HTML

What is HTML?

HTML stands for Hyper Text Markup Language

HTML is the standard markup language for creating Web pages

HTML describes the structure of a Web page

HTML consists of a series of elements

HTML elements tell the browser how to display the content

HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

Why HTML is called as Hyper Text Markup Language?

HTML, which stands for HyperText Markup Language, is named for its key features and functions.

The term "hyper" refers to the ability of HTML to link or connect documents to one another through hyperlinks. Hyperlinks allow users to navigate from one piece of content to another, creating a non-linear way of accessing information. This linking capability is fundamental to the concept of the World Wide Web.

HTML is a markup language because it uses tags to mark or define elements within a document. These tags provide instructions on how elements should be displayed or formatted. The markup tags indicate the structure of the document, such as headings, paragraphs, lists, images, and more.So, the name "HyperText Markup Language" reflects the language's ability to create hyperlinked documents and its use of markup tags to structure and format the content within those documents.

What are HTML Elements?

Elements are the building blocks of HTML that describe the structure and content of a web page. They are the “Markup” part of HyperText Markup Language (HTML).

HTML syntax uses the angle brackets (”<” and ”>”) to hold the name of an HTML element. Elements usually have an opening tag and a closing tag, and

give information about the content they contain. The difference between the two is that the closing tag has a forward slash.

p Element :

The tag stands for paragraph, which is the most common tag used to create lines of text inside an HTML document.

The use of the is compatible with other tags, allowing to add hyperlinks (<a>) and bold () text, among other things.

You can also nest an anchor element <a> within a paragraph.

Example :

My Name is XYZ

Heading Element :

There are six heading elements — <h1>, <h2>, <h3>, <h4>, <h5>, <h6>.

Heading elements are used to signify the importance of the content below them. The lower the number of the heading, the higher the importance.

For example, the <h1> element represents the main heading of the page, and there should only be one per page. This helps search engines understand the main topic of the page. <h2> elements have less importance, and should be below the higher level <h1> element.

Example :

<h1>This is main heading.</h1>

<h2>This is subheading h2.</h2>

<h3>This is subheading h3.</h3>

<h4>This is subheading h4.</h4>

<h5>This is subheading h5.</h5>

<h6>This is subheading h6.</h6>

a Element : The anchor (<a>) element creates a hyperlink to another page or file. In order to link to a different page or file the <a> tag must also contain a href attribute, which indicates the link's destination.

The text between the opening and closing <a> tags becomes the link. This text should be a meaningful description of the link destination, and not a generic phrase such as "Click here". This better enables users with screen readers to navigate among the various links on a page and understand what content each one will link to.

By default, a linked page is displayed in the current browser window unless another target is specified. The default link styles are as follows:

An unvisited link is underlined and blue

A visited link is underlined and purple

An active link is underlined and red

Examples :

freeCodeCamp

You can also create a link to another section on the same page:

<h1 id="top"></h1>

Go to top

An image can also be turned into a link by enclosing the <img> tag in an <a> tag:

<img src="logo.svg">

Meta Tag :

The <meta> tag provides the metadata about the HTML document.

This metadata is used to specify a page’s charset, description, keywords, the author, and the viewport of the page.

This metadata will not appear on the website itself, but it will be used by the browers and search engines to determine how the page will display content and assess search engine optimization (SEO).

Example :

```
<head>
 <meta charset="UTF-8">
<meta name="description" content="Short description of website content here">
 <meta name="keywords" content="HTML,CSS,XML,JavaScript">
 <meta name="author" content="Jane Smith">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
<!-- HTML5 introduced a method to let web designers take control over the viewport, through the <meta> tag. The viewport is the user's visible area of a web page. A <meta> viewport element gives the browser instructions on how to control the page's dimensions and scaling. -->
</head>
```

HTML File Elements :

HTML (Hypertext Markup Language) file elements are the building blocks used to create the structure and content of a web page. HTML is composed of various tags, and each tag represents a different element on the page. Here are some common HTML file elements:

<!DOCTYPE html>: Defines the document type and version of HTML being used. It should be placed at the beginning of an HTML document.

<html>: The root element that wraps all the content on the page.

<head>: Contains meta-information about the HTML document, such as the title, character set, linked stylesheets, and scripts.

<meta>: Provides metadata about the HTML document, such as the character set, viewport settings, and more.

<title>: Sets the title of the HTML document, which appears in the browser's title bar or tab.

<link>: Links external resources, such as stylesheets or icons.

<style>: Contains CSS (Cascading Style Sheets) code to define the style of the document.

<script>: Includes or references JavaScript code.

<body>: Contains the main content of the HTML document.

<header>: Represents the header section of a document or a section, typically containing headings, logos, and navigation.

<nav>: Defines a container for navigation links.

