

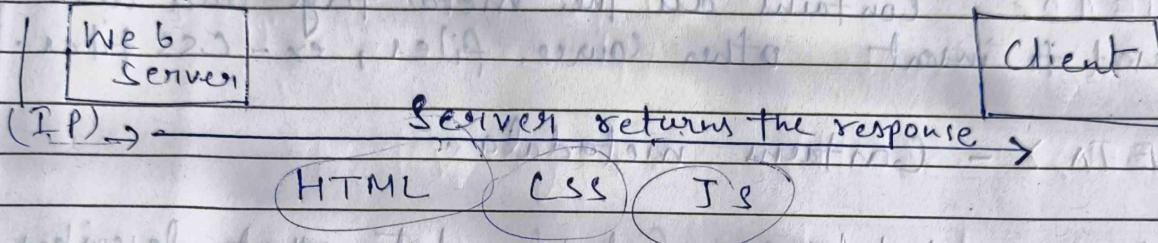
#1 How does a website work (HTML, CSS, JS)

Basic requirements

Date _____

- HTML → Body (Skeleton)
- CSS → Add beauty (Skin) ↘
- Javascript → power (brain) or working.

if exist ← Client request for website ex - google.com



- HTML :-
[80%]
 - HTML stands for Hypertext Markup Language.
 - It is basically a standard markup language for giving a static skeleton to web application and websites.
 - It is well standardized system.

• CSS :-

- [40-60%]
- Cascading Style Sheet which is known as CSS Style sheet language that used to handle the presentation of the web page containing HTML.
 - It makes our websites beautiful and modern looking.

• JAVASCRIPT :-

[50-70%]

- 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 and websites.

Spinal

#2 Basic Structure of a website

Date _____

- `<!DOCTYPE Html>` - It defines that the document belongs to HTML 5 and tells the browser so the browser can understand.
- `<HTML>` - defines or start an HTML file.
- `<HEAD>` contains all the meta tags and also used to insert other source files, ex- CSS files.
- `<META>` - Contains metadata.

Q. What is metadata :- Set of data that describes and gives information about other data.

• `<BODY>` - Whole Content of the website's body.

→ Comment - `<!-- Comment here -->`

Toggle Comment in vs code - `Ctrl + /` (forward slash)

(@Significant)

(Ex-02)

3. Title, Head, Script, meta tags, SEO

Date _____

- <META> tags are mostly used for SEO.

→ <meta charset = "UTF-8"> → Describes the character used.

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

↳ Used for a dynamic, responsive adaptability for the diff. devices.

→ <meta http-equiv = "X-UA-Compatible" content = "ie=edge">

↳ For internet explorer(B) adaptability.

→ <meta name = "description", content = "This is desc.">

↳ For writing desc. (SEO).

→ <meta name = "keywords", content = "html, methods....">

↳ For ranking tags (SEO).

→ <meta name = "robots", content = "Index, Follow/unfollow">

→ For letting search engines show your wanted content on search results or not.

→ Including CSS (external) :-

<link rel = "stylesheet" href = "first.css">

Inside
Head tag

→ Including external JS file:-

<script src = "first.js"> </script>

Special

HF 4. Heading and Paragraph tags

Date _____

- Heading tags can make the headings of different sizes from `<H1>` to `<H6>` have different sizes
- `<P>` This is a paragraph tag `</P>`
- `` This is an emphasized code ``
- `<bold>` `</bold>`, `` ``
- `
` line break.
- `<hr>` horizontal rule.
- `<i>` italic `</i>` @Sigma Hostile
- Tag replication (em et) — fast writing, not edit.
 - Tag * Number = replicated by the number.
 - Alt + mouse click = multi line write / edit.



Date _____

- Anchor tag - `<A>...`

Ex -

` Go to google. `

` Go to github. `

→ For opening links in new tab :-

Attribute `la target = "blank"` ~~l href = "link"~~

→ For internal link files :-

` Doc `

- Image tag - `...`

Ex -

``

``

• For dynamic image - website - "Source unsplash".

• Random images unsplash url -

"`https://source.unsplash.com/weekly/?Search term`".

`width = "900px"
height = "600px"`

Attributes of `img tag`] but most performed with CSS.

#6 Lists and Tables

Date _____

- List
 - unordered list (..., ***, →→→, 000)
 - ordered list (1, 2, 3, 11, 111, a, b, c)

 — ordered list: "apple"

 (type a, i, I, l)

 . . . — unordered list.

