01: HTML 5 & CSS 3

* web/www is an info space where interlinked hyprtxt doc & othr web resources are identified by URL's.
* webpage is a web doc availalole on www. It is stored on webservr & can be viewed using a web browser.
It is grouped into 3:

a) Static web doc:
* web pg that is delivered to the user exactly as stored.
* Displays the same info for all users, frm all contexts.
* They are loaded on client's browsr as exactly they are stored on web servr.
* user can only read the info bt cannot do any modification with the info.
* Adv → time saving, cost effective.
* Disadv → difficult to change.

b) Dynamic web doc:
* Contnt on a webpg can change in response to diffrnt condin
* It is developed using advanced servr-side or client-side technologies.
* Servr side D.w.pg = created by using servrsidescripting
* Client Side D.w.pg = processed using client side scripting like JS. Either passed into DOM.

* Adv → easy to update, interactive
* Disadv → high cost & slow processing.

* static	Dynamic
* simple to construct	complx to construct & design.
* uses HTML, CSS, JS for its construction	uses CGI (Common Gateway Intrface) & lang like AJAX, PHP, PERL.
* HTML pgs loaded quickly with less time.	dynamic web pgs take more time while loading.
* No database used	A database is used in at servr end.

c) Active web doc :
* consists of a comp prgm (JS, Applet, etc) that the srvr snds to the browsr & that the browsr must run locally.
* whn it runs, the active doc prgm can interact with the user & change the display continously.

* Here, the browsr performs the logic instead of the srvr. *

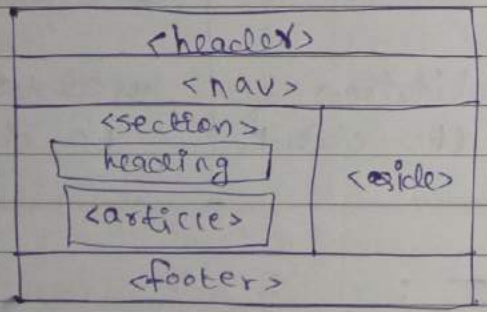
⇒ Scripting lang :
Allow us to write prgm in form of script.
2 types -

(a) client side (S)	(b) servr side (S)
* script is copied to the client browsr	script remains in the web srvr.
* script is executed in the client browsr	script is executed in web srvr.
* users can block client side (S)	cannot block.
* used for validation of data at the client	used to connect to databases.

⇒ HTML 5 :

* cross-platform (it does not care whether you are using tablet, phn / notebook).

* HTML 4	HTML 5
* 4th iteration of HTML * not available in HTML4	5th iteration new tags including <audio>, <video>, <header>, <footer>, <article>, etc. handle inaccurate s/x errors.
* difficult for developers to handle inaccurate syntx errors	



→ HTML 5

→ HTML elements & attributes :

* Elements are the building blocks of HTML
* Attributes provide additional info about elements

* header = represnts the header of whole pg or section of it.
* nav = defines a section of navigation links
  <nav> ... </nav>
* section = represnts a generic of a doc.
  <section> <h1>   </h1> </section>
* article = represnts an independnt item section of content.
  You can display the info in this element in various formats like a news article, a blog post or user's cmnt section.
  <article><h1>   </h1> </article>
* aside = Allows you to create a section that is used to display info about the contnts of other elements like time & date, cmnt news & weather report.
* footer = represnts the footer of a doc mainly contains info about the authr of the doc, copyright info, etc.
* address = specifies the contact info for the author of a doc / article.
  <address> write address </address>.

=> Types of HTML5 elements :

I Structural elements :

* grp of HTML elements that are used to organize pg content to give the doc more str.
* eg → <a>, <article>, <aside>, <body>,
, <details>, <div>, <head>, <html>, <footer>, <nav>,

* **details** =

<details>
<summary> details </summary>
<p> anything you want </p>
</details>

→ ▸ details (if we click on details it display anything you want).

* **span** =

used to apply css property.

<p> my mthr has red .

* **h group** = used to grp html headings.

<hgroup>
<h1>
<h2> </hgroup>.

