THE PRIME STEP

Holding Your Hand Till The End

THE FULL STACK Development

Consider these notes your guiding light to better understanding, but remember, it's your dedication and passion for this field that will truly illuminate your journey.

Questions are the keys that unlock the treasure chest of knowledge. So, let's embark on our journey of discovery by asking:

What Is Full Stack Development?

For answering this first understand some basics

See, In Computer Science We Have Too Many Options Where We Can Build Our Career:

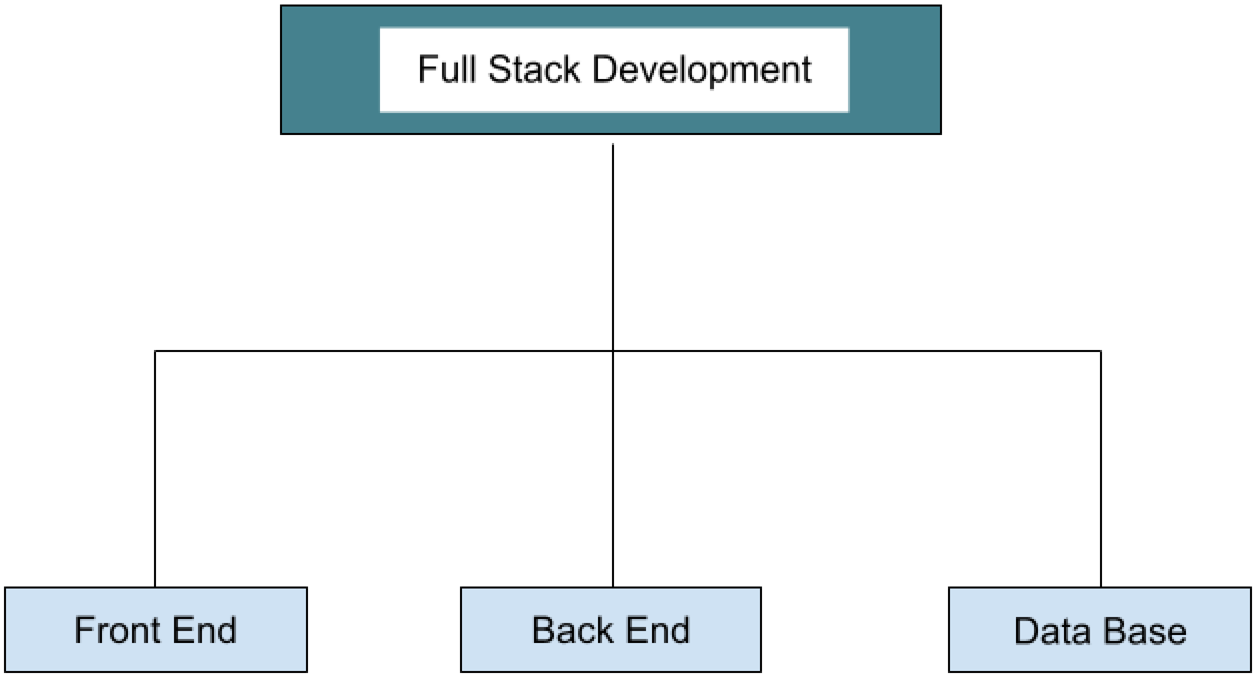
Mainly Categorized Into Two Branches:

1. **Hardware**:- All The Physical Thing Of Computer, Circuit, Physical Components, Processors Etc Etc…
2. **Software**:- Games, Apps, Websites Etc Etc…

So as a Full Stack Developer / WebDeveloper falls in the Software Category, We Do Not Have Any Relation With the Hardware Industry At least For Now.

Okay, Now a Fullstack Developer is a Developer Who Can Develop a Complete Website From Start to End.

Full Stack Development Can Be Further Categorized In:



Front End:- The **'front' of a website** is like the 'face' of it – it's what we see and interact with. This includes everything visible, such as text, images, videos, animations, the sidebar, the buttons and more. So all The Things We See On a Website, The Visible Entity Is the Front-End

Back End:- All the Processing Behind The Scene Which is Not Visible. Think of it as the **engine room of a website**, where all the important work takes place behind the scenes.

For example, when you log in to Facebook by *entering your username and password*, Don't you think there's a hidden process in the background that checks and verifies your information? It ensures that you're a valid user and that your details are correct or not – that's the back-end

Database:- It's simply responsible for storing and managing all the data that the website needs to function properly

OK Now Understand The Subjects We Have To Study In This Whole Course:

Front End:

If Ones Want To Learn Front End He/She Should Learn:

1. HTML - (Basic Structure)
2. CSS - (Style)
    a. BootStrap
    b. CSS Preprocessor
    c. Tailwind

    → **Helpers Of CSS**

3. Java Script - (The Logic Of Front End)
    a. J Query
    b. GSAP
    c. Basic TypeScript
    d. ES6

    → **Helpers Of Java Script and More**

4. Advance (FrameWork):
    a. React Js / Angular Js
    b. Next Js

Remember, This is a basic requirement, It Can Be Further Extended or vary According To the Course.

BackEnd

If one wants to Learn the Back End End He/She Should **Learn a Programming Language First:**

1. Python
2. Java
3. Java Script

These 3 Are Good Options To Start With,

We Still Have Many Languages Which Can Help You In the Back End Like

(C#, Ruby, Go, PHP etc.)

After That,

He/She Should have To Learn The framework Work For the Back End:

Frameworks = Tools / or "Advance Chapters In Programming"

Different Languages Have Different Frameworks

- **Python → Django/Flask**
- **Java → Java EE / Spring / Hibernate**
- **Java Script → Node & Express Js**

We Have To Choose any One Of Them

Data Base:

We Have Many Options, But For Starting:

- SQL
- MongoDB

We Can Start With Any One of These

As We Saw, **The Front Is The First Thing We Have To Learn:**

And for That, We Have To Start With The Subject Called

HTML:

HTML, which stands for "**Hypertext Markup Language**" is like the building blocks of websites. It's a set of instructions that **tells your web browser how to display things on a webpage,** kind of like how a recipe tells you how to make a cake. We use HTML to create web pages by **using the**

help of tags, we have different types of **tags(words of HTML) to make the basic structure of the website like paragraph tags, image tags, video tags etc.** So, in simple terms, HTML is what makes the website work, helping us to make the basic structure of a webpage.

Let’s Start The Journey With HTML

Hyper Text Markup Language

HTML

In the world of Web development,

HTML is where it all begins. It's the canvas

on which we paint our digital masterpieces.

- Satyam Rana

We're here to explore the latest HTML newest version, HTML5, and understand all the important tags it offers. Our goal is to use these tools to create amazing web experiences

CHAPTER - 1:

Journey Through Time: The Fascinating History of the Internet and HTML

After 1980, the internet underwent a significant transformation. Before that, it was mainly used by researchers, scientists, and the military for specific tasks. However, as the 1980s progressed, the Internet started to become more accessible to the general public.

The real **breakthrough** came **in 1989** when **Tim Berners-Lee,** a British computer scientist, **invented the World Wide Web.** This invention made it much easier for people to share and access information on the internet. Instead of needing specialized knowledge to connect to the internet, **people could now use user-friendly interfaces and web browsers**.

In 1991, Tim Berners-Lee also created HTML, Hypertext Markup Language. **HTML was a fundamental innovation** because it allowed people, even those without advanced technical skills, **to create web pages**. With HTML, you could define the structure and content of a webpage using simple tags. It was like giving everyone the tools to build their corner of the internet.

As a result, websites started to proliferate rapidly. People could share their thoughts, knowledge, and creations with a global audience. HTML became the cornerstone of web development because HTML's **simplicity and ease of use** made it accessible to a wide range of people, allowing them to create web pages and share information on the internet **without needing advanced programming skills.**

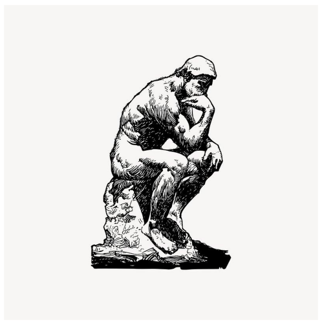
Over the years, web technology has evolved, and websites have become more sophisticated with interactive features, multimedia, and complex functionality. **However, HTML remains the essential language for building web pages.**

CHAPTER - 2:

Diving into HTML: Unveiling the Mysteries & Magic of Tags

So, to learn HTML we need to understand,

What is a **TAG?**



See In Human Language We Have Words Right?