- type attribute :- (disk, circle, square etc)

<ul type = "disc">

 hollow → hollow

- Sublists :- (list inside list)

 hi

 . . .

 hollow

 bye

Tables in HTML

Date _____

- Table in HTML → Combination of Rows and columns.

→ It is made up of Table Head and Table body.

<Table>

<thead>

<tr>

<th> Name </th>

<th> Class </th>

<th> Rollno </th>

</tr>

</thead>

<tbody>

<tr>

<td> Hostel C </td>

<td> 3rd Year B </td>

<td> 34, 33, 32, </td>

</tr>

</tbody>

</Table>

<table border="1">

<tr> <td> " " </td> <td> " " </td> <td> " " </td> <td> " " </td> </tr>

"border" of = 2px "text-align" = center <td> </td>

<tr> <td> " " </td> <td> " " </td> <td> " " </td> <td> " " </td> </tr>

#8 Forms and Input tags

Date _____

- Form → form is a block area where we can fill different formats of data.

1st step → if form action = "backend-PHP" → For saving data in backend

Input tags :-

<div> → for block occupation.

Name : <input type = "text">

</div>

<div>

Role : <input type = "email" name = "myemail">

</div>

My :-

• <input type = "Date" name = " " >

input : submit → <input type = "Submit" value = "Submit now" >

• <input type = "number", name = " " >

• <input type = "checkbox", checked >

• <input type = "radio", name = " " >

input : reset → <input type = "reset", value = " " >

• <input type = "text", name = "mytext", cols = "30", rows = "10" >

</textarea>

and the margin will be... by CSS.

- **label tag :-** for labeling the text for one or more particular forms using `id = ""`.

• `<label for = "name"> Name </label>`

~~{div}~~

~~`<input type = "text" name = "Myname" id = "name">`~~

- The label selects the form after clicking on the labeled text.

- **Select tag :-**

`<select name = "myCar" id = "car">`

`<option value = "ind"> Indica </option>`

`<option value = "toyota" selected> toyota </option>`

By default selection.

#10 Inline and Block elements (take all space):

Date _____

- Inline element - ~~size Apni Tagah ke~~
- Block element - ~~pure horizontal space ke~~

Inline - ~~
 <hr>
~~
~~i, em, div, img,~~ ~~text = object fragment~~

Block - ~~<p>~~

#11. IDs and Classes:

<!DOCTYPE HTML>

<h3> IDs and Classes tutorial </h3>

<div id="mainBox" class="redBg blackborder">
this is mainbox
</div>
.

<HTML>.

→ IDs :- The id global attribute defines an identifier (ID) which must be unique in the whole document. Its purpose is to identify the element when linking (using a fragment identifier), Scripting, or styling (with CSS).

→ Class - The HTML class attribute specifies one or more class names for an element. Classes used by CSS and JavaScript to select and access specific elements.

Event : (div is default).

Span . RedBg = Span of class RedBg.

span # mainBox = Span of id mainBox.

Date _____

• black \Rightarrow `<div class = "black"></div>`

#12. HTML Entities:-

An HTML entity is a piece of text ("string") that begins with an ampersand (&) and ends with a semicolon (;). Entities are frequently used to display reserved characters (which would otherwise be interpreted as HTML code.), and invisible, characters (like non-breaking spaces).

• This is another paragraph with two

\Rightarrow `< p >` \Rightarrow `<p>`

$\& pound;$ = £

$\©$ = ©

similar - go to HTML entities reference. - google

#13. Semantic tags in HTML

Semantic - giving meaning to anything.

* It is important for SEO.

→ Semantic and non semantic tags — google.

`<detail> content ... </detail>`. (for collapse and expand).

`<summary> .. </summary>`. (for summary).

#14 Introduction to CSS - Cascading Style Sheet

Date _____

Q. What is CSS?

Ans

- CSS gives style to your HTML.
- CSS stands for Cascading style sheets.
- CSS is used to give style to our web pages.
- CSS is used to make websites responsive.
- HTML is used to structure a website.
- CSS takes responsibility of design.

• CSS Syntax →

selector property value.
P color : black ;
where? this | set this value.

• Class — for class

idname — for name.

header, P.intro { background-color : red ;
border-radius : 8px ; }

Group of Selectors

Declaration block.

→ Three ways to add CSS to markup - (HTML code).

