

Web Development

Hyper Text Markup Language

HTML →

CSS →

Cascading Style Sheet

JavaScript

JS →

* must i.e. body beauty.

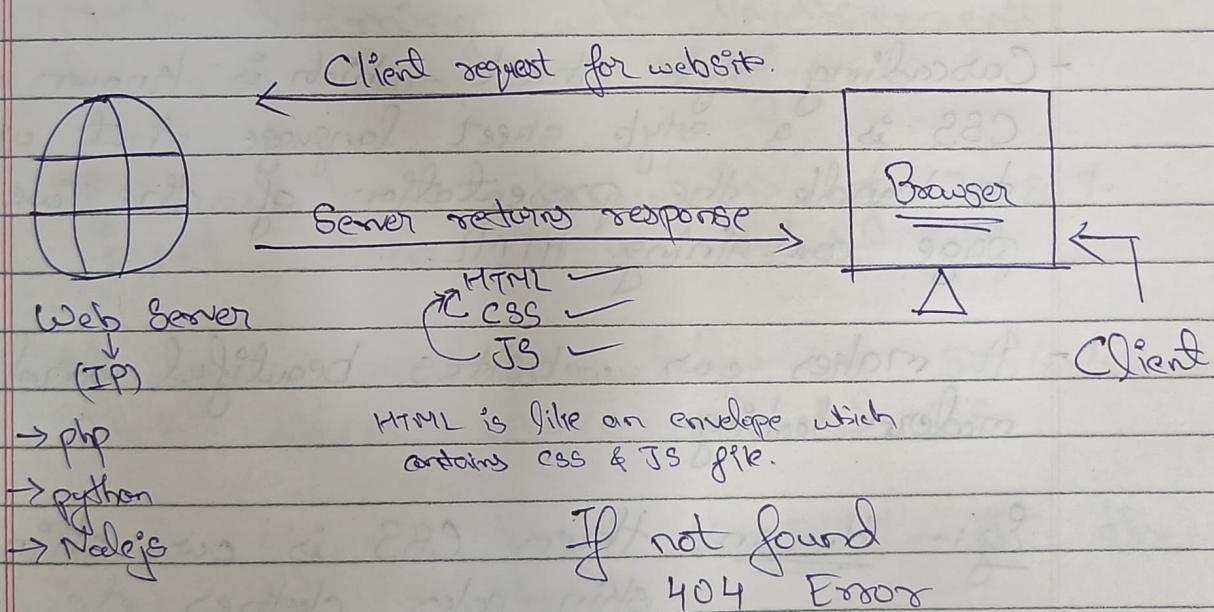
Eg:- Car

Metallic body → HTML

Colors, designs → CSS

Engine. → JS.

* How WEBSITE WORK?



* HTML

- HTML stands for Hypertext Markup Language.
- It is basically a standard markup language for giving a static skeleton to web application & websites.
- Its a well standardized system.

Eg:- Human then HTML is like skeleton with just bones.

* CSS

- Cascading Style Sheet which is known as CSS is a style sheet language that used to handle the presentation of the web page containing HTML.
- It makes our websites beautiful and modern looking.

Eg:- Human then CSS is our skin, skin color, clothes, etc using which we looks beautiful.

* JS

- JavaScript which is often known as JS, is a high-level dynamic interpreted programming language.
- It allows client-side scripting to create completely dynamic web applications & websites.

Eg:- Human then JS is our brain using which we ~~can~~ can think and do different work.

While learning web development do not learn 100% of all.

- If you have knowledge of 70-80% HTML then move on to CSS.
- If you have knowledge of 50-60% CSS then move on to JS.
- And if you have 60-70% JS then that is enough.

* Remaining percentage of learning can be done while you make some projects.

* HTML:-

- Boiler plate (minimum code required to run proper HTML code)
 - i.e. Syntax.
- Code is ~~written~~ in tags. ('<' opens a tag, '>' closes the tag)

* `<!DOCTYPE html>`

`<html lang="en">`

~~content~~

`<head>`

`<meta charset="UTF-8">`

`<meta name="viewport"`

`content="width=device-width initial-scale=1.0"`

`<meta http-equiv="X-UA-Compatible"`

`content="ie=edge">`

`<title> Document </title>`

`</head>`

`<body>`

`<!-- Web content goes here -->`

`</body>`

`</html>`

Comments
tag

in html

- It doesn't get executed

- Used for writing notes.

* `<meta>` :- Metadata means information about data.

`<meta>` tag is mainly used for SEO (Search Engine Optimization) which is used to increase your web traffic.