<main>: Contains the main content of the HTML document.

<section>: Defines a section in a document.

<article>: Represents an independent piece of content that can be distributed and reused.

<aside>: Represents content that is tangentially related to the content around it, like a sidebar.

<footer>: Represents the footer section of a document or a section, typically containing copyright information, links, and other metadata.

<p>: Defines a paragraph.

<h1> to <h6>: Headings, with <h1> being the highest level and <h6> the lowest.

: Defines an unordered (bulleted) list.

: Defines an ordered (numbered) list.

: Represents a list item in a list.

<a>: Defines a hyperlink.

<img>: Embeds an image.

<div>: Defines a generic container that can be used for layout or grouping content.

<span>: Defines a generic container that doesn't affect the layout.

These are just some of the most commonly used HTML elements. The combination of these elements allows web developers to create a structured and semantically meaningful document.

Attributes :

Attributes in HTML provide additional information about HTML elements. They are always included in the opening tag of an element and are usually in the form of name/value pairs. The presence of attributes can modify the behavior or appearance of an HTML element. Here's a breakdown of how attributes work in HTML:

Syntax:

Attributes are added to an HTML element within the opening tag.

The syntax is typically name="value".

Common Attributes:

id: Specifies a unique identifier for an HTML element. The value must be unique within the HTML document.

<div id="uniqueID">Content goes here</div>

class: Specifies one or more class names for an element. Used to apply styles or select elements with JavaScript or CSS.

<p class="important-text">This is important!</p>

style: Defines inline CSS styles for an element.

Styled Text

src: Specifies the source URL of external resources, such as images, scripts, or iframes.

<img src="image.jpg" alt="Description">

href: Specifies the URL of a linked resource, often used with <a> (anchor) and <link> elements.

Visit Example

alt: Provides alternative text for images, displayed if the image cannot be loaded.

<img src="image.jpg" alt="Description of the image">

width and height: Specifies the width and height of an element, such as an image or table cell.

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

Boolean Attributes:

Some attributes don't require a value and are considered boolean. Their presence implies a true value, and their absence implies false.

<input type="checkbox" checked>

<input type="text" disabled>

Custom Data Attributes:

Developers can create custom data attributes prefixed with data- to store extra information.

User Details

Event Attributes:

Used to attach JavaScript code to specific events.

<button onclick="myFunction()">Click me</button>

Deprecated Attributes:

Some attributes have been deprecated in HTML5 and should be avoided.

Example: align attribute for text alignment.

Attributes play a crucial role in enhancing the functionality and appearance of HTML elements. It's essential to use attributes appropriately and follow best practices to create well-structured and accessible web content.

Absolute And Relative Paths :

In HTML, paths are used to specify the location of resources such as images, stylesheets, scripts, or links to other web pages. There are two main types of paths: absolute paths and relative paths.

- **Absolute Path:**
    - An absolute path provides the complete URL or file system path to the resource, starting from the root directory.
    - It includes the full web address or file system path, making it independent of the location of the HTML document.
    - Example of an absolute path to an image on the web:

 alt "An example image"

    - Example of an absolute path to an image on the local file system:

 alt "An example image"

- **Relative Path:**
    - A relative path specifies the location of the resource in relation to the location of the HTML document.
    - Relative paths are shorter and often more flexible because they adapt to the file structure.
    - Relative paths do not include the protocol (http/https) or the domain for web resources.
    - There are different ways to define relative paths:
        - **Relative to the current document:**

```
<img src="images/image.jpg"/> alt "An example image"
```

        - **Relative to the root directory:**

```
<img src="/images/image.jpg"/> alt "An example image"
```

        - **Relative to the parent directory:**

```
<img src="../images/image.jpg"/>  alt "An example image"
```

Choosing between absolute and relative paths depends on the specific use case and the organization of your website's files. Absolute paths are useful when linking to external resources or when the file structure might change, while relative paths are often more convenient for files within the same website or project.

List Elements :

There are three types of lists in HTML: unordered lists (**<ul>**), ordered lists (**<ol>**), and descriptive lists (**<dl>**).

Unordered list :

The unordered list starts with <ul> tag and list items start with the <li> tag. In unordered lists all the list items marked with bullets by default.

Example :

```
<ul>
  <li>Item</li>
  <li>Item</li>
  <li>Item</li>
</ul>
```

Output:

Item
Item
Item

Ordered list :
The ordered list starts with <ol> tag and list items start with the <li> tag. In ordered lists all the list items are marked with numbers.
Example :

```
<ol type=”1”>
  <li>First Item</li>
  <li>Second Item</li>
  <li>Third Item</li>
</ol>
```

Output:
1. First Item
2.Second Item
3.Third Item
Descriptive List :

A descriptive list is created using the **<dl>** element, which stands for "description list." It typically consists of a series of term-description pairs. Each term is represented by a **<dt>** (description term) element, and each description is represented by a **<dd>** (description definition) element.