Ex) Inline CSS - CSS is added to the elements directly using style attribute.

Ex - <P style = " " > .

iii) Internal CSS - CSS is kept inside the head tag in <style> tags.

iv) External CSS - CSS is kept separately inside a CSS style sheet.

↳ S1 → write in CSS file.

↳ S2 → include that CSS file to markup.

#15. Inline, Internal and external CSS.

Visit inlineint ext CSS.html.

Precedence. = Inline. ①

② Internal or external depends on koun baad me likha tha hai.

! important ⇒ for ~~imp~~ highest precedence.

#16. What is CSS selector:

* CSS selectors are used to find the element whose property will be set.

* Selectors are used to target the HTML elements

* Selectors makes it easy for us to easily target single / multiple HTML elements in the markup.

```
P { color: purple; }
```

 selector	 property	 value .
--------------	--------------	-------------

→ Types of basic CSS ~~and~~ selectors:

- i) CSS element selector.
- ii) CSS id selector.
- iii) CSS class selector.

→ The CSS grouping selector,

[visit cssselectors16.htm](#)

group the elements
together and
assign the styles

Some

✓ Class can be written in multiple elements
and multiple classes can be written in single element.

- Ids are unique and only one id is given to the one element and one id cannot be written in two or more elements.

CSS Comments - /* comment */

#17 Using chrome developer tools.

Date _____

Console — writing JS.

Element — Edit HTML

Source — contains all the files.

Style — Writing CSS, edit etc.

You can change everything of a webpage only for your browser view with these tools offered by your browser.

For more visit — chrome devtools17.html.

#18. CSS Fonts

- web safe fonts — installed, i.e. to longer
- web fonts — have to import.
- font stack — a list of font ... to available we apply no mega.

website — CSS ~~web~~ font stack

— google fonts.

→ Font Size: — font-size: 20px;

pixel — 1/196th of an inch

→ line height — space b/w two lines.

line-height: 8.3em;

→ font weight — Bold

font-weight: bold;

→ Font style —

font-style: italic;

Spiral

#19 Colors in CSS

Three types of colors property in CSS -

- By name -

Color: blue.violet;

- By RGB value -

Color:rgb(15, 183, 48);

- By Hex value -

Color: #ffff;

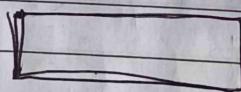
Some more properties → visit internet.

#20 Height, width, borders and backgrounds.

- Height - Height: 100px | defines the spaces or height

- Width - width: 445px | defines the spaces or width

- border -



= border: 4x solid green;

border-radius: 10px;

- border-top, border-right, border-bottom, left also
they can have different colour and width.

- border-top-left-radius: ... and so on.

- background-image: url(...);



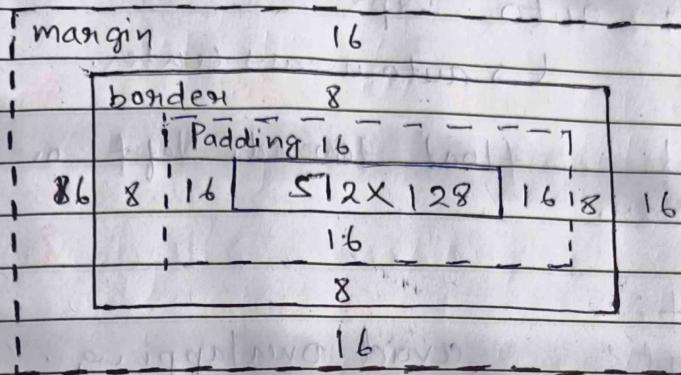
- background-position: center center;

center bottom;

- background-repeat: x or y or no repeat; and so on...

#21. CSS Box Model, Margin, Padding.

Date _____



- We can set margin/padding for top, bottom, left and right like this.

`padding-top: 79px;`

`padding-bottom: 79px;`

`padding-left: 34px;`

`padding-right: 79px;`

- ~~Padding /margin = top right / bottom left;~~
~~padding: 23px 56px 6px 78px.~~

- ~~padding 4 (top/bottom) x (left/right);~~
~~padding: 34px 19px.~~

- ~~box-sizing: border-box;~~
~~(width does not change at all).~~

* {

for all elements.

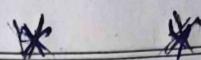
}

#22.

Alignment (Float and Clear).