II metadata elements:

* It is the info about data
* It willnot be displayed on the pg.
* eg → <base>, <link>, <meta>, <style>, <title>

* **base** = specifies default URL, has either href or a target attribute | both. i.e. single <base> element in a doc i.e. inside the <head> tag.

<head>
<base href = "google.com" target = "blank">
</head> <body>
<img src = "img.com">

→ if we click img it will go google.com i.e. not img.com.

III Form elements:

* used to collect data from the site visitor i.e. to post it to a backend appl^n
* eg → <button>, <input>, <datalist>, <form>, <label>, <option>, <select>, <textarea>, <legend>.

* **datalist** = specifies a list of predefined options for an input > tag.

<form>
<label for="bro"> choose: </label>.
<input list="brow". name= "bro" id = "bro">.
<datalist id = "bro">
<option value = "Edge">
</datalist>
<input type = "submit">.

→ choose: [ ] submit

if we click here it show wht we had gvn in <option>.

IV Formatting elements:

* used to format txt in the HTML → bold, i, underline
* eg → <b>, <i>, <address> <big>

* **code** = specifies txt as comp code.

<p> css <code> color </code> property </p>

→ color [illegible] code [illegible] (font)

* <del> → ~~hue~~
* **output**.
* **mark** = highlite txt.

<p> Do not <mark> forget </mark> today </p>

→ forget

* **pre** =

<pre> hello   ansar </pre>

([illegible] [illegible]).

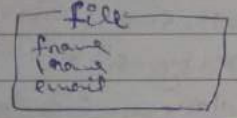
* **em** = emphazised txt.

<p> you hue harry </p>.

hue → italic

* field set =

<form>
<fieldset>
<legend> file </legend>.
<label> : fname
— lname.
— email
</fieldset>.

→ 

* progress = completion progress of a task.

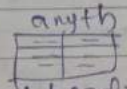
<label for="wht">
downloading : </label>
<progress id="wht"
value="50" max="100">
</progress>.
→ downloading: ▬▬

* strong → bold text.
* <sup>, <sub>

V list elements:

* specifying list of info
* can be unordered, ordered, definition.
* eg →
<dd>, <dl>, <dt>,
, <ul>,
<menu>.

VI Table elements:

* used to define 2D table comprised of rows & colmns.
* eg → <table> <td>,
<th>, <tr>.
* caption =
<table> <caption> anyth </caption>
<tr> <th> — —
</tr> - -
→ 

* tbody, <thead>, <tfoot>,
<col>

VII Scripting elements:

* used to reference executable code typically js code.
* eg → <script> =
used to refernce js code within an HTML doc.
It can be placed in
<head> or <body>

inline js
- <script>
alert ("hello");
</script.

ex js
- <script src="sr.js>
</script>

* <noscript> = used to define content that should be displayed when a browser doesn't support js.
<noscript>
 your browser doesn't support this file
</noscript>

VIII Embedded content elements:

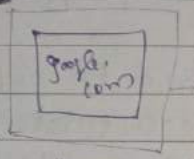
* used for embedding (embed) external applin into HTML doc.
* eg → <area> = defines a specific area within an img map

<img src="hi.jpg"
usemap="#name" />
<map name="name">
<area shape="rect"
coords="0,20,30,40"
href="google.com"
alt="img" title="hi" />
</map>.

- shape → rect → L, T, R, B
→ circle → x, y, r
→ poly → x, y, x, y ...
coords

* you can define multiple <area> tags within single <map>.
* <audio>, <video>, <img>,
<iframes> <frame>, <canvas>
↓
draw graphics by JS.

* <iframe> =



<iframe src="google.com"
width="500" height="500">
</iframe>

=> Types of HTML5 Attribute:

3 Categories —

a) Required / optional (A):
Element specific (A) that defines additional (pro) of an element.