- charset = "UTF-8"
 - characters are encoded using UTF-8 which is default.
- name = "viewport" content = "width=device-width, initial-scale=1.0"
 - it is used for responsiveness of website.
i.e. it by default detects the device width that we are using and sets its to website ~~device~~ width.
- http-equiv = "X-UA-Compatible" content = "ie=edge"
 - Internet Explorer can also be supported using this meta tag.
- name = "description" content = "Site description"
- name = "Keywords" content = "HTML, HTML tutorials"
- name = "robots" content = "INDEX, FOLLOW"

* `<link rel="stylesheet" href="style.css">`

Used for including external CSS

* `<script src="index.js"> </script>`

Used for including external js.

* Heading tag

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

* paragraph tag

< p > < /p >

* Emphasis tag (Instead of italic)

< em > < /em > ↓
 < i > < /i >

* Strong tag (Instead of bold) → < b > < /b >

~~< b >~~ < strong > < /strong >

* Line break (self-closing)

< br >

* Horizontal rule

< hr >

* Anchor tag: (links to other website or pages)

* Anchor tag :-

~~< a href="#" > < /a >~~
< a href="#" > < /a >

Attributes :-

href :- Where to go.
(req) (internal & external)

target :- "-blank"
opens in new tab.

(Source.unsplash.com) → for
* Image tag :- random

< img src="#" alt="#" >

Attributes :-

src (path) :- Source of
(req) img :- Which img to open.

alt :- Alternative if
(req) image not found.

width :- width for img
(will use CSS)

height :- height for img
(will use CSS)

* Lists tags:-

1) Unordered List: (ul) 2) Ordered List: (ol)

Attributes:-

li → list items.

type :- Used for different types of bullets/ numbers in ul & ol lists.

values for type attributes :-

ul ⇒ "circle", "square", "disc"
 → default

ol ⇒ "1", "a", "A", "i", "I"

3) Definition List:

<dl>

dl ⇒ definition list.

<dt> </dt>

<dd> </dd>

dt ⇒ definition term.

<dt> </dt>

<dd> </dd>

dd ⇒ items in a

</dl>

definition list.

* Tables tags:-

<table>

<thead>

<tr>

<th> </th>

<th> </th>

<tr>

<thead>

<tbody>

<tr>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

</tr>

</tbody>

</table>

* * Forms tags :-

<form action="" method="">
</form>

- Use <div> </div> to divide the fields in form.
- <label> </label> :- "for" is linked with "id"
- <input type="text"> - default.
- * <input type="password">
- * <input type="email">
- <input type="tel">
- * <input type="date">
- * <input type="time">
- * <input type="file">
- <input type="color">
- <input type="range">
- * <input type="checkbox">
- * <input type="radio">
- * <input type="number"> - exception → b-q, e'
- * <input type="submit">
- <input type="button">
- <input type="reset">
- <input type="search">
- * <textarea name="" cols="" rows=""> </textarea>
- input tag attributes:-

- 1) type (req) :- discussed above.
- 2) name (req) :- used for ~~describ~~ describing input tag for backend purpose.
- 3) placeholder :- non-selectable text that only appears when it is empty
- 4) required :- compulsory field.
- 5) disabled :- cannot enter any value.
- 6) readonly. → for keyboard inputs
- 7) checked → for checkbox & radio.

- ~~& dropdown list in form~~

```

<select name = " " >
    <option value = " " > </option>
    <option value = " " > </option>
</select>

```

* Inline & Block elements:-

- Inline elements means it will only take the required space and the next element will continue in the same line only.

e.g.: span, strong, a, br, hr, etc.

- Block elements means it will consume all the space available in the line and the next element (even though you have written it in same line in code editor) will start from next line.

e.g.: p, div, h1...h6, etc.

(#) (.)

* Ids and classes:-

- Id is an unique identifier for all tags.
- Same page cannot contain same id's.
- One element can only contain one id.

- Mainly Id is used for grabbing the tag while using CSS or JS.
- Class can be same for multiple tags.
- Class is used in CSS.
- We can change style of multiple tags by giving them same class.
- One element can also have multiple classes.

* HTML by default removes white spaces.

- If we want to get white spaces then we use HTML entities

e.g. 	→ non breaking space ()
<	→ less than (<)
>	→ greater than (>)
£	→ Euro (£)
©	→ ©

- For more information :- Search "HTML entities" on google

* Semantic tags :-

- Previously in HTML we use div tags with diff' id's and doing some CSS to place it in proper places.
- But ~~it is not preferable for SEO~~ it is not preferable for SEO.
- So, Now in HTML5 we use new semantic elements that automatically understands its own role.

- Different semantic tags are :-

<article>

<aside>

<details> ⇒ automatically create a collapsable menu.

<figcaption>

<figure>

<footer> ⇒ always stays at bottom.

<header> ⇒ always stays at top.

<main> ⇒ main body.

<mark>

<nav> ⇒ navigation bar of website.

<section>

<summary> ⇒ Heading line for details.

(.) <time>