11

Frond End + Back End ⇒ Full Stack

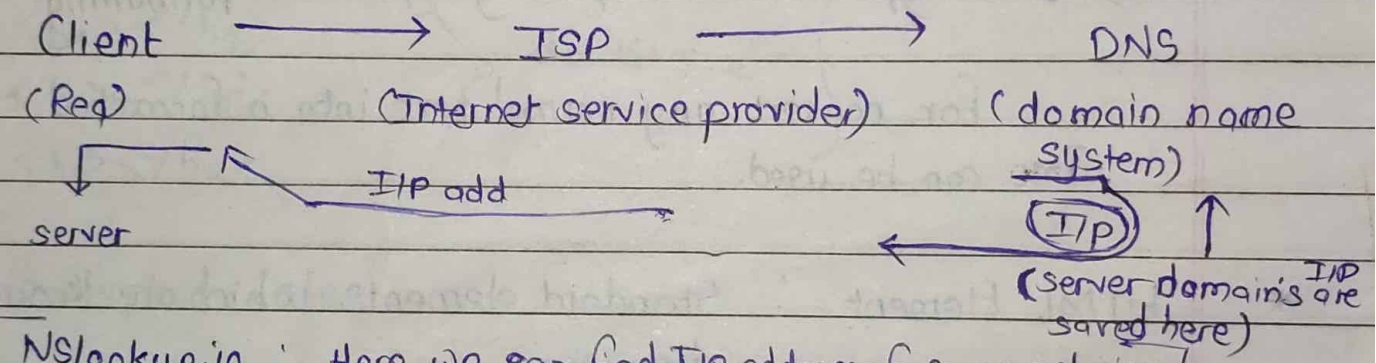
Page No.
Date

Stack means technologies

stackoverflow

Internet : Interconnected network of computers around world

Server : It is 24/7 online. (any device or system)
Clients : Asks for data ie. request for information
→ Server: Server serves this information thats why it is called server. (serves info)
Client: (Requests info)
IP address : There is a different IP address of any comp or website or server or any mobile.



Nslookup.io : Here we can find IP address of some website
Usually we type name cause we can't remember these addresses

Web development: Build websites for internet

* ~~Request Response Cycle~~
we request (access) to any server → request → It goes to the server → in response the code file is given which includes 3 files [HTML, CSS, JAVAScript]
eg: amazon.in
eg: amazon server
← To browser any chrome safari (renders) (interpret)
finally client can access that file in form of website
(the browser creates ~~website~~ website from that files)

HTML - Structure
CSS - Style
JS - functionality (working) ie (apply working, apply functns)

Frontend : HTML, CSS, JS, Bootstrap, Tailwind
Backend : Express, NodeJS
Database : SQL, MongoDB
+ React

HTML : Hyper text markup language — [Structure, formatting

for convert any normal text into a formatted text the HTML can be used.

* HTML Element : Standard elements which are known to browser.
eg: Paragraph, heading, image

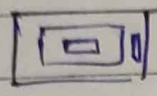
* HTML tags : The container for some content
The HTML tags are used to create diff elements.

elements	(i) Closing tags	(ii) Opening tags
Paragraph		
img	</img>	<img>
Heading	</h1>	<h1>

(opening tag) → This is paragraph ← (closing tag)
content
element (paragraph)

* Paragraph Element --

* If we use another tages in a single tage they are called as nested tags & this process is called as nesting.



 Hello I am Anu

* Heading Element
h1 to h6 [6 Levels]

<h1> </h1>
<h2> </h2>
<h3> </h3>
<h4> </h4>
<h5> </h5>
<h6> </h6>

default size

size decrease ↓

* HTML Boilerplate Code

```
<!DOCTYPE html>
<html>
  <head>
    <title> My page </title>
  </head>
  <body>
    <p> Hello </p>
  </body>
</html>
```

Root Tag

Metadata

display content

{ Indentation: Proper Spacing }

* Lists in HTML

- A
- B
- C

} unlisted

1.
2.
3.

