Frontend Development

Besant Technologies

Dhanabalan.cs

WWW

- The World Wide Web -- also known as the web, WWW or W3 -- refers to all the public websites or pages that users can access on their local computers and other devices through the internet.
- These pages and documents are interconnected by means of hyperlinks that users click on for information.
- This information can be in different formats, including text, images, audio and video.

- **URL:** https://www.google.com/search?q=angular
- So, this URL is redirecting your web browser to the Google search page with a query parameter that specifies the term "angular" to be searched for.
- **http/https** : HTTP/HTTPS is the standard protocol used for transferring data between a web browser (client) and a web server.
- **WWW** : It is often denotes the World Wide Web.
- **google.com** : Domain. It identifies the location of the web server that hosts the site.
- **/search** : The path indicates the specific location or resource on the web server. In this case, it points to the "search" page or resource on the Google website
- **?q=angular** : Query parameters are used to send data to the web server as key-value pairs. In this example, "q" is the key, and "angular" is the corresponding value. This indicates a search for the term "angular" on the Google search page.

EXERCISE

- Understand about website
- What is web pages
- How each webpage interlinked
- What are the types of content can be used in a webpage
- What is W3C and its role?

Frontend Development :

- Front-end development, often referred to as **client-side development**
- It is the process of designing and creating the user interface and user experience **(UI/UX) of a website** or web application that users
- interact with directly in their web browsers
- It involves crafting the **visual elements, layout**, and functionality that make a website or web app both visually appealing and interactive.
- A **Front-End Developer** is someone who creates websites and web applications.
- The **difference** between Front-End and Back-End is that Front-End refers to how a web page looks, while back-end refers to how it works.
- You can think of Front-End as client-side and Back-End as server-side.

Frontend :

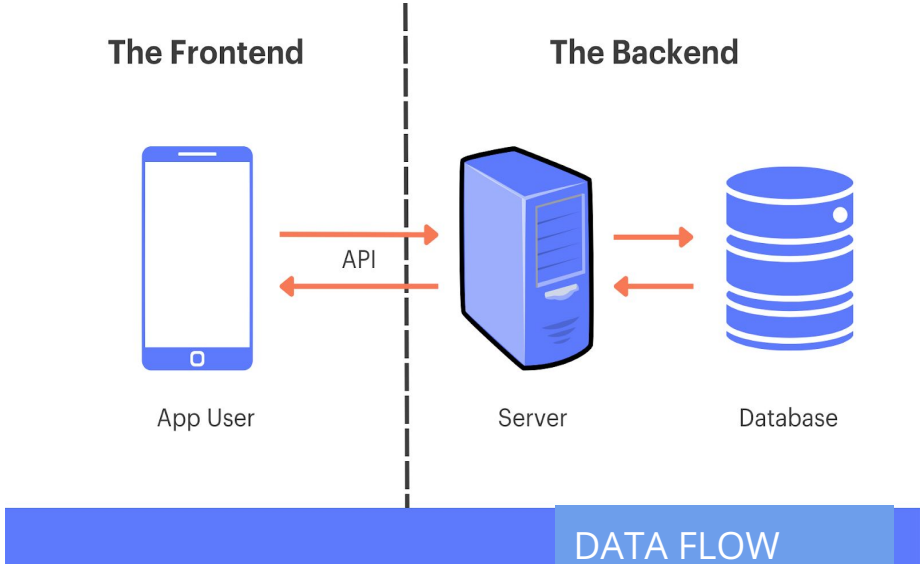
Definition: Front-end development, also known as client-side development, involves creating the user interface (UI) and the user experience (UX) of a website or web application that users interact with directly in their web browsers.

Responsibilities:

- ❑ Design and create web pages, including layout, content, and visual elements.
- ❑ Use HTML for structuring content, CSS for styling, and JavaScript for interactivity.
- ❑ Ensure the website is responsive and works well on various devices and screen sizes.
- ❑ Focus on user-friendly design and accessibility.
- ❑ Implement animations, transitions, and other visual effects.

Skills:

- ❑ Proficiency in HTML, CSS, and JavaScript.
- ❑ Knowledge of front-end libraries and frameworks (e.g., React, Angular, Vue.js).
- ❑ Understanding of responsive web design principles.
- ❑ UI/UX design skills are a plus.



Backend:

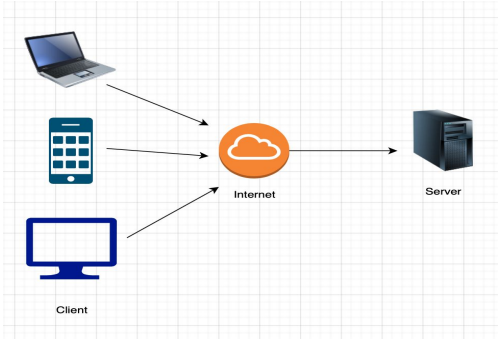
Definition: Back-end development, also known as server-side development, involves creating the server, databases, and server-side logic that enable a web application to function. It handles data processing, user authentication, and more.

Responsibilities:

- ❑ Develop server-side applications and APIs to handle client requests.
- ❑ Implement business logic and data processing.
- ❑ Create and manage databases, including data storage and retrieval.
- ❑ Handle user authentication and authorization.tion, and more

Skills:

- ❑ Proficiency in server-side programming languages (e.g., Node.js, Python, Ruby, Java, PHP, C#).
- ❑ Knowledge of back-end frameworks (e.g., Express.js, Django, Ruby on Rails).
- ❑ Database management skills (SQL or NoSQL databases like MongoDB).
- ❑ API design and development.



Frontend vs Backend Languages :

Frontend Technologies	Backend Technologies	Database
❏ HTML ❏ CSS ❏ JavaScript ❏ TypeScript ❏ Smarty	❏ PHP ❏ JAVA ❏ NODE ❏ C# ❏ Python	❏ MySql ❏ Sqlite ❏ Oracle ❏ MongoDb

Tech Stacks :

- **LAMP** Stack : (Linux, Apache, MySQL, PHP)
- **MEAN** Stack : (MongoDB, Express.js, Angular, Node.js)
- **MERN** Stack : (MongoDB, Express.js, React, Node.js)
- **Full Stack** : Developers who have proficiency in both front-end and back-end development using a variety of technology stacks. They are adaptable and can work with different combinations of front-end and back-end technologies based on project requirements.

HTML

What is HTML?