(A)	belongs	Descr.
1) accept	<input>	types of files that the srvr accepts (only for type file)
2) action	<form>	Specifies where to send the form-data when a form is submitted.
3) auto focus	<button>, <input>, <select>, <textarea>	element should automatically get focus when the pg loads.
4) cols	<textarea>	Specifies the visible width of a txt area.
5) content	<meta>	gives the value associated with the name attribute.
6) controls	<audio>, <video>	Specifies audio/video controls.
7) headers	<td>, <th>	
8) height	<embed>, <canvas>, <img>, <input>	
9) href	<a>, <area>, <base>, <link>	
10) list	<input	refrs to <datalist> element that contains predefined options for an <input>
11) name	<button>, <input>, <select>	
12) src	<audio>, img, source, video	
13) usemap	img, object	
14) width	<img>, input, <video>, object	

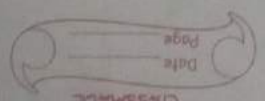
b) Standard / Global (A) : Common to all elements.

(A)	value	Descr.
1) class	clsname	Assigns a clsname.
2) id	name	
3) title	txt	
4) translate	ys / no	whether txt content is translated / not.
5) style	style.	

c) Event (A) :

Are global & can be applied to most of the elements.

1) window events → ononline, onoffline, onload, onunload, onstorage.
2) form events → onblur, onreset, onselect, onsearch, onsubmit.
3) Mouse events → onclick, ondrag, ondrop, onscroll, onshow.
4) keyboard events → onkeydown, onkeypress, onkeyup
5) clipboard events → oncopy, oncut, onpaste.
6) Media events → onplay, onpause, onplaying, onerror, onwaiting.



⇒ HTML 5 Datatypes =

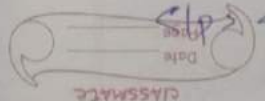
1) String = A str is defined as normal char data
2) Token = It is defined as a str that does not contain any space char.
3) ID = Any str that must be atleast 1 char long & must not contain any space char.
4) Name = Any str same as ID
5) intgr = range 0-9
6) float = 0-9
7) Date-time = A valid d-time as defined in RFC 3339.

eg → 1990-12-31T16:39:57

⇒ Form elements =

I button :

```
<form>
<p>
firstname:<input type="text" name="fname">
<button type="submit" value="submit">
Submit </button>
<button type="reset" value="reset">
Reset </button>
</p></form>.
```



II multiple-choice elements = → Select
→ option
→ optgroup.

* <form>
<select>
<option value="volvo"> volvo </option>.
— — —
— — —
</select> </form>.

* <select>
<optgroup label="cars">
<option value="volvo"> volvo </option>
— —
— —
</optgroup>
<optgroup label="bikes">
option value="duke"> Duke </option
— —
</optgroup> </select>.

*<optgroup> used to group related options in a drop down list.
*<select> used to create drop down list.
*<option> used to define an option in a select list

III textarea & label =

*<textarea> defines a multi line txt input control within a form.
*<label> defines a label for a <button>, <input>, <meter>, <output>, <progress>, <select> or <textarea>.
* <form>
<textarea cols="50" rows="5">
write here ... </textarea>.
</form>
* label syn →
<label for="control id"> ... </label>.

IV radio type, <fieldset> & <legend> =

* <form>
<input type="radio" name="gender"
value="male" id="male">
<label for="male"> male </label>.
— — female. —
</form.

* <form>
<fieldset>
<legend> Gender </legend>
<input type="radio" — — —
<label for="male"> male </label>.
</fieldset> </form>. → Gender • male • female

V <datalist> =

Specifies of list of predefined options for an <input> elemnt.

<form> ~~<datalist id>~~

<input type="text" list="bro">

<datalist id="bro">

option value="chrome">

— — —

</datalist>.

VI <keygen> =

generates an encryption key for passing encrypted data to a srvr.

<label> encryption : <keygen name="key"> </label>

VII <output> = represnts rslt of a calculation.

VIII <progress>

IX <meter> → scalar measurmnt within a known range.

X File uploading =

<input type="file" name="nyname">

XI frame & frameset =

(not supported on HTML5).

<frameset cols = "25%, 50%, 25%">

<frame src = "frame1.html"> top

< " frame2.html> main

< " frame3.html> bottom

</frameset

* used to ÷ web browsr window into multiple sections where each section is loaded seperatly.

⇒ CSS = [cascading style sheet].