In Languages Of Computer (HTML), We Have TAGS (Words of HTML)

So, Like In Human Language When We Want To Express Our Thoughts We Use Words, Similarly If We Want To Make a Website We Have To Use Tags To Express What We Want To Create What We Want To Make.

By Definition:

A "Tag" is like a label or a special code that you put around content to tell web browsers how to display that content on a webpage.

Simply The Web Browsers (Chrome, Safri, UC Browser) are responsible For Displaying/Showing The Website Right So HTML is the Language To Tell them what To Show and Where To Show it.

But How To Write a Tag:

Okay Let's Make a TAG of the Name “TPS” It Is Not an Official Tag but For Example Let's Do It:

1. We Have To Write The Name Of The Tag Inside (<>) These Angular Brackets or We Can Say Less Than Greater Than
2. SO, <TPS> and Done We Have Successfully Written a Tag Named TPS.
3. Now, Let's Write a Tag With the Name Of the **title** tag, as the name subject is used for the title of the website, we will explore it more later,

4. So, <title>, Perfect? Hmmm, But don’t you Think, You also Have To Write The Name Of the Title? Obviously

   Yes. So, <title>This is My First WebPage, Perfect? Hmmm mmmmm, No Don’t You Think How HTML will Understand where the Tile Tag starts and Where it Ends?

   So, <title>My First Web Page (Now We Have To Close The Tag Also)

   Finally,

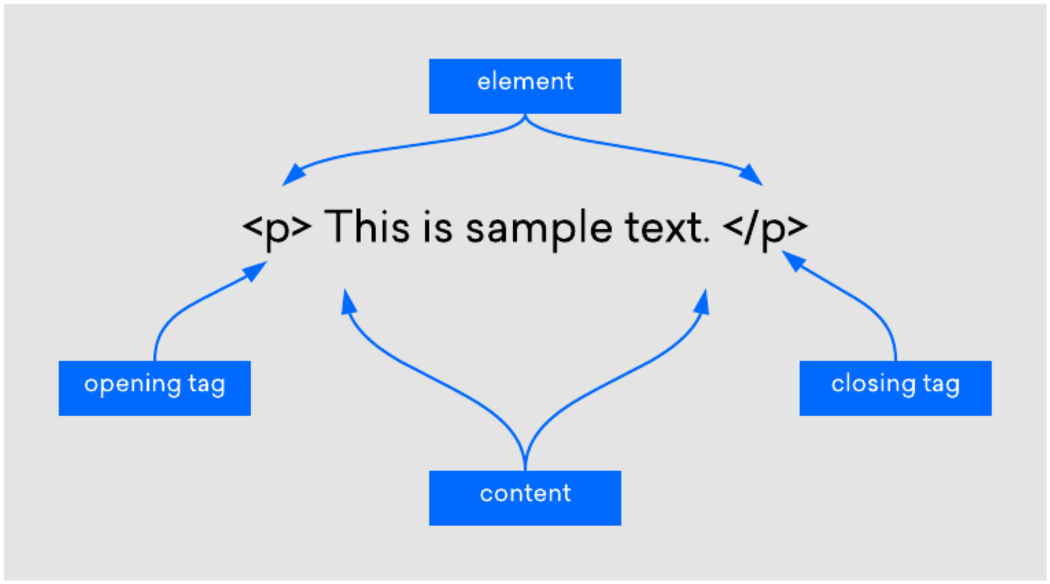
   Complete Tag = <title> My First WebPage </title>

   So, always remember **In Most Cases** We Have To Always Close The Tag Also

For That, We Just Have To Write The Name Of Tag Again with (/) with a forward slash Behind It,

Let’s Understand By a Diagram:

Parts Of a TAG:



<P> Is Used For Paragraph Tag, We Use It Whenever We Want To Write a Paragraph. We Just Use **P** Not a Complete Paragraph

---

Okay, Now After The Basics Understanding Of Tags, Let's Jump Into The **Boilerplate Code** Of HTML.

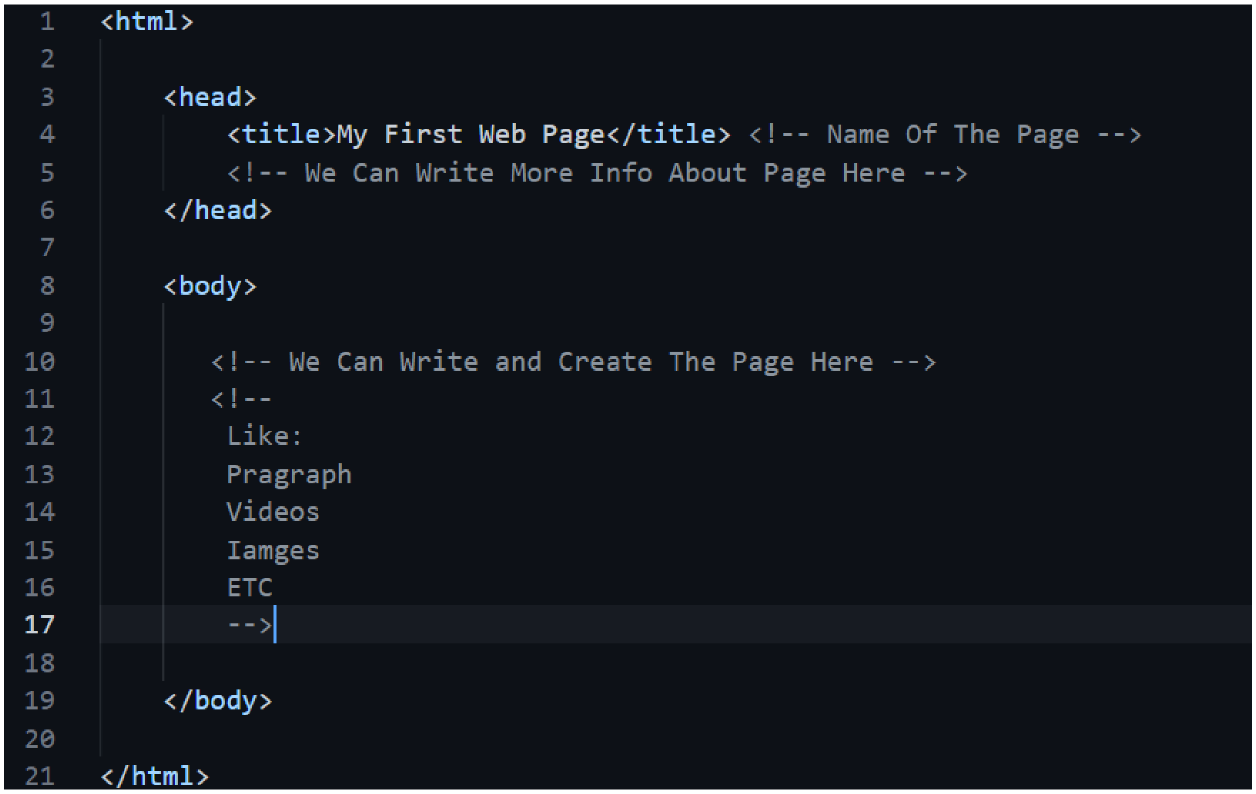
Wait Which Code?

So let us Understand With an Example:

Imagine you're making a new document in HTML, and it's like writing a letter. **Just like you need to start a letter with some basic information,** an HTML document also needs some essential information to work properly.

In HTML, this basic information is called the "boilerplate code." It's like the standard introduction for your webpage. **Simply** (**Prerequisite Code You Always Have To Write**)

Here's a simplified example:



```html
<html>

    <head>
        <title>My First Web Page</title> <!-- Name Of The Page -->
        <!-- We Can Write More Info About Page Here -->
    </head>

    <body>

       <!-- We Can Write and Create The Page Here -->
       <!--
        Like:
        Pragraph
        Videos
        Iamges
        ETC
        -->

    </body>

</html>
```

Now Let’s Understand These Tags:

First Of All You Can See All The Tags Are Closed If They Open You Have To Close (In Most Cases)

The HTML Tag Is The First Tag You Have To Write, Inside the Tag All The codes of HTML will Be Written on this Tag.