Date _____

- margin: auto 33px



→ auto means center.

- float: left float toward left or right
width = 48%.

clear: left;

clear: right; avoid overlapping.

clear: both;

text-align: right;

text-align: left;

" " : center;

" " : justify;

Text alignment.

For more - visit - [A float and clear 2. HTML](#).

H23. Styling links and Buttons.

Date _____

→ ` Read more `

`a:hover {`

`color: #ff0000;`
`background-color: red;`

}

Pseudo
selectors.

`a:active {`

`color: yellow;`
`border: 2px solid black;`

}

`a:visited {`

`color: blue;`
`background-color: #ff0000;`

}

For more - visit : [pseudoselectors 23.html](#).

- tipi draw ↗

- INT H. Pseudos

Spiral

24 Creating a Navigation menu.

Date _____

→ Steps: ① HTML
 <body>
 <Header>

 <nav class="navbar">

 Home

 <div class="Search">
 <input type="text" name="search" id="search".placeholder="Search" />
 </div>

 </nav>
 </header>
 </body>

Step 2: — CSS for styling.

- navbar { }
- navbar ul { }
- navbar li a { }
- navbar li a :hover { }
- Search { }
- navbar input { }

For more visit:-

navbar 24. HTML

#²⁵ Display property : CSS

Date _____

* Display inline :- Jitna bada element hii utni space le lega.

* Display block :- Wo element ~~space~~ space le lega but ham utni width ~~ko~~ ko set kar saktein hain. (margin auto kaise usko centre bhi kar sakte hain).

* Inline-block - Element inline bhi rhega aur iski "width" ~~ko~~ set kar saktein hain.

=> Box-sizing : border-box;

Ye karne se width margin aur padding ke wajah se change nahi hoti jo ~~se~~ ~~ki~~ hai wali shati hain. (9)

For more visit : [displayproperty.html](#)

(studios) : writing <

(xgns) : got

(xgn) : tiger

box : writing <
(xgs) : got

#26 CSS Position property.

Date _____

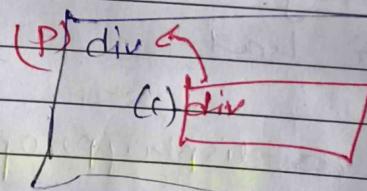
* Static : static means by default position means normal position.

> `position: static;` — should put space

* Relative : positions the element relative to the its normal position and will leave a gap at its normal position. — should - width

> `position: relative;`
`top: 34px;`
`left: 334px;`

* Absolute : positions the element relative to its first parent



> `position: absolute;`
`top: 34px;`
`right: 34px;`

* Fixed : position the element relative to the browser window.

> `position: fixed;`
`top: 3px;`

Windows Map

Date _____

* sticky: A sticky element is together b/w relative and fixed, depending on the scroll position.

for more visit : [Position26.html](#).

27

Visibility and Z-index.

Date _____

visible

* Visibility default :- Kuchh bhi change nahi hoga.

* Visibility hidden :- Content will be hide and but still occupies the space.

visibility: hidden;

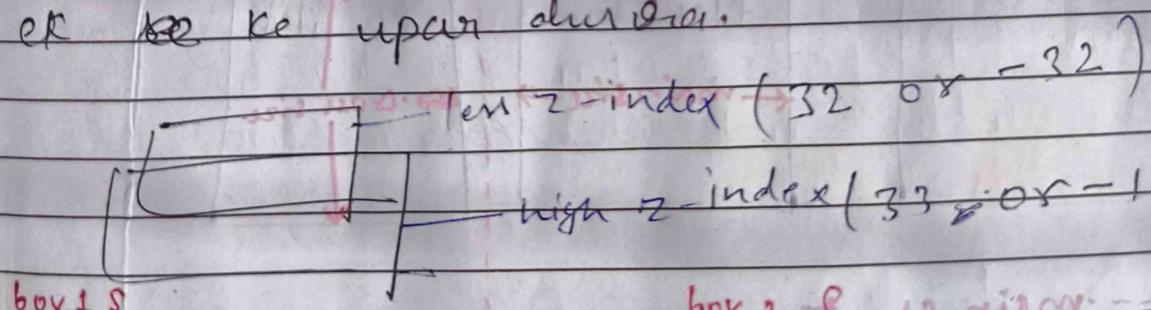
* Display none :- Content will hide and also space will be destroyed (free).

