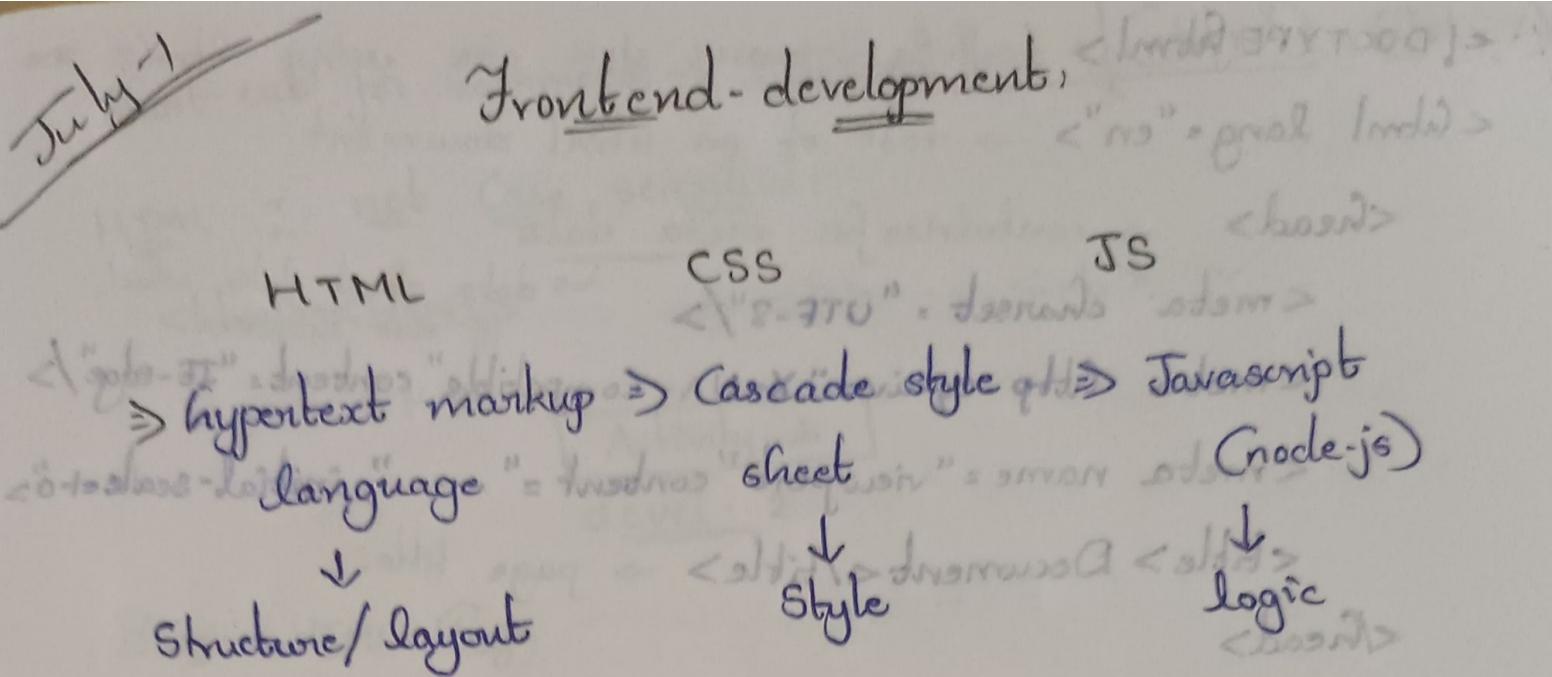


feel



1) HTML

↳ It is the code that is used to structure a web page and its content.

→ The components used to bring the design of the structure of websites are called HTML Tags.

HTML file: File_name.html

Note:- Extension used for html in vs code is open server. which is used to run html on saving the code if you run it on live server.

Emmet tool:-

1. !! → Emmet Abbreviation → Gives the basic html code in visual studio code.

→ The basic or sample code is known as

Boiler Plate code.

⇒ The sample code is,

<div> JMTT no or sint --!>

<!DOCTYPE html> → tells Browser you are using HTML document

<html lang="en"> → root of an HTML document

<head> → container for meta data → data about data which is not shown

<meta charset="UTF-8"/>

<meta http-equiv="X-UA-Compatible" content="IE-edge"/>

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<title> Document </title> → page title.