1. HTML, short for **H**yper **T**ext **M**arkup **L**anguage
2. HTML is a standard markup language for web page creation.
3. It defines the meaning and structure of web content
4. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (**CSS**) or functionality/behavior (**JavaScript**).
5. First published by **Tim Berners-Lee in 1989**, HTML is now used by 94% of all websites

Hypertext: Think of "hypertext" like a web of interconnected information. It's a way to organize and link different pieces of information together, like web pages connected by clickable links.

Markup: "Markup" means adding special codes (tags) to text to give it structure and meaning. These codes tell web browsers how to display the text. It's like giving instructions to a web browser on how to format content.

Language: "Language" here refers to a set of rules and symbols used for communication. In this case, HTML is a language that web browsers understand to display web pages correctly.

Is HTML a programming language ??

1. HTML is not a programming language in the traditional sense.
2. HTML is a markup language used to structure and present content on the web.
3. HTML is essential for web development, it's often paired with programming languages like JavaScript to add functionality and interactivity to web pages.

What is HTML used for ?

1. HTML is primarily used for creating web pages.
2. HTML is free to use and ensures your text, images, and other elements are displayed as intended.
3. With HTML, not only can you add headings, paragraphs, lists, and other elements to your page — you can also embed images, videos, audio files, and other multimedia.
4. make tables for organizing data
5. create forms
6. create emails with HTML

Important:

1. Even after adding headings, images, and hyperlinks, you'd still have a very basic web page — and that's by design.
2. HTML is purely for the content of a page.
3. It creates a simple base upon which you can add styling with another language called CSS (which stands for Cascading Style Sheets).
4. With CSS, you can customize your styling and layouts, changing the color, font, and alignment of elements.

Where to use HTML:

HTML is used in various contexts and scenarios in web development

- **Websites**:HTML forms the backbone of websites. It's used to create the layout, structure, and content of web pages.
- **Blogs**: HTML is used to structure blog posts, format text, embed media, and create links within blog content.
- **E-commerce:** HTML is used to create product listings, shopping cart interfaces, and checkout processes on e-commerce websites.
- **Web Forms**: HTML provides form elements for collecting user data, such as registration forms, login forms, and feedback forms.

HTML VERSIONS

- 1991- Tim Berners-Lee invents HTML 1.0.
- 1993- HTML 1.0 is released. Not many developers are creating websites at this time.
- 1995- HTML 2.0 is published.
- 1997- HTML 3.2 was invented.
- 1999- The widely-used HTML 4.01 comes out. ...
- 2014- HTML 5.0 is released and used worldwide.

PREREQUISITES

EDITOR: Notepad / Notepad++ / Visual studio code Eclipse

BROWSER: Chrome/ IE /Microsoft Edge / Safari and other browsers..

HTML VERSIONS Detailed in NEXT PAGE

REFERENCE WEBSITES

https://overapi.com/html

https://developer.mozilla.org/en-US/docs/Web/HTML

https://html.spec.whatwg.org/multipage/ - **OFFICIAL HTML DOC**

HTML 1.0

HTML 1.0 or first version of HTML was a version of SGML that had ability to link different document or pages using 'href'.HTML 1.0 had 20 elements or tags, now latest version of HTML, ie HTML5 has a lot more.

HTML 2.0

After HTML 1.0, the second version of HTML was released in 1994. HTML 2.0 was an expansion of HTML 1.0.Internet Engineering Task Force (IETF) was behind it's creation.

HTML 3.2

HTML 3.2 was released In 1997. HTML 3.2 had many new features like tables, superscript, subscript etc.Two most important features introduced in HTML 3.2 were tables and text flow around images.Tables were widely used and programmers still use them but it is not recommended anymore. In HTML5 div tags and other semantic elements are used more frequently instead of table element.

HTML 4.01

HTML 4.01 was released In 1999. HTML 4.01 introduced features like scripting, style sheets, better tables, better forms frames and embedding objects.HTML 4.01 was a revised version of HTML 4.0, it also included features for the disabled people to enhance their interactivity with the Global world through Internet.

XHTML

In 2000 XHTML was released. XHTML stands for Extensible Hyper Text Markup Language. XHTML has strict set of rules and it is basically an XML application of HTML.

HTML5

So all of this added up and then after so many year HTML5 was released in 2014. HTML5 is the best version of HTML up till now. HTML5 improved user interactivity so much and also lessened the burden of devices.

HTML5 fully supports all kind of media application that are there. HTML5 supports both audio and video media content. HTML5 also provides full support for JavaScript to run in the background.

Create My First HTML FILE

Step 1: Choose a Text Editor

You'll need a text editor to write your HTML code. You can use simple text editors like Notepad (Windows) or TextEdit (Mac), or more specialized code editors like Visual Studio Code, Sublime Text, or Atom.

Step 2: Create a New Folder (Optional)

You can organize your HTML files in a dedicated folder. To create a new folder:

Windows: Right-click in the location where you want to create the folder, select "New," and then choose "Folder." Give it a name, like "MyWebsite."

Step 3: Create the HTML File

Inside your chosen folder, follow these steps to create your HTML file:

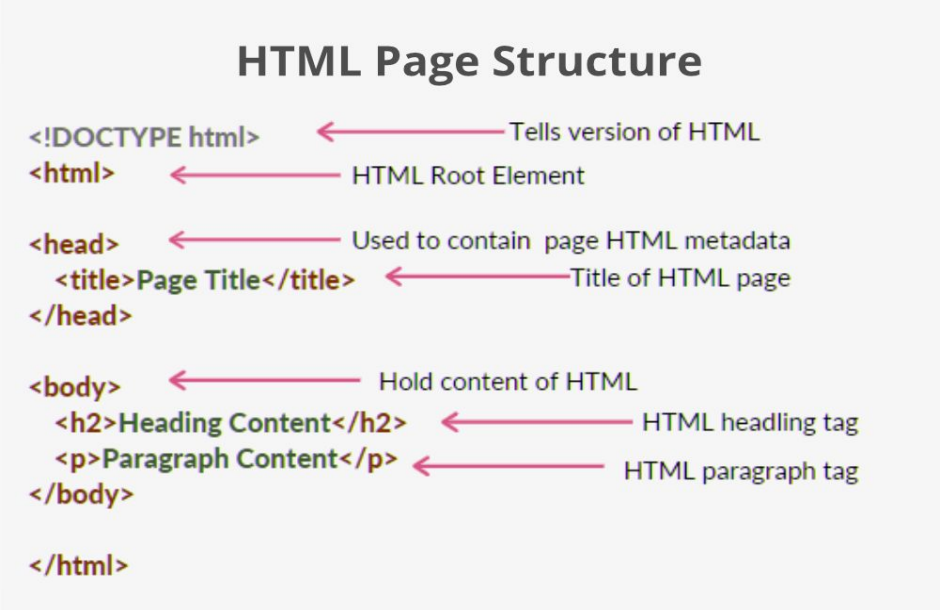
Windows:

- ❏ Open your text editor (e.g., Notepad).
- ❏ Click "File" in the menu and select "New" to create a new document.
- ❏ Write your HTML code
- ❏ Click "File" and select "Save As."
- ❏ Give file name as "**index.html**" (or any name you prefer). And Click "Save.