display: none;

real 3 4

Z-index :- works only for positions - relative, absolute, fixed and sticky.

z-index jiski yada hogi wo upar dikhega ek ke upar duraega.



z-index: 34;

z-index: -32

3

box 2 - front view

z-index: 33

z-index: -16

3

For more visit [visibility27.html](#)

#28

Flexbox in CSS

Date _____

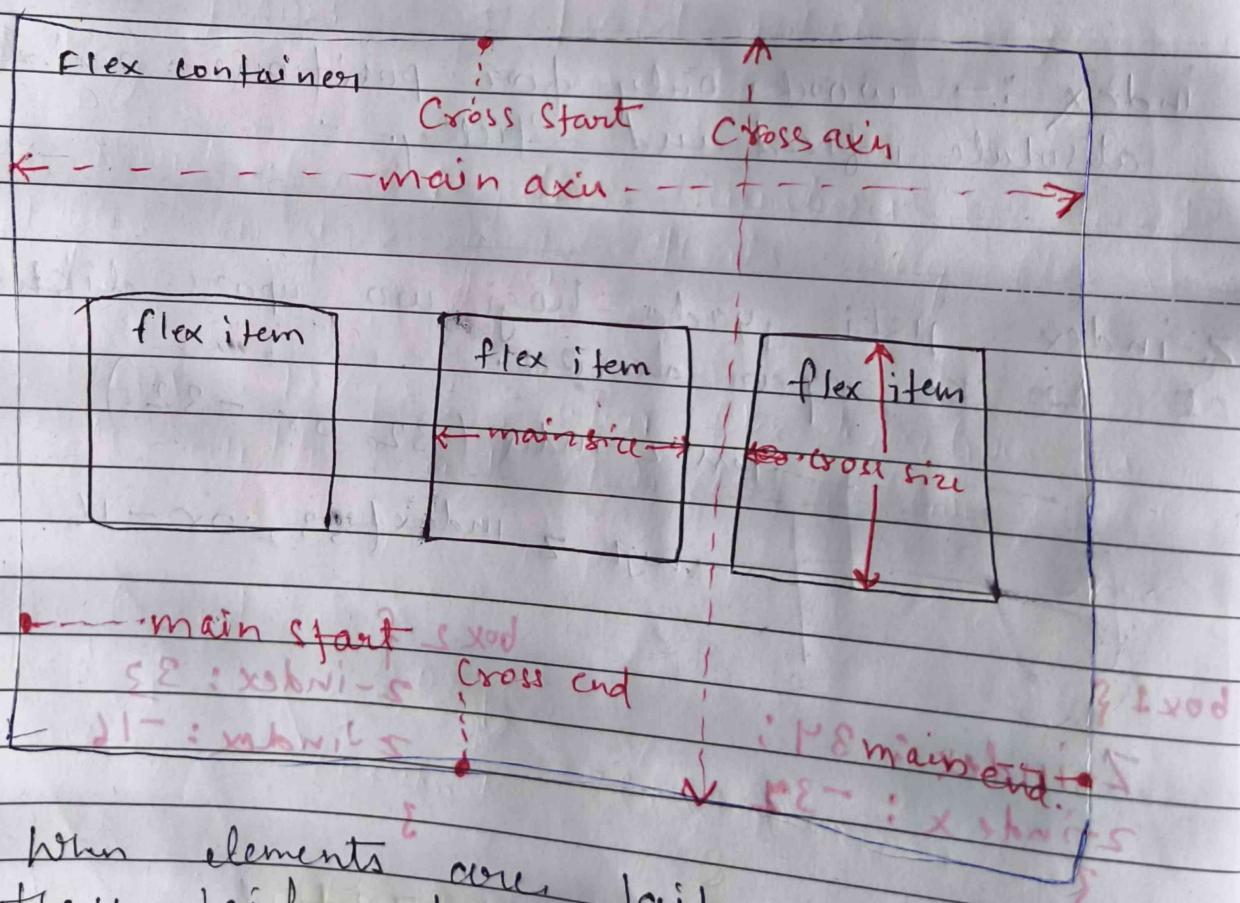
* Flexbox — is a one-dimensional layout method for laying out items in rows or columns.

* CSS Flexbox — is a better way to align items into a container. *(nesting = problem)*

* Flexbox = Flexible box.

selector property value.