</head>

<body></body> → contains all data rendered by the browser

</html>

HTML Tags:
A container for some content or other HTML Tags.

<> → Opening Tag
↳ Angular brackets

</> → closing Tag

Example: <p> This is a paragraph </p>
↳ Content of tag is in bold

Element → Tags + content.

* In VS Code just type the opening tag and it will give you a closing tag

→ It is not compulsory for a tag to have content in between, eg -

Comments in HTML

→ This is the part of code that should not be parsed.

<!-- This is an HTML Comment -->

** Short-cut for comment → Selected text + [Ctrl+?] @ [Ctrl+]

HTML is not Case sensitive:

<html> <HTML> → <p> - - - </p> ✓

<p> = <P>

Activity - 1

Level - 2

Basic HTML Tags:

HTML ATTRIBUTES:

→ Attributes are used to add more information to

The tag. value → Single quotes or double quotes can

Ex. <html lang="en">
language Attribute → English.

Heading Tag:

→ Used to display headings in HTML.

h1 (most important)

h2

h3

h4

h5

(size & importance).

Note: Use this tag for showing importance not

the size.

h6 (least important)

Paragraph Tag:

→ Used to add paragraphs in HTML

<p> This is a sample program </p>

Note: If you have a big paragraph in one line or more than one line and you are getting a horizontal scroll bar.

Then, do the following,

→ Click Settings after selecting the text → Command palette → View: Toggle Word Wrap for all JMTA

It doesn't add extra numbers in code
↓
Anchor Tag: <9> = <9>

→ Used to add links to your page.

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

↓ link displayed on the website.
It is opened in new webpage of absolute

! ⇒ You can even give file names in href like

<u href = "/Hello.html"> Go to Hello

↓ saved code or file. → Relative link or path

⇒ If the file is in another folder then anchor tag

↓ will be
<u href = "/HelloFolder/Hello.html"> Go to Hello
↓ name of the folder.

Image Tag:

→ Used to add images to your page

<img scr = "relative url or absolute url" alt = "Random img"

Attribute.

↓ either saved images or optional
from image addresses.

alt = "Random img" } → used to print the given line

↓ tag if no image is present in
scr or when the image fails
to load in.

Br Tag:

- Used to add next line (line breaks) to your page.
- ↳ Doesn't have any attributes.

Bold, Italic & underline Tags:

- Used to highlight or change text in your page.

** Bold ** → **Bold**

<i> Italic </i> → *Italic*

<u> Underline </u> → Underline

↳ You can use these tags inside other tags.

Big & Small Tags:

- Used to display big & small text on your page.

<big> BIG </big> → **BIG**

<small> SMALL </small> → **SMALL**

Hr Tag:

- Used to display a horizontal ruler, used to separate content.

<hr>

Subscript or Superscript Tag:

_{subscript}

H_2O

^{superscript}

$\text{A}^n + \text{B}$

⇒ H_2O

⇒ $3^2 = 9$

Pre Tag

⇒ Used to display text as it is (without ignoring spaces & next line).

* → → Gives a big sample text.

<pre> This is a book > This is a book

<pre> This is a book > This is a book

→ Even works for next line. > another book </u>

Activity - 2

Level-3

Page layout Techniques

Using Semantic tags for layout

→ more easy to read and

→ more structured

→ Good user experience.

Meaning is from the word.

Semantic

header, footer.

Non-Semantic

span, div.

using the Right Tags

<header> Tag: → Inside body tag

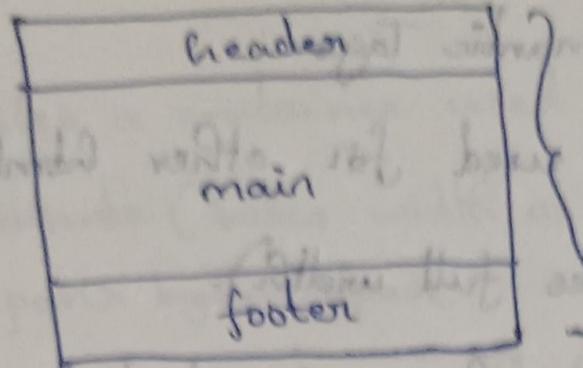
↳ Gives the main head part for the website

<main> Tag:

↳ Gives the body of the website

<footer> Tag:

↳ Gives the main bottom part of the website



Inside Main Tag:

Section Tag: → For a Section on your page
`<section>`

Article Tag: → For an article on your page
`<article>`

Aside Tag: → For content aside main content (ads)
`<aside>` ↳ used to show ads.

