**CHAPTER 2: INTRODUCTION**

1: HTTP is the protocol (A set of rules) by which we send and receive information through internet. (VIDEO 13)

2: Frontend code is the code that runs in our browser (HTML, CSS, JS). Backend code is the code that runs in the server. (VIDEO 14)

**CHAPTER 3: HTML ESSENTIALS**

1: HTML (Hypertext markup language): The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. (Use MDN docs for more information about web development)

Common elements include:

1. <p> element - represents a paragraph
2. <h1> element – represents header
3. <img> element – embeds an image
4. <form> element – represents a form

2: We should always have only one H1 tag at most on a page and it should always be the top level main heading (VIDEO 23)

3: Html elements like <p> ,heading tags, <ol>, <ul>, <li> etc start on a new line. Other tags cannot use the line occupied by these tags even if there is some space left. The other tags will start on a new line as shown below with the <b> tag.

<p>

  The HyperText Markup Language, or HTML is the standard markup language for

  documents designed to be displayed in a web browser.

</p><b>some random text </b>

<p>

  It can be assisted by technologies such as Cascading Style Sheets and

  scripting languages such as JavaScript

</p>

It will be displayed in the browser like this.

The HyperText Markup Language, or HTML is the standard markup language for doc

uments designed to be displayed in a web browser.

some random text

It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript

(VIDEO 23)

And some elements like <b>, <a>, <img> etc., do not start on a new line.

4: We should never have a h3 tag without h2 or h1 or h4 without h3 and so on. We should proceed sequentially.  (VIDEO 23)

5: **HTML SKELETON**

<!DOCTYPE html>

<html>

  <head>

    <title>My Document</title>

  </head>

  <body></body>

</html>

**Every HTML document should have this structure**

1. <!DOCTYPE html> -> It indicates that we are using HTML 5  -> <!DOCTYPE html> is not a tag. We can think of it as a flag which says we are using HTML5.
2. The <html> tag is the root element. All other elements must be descendants of this element. Permitted content in the html element is one <head> element, followed by one <body> element
3. The head tag contains machine readable information (metadata) like title, scripts and style sheets which do not get displayed in the document. In the head element, there is a title element “My Document” which means the browser tab’s name will be “My Document”. (Meta data is the data which provides information about the main content (data))
4. The <body> HTML element represents the content of an HTML document. There can be only one <body> element in a document.

Shortcut for this in VS code: Just type ! and hit tab. (VIDEO 26)

6: Lists in html can be ordered and unordered. Ordered list is represented as <ol> and unordered list is <ul>

    <ol>

        <li>ordered list item1</li>

        <li>ordered list item2</li>

        <li>ordered list item3</li>

    </ol>

    <ul>

        <li>unordered list item1</li>

        <li>unordered list item2</li>

        <li>unordered list item3</li>

    </ul>

<li> tag represents list item.

The ordered list or unordered list cannot contain nested **ol** or **ul** tag directly but we can do so by inserting the **ol** or **ul** tag in the **li** tag as shown below.

    <ul>

        <li>unordered list item1

            <ol>

                <li>nested item1</li>

                <li>nested item2</li>

            </ol>

        </li>

        <li>unordered list item2</li>

        <li>unordered list item3</li>

    </ul>

(VIDEO 27)

6: Anchor tags (<a> tag): the <a> tag is mostly used to create links

Syntax:

<a href="https://www.google.com">Google</a>

href is the attribute which contains the destination website. And the text between the opening and closing tags is displayed. Currently the document is in “File” protocol. Therefore to make http requests we need to include http in the website. So the following will not work and we will end up in going to currentFolder/www.google.com file.