P { color: blue; } *(non-selectable)*



When elements are laid out in flexible box manner:

div.items { display: flex; }

for more visit: [flexbox28.html](#)

H29 Flex Properties - For flex Container

Date _____

- `display: flex;` - Initialize the container as flex box
- `flex-direction: row;` - By default.
- `flex-direction: column;`, column rev. and ^{row} rev.
- `flex-wrap: nowrap;` - By default used for responsiveness wrap the elements ~~like page~~
- `flex-wrap: wrap;` - By default do nothing.
- `flex-wrap: wrap-reverse;` reverse the wrap use less likely.
- `flex-flow: row reverse wrap;` - It's a ~~too~~ shorthand to write both properties together flex direction and flex wrap.
- `justify-content: center;` - Content will be centered.
- `justify-content: space-between;` ~~+ no spaces in corners but equal spaces b/w same boxes.~~
- `justify-content: space-around;` ~~= equal spaces b/w boxes and corners del 2017~~
- `justify-content: space-around;` ~~= equal space of every box on both sides~~
~~(left right both).~~
- `align-items: center;` - vertically centers item.
- `align-items: flex-start;` - By default - bring up items ~~in flexbox~~ ^{Spiral} first row ~~2017~~

- align-items: flex-end; → Go down.
- align-items: stretch; → Create vertically evenly space b/w elements when responsive testing.

flex properties - For flex items :-

- Order: 2; → Higher the order, later it shows up in the container.
- ~~flex-grow: 3;~~ → Stretch while responsive testing.
- flex-shrink: 3; → Negative numbers are not acceptable in ~~flex-shrink~~ that's why flex-shrink is used.
- flex-basis: 150px; → when flex-direction is row sets width, and when flex-direction is column - sets height.
- flex-grow: 1 3 205px;

flex grow flex shrink flex basis.

- align-self: flex-start;
- align-self: flex-end;
- align-self: flex-center;

used to align a single element in flexbox.

For more visit: [flexbox28.html](#).

#30 Responsive design and font units

Date _____

→ What is responsive design?

* Endless new resolutions and devices are difficult to support separately for a web developer.

* Responsive design is a way for a web developer to make the website adapt to all the devices and resolutions.

* Responsive design is a necessity and not a luxury anymore!

Responsive design is easy

→ Various ways to achieve responsive design

1. Setting up the viewport.

2. Use max-width / max-height.

3. Using CSS Media Queries.

4. Using rem / vh / vw units over pixels.

→ $10\text{px} \rightarrow 10\text{em}$

Related
Responsive
 $\text{padding} \Rightarrow 3\text{em} = 300\text{px}$

~~Don't use ems when you : they won't~~

• EM

Spiral

#Viewable in computed size in developer tool window.

• Rem

Date _____

Rem HTML like font size ka base pe depend karta.

HTML f

font-size: 25px

font-size: 3rem = 75px

padding: 3rem = 75px

while in px padding does not work like that too.

(This is right answer).

• Vh :- Stands for viewport height and if values 0% to 100vh used to give size to any container or background to your viewport screen height.

height: 100vh; (occupy whole screen).

• Vw :- Stands for viewport width and if values 0% to 100vw and used to give size to any container or background to your viewport width screen width.

width: 50vw; (occupy 50% width).

For more visit: [responsive.29.html](#)

responsive.29.html

#30 Media Queries:

Date _____

- **Media Query:-** A media query consists of a media type and can contain one or more expressions, which resolves to either true or false.

@media query is composed of an optional media type and any number of media feature expressions which may optionally be combined in various ways using logical operators.

→ They are case sensitive.

→ Use to create responsive design.

@media only screen and (min-width: 100px) and (max-width: 500px)

display: block;

background-color: red;

For more: visit: [mediaQ30.html](#)

#31 More on CSS Selectors

Date _____

• `div p {`

color: yellow;
font-weight: bold;

}

→ All the P tag will get affected which parent is div, even if the P is not direct child of P.

siblings

• `div > p {`

}

→ Only those P will be affect who are the direct child of div. { (x) (y) (z) }

• `div + p {`

}

→ Only those P will get affected who are written just after closing div tag. { (x) } (y) (z)

→ P is the next sibling of div.

for more visit:

more on selectors 31.html

#32 Attribute & nth child pseudo selection.

Date _____

- `a[target]` { = }

→ only affect link which is targeted.

- `a[target = "-blank"]`

{
=

→ only affect those link which has target = "-blank" set.

