Introduction to Web Development

Trainer's Name

Mukhtar Mahamed

Learning Objectives

By the end of this session, you should be able to:

1. Understand the Web Development Technologies.
2. Define key Web Terminologies.
3. Install required Tools and Software's
4. Understand Basics of HTML
5. Understand the Basic HTML Document Structure.
6. Get familiar with the Elements of HTML
7. Write Your first HTML Page.

What is a Web

What is a Web?

The Web is the common name for the **World Wide Web**, a subset of the Internet consisting of the pages that can be accessed by a Web browser.

- The Web is the most popular way of accessing the Internet,

Web Terminologies ...

What is a Web Page?

A document which can be displayed in a web browser such as Firefox, Google Chrome etc. written in HTML, often called just "pages.“

What is Website?

A collection of linked web pages which are grouped together and usually connected in various ways. Often called a "web site" or simply a "site.“.

What is Web Server?

A computer that hosts a website on the Internet.

What is Web Development?

Web development refers to building, creating, and an maintaining websites. It includes aspects such as web design, web publishing, web programming, and database management.

Web Terminologies Continues…

What is Domain name?

A domain name is a unique name (e.g. **google.com**) used to identify the location of a website on the Internet.

What is DNS?

The Domain Name System (or DNS) converts human readable domain names (like: **www.google.com**) into Internet Protocol (IP) addresses (like **173.194.39.78**).

HTTP or HTTPS?

The Hypertext Transfer Protocol (**HTTP**) is the foundation of World Wide Web and is used to load web pages using hypertext links.

A typical flow over **HTTP** involves a client machine making a request to a server, which then sends a response message

What is URL?

A Uniform Resource Locator (**URL**) is the file address of a resource on the Internet. A URL can represent a web page, an image, a video, a style sheet, and much more.

3 Pillars of the Web

HTML

The **'HTML'** part contains all the content, organized into a logical structure. This is the part that an author might be most concerned with: the words, chapter headings, figures, diagrams, etc.

CSS

The **'CSS'** part (version 3 being current) is all about the presentation or style of the page.

JavaScript

The **'JavaScript'**, or **'JS'** for short, part is about the actions a page can take such as interaction with the user and customizing and changing the page according to any number of parameters.

Installing Required Tools/Software

Basic Tools for Web Developers

Code Editor

A text editor that is specialized for writing software.

Most Used Code Editors.

- Visual Studio Code (Recommended)
- Sublime Text
- Atom
- Notepad++

Modern Web Browser

to test our code we need a modern Browser.

Currently, the most-used browsers are Firefox, Chrome, Opera, Safari.

Recommended Browsers are:-

- Chrome (**Recommended**)
- Firefox

Introduction To **HTML**

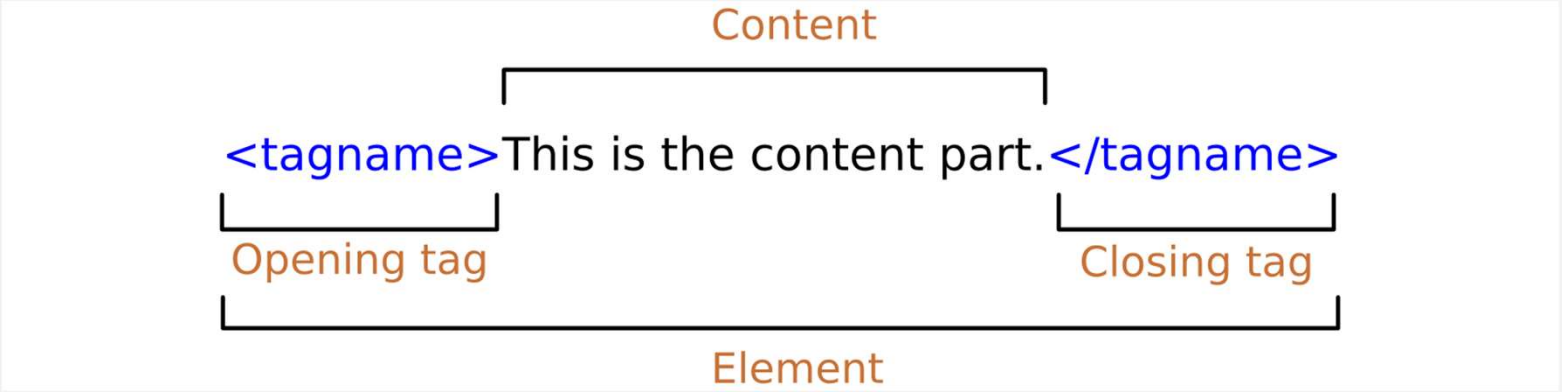
What is HTML?

