading Style Sheets) is used to apply styles to HTML elements, and there are various methods to add styles, including inline styles, internal styles, and external stylesheets.

Formatting

Formatting in HTML refers to the visual presentation and arrangement of content on a web page. HTML provides several elements and attributes that allow you to control the formatting of text and other content. Here's an overview of formatting options in HTML along with examples:

1. Text Formatting Elements:

a. Bold Text:

The `` element is used to make text bold.

Example:

```
<p>This is <b>bold</b> text.</p>
```

b. Italic Text:

The `<i>` element is used to make text italic.

Example:

```
<p>This is <i>italic</i> text.</p>
```

c. Underlined Text:

The `<u>` element is used to underline text.

Example:

```
<p>This is <u>underlined</u> text.</p>
```

d. Strong Text:

The `` element is used to emphasize text, typically displayed in bold.

Example:

```
<p>This is <strong>strong</strong> text.</p>
```

e. Emphasized Text:

The `` element is used to emphasize text, typically displayed in italics.

Example:

```
<p>This is <em>emphasized</em> text.</p>
```

Comments

HTML comments are used to add notes or explanations within the HTML code that are not displayed in the web browser. They are useful for providing context, documenting code, or temporarily disabling sections of code. HTML comments are enclosed within `<!--` and `-->` tags. Here's an overview of how to use comments in HTML along with examples:

1. Syntax:

HTML comments are written using the `<!--` opening tag and the `-->` closing tag. Everything between these tags is treated as a comment and is ignored by the web browser.

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

2. Example:

HTML comments can be added anywhere within the HTML document, including within the **<head>** and **<body>** sections.

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Comments Example</title>
</head>
<body>
  <!-- This is a comment -->
  <h1>Welcome to my website</h1>
  <!--
  This is a multi-line comment.
  It can span across multiple lines.
  -->
  <p>This is the main content of the page.</p>
</body>
</html>
```

Colors

In HTML, colors can be specified using several methods, including color names, hexadecimal color codes, RGB values, RGBA values, HSL values, and HSLA values. Colors are commonly used to style text, backgrounds, borders, and other elements on a web page. Here's an overview of how to specify colors in HTML along with examples for each method:

1. Color Names:

HTML provides a set of predefined color names that can be used to specify colors. These color names include common colors such as "red," "blue," "green," etc.

Example:

```
<p style="color: red;">This text is red.</p>
```

2. Hexadecimal Color Codes:

Hexadecimal color codes represent colors using a combination of six hexadecimal digits (0-9 and A-F) preceded by a pound sign (#). Each pair of digits represents the red, green, and blue (RGB) components of the color.

Example:

```
<p style="color: #FF0000;">This text is red.</p>
```

3. RGB Values:

RGB values specify colors using three integers representing the red, green, and blue components of the color, each ranging from 0 to 255.

Example:

```
<p style="color: rgb(255, 0, 0);">This text is red.</p>
```

Cascading Style Sheet (CSS)

CSS (Cascading Style Sheets) is a style sheet language used to describe the presentation of a document written in HTML or XML. CSS controls the layout, formatting, and appearance of elements on a web page, allowing designers and developers to customize the look and feel of their websites.

Links

In HTML, links are used to navigate between different web pages or resources. They are created using the `<a>` (anchor) element and can point to URLs, files, sections within the same page (using anchors), or email addresses. Here's an overview of how to create links in HTML along with examples:

1. Basic Link Syntax:

The `<a>` element is used to create links in HTML. It has an **href** attribute that specifies the destination URL or resource.

Example:

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

2. Relative URLs:

Relative URLs are URLs that are relative to the current page's location. They are useful for linking to pages within the same website or directory structure.

Example (Relative URL):

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

3. Linking to Sections within the Same Page:

Links can be used to navigate to specific sections within the same page by using anchor tags (`<a>` elements) with corresponding IDs on the target sections.

Example:

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

...