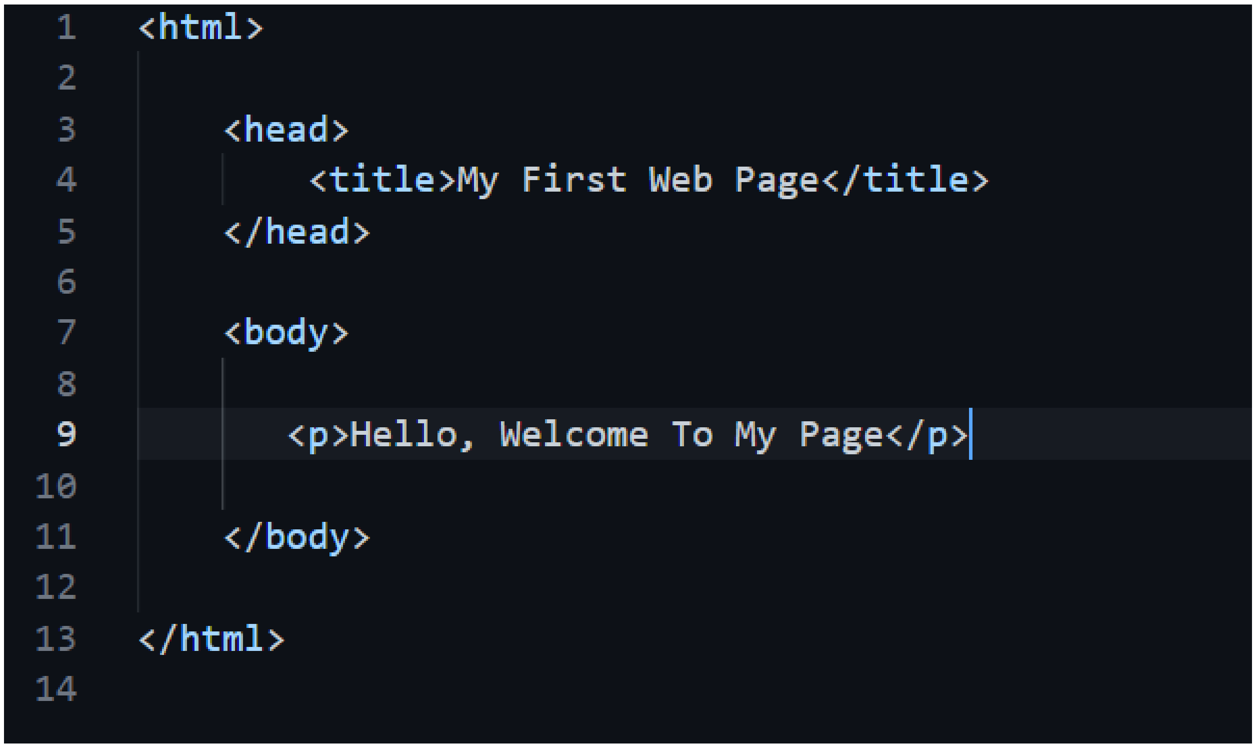
After That, You Can See There are **two Tags Inside the HTML,**

1. **Head** (Contains The Information About Pags, Like Creator Name, Title of Page Settings)
2. **Body** (Here We Create The Page, Inside The Body We Can Create Paragraphs Images Videos Etc)

So, in a nutshell, the boilerplate code is a standardized structure you use to begin an HTML document. It's like the blueprint for your webpage, ensuring that browsers can understand and display your content correctly. Remember, just like in a letter, every tag **you open (like <html>) needs to be properly closed (like </html>) for your code to work as expected.**

So All The knowledge, We Have We Can Now Build a Basic Website:

Okay, So I Think We Can Understand this code Right?



```
<html>

    <head>
        <title>My First Web Page</title>
    </head>

    <body>

       <p>Hello, Welcome To My Page</p>

    </body>

</html>

```

We Know We Have Written The BoilerPlate Code (Prerequisite)

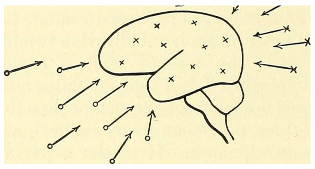
And Then Inside Body Tag We Have Written Something.

Yes, It's The Paragraph Tag, Used For Paragraph

Congrats You Have Successfully Built Your First Page,

You Can Add More (P) Tags

Now Key Points To Remember:



1. **HTML is The First Thing In The Journey Of Front End**
2. **HTML Is Used To Create The Structure Of Website**
3. **The Boilerplate Code Is Must You Have To Always Write It Before You Start Writing HTML**
4. **The Tags Are Simply Words Of HTML, Different Tags Different Uses.**
5. **Most Tags Have a Closing TAG**
6. **For Closing a Tag Just Write The Name Of the Tag with a forward slash**

   (<Name Of Tag> Content </Name Of Tag>)

7. Head Tag Is Responsible For The Information About The Website
8. Body Tag Is Responsible For The Content Of WebSite
9. We Can Write Tags Inside Tags
10. A Tag Can Contain More Information Like (**Attributes**),

Wait What, What Are Attributes?

So,

Let's Understand The Concept Of (**Attributes**)

Attributes are like additional information or instructions that you can add to HTML tags to modify their behavior or provide extra details about an element.

Example: You Want To Add an Image To Your Web Page:

So, First Of All, We Have an Image Tag For That:

But the fun fact is that you don't have **TO CLOSE THIS TAG,** This is a Self Closing Tag.

Some Tags Are Self Closing, We Don't Have To Manually Close Them.

So Let's Add an Image:

<img>

That's It, OK Wait, We Don't Have Text Content, We Want an Image, Right?

Then We Have To Write the Address/Path Of The Image. Sometimes We Want To Adjust the Height or Width Of a Picture, The Attributes come in the Game.

So,

< img src = "Flower.jpg" height = "300" width = "400" >

Attributes Are Always Written In Opening Tag Before The Closing Bracket ">" And After The Name Of Tag, Different Tags Have Different Attributes

- Src = Source (The Path Of Picture)
- Height and Width Attributes Help in setting the Height and Width Of the Picture

Attributes Are Always Written As:

<TagName AttributeName = " " **SecondAttribute** = " " **> Content </TagName>**

So, Attributes are just extra information given to a tag.

OK Now Let's See Some Tags That Can Help In Styling Paragraph:

1. **Bold Tag – <b> Content </b>**
2. **Italic Tag – <i> Content </i>**
3. **Underline Tag – <u> Content </u>**

4. **Delete Tag - <del> Content </del>**
5. **Quote Tag – <q> Content </q>**
6. **Superscript Tag - <sup> Content </sup>**
7. **Subscript Tag - <sub> Content </sub>**

Some Tags Are Just Alternative Of Each Other They Are Just Used For Better Readability:

1. **<Strong>**Content**</Strong>** Is Same As **<b> Content </b>**
2. Emphasis Tag **<em>Content</em>** Is Same As **<i> Content </i>**
3. Insert Tag **<ins>Content</ins>** Is Same As **<u> Content </u**

We Can Use These Tags Inside P Tag,

We Can Use As Many As We Want All At Once If Required

Let's Use Them:

```
<html>


  <head>

    <title>My First Web Page</title>

  </head>


  <body>

        <p>

            Hi This is  <b>Para</b>
```

```
      </p>

      <p>

         Hi This is  <i>Para</i>

      </p>

      <p>

         Hi This is  <u>Para</u>

      </p>

      <p>

         Hi This is  <q>Para</q>

      </p>


      <p>

         E = MC <sup>2</sup>

  <!-- This is Used For Writing Things Like Square In Maths -->

      </p>

      <p>

         H <sub>2</sub>  O

     <!-- This is Used For Writing Things Like 2 Of H2O(Water) -->

      </p>
```