Step 4: View Your HTML File

Now that you've created your HTML file, you can view it in a web browser:

- ❏ Navigate to your folder (e.g., "MyWebsite").
- ❏ Find the "index.html" file you created.
- ❏ Double-click the "index.html" file to open it in your default web browser.
- ❏ You should see your "Hello, World!" message displayed in the browser.



Syntax

- ❏ The <!DOCTYPE html> declaration defines that this document is an HTML5 document
- ❏ The <html> element is the root element of an HTML page
- ❏ The <head> element contains meta information about the HTML page
- ❏ The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- ❏ The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- ❏ The <h1> element defines a large heading
- ❏ The <p> element defines a paragraph

<!DOCTYPE html>

- All HTML documents must start with a <!DOCTYPE> declaration.
- The declaration is not an HTML tag.
- <!DOCTYPE> tag is used to inform the browser about the version of HTML used in the document. It is called as the document type declara (DTD).

HTML5: <!DOCTYPE html>

HTML 4 : <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

XHTML: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

Valid HTML Elements in Different DOCTYPES:

Tag	HTML5	HTML4	XHTML
<font>	NO	YES	NO
<picture>	YES	NO	NO
<strike>	NO	YES	NO
<video>	YES	NO	NO

IMPORTANT:

- In summary, while it may not be technically mandatory to include a doctype declaration in your HTML documents,
- it is highly recommended to do so. It's a fundamental part of document structure and helps ensure compatibility, consistency, and proper rendering across different platforms and browsers.

Building Blocks of HTML

- ❖ Tags
- ❖ Attributes
- ❖ Elements

Tags:

- ❑ Tag tells the browser how to present the content inside it
- ❑ Every tag performs a different task,and for every different tasks there are different HTML tags available

  Example:

  <img /> - Display Image

  **** Welcome to HTML5 **** - Make text as bold
- ❑ Tag starts with a less-than angle bracket (<) and ends with a greater-than angle bracket (>) with the tag name in-between. Example , , <h1>, etc.
- ❑ Generally, there are opening and closing tag for each tag. A tag is close by a less-than angle bracket (<), forward-slash (/), tag name and greater-than angle bracket (>). i.e (</tag>)

Tags Example:

 Paragraph Tag

<h2> Heading Tag </h2>

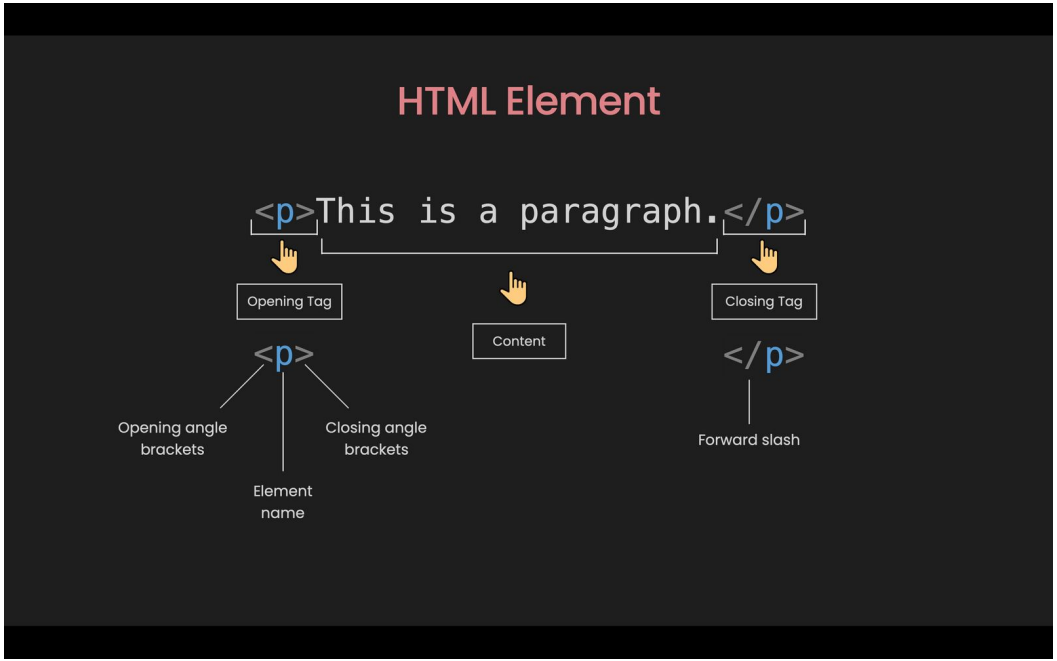
 Bold Tag

 Italic Tag

<u> Underline Tag</u>

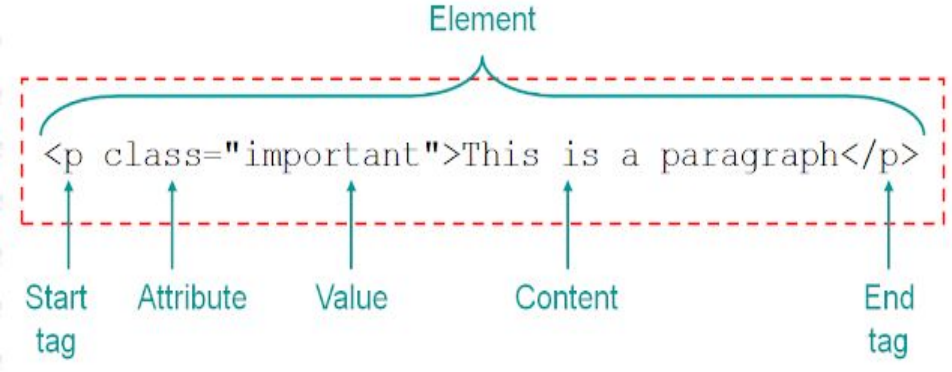
ELEMENTS:

- Technically, an element is a collection of start tag, attributes, end tag, content between them.
- Elements are responsible for creating web pages and define content in that webpage



ATTRIBUTES:

- Attributes attach extra information to the HTML elements in the form of key-value pair.
- This extra information can be adding the title to the element, adding class or id, adding inline style, etc.
- The attributes are used to find HTML elements to style it using CSS selectors and is also used by JavaScript to select HTML elements by ID or other attribute or then perform some task over it.
- Attributes have two parts: the name and the value. value part should be written in the quote; either in single quote (') or in the double quote (""). Example title="this is my website"
- Attribute is written only in opening tag shown as follows:
- My First HTML Markup tag



HTML Attributes

Attribute	Description
alt	Specifies an alternative text for an image
disabled	Specifies that an input element should be disabled
href	Specifies the URL (web address) for a link
id	Specifies a unique id for an element
src	Specifies the URL (web address) for an image
style	Specifies an inline CSS style for an element
title	Specifies extra information about an element (displayed as a tool tip)
value	Specifies the value (text content) for an input element.