Here's an example of a simple descriptive list in HTML:

```
<dl>
  <dt>Term 1</dt>
  <dd>Description 1</dd>
  <dt>Term 2</dt>
  <dd>Description 2</dd>
   <!-- Additional term-description pairs can be added as needed -->
</dl>
```

In this example:

- **<dl>**: This is the container element for the descriptive list.
- **<dt>**: Stands for "description term." It is used to define the terms in the list.
- **<dd>**: Stands for "description definition." It is used to provide the corresponding descriptions for each term.

Here's a more practical example related to a list of programming languages:

```
<dl>
```

<dt>HTML</dt>

<dd>HyperText Markup Language</dd>

<dt>CSS</dt>

<dd>Cascading Style Sheets</dd>

<dt>JavaScript</dt>

<dd>A scripting language for web development</dd>

</dl>

Formatting Elements:

In HTML, formatting elements are used to control the visual presentation of text and content on a web page. These elements allow web developers to apply various styles, such as bold or italic text, headings, paragraphs, and line breaks. Here are some common formatting elements in HTML:

- **Headings (`<h1>` to `<h6>`):**
    - HTML provides six levels of headings, **`<h1>`** being the highest (most important) and **`<h6>`** being the lowest. Headings are used to structure content hierarchically.

Example :

<h1>This is Heading 1</h1>

<h2>This is Heading 2</h2>

<!-- ... -->

<h6>This is Heading 6</h6>

Paragraphs (`<p>`):

- The **`<p>`** element is used to define paragraphs of text.
- Example :

This is a paragraph of text.

Bold (`<b>` and `<strong>`):

- The **`<b>`** element is used for bold text, but it doesn't carry any semantic meaning.
- The **`<strong>`** element is used for strong importance, and browsers typically render it as bold.
- Example :

This is bold text.

This is strongly emphasized text.

Italic (`<i>` and `<em>`):

- The **`<i>`** element is used for italic text, but like **`<b>`**, it doesn't convey specific semantic meaning.
- The **`<em>`** element is used for emphasized text, and browsers typically render it as italic.

Example :

This is italic text.

This is emphasized text.

Underline (`<u>`):

- The **`<u>`** element was traditionally used for underlined text, but it's not widely recommended in modern HTML and CSS. Instead, CSS is typically used for underlining.

Example :

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

Strikethrough (`<s>` and `<del>`):

- The **`<s>`** element is used for strikethrough text.
- The **`<del>`** element represents deleted text, and browsers often render it with a strikethrough.

Example :

<s>This text has a strikethrough.</s>

~~This text is deleted.~~

Superscript (`<sup>`) and Subscript (`<sub>`):

- The **`<sup>`** element is used for superscript text (e.g., exponent in mathematical expressions).
- The **`<sub>`** element is used for subscript text (e.g., chemical formulas).

Example :

x² + y² = r² -->x^2+y^2=r^2

H₂O

Line Break (`<br>`):

- The **`<br>`** element is used to insert a line break within text.

Example :

This is some text.
It continues on a new line.

Strong Element (`<strong>`):

- The **`<strong>`** element is used to give strong importance to the enclosed text.
- Browsers typically render the content inside **`<strong>`** as bold, but its primary purpose is to convey importance rather than visual styling.

Example :

```
<p>This is <strong>important</strong> information.</p>
```

Emphasis Tag (`<em>`):

- The **`<em>`** element is used to emphasize text. Browsers often render the content inside **`<em>`** as italic.
- Unlike **`<i>`**, **`<em>`** is semantically meaningful, indicating emphasis.

Example :

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

Mark Tag (`<mark>`):

- The **`<mark>`** element is used to highlight parts of text for reference or to draw attention.
- Browsers typically render the content inside **`<mark>`** with a background color.

Example :

```
<p>Search for <mark>relevant</mark> keywords.</p>
```

Horizontal Rule Tag (`<hr>`):

- The **`<hr>`** element is used to create a horizontal rule or line, often to separate sections of content.

Example :

```
<p>This is some content.</p>

<hr>

<p>This is a new section.</p>
```

Small Tag (`<small>`):

- The **`<small>`** element is used to decrease the text size, indicating that the text is small print or fine print.

Example :

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

Insert Tag (`<ins>`):

- The **<ins>** element represents inserted text, indicating content that has been added to the document.

Example :

```
<p>This is <ins>new</ins> content.</p>
```

Bidirectional Override Tag (<bdo>):

- The **<bdo>** element is used to override the text directionality defined by the surrounding content.

Example :

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

<bdo dir="rtl">This text is right-to-left.</bdo>
```