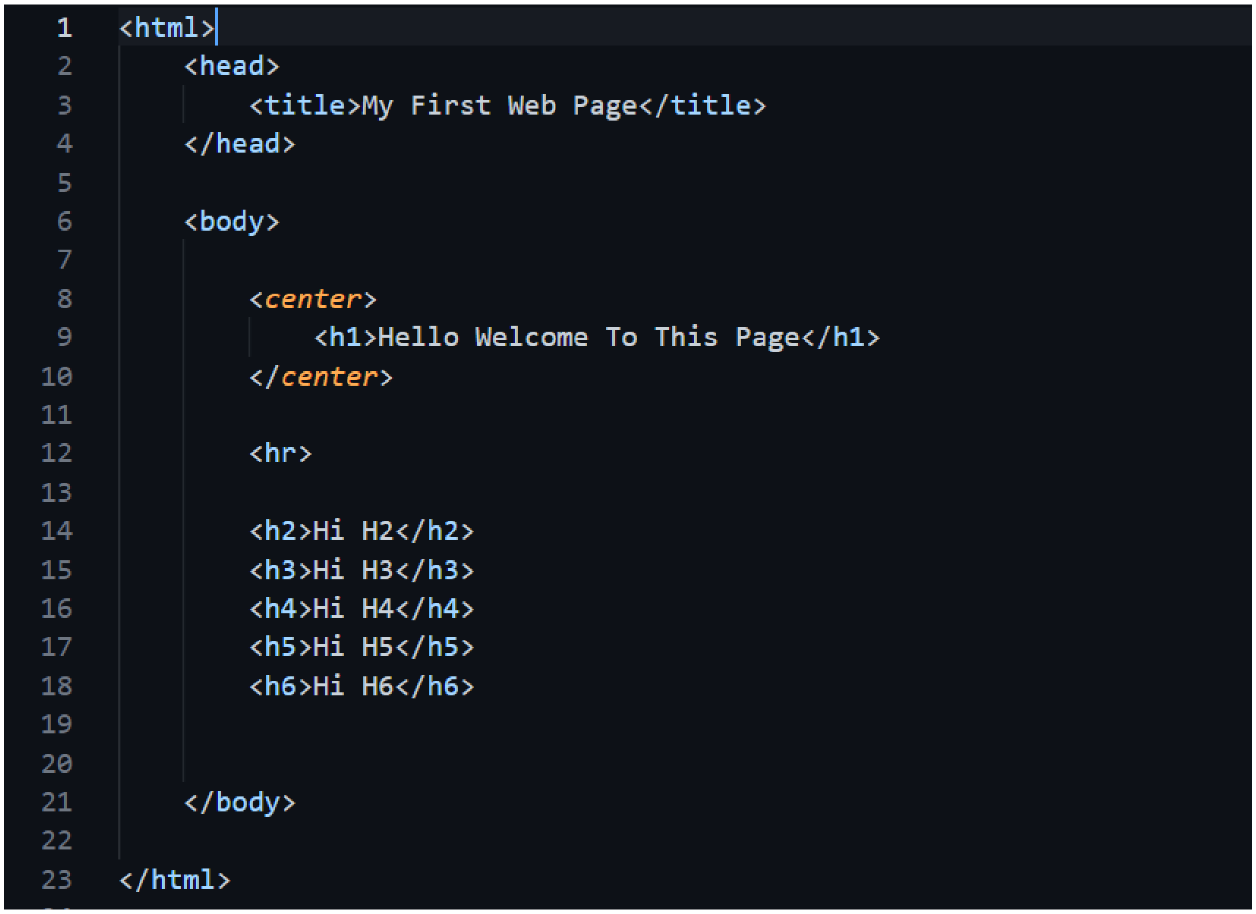
```
        <p>
          <strong>Hi</strong>
          <!-- Same as Bold Tag -->
        </p>
       <p>
          <em>Hi</em>
          <!-- Same as Italic Tag -->
        </p>
        <p>
          <del>Rs:1000</del>
          <ins>Rs:7899</ins>
        </p>
  </body>
</html>
```

After This Let's Understand The Family Of Heading Tags:

Heading tags are like titles for different sections of a web page. Imagine you're reading a book, and each chapter has a **big, bold title at the top.** These titles help you understand what each chapter is about, right? Well, heading tags do the same thing for a web page.

They make it easier for both people and search engines **to figure out what a section of a web page is talking about.** Heading tags are **typically labelled from H1 to H6,** with H1 being the main title or heading of the whole page and H6 being the least important. So, think of them as signposts that guide you through the content on a webpage, making it clear and organized.

In Code:



```html
<html>
    <head>
        <title>My First Web Page</title>
    </head>

    <body>

        <center>
            <h1>Hello Welcome To This Page</h1>
        </center>

        <hr>

        <h2>Hi H2</h2>
        <h3>Hi H3</h3>
        <h4>Hi H4</h4>
        <h5>Hi H5</h5>
        <h6>Hi H6</h6>


    </body>

</html>
```

We Are Using a New Tag **(Center)**, It Can be Used To Place almost anything in the Horizontal Center.

We Also used a New Tag **(hr)** “Horizontal Row” That Is Used For a Horizontal Line Across The Parent (Complete Web Page For Now)

We Have Covered The Tags Which Can Help Us In Writing Paragraphs and Headings

We also have already learned how to add images, and it's as simple as that.

Now You Know:

1. Tags Can Be Written Inside Another Tags
2. For Heading We can Use Heading (H - H6) Tags
3. Tags Can Have Attributes (Extra Information Given To Tag)
4. Different Attributes Are Used For Different Purposes
5. Some Tags Are Self Closing and We Don't Have To Close Them
6. Para Tags
7. Heading Tags
8. Image Tag
9. Hr Tag
10. Center Tag

You're all set to learn about new tags. We just have to tell their names and what they're for, and you're good to go.

Keep practising, watch lectures, create some web pages on these topics, and you'll see it's not rocket science at all! You've got this!

CHAPTER - 3:

Unlocking the Power of Links & Media:

Media TAGS:

In simple terms, media tags can help you add pictures, music, or videos to a webpage. It's a way to tell the web browser, "Hey, here's a picture to show," or "Here's some music to play."

These tags make the web page more interesting by adding different types of media to it.

Media Tags Used To Add:

- Images
- Videos
- Music
-

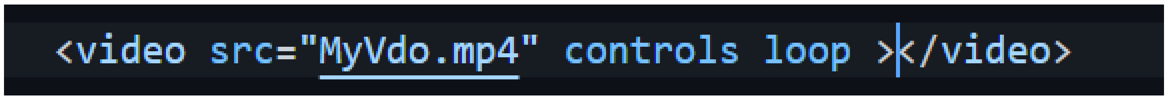
We Already Know How To Add **Images:**

```
<img src="Mountain.jpg" height="300" alt="There Was a Mountain Here">
```

The **'alt'** attribute is like a backup plan for images. It stands for **'Alternate Text**.' If, for any reason, an image can't be shown on a web page, this alternate text will appear instead. It's a way to describe the image in words, so people know what should be there even if they can't see the picture.

And remember, if you set either the height or width of an element, the other dimension will automatically adjust accordingly. You don't need to provide values for both, just one will do the trick.

Video Tag:



```
<video src="MyVdo.mp4" controls loop ></video>
```

As you Can See:

1. **Src** is used for the Source/Path/Location Of The Video
2. **Controls** attribute is used to activate the video controls such as, play, pause, volume, and progress bar. **It Is Necessary To Write**.
3. **Loop** attribute is used to replay the video again and again when it ends

These Attributes are called Boolean Attributes, We Don't Have To Give Them Any Data.

We Also Have:

1. **width**: Sets the width of the video element in pixels or as a percentage of the container.

   Example: <video src="video.mp4" width="640"></video>

2. **height**: Sets the height of the video element in pixels or as a percentage of the container.

   Example: <video src="video.mp4" height="360"></video>

3. **autoplay**: Automatically starts playing the video as soon as it's loaded.

   Example: <video src="video.mp4" autoplay></video>

4. **muted**: Mutes the audio of the video, so it plays without sound.

Example: <video src="video.mp4" muted></video>

5. **poster**: Specifies an image to display as a placeholder before the video is played.

   Example: <video src="video.mp4" poster="video-poster.jpg"></video>

6. **preload**: Defines how the browser should load the video data. Options include "auto," "metadata," and "none."

   Example: <video src="video.mp4" preload="auto"></video>

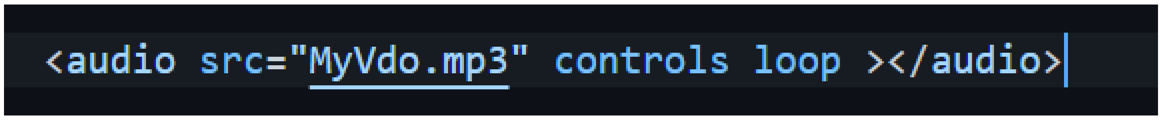
7. **playsinline**: Allows the video to play inline on mobile devices, rather than in a separate full-screen player.

   Example: <video src="video.mp4" playsinline></video>

8. **control list**: Defines the set of controls that should be displayed, allowing you to customize which controls are visible.

   Example: <video src="video.mp4" controls controlsList="nodownload"></video>

Audio Tag:



```
<audio src="MyVdo.mp3" controls loop ></audio>
```

As you Can See same as the Video Tag:

1. **Src** is used for the Source/Path/Location Of The Video
2. **Controls** attribute is used to activate the video controls such as, play, pause, volume, and progress bar. **It Is Necessary To Write**.
3. **Loop** attribute is used to replay the video again and again when it ends

We have all the same attributes available for the <audio> tag as those for the <video> tag, such as 'autoplay,' 'muted,' and 'controlsList etc.'

Now Let's Understand The Concepts Of Adding Links:

For That, We Have an **Anchor Tag.**

Think of the anchor tag <a> as a magical doorway on a web page. When you click on this doorway, it can take you to other places on the same webpage or transport you to entirely different web pages on the internet.