Headings:

- Heading tags define a page's **main heading (<h1>)** as well as the **sub-headings (<h2>-<h6>)** of various content sections.
- Headings help us to get an idea on the content on the web page.

PRIORITY:

<h1>:
- Priority: **Highest**
- Usage: There should be only one <h1> tag per page, and it should represent the mai heading or topic of the page. It sets the primary focus and topic for the content.

<h2>:
- Priority: **High**
- Usage: <h2> tags are used for subheadings that are directly related to the <h1> tag. They provide a logical subdivision of content under the main topic.

<h3> to <h6>:
- Priority: Decreasing from <h3> to <h6>
- Usage: These tags are used for further subdividing content within each level of headings. For example, you might use <h3> for subsections within an <h2> section, and so on. The importance of content decreases as you go from <h3> to <h6>.

Syntax:

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

Paragraphs:

Purpose: The tag is used to define and structure text content as a paragraph on a web page.

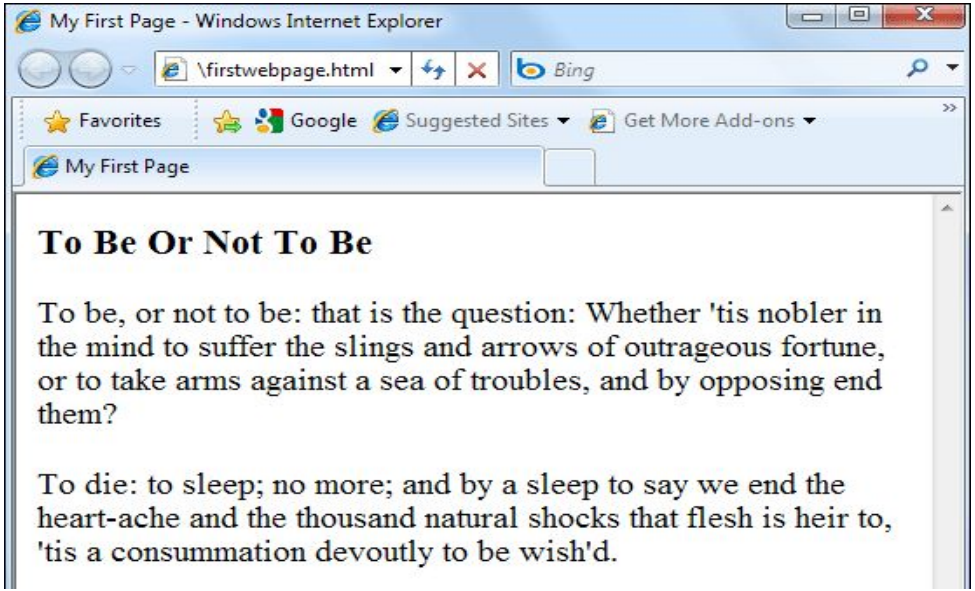
Block-Level Element: It is a block-level HTML element, which means it starts on a new line and creates space both above and below the content it encloses.

Nested Elements: You can include other inline and block-level HTML elements within a tag to format the content further, but it's generally recommended to use other tags like **, , or ** for more complex structures.

Syntax:

This is a paragraph of text. It can contain various sentences and spans multiple lines.

Paragraphs sample output:



HTML Formatting Tags:

There are various types of formatting tags available to us through which we can write texts in different styles and formats. Different types of formatting tags are:

1. **Bold tag** - It does not hold any extra importance other than making text bold. **BOLD TEXT**
2. **Strong tag-** Other than making text bold tag have symantic importance and it is used to give extra stress and emphasize important words. **STRONG TEXT**
3. **Italic tag** - This tag is just used to write text in italic font, it does not hold any special meaning.**ITALIC TEXT**
4. **Em tag -** It is used to emphasize the text. Screen readers put extra verbal stress while reading the text in tags. tag was introduced in HTML5 **Emphasized Text**
5. **Underline tag -** Underline tags are used on important words which require the attention of readers or can be used to underline headings or subheadings **<u>Underlined TEXT</u>**
6. **Mark tag -**
   - This tag was introduced in HTML5.
   - Mark tag is used to highlight a word in a sentence.
   - Mark tag highlights the words by giving them a **yellow background**. It is supported by all major browsers.
   - **<mark>Formatting tags</mark>**

Continues

HTML Formatting Tags:

1. **Strike tag**

   In HTML4 <strike> tag is used to put a horizontal line through words. It can be used in articles to show that some items are deleted or updated from a list or article.

   In HTML5 <s> tag is used to write striked text. It is used when a word is no longer relevent or requierd in an article. We should not use <s> tag to represent deleted text <del> tag must be used in that case.

   NOTE- <strike> tag is not supported in HTML 5. We can use <s> or <del> instead.

   <strike>Text</strike>

   <s>Text</s>

   ~~Text~~

2. **Subscript tag-** A Subscript is a letter, symbol or a number which is usually in a smaller font size and half a character below the normal line. Subscript tags are used to write chemical formulas like

   H_2O

   H₂O

3. **Superscript tag-** A Superscript is a number, letter or symbol which is usually in a smaller font size and half a character above the normal line. Superscript tags are used to write mathematical equations

   x²+ y²

Continues

HTML Formatting Tags:

1. **Deleted and Inserted tag**

- Delete tag is used for text which is no longer required or has to be deleted.
- Browsers render deleted text as strike-through text.
- While <ins> tag is used to insert updated or new text in the article.
- Text enclosed by <ins> tag is renderd as underlined text by browsers.
- Delete and insert tags are used to add and remove words.

~~Text~~

<ins>Text</ins>

~~HTML4~~ HTML5 is the latest version of HTML!

