**Frontend Development:**

the part of an application or website that users directly see and interact with.

**Key aspects:**

* User interface (UI): the design and layout of the application/website
* User experience (UE): how and what users interact with the application.
* Core technologies: HTML, CSS, JavaScript
* Client-side development: FE code runs in the user’s browsers, making it accessible directly to them

**Core Technologies:**

HTML: hypertext markup language

to create structure of the webpage.

CSS: cascading style sheets

To style and layout web page.

JS: java script

Adds interactivity and dynamic behaviour to the web application.

**HTML**

Hyper Text Markup Language

To create the structure of a webpage

Webpage: a page that loads data from the server to client in a web browser

Different resolutions: 1280, 1338, 1536, 1640, 1920

HMTL has elements and attributes.

Boiler plate code – common code for almost any page. Can be obtained by ! + enter key. This is called emmet abbreviation, a tool kit to get the boiler plate code without typing everything.

<!DOCTYPE html> tells browser the html version we are using

<html lang="en"> html tag, root of a html doc, the biggest element of a webpage containing many other elements

<head> container for metadata, data that doesn’t get displayed on the browser

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0"> viewport is for responsive responsiveness (aspect ratio- size of device)

    <title>Document</title> contains the title of the page

</head> contains the data that the browser renders

<body>

</body>

</html>

We can inspect the code in view page source or in inspect. And can also make any changes in inspect.

Comments: <!—single line comment-->

tags - are names used to represent a particular format of content

tags + content => element

1. Block: group of elements

<elementNm></elementNm>

Eg: <p></p>, <div></div>

1. Inline: an element which opens & closes on itself.

<element/>

Eg: <input/>

Attributes – used to add additional characteristics/ properties to the elements

<elementNm attributeName= “attribute” />

<elementNm attributeName= “attribute” ><elementNm/>

Topics:

formatting elements - <strong> bold chars, <br/> break the lines, <u> underline

anchor elements – links <a> <a/>

lists – ordered lists <ol>, unordered lists <ul>, list items <li>, definition list of items <dl>, def term <dt>, def data <dd>

tables - <table>, table row <tr>, table head <th>, table data <td>, <caption> to mention heading of the table, table has rows and rows has columns.

Forms - <input type= “text/number/email” checked required />

checked – sets default value, required – can’t leave it empty, definitely should give a value

field set - <fieldset> <legend> heading for fieldset, <label> for titles, <input />

media objects - <video controls width = “150px” src = “path” />

<audio controls src = “path”/>

Controls – full screen option, play pause options.

semantic elements - <main> <header> <nav> <section> <article> <footer>

**CSS**

Cascading style sheets

Purpose: to design or style a webpage

Types/ methods:

Inline – used to style a sp element

Internal - <style> in head section of html

External – links an external style file in the head section

Selectors:

class (.) – selects html ele with sp class attribute

id (#) – selects html ele with sp id attribute

tags – selects html tag names

universal (\*) – selects entire page.

Color systems:

Rgb – red, green, blue, range: 0-255

Color: rgb(0,255,0) – green, Rgb(255,0,0)- red, Rgb(0,0,255)- blue

Rgb(255,255,0)- yellow

Hexadecimal (hex) hex= 6, decimal-10 🡪 total 16 digits

contains 0-9 (decimal num) + a,b,c,d,e,f (6 alphabets)

color: #ff0000 – red, #00ff00 – green, #0000ff- blue

comments: /\* comments \*/

text-properties:

text-align, text-decoration, font-weight, font-family, font-styles, line-height, text-transformation

font-family: arial, roboto, geneva

-if arial is not supported roboto will be applied, if roboto is not supported Geneva will be applied. This is called fall back mechanism

Box-model:

Margin - gap btw the main element and console or other elements,

Border – boundary,

padding – space around the content,

height & width of content

display properties:

inline – takes only the space required by the element (without any margin/padding)

block – takes full width space

inline-block – similar to inline but we can set margin and padding

none – to remove element from document flow