Revisiting Anchor Tag:

`Google`

→ used to open the link in new tab.
 ↳ for new tab

``

Used as a clickable pic to open the link.
 ↳ link of Image.
 ↳ can be added here.
 ↳ target = "-main"

Revisiting Image Tag:

``
 ↳ set height

``
 ↳ set width

Aspect Ratio = $\frac{\text{height}}{\text{width}}$.

↳ Constant

So only one (i.e. h or w) can be adjusted at Once.

* Div Tag → Non-Semantic Tags

Div is a container used for other html elements.

** Block Elements (takes full width).

Use of Div is any changes applied to the div takes effect on its contained elements.

List of Block Element Tags: (Tags):

<div>

<div>

<div>

<div>

<div>

dot over rot

-dot over ni strikethrough or bear <

→ <div> <div> = degree "mas-algoopl1:egfdw" = font >

Eg: <div>

↳ default width is 0.

<p> bye </p>
<p> goodbye </p>
<p> ciao </p>
</div>

1 div.
red
background-color:red;
not "num_1" = degree
and bebe ad

Output

color is changed to
all the components in div.

bye
goodbye
ciao

<x902=alpian> "div" = ore pmi >

alpian

<x902=Abiau> "div" = ore red. pmi >

Abiau

Span - Tag:

→ Span is also a container used for other HTML elements
 inline Elements (takes width as per size).

Syntax: ` bye `

Eg: `bye` → If background-color = red;
 O/p: `bye`

bye `<il>@qqA <il>` `<il>@qqA <il>`
 List of Span Tags: `<il>` `<il>ognuM <il>` `<il>`

→ background-color defines the color of the entire element.

→ color defines the color of the text.

→ font-family defines the font style of the text.

→ font-size defines the size of the text.

→ font-style defines the style of the text.

→ font-weight defines the weight of the text.

→ text-decoration defines the decoration of the text.

→ text-align defines the alignment of the text.

→ text-indent defines the indentation of the text.

→ text-decoration defines the decoration of the text.

→ text-indent defines the indentation of the text.

→ text-align defines the alignment of the text.

→ text-decoration defines the decoration of the text.

→ text-indent defines the indentation of the text.

→ text-align defines the alignment of the text.

→ text-decoration defines the decoration of the text.

→ text-indent defines the indentation of the text.

→ text-align defines the alignment of the text.

→ text-decoration defines the decoration of the text.

Level - Pro (or) Four

Lists in HTML:

lists are used to represent ordered list data.

class = relax
unordered list : unordered :-
Syntax:

list ← Apple
item. Mango

ordered 1 I a
2 II b
3 III c
4 IV d

 Apple
 Mango

Unordered-list:-

→ We can change the circular bullet points into square, disc.

⇒ * list-style-type: none ; should be written in
disc → ●
circle → ○
square → ■
lower-roman → iv..

Ordered-list:-

⇒ list-style-type: decimal (1, 2, 3)
decimal-leading-zero (01, 02, 03)
lower-alpha (a, b, c)
upper-alpha (A, B, C)
lower-roman (i, ii, iii)
upper-roman (I, II, III)

⇒ Definition-list: → made up of set of terms along with the definitions for each of those terms.

<dl> <dt> <dd>
 ↓ ↓ ↓
definition list. definition term definition

Eg:-

<dl>
 <dt> Sashimi</dt>
 <dd> Sliced raw fish that is served with condiments such as shredded daikon radish</dd>
</dl>

⇒ Sashimi

Sliced raw fish that is served with condiments such as shredded daikon radish or ginger root, wasabi & soy sauce.

* → You can also have nested lists (one list inside another).

⇒ Tables in HTML:

→ These are used to represent real life data.

1) `<table>` ⇒ used to create a table in which contents or elements are added row by row.

2) `<tr>` ⇒ (Table row) indicates the start of each row

3) `<td>` ⇒ (Table data) used to display table data.

4) `<th>` ⇒ (Table header) to represent the heading for a row or column.

↳ `scope` ⇒ It is an attribute used to show whether the given `<th>` is for row or column.

⇒ `<th scope="row">` Row -

⇒ `<th scope="column">` ---

* Note: `<td>` starts from left to right & `<th>` starts from left to right.