2. **Big and Small Texts**

- HTML <big> tag makes the font of text one size bigger than the surrounding text i.e from small to medium, medium to big.
- HTML <small> tag makes font of text one size smaller than the surrounding text i.e from big to medium, medium to small.
- **Note- Big tag is not supported in HTML5 while the small tag is.**
- Big and small tags are used to make the font one size bigger and smaller than the default font size.
- Instead of the big tag, we can change the font size of the text by using CSS.

<big>Text</big>

<small>Text</small>

This is an example of small tag and big tag

Continues

SELF EXPLORE -TAGS

- <kbd>
- <code>
- <samp>
- <var>

- **<pre>**: it represents preformatted text which is to be presented exactly as written in the HTML file
  <pre>SOME TEXT</pre>

- **<hr />**: Horizontal Ruler **<hr />**
- **
**: Create a Line Break **
**

(Block-Level Element):

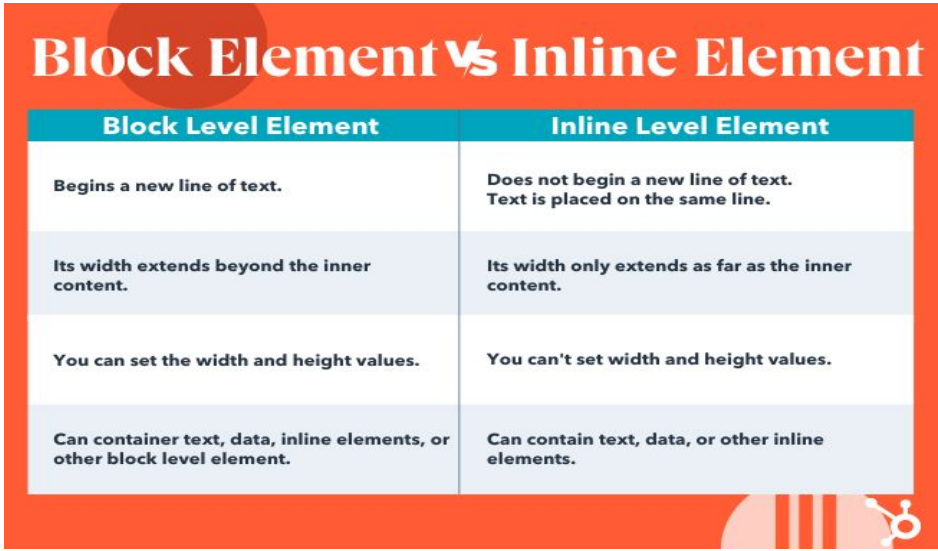
- The element is a block-level **container** used for grouping and structuring content.
- It is often used with CSS for **layout purposes**, such as creating sections, columns, or containers for various elements.

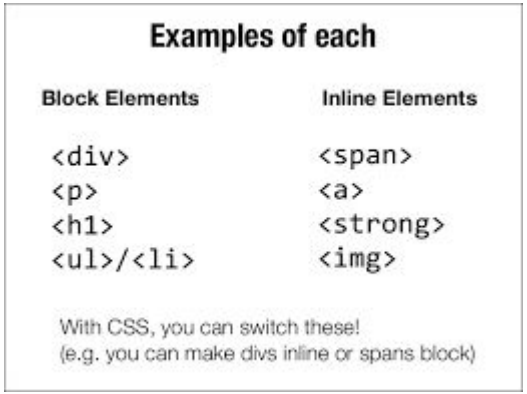
```
<div>
    <h1>Header</h1>
    <b>Bold text</b>
    <p>This is some content.</p>
</div>
```

(Inline Element):

- The element is an inline container used to apply styling or manipulate small portions of text or inline elements within a block-level element..

```
<p>This is a <span style="color: red;">red</span> word.</p>
```





<a> The Anchor Tag

- The <a> tag, also known as the anchor tag and it is used to create hyperlinks on a web page.
- Hyperlinks are clickable elements that allow users to navigate to other web pages, resources, or locations
- The most important attribute of the <a> element is the href attribute, which indicates the link's destination.
- **By default, links will appear as follows in all browsers:**
  - An **unvisited link** is underlined and blue
  - A **visited link** is underlined and purple
  - An **active link** is underlined and red
- **We can create Hyperlinks to:**
  - web pages
  - files
  - email addresses
  - locations in the same page - BookMarking

EXAMPLE:

```
<a href="https://www.amazon.com/">AMAZON</a>

<a href="sample.pdf">DOWNLOAD AS PDF</a>

<a href="mailto:john@gmail.com" title="email">Email</a>

<a target="_self" href="demo.html#bookmark">Navigate to specific part of a page</a>

<a href="javascript:alert('welcome!');">Execute JavaScript</a>
```

The Anchor Tag Attributes

href: This is the most important attribute of the <a> tag, and it specifies the URL or destination to which the link points. It can be an external URL or a relative URL to an internal page or resource.

Visit Example.com

target: This attribute determines how the linked content will be displayed. Common values include:

- _self (default): Opens the linked document in the same window or tab.
- _blank: Opens the linked document in a new browser window or tab.
- _parent: Opens the linked document in the parent frame or window.
- _top: Opens the linked document in the top-level frame or window.

Visit Example.com in a new tab

title: The title attribute provides additional information about the link, often displayed as a tooltip when the user hovers over the link. It's useful for providing context or describing the link's purpose.
Example.com

id and class: You can apply id and class attributes to the <a> tag to target it for styling with CSS or for JavaScript interaction.
Click me

download: When used, this attribute suggests to the browser that the linked resource should be downloaded as a file instead of navigating to it. The attribute's value is the desired filename for the downloaded file.
Download PDF

HTML COMMENTS

- In HTML, you can use comments to add notes or explanations to your code that are not displayed to website visitors.
- Comments are helpful for documenting your code, providing context, or temporarily disabling specific sections of code during development.
- **Single Line Comments**

```
 !-- This is a single-line comment --
```

- **Multi Line Comments**

```
 !-- This is a

multi-line

comment -->
```

<img> Tag

- The <img> HTML element embeds an image into the document

  < img src=”flower.png” alt=”flower” title=”Rose” style=”width:200px; height:200px” />

Attributes:

- The src attribute is required, and contains the path to the image you want to embed.
- The alt attribute holds a text description of the image, which isn't mandatory but is incredibly useful for accessibility — screen readers read this description out to their users so they know what the image means. Alt text is also displayed on the page if the image can't be loaded for some reason: for example, network errors.
- **Supported image formats:**
  - gif
  - jpeg
  - png
  - svg
  - webp

EXAMPLE:

<img src="https://angular.io/assets/images/logos/angular/shield-large.svg" alt="Angular" width="100px" height="200px"> **(absolute URL)**

 (Relative URL)

ABSOLUTE PATH

- An absolute path specifies the **full URL** or file path to the image or assets, starting from the root directory of the website or the entire file system.