Here's how it works:

1. **Linking to Other Pages:** You can use the <a> tag to create a link to another web page. It's like saying, "Hey, click here to go to this other webpage." You provide the web address (URL) of the destination page in the href attribute of the <a> tag.

   Example: **<a href="https://www.example.com">Click here to visit Example.com</a>**

   “Href” is an attribute “HyperText Reference”

2. **Linking to Files:** You can also use the <a> tag to link to files like PDFs, images, or documents. Just provide the file's URL in the href attribute.

   Example: <a href="document.pdf">Download the PDF</a>

Code:

```
<a href="https://www.youtube.com">YouTube</a>

<a href="MyDoc.pdf">Download PDF</a>

<a href="ContactUs.html">Contact US</a>
```

Let’s Understand a Tag Called iframe:

Think of the <iframe> tag as a special window within a webpage. It's like having a little TV screen embedded inside a bigger TV. This little screen can show content from a completely different website or web page while still being a part of the main page you're on.

We Can Add:

1. **Another Pages**
2. **PDFs**
3. **Youtube Video**
4. **Maps**
5. **Social Media Feeds**

Code:

Pdf:

```
<iframe src="Dos.pdf" width="450"></iframe>
```

For Youtube:

You Have To Follow Certain Steps:

Step 1: Visit YouTube.com and select the video you want to add.

Step 2: Click the 'Share' button below the video player, and then click 'Embed' in the options that appear.

Step 3: Copy the provided code and paste it into your website's HTML editor."

These steps should help you easily embed a YouTube video on your website.

CHAPTER - 4:

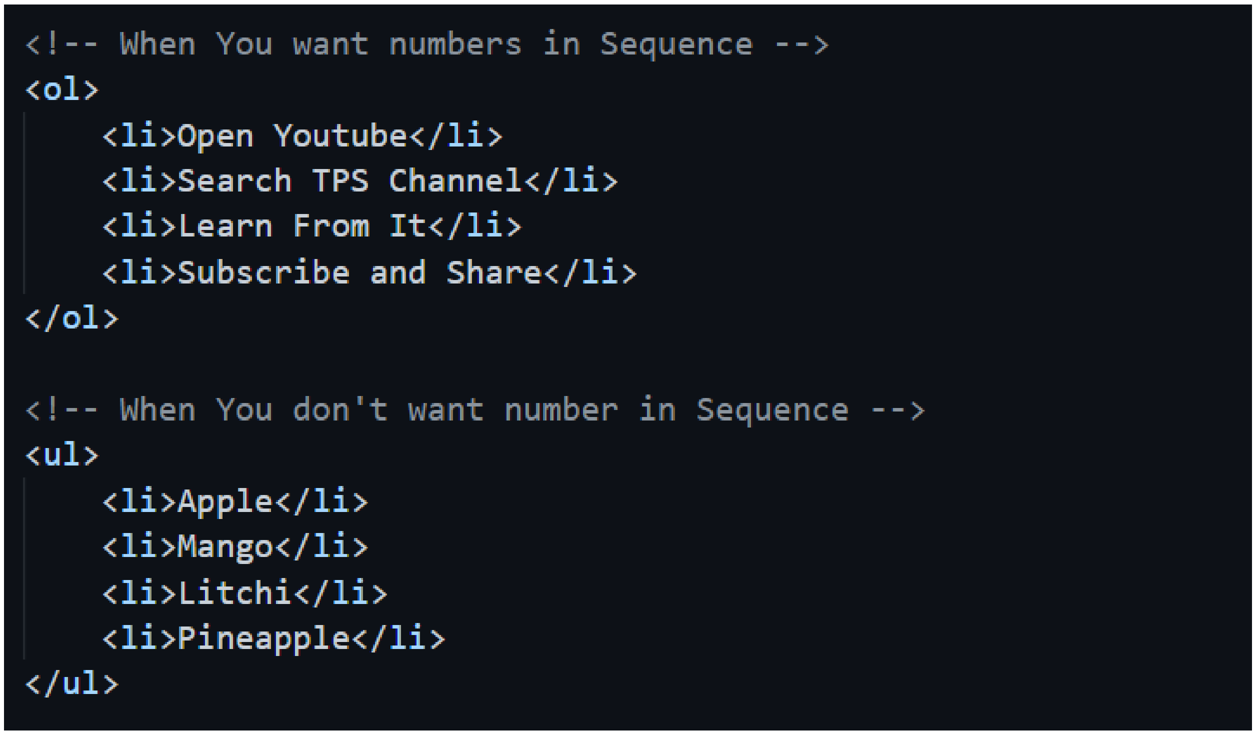
Crafting Data Harmony: The Art of Lists and Tables

In HTML, a "**list**" is a way to organize information in a structured manner, making it easier for people to read and understand. It's like creating a grocery list or a to-do list to keep things organized in your daily life.

There are two main types of lists in HTML:

1. **Ordered List (OL):** An ordered list is like a numbered list. It's used when you want to display items in a specific sequence or order, where each item has a number next to it to indicate its position
2. **Unordered List (UL):** An unordered list is like a bullet-pointed list. It's used when the order of items doesn't matter; they are just a collection of related items.

3. We also have a **definition list**, but we don't use that much. **Let's understand By Code:**



```
<!-- When You want numbers in Sequence -->
<ol>
    <li>Open Youtube</li>
    <li>Search TPS Channel</li>
    <li>Learn From It</li>
    <li>Subscribe and Share</li>
</ol>

<!-- When You don't want number in Sequence -->
<ul>
    <li>Apple</li>
    <li>Mango</li>
    <li>Litchi</li>
    <li>Pineapple</li>
</ul>
```

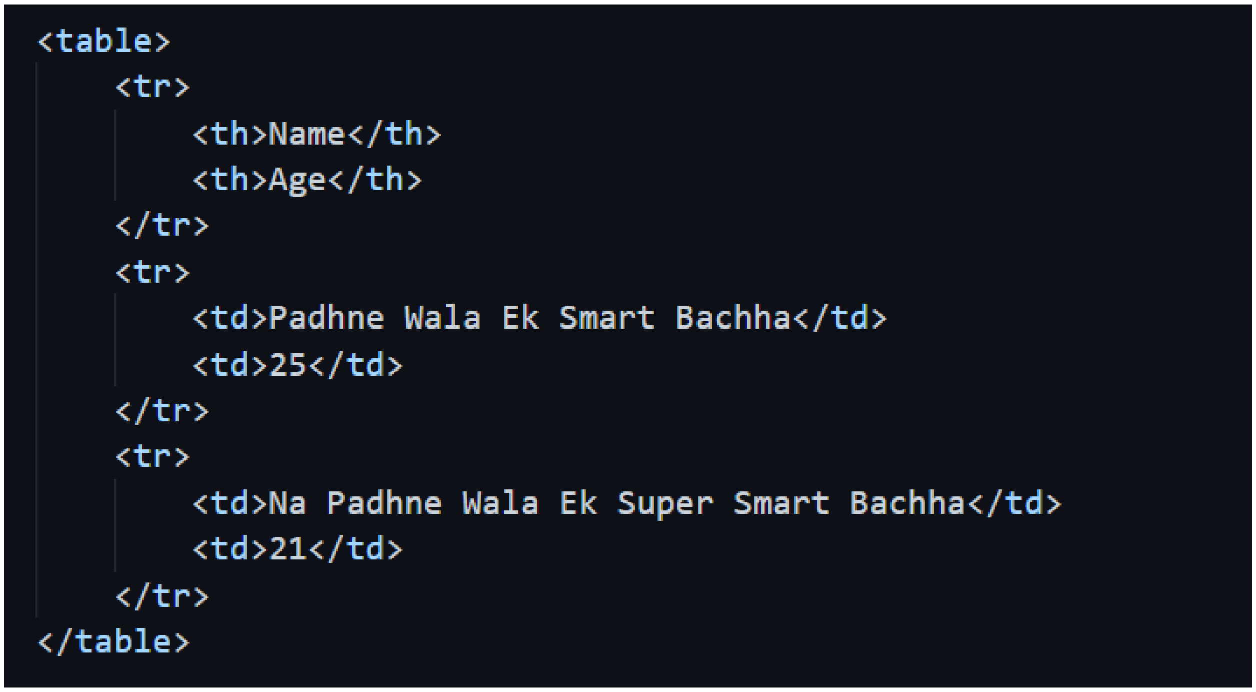
Now, Let's Understand Tables:

In HTML, a "table" is a way to display data in rows and columns, just like you would see in Excel or on a piece of graph paper. It's a helpful tool for presenting information in an organized grid-like format, making it easier to compare and understand data.