- `a[target = "self"]` { = }

→ only affect those link which has target = "self" is set.

- `input[type = "email"]`

{
=

→ only affect those input value type is email.

→ Nth child :-

- li:nth-child(3)

{
=

→ this will change the CSS of third item in list.

Why,

- li:nth-child(2n+0)

child

for odd

- li:nth-child(odd)

{
=

changes odd items

for even

- li:nth-child(even)

{
=

changes in even items

For more visit : alt nth ch pseudo sel 32.html.

#33 Before and After Pseudoselector

Date _____

• Before pseudoselector :

→ ::before and ::after pseudoselectors allows you to insert "content" before and after any non-replaced element (eg. they work on a `<div>` but not an `<input>`). This effectively allows you to show something on a web page that might not be present in the HTML content.

• element ::before {

Content : "this is content"

}

Will set the content before the element.

• element ::after {

Content : "xyz"

Will set the content after the element.

Element = div, section, body, h1...etc,

Ex - header ::before {

Content : "

background : #ccc

For more visit

: before and after pseudos33.html

#34 Box shadow and text shadows

Date _____

→ Box Shadow :-

- box-shadow : offset-x offset-y blur-radius color;

→ box-shadow : 2px 8px green;

- box-shadow : offset-x offset-y blur-radius spread-radius color;

→ box-shadow : 2px 8px 10px 34px rgba(...);

→ box-shadow : -2px -3px 10px 34px rgba(...);

negative values will create
opposite side shadow.

→ Text Shadow :-

- text-shadow : offset-x offset-y color;

" " ; " 2px 4px -2px white;

- text-shadow : offset-x offset-y blur-radius color;

" " ; 2px 3px 9px rgba(...);

for more visit : [box and text shadow 84.html](#)

#35. Variables | and custom properties

Date _____

→ You can create variables globally or
locally declared so that you can use
it to create custom properties where
wanted.

→ For local → Inside the block.

• - {

{ --color: var(--primary-color); background-color: var(--background-color); }

} --color: var(--primary-color); background-color: var(--background-color); }

use :- background-color: var(--primary-color);

→ For globally declared - on the top.

{} ; root {

--primary-color: blue;
--danger-color: red;
z-index: 22px;

for more visit [this link](#) as local.
use same

#36. Keyframes and animations.

Date _____

→ Creating an animation.

• box {

animation-name: hritik;

animation-duration: 8s;

animation-iteration-count: 300; or (infinity)

animation-fill-mode: forward;

(Naage jake, animation aur jaega whi)

animation-fill-mode: alternate;

(animation ek baar chal ke ruk jaega whi).

animation-timing-function: ease-in; (typical)
(Controls the speed on the journey of animation).

animation-delay: 3s;

Animation 3 second ice baad chalgai.

animation-direction: reverse;

(direction of animation will be reversed).

① keyframes hritik {

from {

width: 200px;

}

to {

width: 400px;

}

→ Shorthand for animation:-

animation : animation-name

animation-duration

animation-timing-function

animation-delay

animation-iteration-count

animation-fill-mode

→ Ex-

Animation: hslide 5s ease-in 12 backfor

→ Another way for keyframe:-

@ keyframes hslide {

0% {

top: 0px

left: 0px

25% {

top: 250px

left: 0px

50% {

top: 250px
left: 250px

75% : 1st keyframe

100% {

top: 0px

left: 250px

for more visit : [animations360.com](#)

#37. Transitions in CSS:-

Date _____

box {

transition-property: background-color;
transition-duration: 2s;
~~transition-timing-function: ease-in;~~
transition-delay: 0s;

}

box:hover {
background-color: white;

}

→ Shorthand for transition:-

transition: background-color;

transition: transition-property transition-duration
transition-timing-function transition-delay;

transition: background-color 2s ease-in;

→ For many or all transitions

box {
}

transition: all 2s ease-in;

box:hover {

background-color: white;

height: 400px;

width: 400px;

border-radius: 100px;

font-size: 40px;

Sachin

#38: Transform property in CSS

Date _____

• box S

} transition: all 1s ease-in-out;

• box S : hover S

① transform: rotate(45deg);

② transform: skew(45deg);

③ transform: scale(2) (2x bigger)

④ ~~translate~~ transform: translateX(123px);
(move towards X-axis by 123px).