<a href="www.google.com">Google</a>

We can use this to move to another html page. Let’s say we create a html file “about.html” and on clicking the link we go to this document. We can do so using the anchor (<a>) tag

<a href="about.html">About</a>

Or we have a folder which contains this html file

<a href="folderName/about.html">About</a>

(VIDEO 28)

7: Images

    <img

      src="link to that image or local image address"

      alt="An alternate text"

     >

The alternate text is showed if the image is not loaded. Remember that img has **no closing tag.**

8: Comments in html has the following syntax

<!-- This is a comment -->

**CHAPTER 4: HTML NEXT-STEPS AND SEMANTICS**

1: HTML Standard is a document that describes how HTML should work. (Describes the rules of HTML. Does not contain the implementation)

2: The browsers actually have to do the work and implement HTML according to those rules.

3: HTML5 is the latest evolution of standard that defines HTML. It includes new elements and features for browsers to implement.

(VIDEO 32)

4: **Inline elements and Block Elements**

1. Inline elements fit alongside other elements (Eg: <b>, <img>, <a> etc.,)
2. Block Elements take a block of space (Eg: header tags, <p>, <ol>, <ul>, <li> etc.,)

5: **<div>: Div is the content division element. The HTML Content division element <div> is the generic container for the content.** Div is a block element. The div element just groups together the content it contains.

6: **<span>: Span is also a generic container like div BUT it is inline unlike div**

(VIDEO 33)

7: <hr> tag represents horizontal ruling or horizontal line. It is used in between paragraphs

    <p>First Paragraph about wolves</p>

    <hr>

    <p>Second paragraph about wolves</p>

Note that hr element has no closing tag like img.

<br> element represents the line break element.

    <p>

      Lorem ipsum dolor, sit amet consectetur adipisicing elit. <br>

      iusto aliquam ab, amet necessitatibus. Assumenda mollitia tempore

    </p>

The output is shown below.

Lorem ipsum dolor, sit amet consectetur adipisicing elit.

iusto aliquam ab, amet necessitatibus. Assumenda mollitia tempore

 <sup> and <sub> represents superscript and subscript respectively.

<h2>Pythagorous Theorem</h2>

<p>a<sup>2</sup> + b<sup>2</sup> = c<sup>2</sup></p>

(VIDEO 34)

8: **HTML Entities**

1. Starts with an **ampersand** and end with a **semicolon**.
2. Used to display reserved characters, that normally would be invalid. (Like a diamond, arrow, Greek letters, less than symbol, greater than symbol etc)

Html entity for < is **&lt;**

Htmlentity for & is **&amp;**

1. Also used in place of difficult to type characters
2. The browser interprets them and renders the correct character instead.

(VIDEO 35)

9: **Intro to semantic markup**

 Elements like header, main, nav, section, footer, article, aside, summary, details etc., behave exactly the same as a div element but they add more meaning to the content. Instead of using generic containers like div or span, we should use these elements to add more meaning.

1. <main> element should be unique to the document. Content that is repeated across a set of documents or document sections such as sidebars, navigation links, copyright information, site logos and search forms should not be included unless the search form is the main function of the page.
2. <nav> element provides navigation links in navigation bars.
3. <section> element represents standalone section of a document.
4. <article> element generally includes forum post, a magazine, newspaper article, blog entry etc., Each article should be identified by including a heading as a child to the <article> element.
5. <header> tag represents the introductory content, heading elements, logo, search form, author name, etc., It can also contain navigational content (VIDEO 36, 37)

Generally, the layout is like this

<header></header>

<main></main>

<footer></footer>

**CHAPTER 5: HTML FORMS AND TABLES**

1:

<table>

  <caption>

    Council budget (in £) 2018

  </caption>

  <thead>

    <!--thead: table header-->

    <tr>

      <!--tr: table row-->

      <th scope="row">Items</th>

      <!--th: header for a group of cells defined by its scope-->

      <th scope="row">Expenditure</th>

    </tr>

  </thead>

  <tbody>

    <!--tbody represents the table body-->

    <tr>

      <th scope="row">Donuts</th>

      <td>3,000</td>

      <!--td: table data cell-->

    </tr>

    <tr>

      <th scope="row">Stationery</th>

      <td>18,000</td>

    </tr>

  </tbody>

  <tfoot>

    <!--tfoot represents the table footer-->

    <tr>

      <th scope="row">Total Expenditure</th>

      <td>21,000</td>

    </tr>

    <tr></tr>

  </tfoot>

</table>

    <table>

      <tr>

        <th rowspan="2">Animal</th>

        <!—row span = 2 means that the cell is 2 times the height of a row-->

        <th colspan="2">Average Mass</th>

        <th colspan="2">Maximum Mass</th>

        <th rowspan="2">Flighted</th>

      </tr>

      <tr>

          <td>KG</td>

          <td>Pounds</td>

          <td>KG</td>

          <td>Pounds</td>

      </tr>

      <tr>

        <!--tr represents table row-->

        <td>Ostrich</td>

        <!--td represents a single cell (table data cell)-->

        <td>104</td>

        <td>230</td>

        <td>156.8</td>

        <td>346</td>

        <td>No</td>

      </tr>

      <tr>

        <td>Somali Ostrich</td>

        <td>90</td>

        <td>200</td>

        <td>130</td>

        <td>287</td>

        <td>No</td>

      </tr>

    </table>

(VIDEO 45)

2: **Forms**:

1. In forms, the action attribute specifies where the data should be sent.
2. The method attribute specifies which HTTP method should be used. The most important element in forms is the input element.
3. The type attribute in the input is very versatile. It can be a number, password, colour picker, image picker, email, file picker, checkbox, radio etc.,
4. The **placeholder** in the input element is the text which would show when the input field is empty.
5. The default behaviour of a button in forms is to submit the form. So always specify the type of button like button or submit.
6. The html **label** element is used with the input field. Whenever the label text is clicked, the input field linked to that label is focussed. The linking is done using **id** attribute for input and **for** attribute for label

    <form action="about.html">

      <label for = "password">password</label>

      <input type="password" id = "password">

    </form>

1. The name attribute in the input field goes with the form action when form is submitted.

    <form action="/about.html">

        <div>

            <label for="username">UserName</label>

            <input type="text" placeholder="username" name="username">

        </div>

        <div>

            <label for = "password">password</label>

            <input type="password" id = "password" name = "password" placeholder="password">

        </div>

        <button type = "submit">Submit</button>

    </form>

When we click the submit button, we will be directed to

/about.html?username=superman&password=1234

The general pattern is url?name1=enteredInput1&name2=enteredInput2 where url is the same url in form action attribute. (With radio buttons, we use same **name** field to group the radio buttons so that only one radio button can be clicked. And we use **value** field to send the form data in the **action** address of the form for both radio and checkbox)

If we also hit enter button in the input field, it will submit the form.

Drop down options is created using **select** and **option** element.

<select name="pets" id="pet-select">

    <option value="">--Please choose an option--</option> <!--It will be shown by default-->

    <option value="dog">Dog</option>

    <option value="cat">Cat</option>

    <option value="hamster">Hamster</option>

</select>

Ranges are also very popular inputs used in forms.

3: To allow users to enter paragraph or feedback or any free-form text, we can use **<textarea>** element which is **NOT** an input element.

<textarea id="story" name="story" rows="5" cols="33">

It was a dark and stormy night...

</textarea>

4: **required** attribute:

    <form action="dummy">

        <label for="username">username</label>

        <input type="text" name="username" id="username" minlength="2">

        <button>Submit</button>

    </form>

minLength is 2 that means we can submit usernames of minimum length of 2. But if we keep the username field empty, the form will still get submitted. To avoid this, we use the **required** attribute.

    <form action="dummy">

      <label for="username">username</label>

      <input type="text" name="username" id="username" minlength="2" required>

      <button>Submit</button>

    </form>

5: We can also use the **pattern** attribute in the input fields which specifies a regular expression (regex).