Tags We Use For Creating a Table:

1. **Table:** The **overall structure is defined using the <table>** tag. Think of this as the container for your data, like the table itself in a board game.
2. **Row:** Inside the <table>, you use the **<tr> tag to create rows.** Rows are horizontal segments, like rows of seats in a theatre.
3. **Column Header:** Within each row, you can use the **<th> tag to create column headers.** These headers are typically bold and centred, and they describe what each column represents, like the labels on top of columns in a table.
4. **Cell**: Inside each row, you use the **<td> tag to create cells**. Cells are where you put your actual data or content. They are like the individual squares or cells in a table grid.

In Code:



```
<table>
    <tr>
        <th>Name</th>
        <th>Age</th>
    </tr>
    <tr>
        <td>Padhne Wala Ek Smart Bachha</td>
        <td>25</td>
    </tr>
    <tr>
        <td>Na Padhne Wala Ek Super Smart Bachha</td>
        <td>21</td>
    </tr>
</table>
```

CHAPTER - 5:

Mastering Web Forms and Beyond: Harnessing User Data for Your Needs

---

Think about it this way:

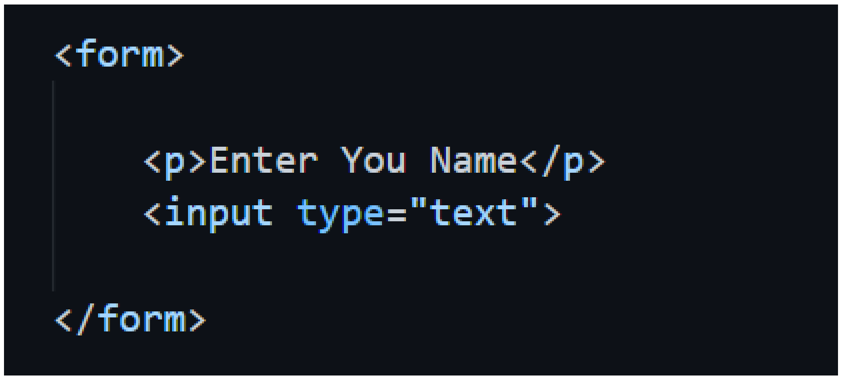
Until now, we've been showing stuff on the screen. But sometimes, we need to do the opposite – we need to get information from the user. Like when we type our name, password, address, or anything else on a website.

That's where the <input> and <form> tags come in Game.

Input Tag: Gives you a space to type in your information, so websites can collect and use it.

Form Tags: The <form> tag in HTML is like a container that holds various elements like Input tags and More.

Code:



```
<form>

    <p>Enter You Name</p>
    <input type="text">

</form>
```

We Can Add As Many input Tag We Need

We have Different Types of Input Tags For Taking Different Types Of Data:

1. **Text**: <input type="text">

   - Used for single-line text input fields, like name or username.

2. **Password**: <input type="password">

   - Creates a password input field where the characters are hidden as you type.

3. **Email**: <input type="email">

   - Specifically designed for email address input, with built-in email validation.

4. **Number**: <input type="number">

   - Provides a field for entering numeric values, like age or quantity.

5. **Checkbox**: <input type="checkbox">

   - Creates a checkbox that allows users to select or deselect an option.

6. **Radio**: <input type="radio">

   - Used for creating a set of radio buttons where users can select one option from multiple choices.

7. **Date**: <input type="date">

   - Provides a date picker for selecting a date.

8. **Time**: <input type="time">

   - Creates a time picker for selecting a specific time.

9. **File**: <input type="file">

   - Allows users to upload files, like images or documents, to a server.

10. **Submit**: <input type="submit">

   - Adds a button to submit the form data to a server.

11. **Reset**: <input type="reset">

   - Adds a button to reset the form fields to their initial values.

12. **Button**: <input type="button">

   - Creates a generic button that can trigger JavaScript functions when clicked.

13. **Hidden**: <input type="hidden">

   - Provides a way to store data on the form that is not visible to users but can be sent with the form submission.

14. **Color**: <input type="color">

   - Allows users to select a color using a color picker.

15. **Range**: <input type="range">

   - Creates a slider control for selecting a value within a specified range.

16. **Search**: <input type="search">

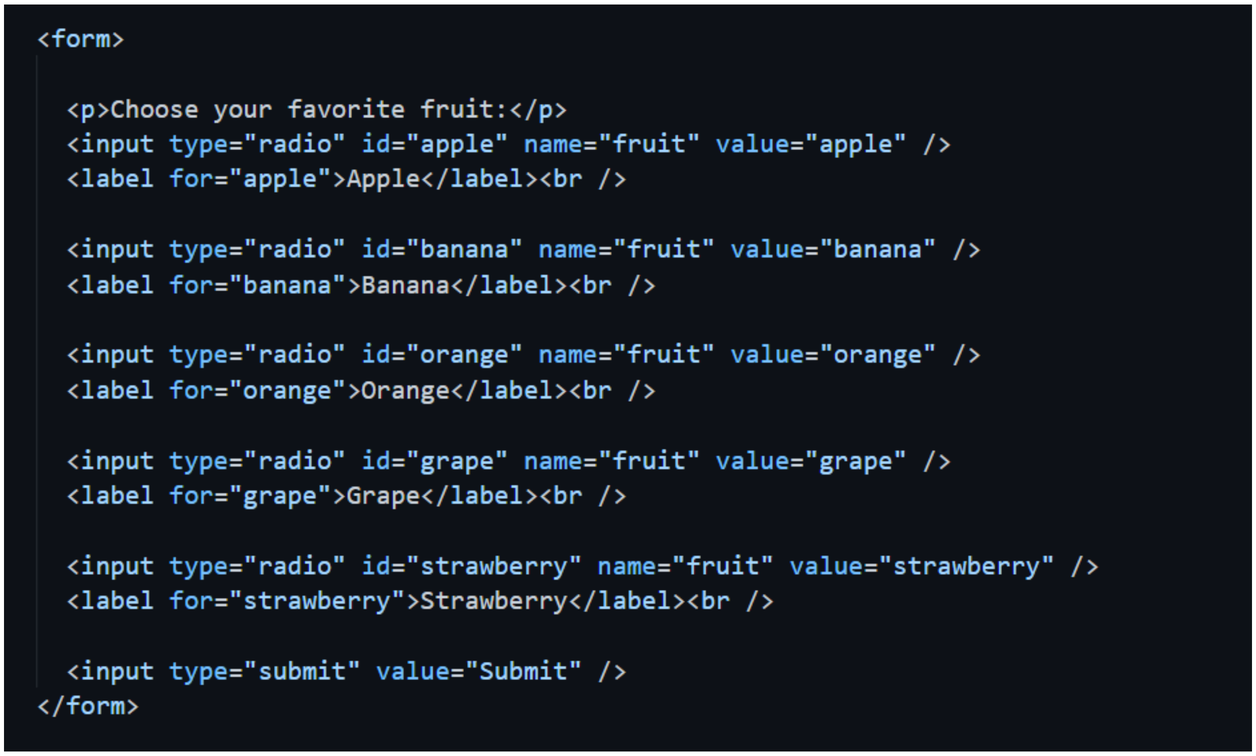
   - Used for search input fields, often with a built-in search icon.

Understanding The Radio Buttons a Bit More:

A radio button is a type of input element in HTML that allows users to select one option from a group of related options. Radio buttons are often used in forms when there are multiple choices, but the user should pick only one of them. They are called "radio" buttons because they work like the buttons on a car radio: when you select one station, it deselects the previously selected one.

Here's how you can create a group of radio buttons with an example:

Code:



```
<form>

  <p>Choose your favorite fruit:</p>
  <input type="radio" id="apple" name="fruit" value="apple" />
  <label for="apple">Apple</label><br />

  <input type="radio" id="banana" name="fruit" value="banana" />
  <label for="banana">Banana</label><br />

  <input type="radio" id="orange" name="fruit" value="orange" />
  <label for="orange">Orange</label><br />

  <input type="radio" id="grape" name="fruit" value="grape" />
  <label for="grape">Grape</label><br />

  <input type="radio" id="strawberry" name="fruit" value="strawberry" />
  <label for="strawberry">Strawberry</label><br />

  <input type="submit" value="Submit" />
</form>
```

In this example:

- We have a <form> element that contains a group of radio buttons.
- Each radio button is created using the <input> element with type="radio".
- The id attribute uniquely identifies each radio button.
- The name attribute groups the radio buttons together. In this case, all radio buttons have the same name attribute, "fruit," which means users can only select one fruit option.
- The value attribute specifies the value associated with each radio button.
- The <label> elements provide labels for each radio button.
- The attribute in the label is associated with the ID of the corresponding radio button, making it easier for users to select the option by clicking the label text.