- **HTML** (**H**yper**t**ext **M**arkup **L**anguage) is the code that is used to create a web page and its content.
- **HTML** is not a programming language; it is a markup language that defines the structure of your content.
- **HTML** consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way or act a certain way.

```
<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
```

HTML Tags

- **HTML tags** are like keywords which defines that how web browser will format and display the content.
- **HTML tags** are used to create HTML documents and render their properties.



HTML Elements

- An **HTML** element is defined by a **start tag**, some content, and an **end tag**.

```
<h1>this is the content </h1>
```

- Some HTML elements have no content **
** element
  - These elements are called empty elements.
  - Empty Elements do not have end tag

- Elements can be placed within other elements. This is called nesting.

```
<div>
    <h1>this is the content </h1>
    <p>this is paragraph </p>
</div>
```

HTML Basic Structure

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>title</title>
</head>
<body>
  <div>
      <h1>this is the content </h1>
      <p>this is paragraph </p>
  </div>
</body>
</html>
```

HTML File extensions are **.html** or **.htm**

What is in the Head

- The HTML Head Element contain metadata about the document
- content in the head are not displayed on the page.
  - The **<title>** Element
    - Title is metadata that represents the title of the overall HTML document
    - The <title> contents are also used in search results, as you'll see below.
  - the **<meta>** element
    - Many **<meta>** elements include **name** and **content** attributes:

```
<meta name="author" content="Mukhtar Mahamed">
```

  - Other Elements in Head <script>, <link>, <style> and <base>

Character encoding

- Character encoding is a method of converting bytes into characters.
- To display an HTML page correctly, a web browser must know which character set to use.
- Use The **<meta>** tag to define character encoding
- The **<meta>** declaration belongs inside the **<head>** element

```
<head>
  <meta charset="UTF-8">
</head>
```

HTML comments

- The purpose of comments is to allow you to include notes in the code to explain your logic or coding.
- Browsers ignore comments, effectively making comments invisible to the user
- Comment syntax <!-- Write your comments here -->

```
<!-- <p>I am comment</p> -->
<p>I am paragraph</p>
```

Attributes

- Attributes contain extra information about the element that won't appear in the content.
- Attributes in HTML are written inside the opening tag like this:

An attribute should have:

- Attribute name followed by equal sign
- Attribute value

```
<a href="google.com">visit google</a>
```

```
<p id="par-1" class="paragraph">
   this is paragraph
</p>
```

Global Attributes

- The global attributes are attributes that can be used with all HTML elements.

- Global Attributes – most used ones
  - **Class** - Assigns a class name or space-separated list of class names to an element.
  - **Id** - Specifies a unique identifier (ID) for an element
  - **Lang** - Specifies the primary language for the element's text content.
  - **Style** - Specifies inline style information for an element.
  - **Title** - Provides advisory information related to the element.

HTML Headings

- Headings help in defining the hierarchy and the structure of the web page content.
- HTML offers six levels of heading tags, **<h1>** through **<h6>**

```
<h1>Heading level 1</h1>
<h2>Heading level 2</h2>
<h3>Heading level 3</h3>
<h4>Heading level 4</h4>
<h5>Heading level 5</h5>
<h6>Heading level 6</h6>
```

- Use the **<h1>** tag to mark the most important heading which is usually at the top of the page.

HTML Paragraphs

- Paragraph element is used to publish text on the web pages.
- Paragraphs are defined with the **<p>** tag.

```
<p>This is a paragraph 1</p>
<p>This is another paragraph 2</p>
```

- A paragraph always starts on a new line.
- to create line break in paragraphs use **
** tag

Nesting elements

- Elements can be placed within other elements. This is called nesting.
- There is a right and wrong way to do nesting.