⑤ sky, ~~translate~~

transform: translateY(123px);
second transform: translateZ(123px);

⑥ transform: translate(123px, 124px)
(move the box towards X,Y direction by
123/124px). Create your own for
understanding more precisely.

for more visit: [transform 38.html](#)

CSS GRID (1)

Date _____

- CSS grid layout excels at dividing page into major regions or defining the relationship in terms of size, position, and layer, between parts of a control built from HTML primitives.

For desktop

navbar		
Sidebar	Main	
Content 1	Content 2	Content 3
Footer		

for mobile

Navbar	
Sidebar	Main
Content - 1	
Content - 2	
Content - 3	
Footer	

• Container {

display: grid;

grid-template-columns: 300px 100px 100px;
 (create 3 columns of items of 300px (1st)
 100px (2nd) and 100px (3rd).)

grid-template-columns: 300px auto 300px;
 (same as previous but it will automatically adjust
 2nd column)

grid-template-columns: 1fr 4fr 1fr;
 (1:4:1 screen column divide)

grid-template-columns: repeat(4, auto);
 (for ~~multi~~ too many items this will
 fill all the items.)

grid-gap: 3rem; set gap between items in a grid.

For more visit: [css grid 39.html](#)

UI # CSS Grid (2)

Date _____

- grid {

display: grid;

(makes the display to grid).

grid-template-columns: 1fr 2fr 4fr;

Same as grid-template-columns but it maintains width of row and size will create rows 1:4:1 and the rest will be default 1fr (means 1).

grid-template-rows: 2fr;

(all the rest items will be set a width of 2fr from default 1fr by this code).

grid-template-columns: 1fr 2fr 2fr;

grid-gap: 1rem;

}

for more visit : [Grid 40.html](#)

#92 CSS Grid (3)

Date _____ #

→ Another way to write grid gap:-

grid-column-gap: 7rem; (only apply on column)
grid-row-gap: 1rem; (" " .. rows).

→ Spans - rows and columns.

• box: first-child {

grid-column-start: 1; grid-column-end: 4;

grid-column-end: 3;

grid-row-start: 1;

grid-row-end: 3;

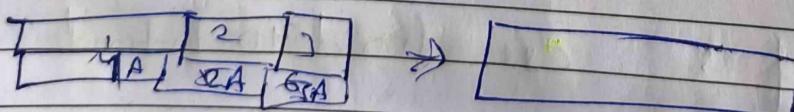
or

grid-column: 1 / span 3;

grid-row: 1 / span 2;

{ Shorthand }

{ spans 1 to 3 items in column
span 1 item in row }



for more visit : [css grid](http://cssgrid.net)
create
fill all
grid-gap

Hybrid-CSS (9) Autofit & Minmax

Date _____

Autofit → Automatically adjust the grid minmax (x, y) → used for responsiveness from x to y change.

→ Autofit and minmax both used together for responsiveness in grid.

→ `grid-template-columns: repeat(autofit, minmax(400px, 1fr))`

For more visit? [css GRID 42.html](#)

#43

CSS Grid (5) Template Area.

Date _____

grid-template-areas:

' navbar navbar navbar navbar '

' section section section section '

' footer footer footer footer '

#navbar {

grid-area: navbar;

}

#section {

grid-area: section;

}

#aside {

grid-area: aside;

}

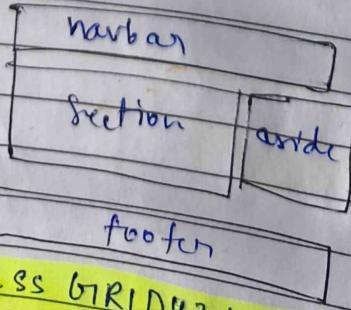
#footer {

grid-area: footer;

}

```
<div id="navbar"> ... </div>  
<div id="section"> ... </div>  
<div id="aside"> ... </div>  
<footer> ... </footer>
```

output ⇒



for more visit: [CSS GRID43.html](#)

#44 CSS Grid (6) - media queries

@media only screen and (max-width: 300px) {
Date _____}

• Containers {

```
display: grid;  
grid-gap: 1rem;  
grid-template-areas:  
  "navbar" . . .
```

grid is implemented with media queries.

}

Span { display: block;
text-align: center }

}

@media only screen and (min-width: 300px) and
(max-width: 500px) {

• Containers { . . . }

Span { . . . }

Section { . . . }

}

{

and so on →

}

For more visit : [CSS GRID DAY 4.html](#)