When the user selects one radio button, it is highlighted, and the others are automatically deselected. When the form is submitted, the selected option's value is sent to the server as part of the form data, allowing you to process the user's choice.

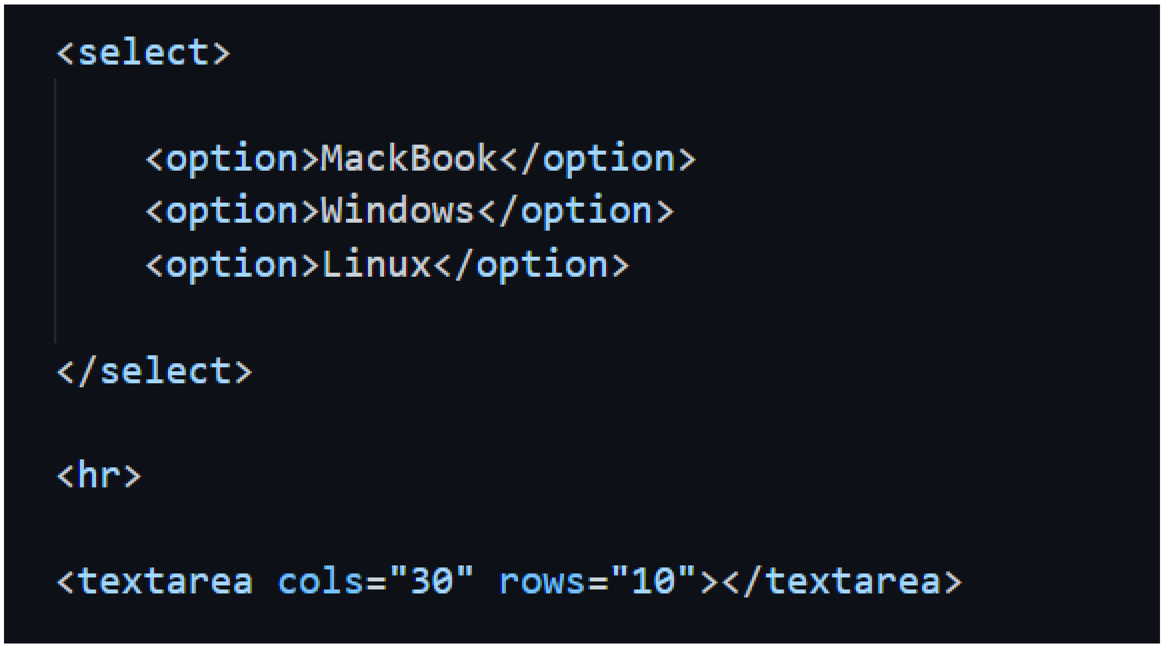
The Same Process is for checkboxes Also, Where We Can Select Multiple Options at a time

Now Remember “Id” is a Global Attribute, It Can Used in Any Tag, It is like an Identification of a tag

Now We Have Some Tags That are Not Input Tags But They Are Also Used For Taking Data From Users:

- **Text Area:** Used For Long Text Like Comments and Review.
- **Select & Option:** Used For Dropdowns.

Code:



```
<select>

    <option>MackBook</option>
    <option>Windows</option>
    <option>Linux</option>

</select>

<hr>

<textarea cols="30" rows="10"></textarea>
```

In the Text Area, There are Two Attributes “cols” and “rows” as the name subject they are,

Rows and columns are used to set the size of the text area.

Now After Learning All This, we can say have Covered a very Important Chapter Of HTML

And Now It’s The Perfect Time For Learning MetaData.

CHAPTER - 6:

Unlocking the Concept of Metadata: For Crafting Captivating Descriptions, Perfect Encoding, Dynamic Viewports, and Beyond

Metadata is like a hidden message that helps the internet work better. It's like a note t**hat tells web browsers and search engines** important things about a web page, such as **what the page is about, how it should be displayed, and other special instructions.** This hidden information makes sure that websites look right, show up in search results, and work well on different devices like phones and computers. *So, think of metadata as the secret sauce that makes the web more organized and user-friendly.*

These Tags are Written inside the Head Tag

Here's a list of common meta tags used in HTML and their explanations:

You don't have To Use & Remember all, There are a Few Tags Which are often used.

1. **<meta charset="UTF-8">**

Explanation: Specifies the character encoding for the document, ensuring that special characters and symbols are displayed correctly.

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

Explanation: Sets the viewport settings for responsive web design, ensuring the web page looks and scales properly on various devices like smartphones and tablets.

3. **<meta name="description" content="...">**

Explanation: Provide a brief description of the web page's content. This description is often used by search engines in search results to help users understand what the page is about.

4. **<meta name="keywords" content="...">**

Explanation: Lists keywords or phrases related to the content of the page. While less important for search engines today, it can still provide context for page content.

5. **<meta name="author" content="...">**

Explanation: Specifies the author's name or the entity responsible for the web page's content.

6. **<meta name="robots" content="...">**

Explanation: Instructs web crawlers (like search engine bots) on how to index and handle the page, controlling aspects such as whether it should be indexed or follow links.

7. **<meta http-equiv="refresh" content="5;url=https://example.com">**

Explanation: Redirects or refreshes the page after a specified time (in this case, 5 seconds) and directs the browser to load a different URL.

8. **<meta name="theme-color" content="#RRGGBB">**

Explanation: Defines the theme color for mobile devices' browser interface when the web page is added to the home screen or as a progressive web app (PWA).

9. **<meta name="apple-mobile-web-app-capable" content="yes">**

Explanation: Indicates that the web page can be viewed as a standalone app when added to the home screen on Apple devices.

10. **<meta name="apple-mobile-web-app-title" content="App Title">**

Explanation: Specifies the title that should appear when the web page is saved to the home screen as a web app on Apple devices.

11. **<meta property="og:title" content="...">**

Explanation: Part of the Open Graph protocol for social media sharing, defines the title when the page is shared on platforms like Facebook.

12. **<meta property="og:description" content="...">**

Explanation: Part of the Open Graph protocol, describes social media sharing.

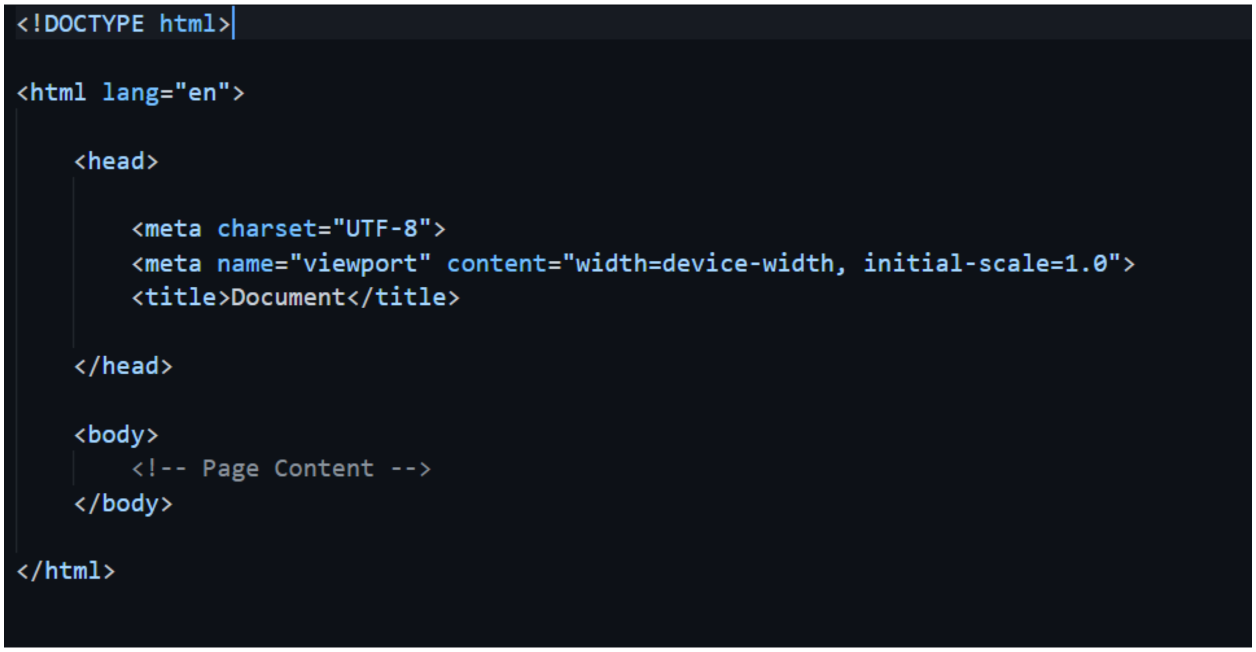
13. **<meta property="og:image" content="https://example.com/image.jpg">**

Explanation: Part of Open Graph protocol, specifies an image to be displayed when the page is shared on social media.