} Listed

```
<ul>
    <li> Bread </li>
    <li> Butter </li>
    <li> Bat </li>
</ul>

<ol>
    <li> Bread </li>
    <li> Butter </li>
    <li> Bat </li>
</ol>
```

* HTML attributes

To add more information to tag.

`<ol type="a">` — type: attribute

`<html lang="en">` — lang: attribute

* Anchor element / anchor tag

This is used to add link to your page.

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

href: attribute

(href = hypertext reference)

Here after clicking we will reach to google site.

Page No.
Date

links → absolute link : Site of any original present link e.g. google, netflix

→ Relative link : To go from one HTML page to another HTML page. in e.g. in our system.

* Image Element : to add image on our page.

<img src = "image.png" alt = "Name of image">

src → Source attribute (where is our image)

alt → alternative attribute (alternative name in case some image is removed)

relative link : Here We also can used direct i.e absolute link on any site

* Br tag & hr tag.

Br = ~~Here~~ ~~break~~ line goes to next one

hr = one line is drawn

 -- ⇒ =

<hr> ⇒ —

* Bold italic & underline

 I ⇒ **I**

~~<b>~~

 I ⇒ *I*

<u> I </u> ⇒ I

/ forward slash } \ Backward slash.
(mdn) website

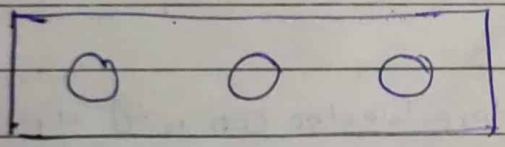
* Comments in HTML
  It is not included in part of code. It is used for adding more information to the code.

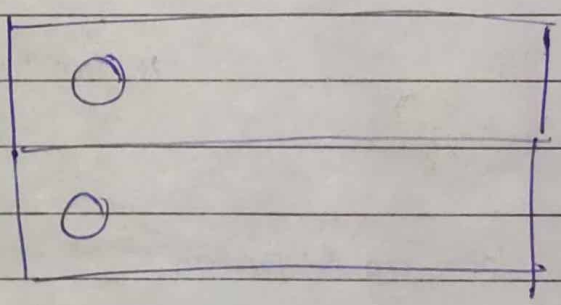
single line:
<!-- comment -->

Multiple line:

* HTML is non-case sensitive language.

Inline Elements	Block Elements
• Don't start from new line • takes necessary width.	• Takes full width available ie. whole block • Starts new line





Heading element : Block element
Paragraph element : Block
Image element : Inline element
Anchor element/ Tag : Inline element

* Div element (container) (called as content division element)
  : Group the elements differently.
  : It is a block element.
  : It is a container which can used for different elements grouping
  : (used for making group of elements)

* Span Element
  - Inline element
  - Similar to div element but it doesn't occupy whole line. because it is inline element.

* Hr Tag (horizontal Rule element)
  : <hr> —
  : <hr> —
  ∴ </hr> —

* Br Tag (Break Rule element
  Br ↴ goes to next line

* Sub & sup script.

a^2 ⇒ superscript
a ²

H_2 ⇒ subscript.
H ₂

* Sementic Markup
  (Meaning of content)
  [Used for only understanding like to understand content i.e. header, footer]

<section>
<header>
<footer>
<main>

} meaningful (makes meaningful)
seo friendly (search engine optimization)

↓
improves website ranking

- Layout becomes structured :
- Readable + screen readers
  UX & UI improves

(i) <header> upper part / top part.
<!header> (mostly includes name & some other standards)

(ii) <main><!main> (imp part / content)

(iii) <footer> <!footer> (mostly contact us or down / last part is footer)

(iv) <nav> <!nav> (for making navigation part) (to reach out to that part)

(v) <article><!article> (

(vi) <section><!section> (to group together related content)
(related content)

(vii) <aside><!aside> (indirect related content)

* HTML entities
(& starts ; ends) ♥ ⇒ ♥
& ⇒ & ampersent
< ⇒ less than < > ⇒ greater than >
  ⇒ non-breaking space
&quot ⇒ "
: which type doesn't prints directly then we use HTML entities for their use.
HTML entities are piece of text ("string")
: Used to display reserved characteristic or invisible character (like) or special characters