⇒ `<td>` ⇒ The content in this element starts from left of the cell & goes on. (left-indenting)

⇒ `<th>` ⇒ The content in this cell occupies the middle space and is bold.

Eg:

```
<table>
  <tr>
    <th></th>
    <th scope="col"> Saturday </th>
  </tr>
  <tr>
    <th scope="row"> Tickets sold: </th>
    <td> 120 </td>
  </tr>
</table>
```

Spanning - columns

- columns does nothing or don't do it and both
 → used to create cells which spans over multiple columns.

→ `<td colspan="2"> ... </td>`

| | | |
|--|--|--|
| | | |
| | | |
| | | |

occupies 2 columns

worth of size = 2 cells

works to drop out width (or id) < `<td>` (

Spanning - rows

- used to create cells which spans over multiple rows -

→ `<td rowspan="2"> ... </td>`

| | | |
|-------------------------|--|--|
| | | |
| occupies 2 rows | | |
| worth of size = 2 cells | | |

Long - tables

- 1) `<thead>` → Headings of the table should be kept inside `<thead>`
- 2) `<tbody>` → It represents the body of the table
- 3) `<tfoot>` → It represents the footer of the table

Table - attributes

- 1) `width="400"` → Indicate how wide the table should be
- 2) `cellpadding="10"` → Used to add space inside the cell
- 3) `cellspacing="5"` → Used to create space btw cells on all sides
- 4) (Border) `border="2"` → Used to create or provides borders for the table
- 5) `bgcolor="Blue"` → Gives background color to the table cells.

Forms in HTML

→ Forms are used to collect data from the user and also allow the user to perform other actions -
Eg: sign up/login/help requests/contact me.

Form controls:

→ a user interface or a point of connection b/w the user and the server.

→ Adding text.

→ Text input, Password input, Text Area

→ Making choices

→ Radio buttons, Checkboxes, Drop-down boxes.

→ Submitting forms

→ Submit buttons, Image buttons.

→ Uploading files

→ File upload.

Action in form:-

→ Used to define what action needs to be performed when a form is submitted.

Eg: <form action="/action.php"></form>

= TEXT-Input-

How Form Works

User fills the
form & clicks
Submit.

②

form control is
sent to the
server along with
values given by user.

③

Server processes
info using a
prog lang &
may store info in
a DB -

④

Server creates a
new page to send
back to the browser.

Form methods
and more values are added
to the end of url upon
submitting or now post. Values are set in
HTTP headers.

→ short forms

→ long forms with user uploads.

→ contain sensitive info (pwd)

→ Just retrieving data.

→ not sending personal info to confidential user etc.

⇒ Text-inputs (input)

→ <input type="text", name="broccoli", size="15", type="text">

words present
depending upon
type.

1) text

< input type="text" name="Username" size="15" maxlength="30" placeholder="Enter text" />

→ Enter text.

max no. of
char's.

2) Password

< input type="password" name="PassWord" />

→ Enter pwd →

→ hidden.

3) Text-Area

< textarea name="textarea" rows="3" cols="20" > Type </textarea>

→ Type...
placeholder
rows
cols

3 charrows
shown.

Not placeholder.
but text written
by default.

3) Radio-buttons: (Only 1 at a time)

`<input type="radio" name="A" value="male" checked="checked"/> Male`

`<input type="radio" name="A" value="Female"/> Female.`

A } (denotes group)

B { `<input type="radio" name="B" value="Apple"/> Apple.`

auto selected.

⇒ Male Female | Apple.

Individual selector → "Male" "Female" "Apple".

One group - Other group. → "Apple".

Can only select one.

4) Check-box:

`<input type="checkbox" name="A" value="A"/> A`

`<input type="checkbox" name="B" value="B"/> B.`

⇒ A B. → A B

both can be selected.

5) Select (`<select></select>`)

`<select name="Fruits" size="2">`

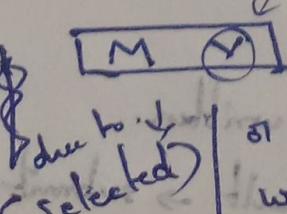
→ `<option value="M" selected> M </option>`

`<option value="P"> P </option>`

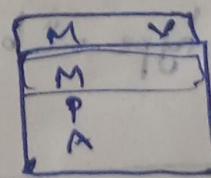
`<option value="A"> A </option>`

select multiple with Ctrl + click.