```
<div>
    <h1>this is the content </h1>
    <p>this is paragraph </p>
</div>
```

Lists

- HTML lists allow web developers to group a set of related items in lists.
- There are 3 types of Lists in HTML Elements
  - Ordered Lists - ** **
    - for wrapping, <li> for items
  - Unordered Lists - ** **
    - for wrapping, <li> for items
  - Description Lists - **<**dl> <dt> <dd>
    - The <dl> defines the description list, the <dt> defines the term (name), and the <dd> tag describes each term:

Special characters in HTML

- In HTML, the characters **<**, **>**,**"**,**'** and **&** are special characters.
- They are parts of the HTML syntax itself.
- So how do you include one of these special characters in your text?
  - Use character references
  - These are special codes that represent characters.

Literal Character	Character Reference
<	<
>	>
"	"
'	'
&	&amp

Emphasis and importance

- we often want to mark certain words as important or different in some way.
- HTML provides various semantic elements to allow us to mark up textual content with such effects.

Emphasis

- In HTML we use the **** to indicate Emphasis.

Strong importance

- In HTML we use the **** element to mark up important part of sentence.

- **Italic, bold, underline** Elements.
  - **<i>** for Italic
  - **<b>** for Bold
  - **<u>** for underline

What is a hyperlink?

- Hyperlinks are most important parts of the Web.
- Hyperlinks allow us to link documents to other documents or resources
- Hyper Links are what makes the Web a web

- **structure of a link**
  - To create Link, we use **<a>** element. Also known as **Anchor Tag**
  - Add **href** attribute that has web address as value.

```
<a href="google.com">visit google</a>
```

- **Title** attribute can also be added to give additional information

Document fragments

- It's possible to link to a specific part of an HTML document, known as a **document fragment.**
- To do this you first have to assign an **id attribute** to the element you want to link to.

```
<p id="contact">Tell: 123456</p>
```

- Then to link to that specific id, you'd include it <a> href attribute preceded by **#** symbol

```
<a href="#contact">contact us</a>
```

E-mail links

- It's possible to create links or buttons that, when clicked, open a new outgoing email message rather than linking to a resource or page.
- This is done using the **<a>** element and the mailto: URL scheme.

```
<a href="mailto:example@gmail.com">Send email to Us</a>
```

- When Link is clicked, the computer will open the default mail app.

A quick Intro on URLs and paths

- To fully understand link targets, you need to understand URLs and file paths.
- A **URL**, or Uniform Resource Locator is simply a string of text that defines where something is located on the Web
- **URLs** use **paths** to find files. **Paths** specify where the **file** you're interested in is in the filesystem

- **Absolute** versus **relative** URLs
  - **absolute URL**: Points to a location defined by its absolute location on the web
  - **relative URL**: Points to a location that is relative to current location.

Block versus inline elements

There are two important categories of elements to know in HTML:

1. **block-level elements**
   - Block-level elements form a visible block on a page
2. **inline elements**
   - An inline element does not start on a new line and it only takes up as much width as necessary.

Quotations

- There are two ways to create Quotations in HTML.
  - **Blockquotes**
    - To create block quote use **<blockquote>** element with **cite** attribute pointing to the source of the quote.
  - **Inline quotations**
    - To create Inline quote use the **<q>** element
    - With cite attribute pointing to source.

Marking up contact details

- HTML has an element for marking up contact details — **<address>.** This simply wraps around your contact details.

```
<address>
  <p>
    <strong>Shaqodoon Office (HQ)</strong><br>
      Shacabka<br>
      Hargeisa, Somaliland<br>
  </p>

  <ul>
    <li>Tel: +252 2 515 777</li>
    <li>Email: <a href="mailto:info@shaqodoon.org">info@shaqodoon.org</a></li>
  </ul>
</address>
```

Superscript and subscript

- use **superscript** and **subscript** when marking up items like dates, chemical formulae, and mathematical equations so they have the correct meaning.
  - The **<sup>** for Superscript
  - and **<sub>** for subscript.

    - **QUIZ**: create the html code that display this below page.

My birthday is on the 21^{th} of Dec 2020.

Caffeine's chemical formula is $C_8H_{10}N_4O_2$.

If x^2 is 9, x must equal 3 or -3.

Displaying computer code

- There are several elements available for marking up computer code using HTML:
  - **<code>:** For marking up generic pieces of computer code.
  - **<pre>:** For retaining whitespace and Indentations
  - **<kbd>:** For marking up keyboard short-cuts
  - **<samp>:** For marking up the output of a computer program

<details> and <summary> Elements

- The HTML Details Element **<details>** creates a disclosure widget in which information is visible only when the widget is clicked.
- A summary can be provided using the **<summary>** element.
- A **<details>** widget can be in one of two states.
  - The default **closed** state displays only the triangle and <summary> text.
  - **Open** state, revealing its contents.