⇒ Here we also can use character codes rather than names.
(decimal number can be easy)

Page No.
Date

* Emmet (Name of toolkit)
: shortform for bigger code easily makes available.
eg - when we use ! in VS code the boiler plate code for HTML is generated automatically.
So, ! this can called as abbrivation for that code

* family hierachy ie. Nesting of tags
(i) child & parent relation.

```
<nav>
    <ul>
        <li> </li>
    </ul>
</nav>
```

⇒ nav > ul > li

li ⇒ child & ul is parent for li.
ul becomes child of nav &
nav becomes parent for ul.

(ii) sibling : same as brothers or sisters
: which are at same level are called siblings

```
<p>
    <img>
    <div> </div>
</p>
```

⇒ Theses both are siblings here
here they become child for p tag.
P becomes parent for them.

(iii) ul > li * 5

```
<ul>
    <li> </li>
    <li> </li>
    <li> </li>
    <li> </li>
    <li> </li>
```

So, for creating these types of codes with short abbrevations emmet is used.

Page No.
Date

* HTML 5
: Set of modern web technologies.
: DOC type
: updated version of HTML + other things / features with aren't part of HTML

new features

storage, JS APIs, multimedia

• HTML works with HTML standards.

* HTML standard (living standard)
: HTML standard is a document which tells the browser that how HTML should work.
• This document includes all information about how to process HTML.
• Living standard because it is change / updated with time.

* HTML Tables

Rows & column
Horizontal line — vertical line

* table element

<table> </table>
table tag

<table>
<caption> Table of contents </caption>

<tr> ⇒ creates rows
<th> ⇒ table head
<td> ⇒ table data

<thead>: Header data
<tbody>: Table data body
<tbody>: table footer (final calculations)

* Colspan & Rowspan attributes.

Colspan means no of columns gets combined.
Rowspan means no. of rows gets combined.

Hello	
No	Yes

colspan = "2"

* HTML forms.

① Action attribute: It is used to define what action should be done after form is submitted. i.e. where the data should be sent. [redirection / submits the info]

* ② Input element: To take info of different types from user.

⇒ some popular inputs are:

<input type="text"> | <input type="password">
<input type="date"> | <input type="time">
<input type="color">

* HTML forms

* ③ Placeholders & labels.

① Placeholders : The placeholders are used to display any text ie. any name on blank space which gets erased when user starts typing input text.

• Random text.

* ② labels

: To combine any element & text.

Simply Labels are used to combine to elements with help of "for" attribute "id" attribute.

:- 'id' is unique name which can't be given to other element.

:- 'class' is another which can be given to no. of elements.

④ button

(i) type button submit = info submit

(ii) type button = any button

(iii) type reset = refresh

Name attribute [form(name = value) gets stored]

The name attribute is used to give & store the information by that given name.

[∵ we shouldn't use name attribute to the password.]

eg:-

Student's Name:

<input type = "text" placeholder = "first_name" name = "student_name" >

Student's Name [first-name]

: If here user types name = Anushka.

then 'student_name = Anushka' this gets saved in memory. This can be accessed by that name only.

④ when you define name then value gets submitted.

Page No.
Date

* ⑤ Checkbox element ☐ Hi ☐ Hello.
: select multiple options.
: Here we have to tick mark on box.
: we have to create / associate Label element with input element & attache label with for & element with id.

```
<input type="checkbox" id="age" name="age"/>
<label for="age"> I ame 18 </label>
```

: if we use checked then alredy it is checked / tick marked.

⑥ Radio button
: To select only one option.

there should same name attribute

⑦ Dropdown (select & option)
: from No. of options we can select one by using dropdown.
: can use "selected attribute" for preselection

```
<select name="branch" id="branch">
    <option value="prod"> production </option>
    <option value="CS"> Comp Sci </option>
</select>
```

⑧ Range element

```
<input type="range" min="0" max="100" step="10"
                                value="50">
```

Value is default value. step is increasing & decreasing specific value.

⑨ Textarea element <textarea>