```
<h2 id="section1">Section 1</h2>
```

```
<p>This is the content of Section 1.</p>
```

4. Linking to Email Addresses:

Links can be used to create email links that open the user's default email client with a pre-filled email address.

Example (Email Link):

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

Images

In HTML, images are displayed using the **** (image) element. Images enhance the visual appeal of web pages and are commonly used to convey information or complement textual content. Here's an overview of how to insert images in HTML along with examples:

1. Basic Image Syntax:

The **** element is used to insert images into HTML documents. It is a self-closing tag and does not require a closing tag. The **src** attribute specifies the URL (source) of the image, while the **alt** attribute provides alternative text for the image, which is displayed if the image cannot be loaded or for accessibility purposes.

Example:

```

```

HTML Tables

In HTML, tables are used to organize and display data in rows and columns. Tables consist of one or more rows (**<tr>** elements) containing table data cells (**<td>** elements) and optionally header cells (**<th>** elements) for column or row headings. Here's an overview of how to create tables in HTML along with examples:

1. Basic Table Structure:

The basic structure of an HTML table consists of the **<table>** element, which contains one or more rows (**<tr>** elements), each row containing one or more data cells (**<td>** elements) or header cells (**<th>** elements).

Example:

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

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

2. Table Headers:

Header cells (<th> elements) are used to define column or row headings within a table. They are typically displayed in bold and centered by default.

Example (With Row Headers):

```
<table>
  <tr>
    <th>Month</th>
    <th>Sales</th>
    <th>Expenses</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$1000</td>
    <td>$500</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$1500</td>
    <td>$600</td>
  </tr>
</table>
```


3. Table Data:

Data cells (**<td>** elements) contain the actual data within the table. They are displayed with normal font weight and alignment by default.

Example:

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

4. Table Borders:

Table borders can be styled using CSS to define the appearance of the table, including border width, color, and style.

Example (With Border Styling):

```
<style>
  table {
    border-collapse: collapse;
  }
  th, td {
    border: 1px solid black;
    padding: 8px;
  }
</style>
<table>
  <!-- Table content goes here -->
</table>
```

5. Spanning Rows and Columns:

You can span multiple rows or columns within a table using the **rowspan** and **colspan** attributes on `<td>` or `<th>` elements.

Example (Spanning Rows):

```
<table>
  <tr>
    <td rowspan="2">Row 1</td>
    <td>Cell 1</td>
    <td>Cell 2</td>
  </tr>
  <tr>
    <td>Cell 3</td>
    <td>Cell 4</td>
  </tr>
</table>
```

HTML Lists

In HTML, lists are used to organize and display content in a structured format. There are three main types of lists in HTML: ordered lists (``), unordered lists (``), and definition lists (`<dl>`). Here's an overview of how to create lists in HTML along with examples for each type:

1. Ordered Lists (``):

Ordered lists are used to create lists where each item is numbered or ordered in a sequential manner. Each list item is represented by the `` (list item) element.

Example:

```
<ol>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ol>
```

2. Unordered Lists ():

Unordered lists are used to create lists where each item is bulleted or unordered. Each list item is represented by the (list item) element.

Example:

```
<ul>
  <li>Red</li>
  <li>Green</li>
  <li>Blue</li>
</ul>
```

Div

In HTML, the <div> element is a block-level container used to group and organize content on a web page. It is a versatile and widely used element in HTML and is commonly used with CSS for styling and layout purposes. Here's an overview of the <div> element along with examples of how it can be used:

1. Basic <div> Syntax:

The <div> element is an empty container that does not have any specific meaning or styling by default. It is typically used with CSS to apply styles or define layout structures.

Example:

```
<div>This is a div element.</div>
```

2. Grouping Content:

One of the primary purposes of the <div> element is to group related content together. It allows you to create logical sections within your HTML document.

Example:

```
<div>
  <h2>Section 1</h2>
```

```
<p>This is the content of section 1.</p>
</div>
<div>
  <h2>Section 2</h2>
  <p>This is the content of section 2.</p>
</div>
```

Classes and ID

Classes and IDs are attributes used to uniquely identify or categorize elements, allowing them to be targeted and styled with CSS or manipulated with JavaScript. Both classes and IDs are used to apply styles or functionality to specific elements on a web page. Here's an overview of classes and IDs in HTML along with examples:

1. Classes:

Classes are used to categorize multiple elements and apply the same styles to them. Multiple elements can have the same class, and an element can have multiple classes separated by spaces.

Syntax:

```
<element class="class1 class2 ..."></element>
```

Example:

```
<p class="highlight">This paragraph has a highlight class.</p>
<p class="highlight">So does this one.</p>
```

2. IDs:

IDs are used to uniquely identify a single element on a web page. Each element can have only one ID, and IDs should be unique within the entire HTML document.

Syntax:

```
<element id="uniqueID"></element>
```

Example:

```
<div id="header">This is the header.</div>
```

3. Applying Styles with Classes and IDs:

Classes and IDs can be used to apply CSS styles to elements. Classes are typically used for styling multiple elements with the same styles, while IDs are used for uniquely styling individual elements.

Example (Applying Styles with Classes):

```
<style>
  .highlight {
    background-color: yellow;
  }
</style>
<p class="highlight">This paragraph has a highlight class.</p>
<p class="highlight">So does this one.</p>
```

Example (Applying Styles with IDs):

```
<style>
  #header {
    background-color: lightblue;
    padding: 10px;
  }
</style>
<div id="header">This is the header.</div>
```

HTML JavaScript

HTML and JavaScript are two distinct languages commonly used together in web development to create interactive and dynamic web pages. While HTML (HyperText Markup Language) is used for structuring and presenting content, JavaScript is a scripting language used for adding interactivity, behavior, and dynamic features to web pages. Here's an overview of HTML and JavaScript together, along with examples of how they can be used:

1. Embedding JavaScript in HTML:

JavaScript code can be embedded directly within HTML documents using the `<script>` element. You can include JavaScript code either inline within the HTML document or by referencing external JavaScript files.

Example (Inline JavaScript):

```
<!DOCTYPE html>
<html>
<head>
  <title>Inline JavaScript Example</title>
</head>
<body>
  <h1>JavaScript Example</h1>
  <p>This is a paragraph.</p>

  <script>
    alert('Hello, World!');
  </script>
</body>
</html>
```

Example (External JavaScript File):

```
<!DOCTYPE html>
<html>
<head>
  <title>External JavaScript Example</title>
```

```
<script src="script.js"></script>
</head>
<body>
  <h1>JavaScript Example</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

script.js:

```
alert('Hello, World!');
```

2. Handling Events:

JavaScript is commonly used to handle various events triggered by user interactions (e.g., clicks, mouse movements, keypresses) or changes in the web page (e.g., page load, form submission).

Example (Button Click Event):

```
<button onclick="myFunction()">Click me</button>
<script>
function myFunction() {
  alert('Button clicked!');
}
</script>
```

3. DOM Manipulation:

JavaScript is used to manipulate the Document Object Model (DOM) of an HTML document dynamically, allowing you to add, remove, or modify elements and their attributes.

Example (Changing Text Content):

```
<p id="demo">This is a paragraph.</p>
<button onclick="changeText()">Change Text</button>
```

```
<script>
function changeText() {
    document.getElementById('demo').innerHTML = 'Text changed!';
}
</script>
```

Layouts

HTML has several semantic elements that define the different parts of a web page:

- `<header>` - Defines a header for a document or a section
- `<nav>` - Defines a set of navigation links
- `<section>` - Defines a section in a document
- `<article>` - Defines an independent, self-contained content
- `<aside>` - Defines content aside from the content (like a sidebar)
- `<footer>` - Defines a footer for a document or a section
- `<details>` - Defines additional details that the user can open and close on demand
- `<summary>` - Defines a heading for the `<details>` element

There are four different techniques to create multicolumn layouts. Each technique has its pros and cons:

- CSS framework
- CSS float property
- CSS flexbox
- CSS grid

Responsive

Responsive web design is about creating web pages that look good on all devices. A responsive web design will automatically adjust for different screen sizes and viewports.

Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones).

Setting The Viewport

To create a responsive website, add the following <meta> tag to all your web pages.

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