6) `</select>`

⇒ 

due to (selected) or first option will be selected by default.



- File Input Box (looks out for label & input box placement),
 - occupies entire block.

```
<input type="file" name="file"/>
```

should def
are post)

choose file } File not chosen.

click on this

6) Submit - button;

```
<input type="Submit" name="Button" value="Subscribe"/>
```

\Rightarrow Subscribe

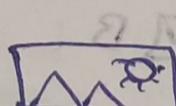
↓
button inside
process text

7) Image - button

~~image~~

```
<input type="file" src="images/1.jpg" alt="Submit">>  
<input color="white" border="1" width="20" height="10">>
```

23



← clickable.

3) Reset - button,

<input type = "reset" value = "Reset"/>

⇒ **Reset** → Resets all input field user values

⇒ Button Tags (just a clickable button)

```
<button type="button"> Click me</button>
```

2) click me.

gf not written

default → submit

⇒ LABELS:

⇒ <label> Age: <input type="text"/> </label>

↓
for packing (name + input)

⇒ <label for="Username"> Username: </label>
<input type="text" id="Username" />

when click
Username it
highlights the
input field.

for better understand.

X July 5th.

⇒ FIELDSET:

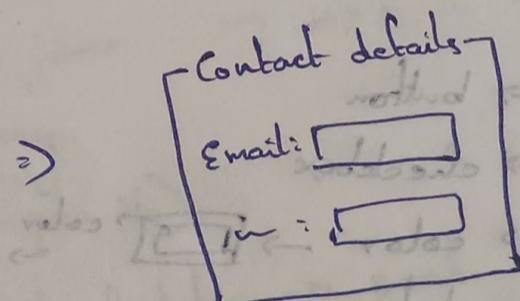
<fieldset>

<legend> Contact Details </legend>

<label> Email:

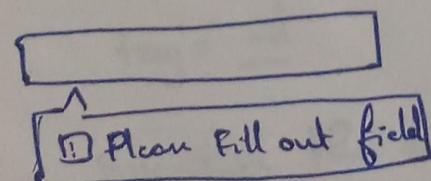
<input type="text" name="name"/> </label>

</fieldset>



⇒ Form - Validations:

<input type="text" required>



→ if submit is clicked without
entering data.

- search
- submit
- tel
- text
- time
- url
- week.

extra fields 9/20/18

EXTRA MARKUP:

HTML 5 → <!DOCTYPE html>

<html> <head> <body>
tells browser which version of HTML
the page is using.

comments → <!-- comment -->

id → unique identifier → not sent to server

class → for grouping
not unique → not sent to server

name → identifier form
data key
(unique - per ip) → sent to browser ✓

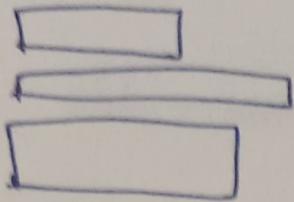
value → stores actual data → sent to browser ✓
of a field → not sent to browser

for → associates label → Unique → not sent to browser
with the input tag's id

type → specifies the kind → not sent to browser.
of input

BLOCK ELEMENTS

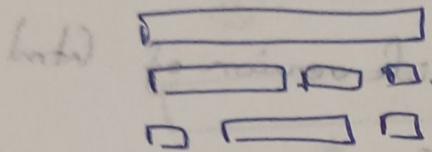
These appear on a new line always.



$\rightarrow <\text{hi}> <\text{p}> <\text{ul}> <\text{li}>$ —
 |
 |
 |
 |
 |

INLINE ELEMENTS

INLINE ELEMENTS
They appear to continue on the same line as others
and they do too! > ←



$\Rightarrow \langle a \rangle, \langle b \rangle, \langle em \rangle, \langle img \rangle$

□ □ □

— 14 —

<iframe

src = "https://en.wikipedia.org/wiki/HTML"

width: "100" don't forget set width

height = "400"
expire. Jan

> </frame>

⇒ INFO ABOUT YOUR PAGE: (inside head)
(Get reg-expire.)

<meta charset="UTF-8"/> below wrote & who

```
→ <meta name="viewport" content="width=device-width,  
initial-scale=1.0" />
```

→ makes name = "description" content = "Installation guide" />

tags mit Stift
bei einer

→ <meta http-equiv="author" content="Jon Duckett"/>