```
<details>
  <summary>what is this</summary>
  Something small enough to escape casual notice.
</details>
```

HTML Tables

- A table is a structured set of data made up of rows and columns (tabular data).
- A table allows you to quickly and easily look up values that indicate connection between different types of data
- HTML tables should be used for tabular data only.
- An HTML table consists of one **<table>** element and one or more **<tr>, <th>,** and **<td>** elements.
- An HTML table may also include **<caption>, <thead>, <tfoot>,** and **<tbody>** elements.
- **colspan** and **rowspan** attributes can expand cells, they take positive number which indicates how many cells to expand

HTML Forms

- Web forms are one of the main points of interaction between a user and a web site or application.
- Forms allow users to enter data, which is generally sent to a web server for processing and storage.
- The HTML **<form>** element is used to create an HTML.
- The **<form>** element is a container for different types of input elements,

Input Element

- The HTML **<input>** element is the most used form element.
- An **<input>** element can be displayed in many ways, depending on the **type** attribute.
  - **text** – display single-line text input
  - **radio** – display radio button – for selecting one of many choices.
  - **checkbox** – display checkbox, zero or more choices.
  - **submit** - display submit button.

The <label> element

- The **<label>** tag defines a label for many form elements.
- The **<label>** element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.
  - The **for** attribute of the **<label>** tag should be equal to the id attribute of the **<input>** element to bind them together.