Describes how HTML elemnts are to be displayed on screen, paper / othr media.

Adv :

* css saves time
* pgs load faster.
* Easy maintanance.
* Global web standards.

Syx :

h1 { color : ~~blue~~ red ; font-size : 12px ; }

selector property value.

* Selector points to the HTML elemnt you want to style.

* using CSS, can be added to HTML doc in -

a) Inline =
* using style attribute inside HTML elemnts
* used to apply a unique style to a single HTML elemnt.
* eg = <h1 style = "color: blue;" > Hi </h1>

b) Internal =
* using <style> elemnt in <head> tag
* used to define a style for a single HTML pg.
* eg =
```
<head>
<style>
body { bgcolor : blue ; }
h1 { font-size : 30px ; }
</style>
<head> <body>
<h1> hi anyar </h1>
</body>
```

c) External =
* used to define the style for many HTML pgs.
* using a <link> elemnt to link to a ex CSS file.

* eg = <body> <h1> hi anyar </h1> </body>
CSS file → h1 { font-size : 3px ; }

→ CSS selectors =

a) CSS elemnt (S) =
* select HTML elemnts based on the elemnt name.
* eg =
```
p {
    text-align : center;
}.
```

b) CSS id (S) =
* uses id attribute of HTML elemnt to select a specific elemnt
* id of an elemnt is unique within a pg.
* eg =
```
# para1 {
    color : red;
}
```
id = para1

c) CSS class (S) =
* selects HTML elemnts with a specific cls attribute
* here, we use period (.) elemnt followed by cls name.
* eg =
```
.para1 {
    color : red;
}
```
class = para1

```
p.para {
    font-size : 3px;
}
```

d) CSS universal (S) =
* Selects all HTML elemnts on the pg
* eg = * {
    text-align : center;
    color : blue;
}

e) CSS grouping (S) =
* Selects all the HTML elemnts with the same style defnitions.
* eg = h1, h2, p {
    font-size : 3px;
}

→ CSS comments =
* used to explain the code.
* eg = /* Comments */

→ CSS lists =
a) ordered list [<ol>]
b) unordered list [<ul>].

* list-style-type : circle;
[circle, square]

* list-style-image : url ('sample.jpg');
* list-style-position : outside;
• This is default

* list-style-position : inside;

* list-style : inside square;
* marker-offset : 2em; → 2em

→ CSS tables =

I Adding borders → border : 1px solid black;

II Collapsing table border → border-collapse : collapse;

III Table layout → table-layout : auto;

Name	Answer
email	mounganga

table-layout : fixed; →

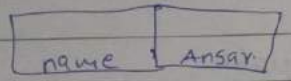
Name	Answer
email	mounganga98

IV Table position → caption {
caption-side : bottom; }
<caption> user details </caption>
→ user details

V table height & width → width: 100%;
height: 50px;

VI Horizontal alignment →
text-align: left;
def → td th
(L, C, R)

VII vertical alignment →
vertical-align: bottom;
(Top / B / M)
(def → content of table is Mid)



VIII Table padding →
padding → border & content.
margin → window & border.
→ padding: 15px;

IX Hoverable table →
tr:hover { background-color: green; }
:hover → to highlight table rows on mouseover

X Striped table → nth-child().
tr:nth-child(even) { bgcolor: red; }

Rollno	name
10	Ak
20	Bk
30	Ck

(odd)

Rollno	name
10	Ak
20	Bk
30	Ck

→ (even)

→ Links with css =
* hyperlink: is a connection frm 1 web resource to anthr.
* Has 4 states — link, visited, active, hover using pseudo-classes of <a> elemnt.
* pseudo-cls used to define a spl state of an elemnt.
* Str → Selector : pseudo-cls { pro : value; }
* Eg → a:link {
color: red; }
a:visited { color: blue; }
a:hover { color: green; }
a:active { color; yellow }
* a:link → set styles for unvisited links that has no mouse pointr ovr it
* a:visited → set styles for the link the user has visited bt has no mouse pointer ovr it.
* a:hover → set styles for a link when user place the mouse pointr ovr it.
* a:active → set styles for a link that is in the process of being clicked.
* Text decoration = used to remove the default underlines frm links.

```
a:link { text-decoration : none ; }
a:visited {  "        : none ; }
a:hover {  "        : underline ; }
a:active {  "        : underline ; }
```

* a:link, a:visited {
    color : white ;
    padding : 14px 25px ;
  }.