These meta tags play essential roles in providing information, enhancing user experience, and improving a web page's discoverability and appearance in search results and on social media platforms.

In Code:

Here We Are Using The Most Used Meta Data Tags:



```
<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Document</title>

    </head>

    <body>
        <!-- Page Content -->
    </body>

</html>
```

Now, You Can See Two More Things Beside Metadata:

- **Doctype:** it tells the web browser which version of HTML the page is written in so it can display it correctly
- **Lang Attribute:** It is like setting the language for your webpage, helping browsers and search engines understand and display content in the right language.

Now MetaData:

The Two Are The One Of The Most Frequent Ones:

- **Charset:** Charset is like telling your web page which alphabet and symbols to use, so it displays text properly. *As a Beginner, You Can Ignore It For Now.*
- **Viewport:** Viewport is like giving your webpage glasses, helping it fit and look good on different-sized screens like phones and computers.

Attributes Inside The Viewport Meta Data

width=device-width: This instructs the browser to set the width of the viewport to be equal to the width of the device's screen. It ensures that the web page's content adjusts to fit the screen's width, making it responsive.

initial-scale=1.0: This sets the initial zoom level of the page to 1.0, which means it's displayed at its original size when first loaded. Users can then zoom in or out as needed.

Favicons:

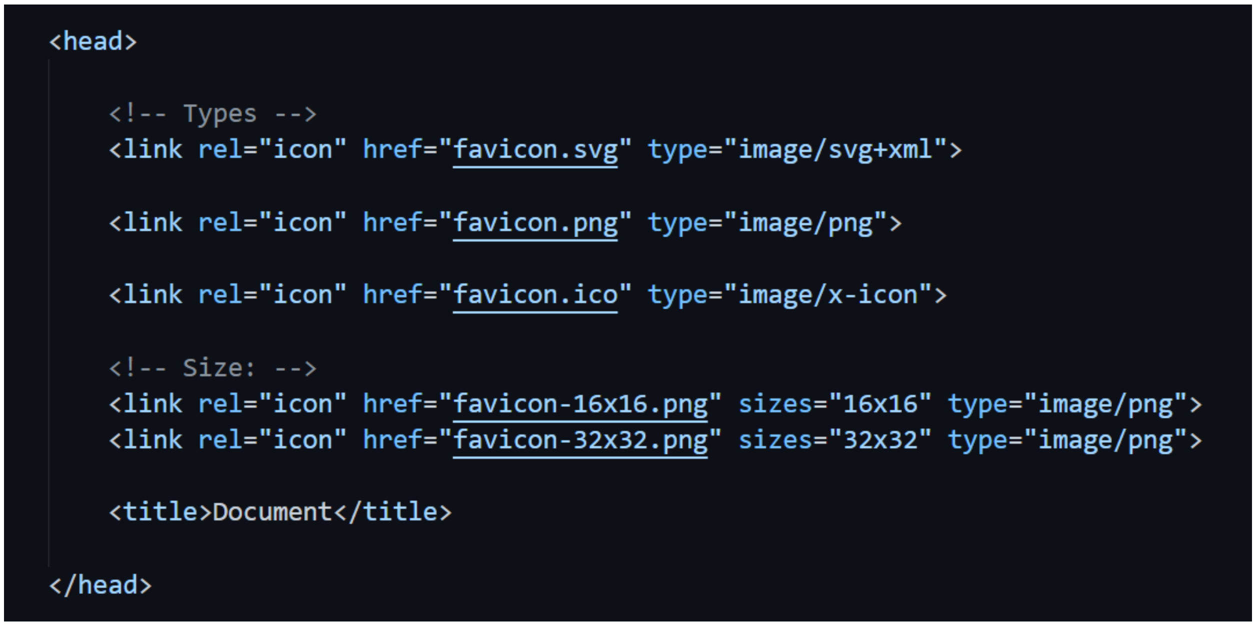
A **"favicon," short for "favorite icon,"** is a small image or icon that is associated with a website and is displayed in various places within web browsers. It serves several purposes:

1. **Browser Tabs:** Favicons appear in the tabs of web browsers, helping users quickly identify and switch between open tabs. They provide a visual representation of the website.
2. **Bookmark Icons:** When users bookmark a webpage, the favicon often becomes the icon associated with the bookmarked link, making it easily recognizable in the user's list of bookmarks.
3. **Address Bar:** In some browsers, the favicon is displayed in the address bar, reinforcing the website's branding or identity.
4. **Search Results:** Some search engines may display the favicon next to the website's search results, enhancing the website's visibility and making it more recognizable to users.

Favicons are typically small and square, often 16x16 pixels or 32x32 pixels in size, although higher resolutions are sometimes used for better quality on high-density screens. They are saved in common image formats like .ico, .png, or .svg and are linked to the website's HTML using a <link> element in the <head> section of the webpage. This allows browsers to associate the icon with the website and display it in the appropriate places.

Favicons are an important part of a website's branding and user experience, as they provide a consistent visual cue for users when interacting with the site.

In Code:



```html
<head>

    <!-- Types -->
    <link rel="icon" href="favicon.svg" type="image/svg+xml">

    <link rel="icon" href="favicon.png" type="image/png">

    <link rel="icon" href="favicon.ico" type="image/x-icon">

    <!-- Size: -->
    <link rel="icon" href="favicon-16x16.png" sizes="16x16" type="image/png">
    <link rel="icon" href="favicon-32x32.png" sizes="32x32" type="image/png">

    <title>Document</title>

</head>
```

Favicons can vary in size and file types. In this example, we'll demonstrate different options.

CHAPTER - 7:

Hidden Gems: Uncommon HTML Tags That Can Be Useful

Here is a list of some uncommon HTML tags that can be incredibly useful in various web development scenarios:

1. **<details>** and **<summary>:** Used to create expandable sections of content with a summary that users can click to reveal or hide additional information.

2. **<mark>:** Highlights text to make it stand out visually.

3. **<abbr>:** Defines an abbreviation or acronym, providing an explanation when users hover over it.

4. **<time>:** Specifies a specific date and time, which can be useful for events or publications.

5. **<meter>:** Creates a visual representation of a scalar measurement within a known range, such as a progress bar.

6. **<progress>:** Displays the progress of a task or process, like downloading a file.

7. **<cite>:** Indicates the title of a creative work, like a book or movie.

8. **<figcaption>:** Provides a caption for a **<figure>** element, commonly used for images or illustrations.

9. **<q>:** Defines inline quotations, adding quotation marks around the enclosed text.

10. **<small>:** Reduces the text size, often used for fine print or legal disclaimers.

11. **<code>:** Displays computer code, typically in a monospace font.

12. **<kbd>:** Represents keyboard input, helpful for displaying key combinations or shortcuts.

13. **<samp>:** Represents output from a computer program, script, or command.

14. **<sub>** and **<sup>:** Renders text as subscript or superscript, useful for mathematical formulas or footnotes.

15. **<ruby> and <rt>:** Used for displaying Ruby annotations, often seen in East Asian typography.

16. **<wbr>:** Suggests a word break opportunity within the text, aiding in better line breaking.

17. **<marquee>:** Element is an uncommon HTML tag that creates scrolling text or images within a web page. It's like a moving banner or ticker tape.

18. **<pre>:** This HTML element allows you to display text exactly as it's written, preserving both spaces and line breaks, without imposing any fixed formatting.

19. **<canvas>:** This HTML element provides a blank, drawable area where you can use JavaScript to create graphics, charts, animations, and other visual elements without any predefined formatting.

Congratulations on completing this journey through the world of Full Stack Development! 🚀

As you reach the end of these notes, remember that your dedication and passion for this field will be the driving force behind your success. Learning is a continuous adventure, and your curiosity will lead you to new horizons.

Always keep asking questions, as they are the keys that unlock the treasure chest of knowledge. Embrace the challenges, celebrate your victories, and keep striving for excellence.

Wishing you the very best in your Full Stack Development endeavors! May your path be filled with exciting projects, valuable experiences, and endless opportunities.

Safe travels on your coding journey, and may your code always run smoothly.

See You In CSS Now.

All the best,

THE PRIME STEP