  Example <img src="https://angular.io/assets/images/logos/angular/shield-large.svg" alt="Angular" width="100px" height="200px"><

- In this example, the image source (src) is an absolute URL, which means the image is hosted on a remote server and can be accessed directly via the internet.

RELATIVE PATH

- A relative path specifies the path to the image relative to the current web page or the HTML file's location.
- Relative paths are often used for images within the same website directory structure.
- Suppose you have the following directory structure:

```
- website/
  - index.html
  - images/
    - relative-image.jpg
```

Example

- In this example, the image source (src) uses a relative path, so the browser looks for the image in the "images" directory located in the same directory as the HTML file.
- When using relative paths, it's crucial to understand the directory structure of your website and specify the correct path to the image relative to the HTML file's location.
- This allows you to keep your image references organized and maintainable as you build your website.

QUESTION- TIME

1 . Which should use absolute Path or relative path ?

2. How to make clickable / i want to redirect to http://www.google.com, when i click an image. How?

3. What is Broken Links?

4. I want to download a zip directory, while clicking a LINK.

5. How do you create a clickable phone number link using the <a> tag for mobile devices?

LISTS in HTML

You can create lists to **organize and display information in an ordered or unordered manner**.
There are **three main types of lists** in HTML

Ordered Lists (<ol>):

- Ordered lists are used when you want to display a list of items in a specific sequence or order, typically with numbers or letters indicating the order.
- Each list item is preceded by a number or letter. Here's an example of an ordered list:

```
<ol>
    <li>Angular</li>
    <li>React JS</li>
    <li>Vue JS</li>
</ol>
```

Output:

1. Angular
2. React JS
3. Vue JS

```
<ol type="A" start="5">
    <li>Coffee</li>
    <li>Tea</li>
    <li>Milk</li>
</ol>
```

Ordered HTML List - The Type Attribute

The `type` attribute of the `<ol>` tag, defines the type of the list item marker:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

Output:

E. Coffee
F. Tea
G. Milk

Continues

Unordered Lists (<ul>):

- Unordered lists are used for lists of items that do not have a specific order.
- Typically, they are displayed with bullet points.

```
<ul>
    <li>BOOTSTRAP</li>
    <li>HTML</li>
    <li>CSS</li>
</ul>
```

Output:

- BOOTSTRAP
- HTML
- CSS

```
<ul type=”square”>
    <li>BOOTSTRAP</li>
    <li>HTML</li>
    <li>CSS</li>
</ul>
```

Output:

- BOOTSTRAP
- HTML
- CSS

- `type="disc"` (default): This is the default value, and it displays list item markers as filled circles.
- `type="circle"`: It displays list item markers as hollow circles.
- `type="square"`: It displays list item markers as squares.

Continues

DEFINITION Lists (<dl>):

- Definition lists are used to create lists of terms and their corresponding definitions.
- A <dl> element contains a series of <dt> (definition term) elements for terms and <dd> (definition description) elements for their corresponding definitions

```
<dl>
  <dt>HTML</dt>
  <dd>Hyper Text Markup Language</dd>
  <dt>JS</dt>
  <dd>Javascript</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheet</dd>
</dl>
```

Output:

HTML
: Hyper Text Markup Language

JS
: Javascript

CSS
: Cascading Style Sheet

HTML *CLASS ATTRIBUTE*:

- The HTML class attribute is used to **specify a class for an HTML element**.
- Multiple HTML elements can share the same class.
- The class attribute is often used to point to a class name in a style sheet.
- It can also be used by a JavaScript to access and manipulate elements with the specific class name.
- **Multiple elements can have the same class**
- An element can belong to **multiple classes by separating them with spaces**.
- To create a class; write a period (.) character, followed by a class name. Then, define the CSS properties within curly braces {}

```
<a href="#" class="menu-item active">Home</a>
<a href="#" class="menu-item ">About Us</a>
<a href="#" class="menu-item ">Products</a>
<a href="#" class="menu-item ">Sitemap</a>
<a href="#" class="menu-item ">Contact Us</a>
```

```
<style>
.menu-item {
color: #000;
}
.active {
background-color: #e74c3c;
text-decoration: none;
}
</style>
```

Output:

Home About Us Products Sitemap Contact Us

HTML *ID ATTRIBUTE*:

- The id attribute is used to uniquely identify a specific HTML element on a page.
- Unlike classes, each ID should be unique within a document.
- This uniqueness allows you to precisely target and style or manipulate a single element
- The syntax for id is: write a hash character (#), followed by an id name. Then, define the CSS properties within curly braces {}.

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

```
<style>
#header
{
background-color: yellow;
font-weight: bold;
}
</style>
```

Output:

This is a header.

ID Selector	Class Selector
• ID selector uses ID to select elements	• The class selector uses the CSS class to select elements.
• When you just need to select only one element, use ID selector.	• When you want to select a group of elements, having the same CSS class
• ID selector uses # character.	• The class selector uses “.” character.

HTML *FAVICON*:

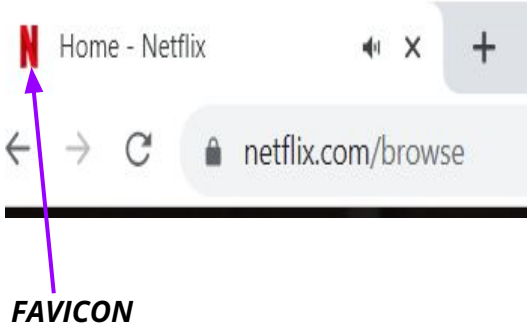
- A favicon is a small image displayed next to the page title in the browser tab.
- **NOTE**: A favicon is a small image, so it should be a simple image with high contrast.

Create Your Favicon Image:
First, create an icon image that you want to use as your favicon. Favicon images are typically square and can be in various formats such as ICO, PNG, or SVG. The most common format is ICO, but modern browsers also support PNG and SVG favicons.

Choose an Appropriate File Format:
Decide which format you want to use for your favicon. If you choose ICO, you'll need to create an ICO file. If you prefer PNG or SVG, make sure your favicon is saved in one of those formats.

Name Your Favicon File:
Save your favicon image with an appropriate name. The recommended name is "favicon.ico" for ICO files, but if you're using PNG or SVG, you can still name it "favicon.png" or "favicon.svg" for clarity.