```
<input type="radio" id="male"  value="male">
<label for="male">Male</label><br>
```

The <textarea> Element

- The **<textarea>** element defines a multi-line input field (a text area):

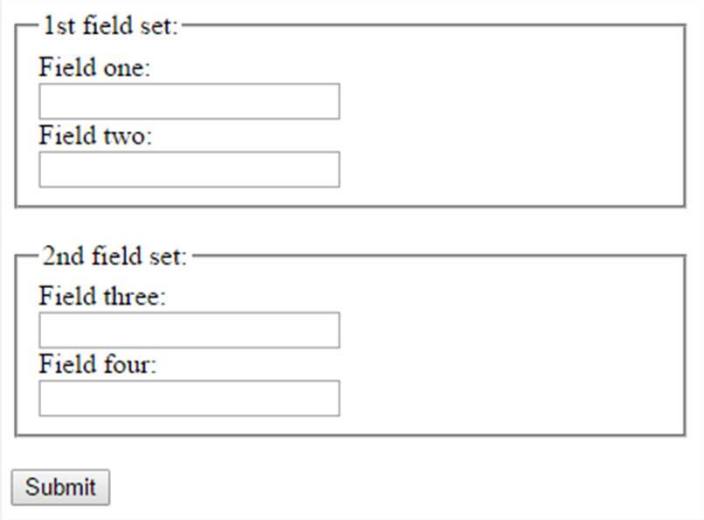
```
<label for="comment">Post your Comment</label><br>
 <textarea id="comment" rows="10" cols="30">
</textarea>
```

The <datalist> Element

- The **<datalist>** element specifies a list of pre-defined options for an <input> element.
- The **list** attribute of the **<input>** element, must refer to the id attribute of the **<datalist>** element.

The **<fieldset>** and **<legend>** elements

- The **<fieldset>** tag is used to group related elements in a form.
- The **<fieldset>** tag draws a box around the related elements.
- The **<legend>** tag is used to define a caption for the <fieldset> element.



Semantic Elements

- A semantic element clearly describes its meaning to both the browser and the developer.
  - Example of non-semantic elements: ** ** and **** - Tells nothing about its content.
  - Examples of semantic elements: **<form>, <table>,** and **<article>** - Clearly defines its content.

  Semantic Elements include:-

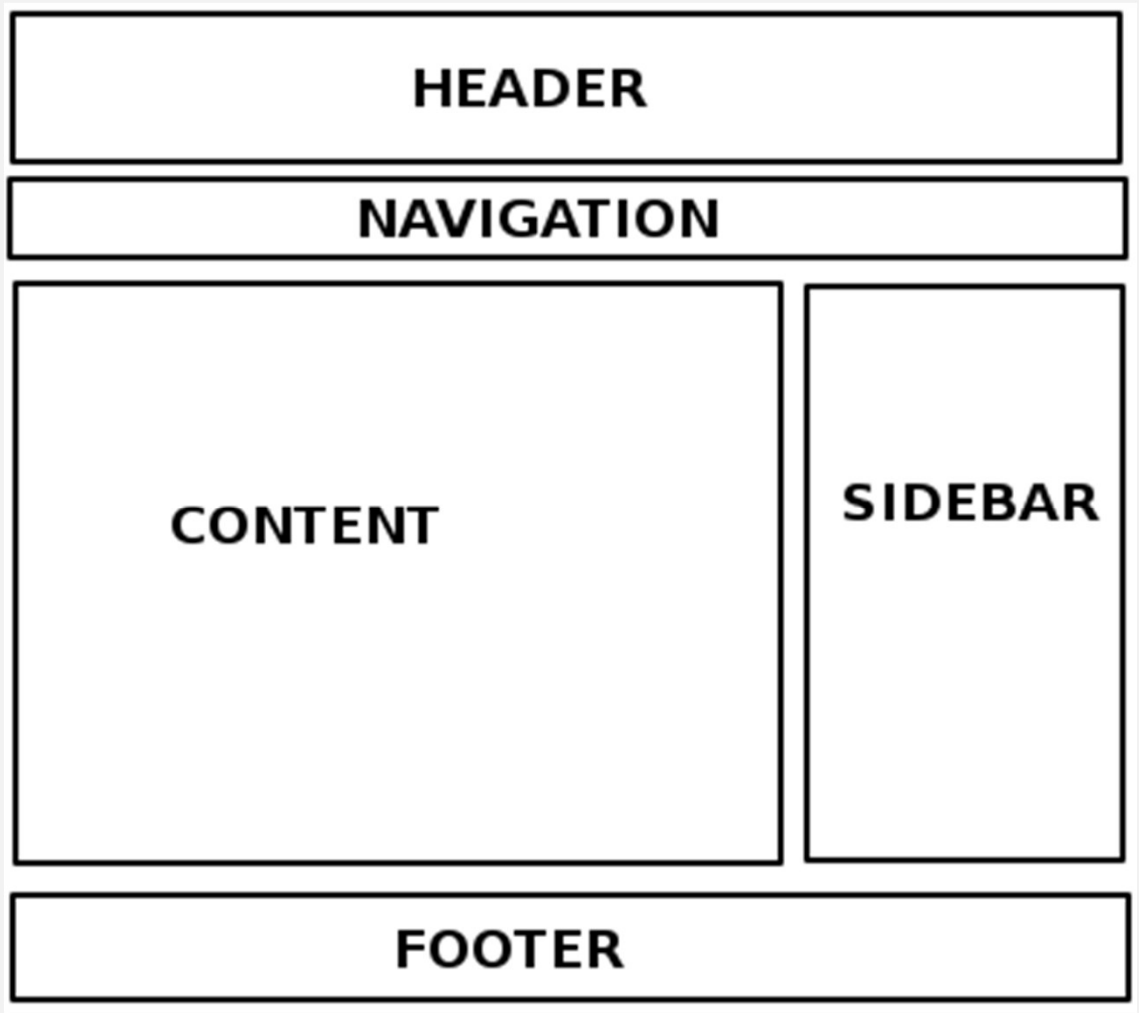
  - <article>, <aside>, <details>, <footer>, <header>, <main>, <mark>, <nav>, <section>, <summary>, <time>, <address>

Structuring document using semantic

- Webpages can look different from one another, but they all tend to share similar standard components.

- Basic sections of a document
  - **header**: <header>.
  - **navigation bar**: <nav>.
  - **main** content: <main>, with various content subsections represented by **<article>**, **<section>,** and elements.
  - **sidebar**: <aside>; often placed inside **<main>**.
  - **footer**: <footer>.

Basic Page Layout



Debugging HTML

- **HTML** is not compiled into a different form before the browser parses it and shows the result (it is interpreted, not compiled).
- **HTML's** element syntax is lot easier to understand than a "real programming language" like Rust, JavaScript, or Python.

  - there are two main types of error in Programming:
    - **Syntax errors**: These are spelling errors in your code that actually cause the program not to run, like the Rust error shown above.
    - **Logic errors**: These are errors where the syntax is actually correct, but the expected result is incorrect.