→ Navigation bars =

```
<ul>
<li> <a href = home.html> Home </a> </li>
</ul>.
```

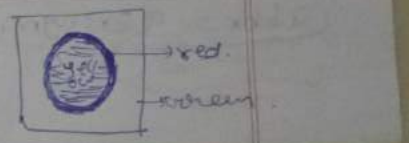
ul {
  list-style-type : none ; → removes bullets.
  margin : 0 ;
  padding : 0 ; } remove browser default settings.
}

* display : inline → build a horizontal navigation bars.

→ HTML tags =

1) ruby =
```
<body>
  my youtube channel is .
  <ruby>
  <rb> ATT </rb>
  <rt> all type tutorials </rt>.
  </ruby>
```
→ ... is ATT

2) picture =
```
<body>
<picture>.
  <source media = "(min-width : 650px)"
          srcset = "logo.jpg">.
  <img src = "logos.jpg">.
</picture> </body>.
```
→ logos.jpg Then min-width → 650px logo.jpg

3) SVG = (Scalar vector Graphics)
```
<body>
<svg width = "100" height = "100"
  style = " border : 1 px solid green ";>
  <circle cx = "50" cy = "50" r = "40"
    fill = " yellow " & stroke = " red " />
</svg> </body>
```
→ 

* <rect x="10" y="10" width="80" height="50" fill="yellow" stroke="red" />.

4) download =

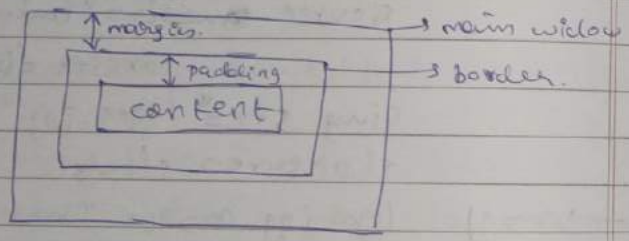
<img src="fox.jpg" width="200">

download

(filename)

→ [fox] download.

5) margin & padding =



margin → left, R, T, B.
padding → L, R, T, B.

6) abbr =

 I have a <abbr title="pen drive"> PD </abbr>.

→ I have a PD (if we hover the mouse over PD it shows pendrive).

abbr = acronym

7) address =

<address>
written by:
<a href="mailto: mounasarraf..">
visit us at <a href="...">.
up, India.

8) article & aside =

<article style="width: 65%; float: left; padding: 10px;">
<h1> comp </h1>
<p> ~~~ </p>.
</article>
<aside style="width = 25%; float: left; padding: 10px;">
<h1> Laptop </h1>
<p> ~~~ </p>.

→ [comp ~~~ | Laptop ~~~]

9) audio =

<audio controls>
<source src="song.mp3
type="audio/mp3">.

</audio>.

10) base =

```
<head>
<title> Base tag </title>
<base href = "media/images/">
</head>
<body>
<img src = "cat.jpg">
<img src = "fox.jpg">.
</body>
```

11) bdo = (bi-directional override)

```
<body>
<bdo dir = "ltr"> This is bdo </bdo>
<bdo dir = "rtl"> This is bdo </bdo>.
</body>.
```

→ This is bdo.
odb si sihT

→ CSS img Gallary =

* <img> tag is used to insert images in HTML doc
* <img src = "url" alt = "text">
* src → tells the browsr where to find the img you want to display.

* alt attribute is the alternative description for an imag, if the img cannot be displayed

→ CSS opacity =

* Specifies the transparency of an elemnt.
* opacity property value must be a no. b/w 0.0 (fully transparnt) & 1.0 (fully opaque)
* opacity : 0.2; → transparnt. (less) img
* opacity : 1.0; → default.
* Transparnt hovr →

```
img : hover {
    opacity : 1.0;
}
```