Upload Your Favicon Image:
Upload your favicon image to your website's server or a web hosting service. Make sure it's accessible via a URL (e.g., "https://example.com/favicon.ico").



Add the Favicon to Your HTML Document:
Open the HTML document where you want to add the favicon and include the following code inside the <head> Section:

Syntax:

```
<head>
  <title>My Page Title</title>
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
```

HTML Practicals#1 : Please read the requirement carefully and implement

Create a webpage using following requirements:

- ❏ Create a Header Part of the webpage, it should contain your company logo (use div and img tag and other styles as per your creativity)
- ❏ Then Create a Navigation section using anchor tag (Home, About, contact, products and some download Link). **Hyperlink Should Not be broken.**
- ❏ Add FAVICON to your page
- ❏ Try to apply Background color for an entire page
- ❏ Please go to the URL: https://scribemedia.com/book-subtitle/
  - ★ Create Some headings and subheadings
  - ★ Under every headings i i need sample paragraph contents
  - ★ Need Every headings & Subheadings should be underlined
  - ★ Some Important keyword should be highlighted in paragraph
  - ★ Formatting tags should be used across this page
  - ★ Use Some valid images in your website
  - ★ Add Lists in your webpage

Note: (The following coding practice should follow)

- ★ Proper directory Structure and use proper fileName
- ★ Include doctype in your file
- ★ Add Comments (Author/Createddate/Purpose)
- ★ **Don't use inline Styles**, except some scenarios
- ★ Practice attributes like ID,CLASS
- ★ All html tag should be in small letters

HTML5 *TABLES*

- HTML tables are used to display data in rows and columns on a web page.
- They consist of several HTML elements, including the <table>, <tr>, <th>, and <td> elements.

```html
<table>
  <thead>
    <tr>
      <th>Name</th>
      <th>Job Title</th>
      <th>Salary</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>John Doe</td>
      <td>Software Engineer</td>
      <td>$80,000</td>
    </tr>
    <tr>
      <td>Jane Smith</td>
      <td>Product Manager</td>
      <td>$95,000</td>
    </tr>
  </tbody>
</table>
```

```css
<style>
  table {
    width: 100%;
    border-collapse: collapse;
  }

  th, td {
    border: 1px solid #ddd;
    padding: 8px;
    text-align: left;
  }

  th {
    background-color: #f2f2f2;
  }
</style>
```

Name	Job Title	Salary
John Doe	Software Engineer	$80,000
Jane Smith	Product Manager	$95,000
Robert Johnson	Marketing Specialist	$70,000
Susan Lee	Graphic Designer	$60,000

In this example:

- The <table> element is used to create the table.
- The <thead> element contains a row of table headers (<th>) to define column headings.
- The <tbody> element contains the actual data rows.
- Each row of the table is defined using the <tr> (table row) element. Within each row, you can use <th> (table header) elements for headers in the table's header row and <td> (table data) elements for data cells in the data rows.

Here's a breakdown of what the CSS styles in the <style> section do:

- width: 100%;: Makes the table span the entire width of its container.
- border-collapse: collapse;: Collapses the borders of adjacent table cells into a single border, creating a cleaner appearance.
- th, td: Styles both header and data cells with a border, padding, and left-aligned text.
- th: Styles header cells with a background color to distinguish them from data cells.

Add TABLE BORDER

To add a border, use the CSS border property on table, th, and td elements.

```
table, th, td {
  border: 1px solid black;
}
```

TABLE CELL PADDING

- Cell padding is the space between the cell edges and the cell content.
- By default the padding is set to 0.
- To add padding on table cells, use the CSS padding property:

```
th, td {
  padding: 15px;
}
```

Add TABLE CAPTION

You can add a caption that serves as a heading for the entire table.

```
<caption>Students Result</caption>
```

COLLAPSED TABLE BORDER

To avoid having double borders, set the CSS border-collapse property to collapse.

```
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
```

TABLE CELL PADDING

- Cell spacing is the space between each cell.
- By default the space is set to 2 pixels.
- To change the space between table cells, use the CSS border-spacing property on the table element

```
table {
  border-spacing: 30px;
}
```

TABLE STYLE : ZEBRA STRIPES

- If you add a background color on every other table row, you will get a nice zebra stripes effect. (even / odd rows)
- To style every other table row element, use the :nth-child(even) selector like this:

```
tr:nth-child(even) {
      background-color: darkgrey;
}
```

TABLE ROWSPAN

- you can use the rowspan attribute to make a cell span multiple rows.
- This attribute is applied to a <td> or <th> element and specifies how many rows the cell should span.
- This is useful when you want to create a table with rows that have cells that cover multiple rows.

```
<table>
 <thead> <tr><th>Name</th><th>Department</th><th>Skills</th></tr>
 </thead>
 <tbody>
  <tr> <td rowspan="2">John Doe</td> <td>Development</td><td>JavaScript, HTML, CSS</td>
  </tr>
  <tr>
   <td>Testing</td> <td>Manual testing</td>
  </tr>
  <tr>
   <td  rowspan="2">Jane Smith</td><td>Design</td><td>Photoshop, Illustrator</td>
  </tr>
  <tr >
   <td >UI/UX</td>
   <td>Sketch, Figma</td>
  </tr>
 </tbody>
</table>
```

Name	Department	Skills
John Doe	Development	JavaScript, HTML, CSS
	Testing	Manual testing
Jane Smith	Design	Photoshop, Illustrator
	UI/UX	Sketch, Figma

TABLE COLSPAN

- In HTML tables, you can use the colspan attribute to make a cell span multiple columns.
- This attribute is applied to a <td> or <th> element and specifies how many columns the cell should span.
- This is useful when you want to create a table with cells that cover multiple columns

```
<table>
 <thead>
  <tr> <th>Name</th><th colspan="2">Skills</th> </tr>
 </thead>
 <tbody>
  <tr>
   <td>John Doe</td>
   <td>JavaScript</td>
   <td>HTML, CSS</td>
  </tr>
  <tr>
   <td>Jane Smith</td>
   <td colspan="2">Photoshop, Illustrator, Sketch</td>
  </tr>
 </tbody>
</table>
```

Name	Skills	
John Doe	JavaScript	HTML, CSS
Jane Smith	Photoshop, Illustrator, Sketch	

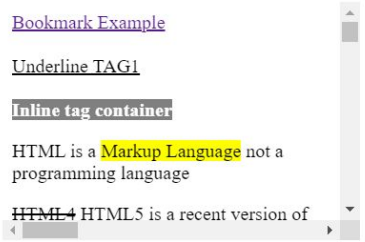
Attribute	Semantic Tags	Non-Semantic Tags
Purpose	Convey the meaning and structure of content	Primarily used for layout and presentation
Examples	`<header>`, `<nav>`, `<article>`, `<footer>`, `<section>`, `<aside>`, `<main>`, `<figure>`, `<figcaption>`, `<details>`, `<summary>`, etc.	`<div>`, `<span>`, ` `, `<b>`, `<i>`, `<u>`, `<font>`, `<center>`, `<table>`, `<tr>`, `<td>`, `<th>`, etc.
Accessibility	Enhance accessibility by providing clear content structure	May not enhance accessibility unless used with ARIA attributes
SEO Impact	Can improve search engine optimization (SEO) by providing semantic structure to content	May not directly impact SEO, as search engines may give less weight to non-semantic elements
Use Case	Used for content organization and meaning	Used for layout styling and presentation
Screen Readers	Provides clear structure for screen readers	May not convey content structure effectively to screen readers
Example Code	`<header><h1>Welcome to My Website</h1></header>`	`<div><font color="blue"><b>Important Text</b></font></div>`
Recommended Usage	Encourage their use for better code quality and accessibility	Use sparingly for layout purposes, avoid overuse

HTML iframe Tag:

1. The HTML <iframe> tag specifies an **inline frame**.
2. It is used to **embed another HTML document or web page within the current web page**.
3. It allows you to display content from another source, such as an **external website, video, map, or document,** directly within your web page.
4. Here's an example of how to use the <iframe> element:

```
<html>
 <head>
  <title>DEMO</title>
 </head>
 <body>
  <!-- HTML Content loaded within iframe -->
  <iframe src="./080902023-html-demo.html" height="200" width="300" title="Iframe Example"
  style="border:none;"></iframe>
  <br /><br/>
  <!-- Embed YouTube video's URL in iframe -->
  <iframe width="560" height="315" src="https://www.youtube.com/embed/5ccq_nLHneE?
  si=IpRzHHpUm3Z8EdKQ" title="YouTube video player" frameborder="0" allow="accelerometer;
  autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
  allowfullscreen></iframe>
 </body>
</html>
```

OUTPUT:



Three Primary ways to apply *CSS to HTML DOCUMENTS*:

- Inline CSS
- Internal CSS
- External CSS.

Inline CSS:

- Inline CSS is applied directly to individual HTML elements using the style attribute within the HTML tag.
- Inline CSS is useful for making quick, specific style changes to individual elements but is generally not recommended for larger-scale styling.
- Example of Inline CSS:

```
<h1 style="color: blue; font-size: 24px;">
 This is a heading with inline CSS
</h1>

<p style="color: red;">
  This is a paragraph with inline CSS.
</p>
```

Internal CSS:

- Internal CSS is defined within the **<style>** element in the HTML document's **<head>** section.
- It applies to all HTML elements within the same document.
- Internal CSS is useful when you want to apply styles to multiple elements within a single HTML document.
- Example of Internal CSS:

```
<!DOCTYPE html>
<html>
<head>
    <title>Internal CSS Example</title>
    <style>
        h1 {
            color: blue;
            font-size: 24px;
        }
        p {
            color: red;
        }
    </style>
</head>
<body>
    <h1>This is a heading with internal CSS</h1>
    <p>This is a paragraph with internal CSS.</p>
</body>
</html>
```

External CSS

- External CSS is stored in a **separate .css file and linked to the HTML document using the <link> element**.
- This approach allows you to maintain styles separately from the HTML content and reuse styles across multiple pages.
- External CSS is recommended for larger websites with multiple pages and for better organization of styles.
- Example of Internal CSS:

styles.css (External CSS File):

```css
/* styles.css */
h1 {
    color: blue;
    font-size: 24px;
}

p {
    color: red;
}
```

HTML File (index.html):

```html
<!DOCTYPE html>
<html>
<head>
    <title>External CSS Example</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <h1>This is a heading with external CSS</h1>
    <p>This is a paragraph with external CSS.</p>
</body>
</html>
```

- In this example, the CSS rules are defined in an external .css file (styles.css), and the HTML document links to this file using the <link> element.
- The styles are applied to the HTML elements according to the rules defined in the external CSS file.
- Using external CSS is a recommended practice for maintaining scalable and maintainable stylesheets in larger web projects, as it promotes separation of concerns between content and presentation.

Three Primary ways to apply *Javascript to HTML DOCUMENTS*:

- Inline JS
- Internal JS
- External JS.

Inline JS:

- Inline JavaScript is placed directly within the HTML document
- Inline JavaScript is useful for making quick, specific script implementations directly within HTML elements.
- Example of Inline JS:

```
<button onclick="alert('Hello, inline JavaScript!')">Click me</button>
```

In this example, the `onclick` attribute contains inline JavaScript code that triggers an alert when the button is clicked.

Internal JS:

- Internal JavaScript is defined within the **<script>** element in the HTML document's <**head**> or **<body>** section. It applies to all HTML elements within the same document.
- Internal JavaScript is useful when you want to include script logic specific to a single HTML document.
- Example of Internal CSS:

```html
<!DOCTYPE html>
<html>
<head>
    <title>Internal JavaScript Example</title>
    <script type="text/javascript">
        function showMessage() {
            alert('Hello, internal JavaScript!');
        }
    </script>
</head>
<body>
    <h1>This is a heading</h1>
    <button onclick="showMessage()">Click me</button>
</body>
</html>
```

External JS

- External JavaScript is stored in a separate **.js file** and linked to the HTML document using the **<script>** element with the **src** attribute.
- External JavaScript allows you to maintain your scripts separately from HTML content and reuse them across multiple pages.
- Example of Internal CSS:

script.js (External JavaScript File):

javascript

```javascript
// script.js
function showMessage() {
    alert('Hello, external JavaScript!');
}
```

HTML File (index.html):

html

```html
<!DOCTYPE html>
<html>
<head>
    <title>External JavaScript Example</title>
    <script type="text/javascript" src="script.js"></script>
</head>
<body>
    <h1>This is a heading</h1>
    <button onclick="showMessage()">Click me</button>
</body>
</html>
```

- In this example, the JavaScript function **showMessage()** is defined in an external .js file (script.js) and linked to the HTML document using the **<script>** element's **src attribute.**
- The function can be reused across multiple HTML files.
- Using external JavaScript is a common practice for maintaining clean and reusable code in larger web projects, similar to using external CSS for styles.