Homework
1) Embed YouTube videos using iframe
2) Audio & video tag

what you want to embed → To apply border

<iframe src = "https://__" frameborder = "0"
height = " " width = " " > </iframe>

iframe tag is inline element.
Embeding YouTube videos in our webpage
1) Choose the desired video.
2) Click 3 dots & then click share
3) Now click on embed
4) Copy code & put in .html file

Forms
1) It is a good practice that we first add label and then input tag. In for attribute of label tag, we need to add the id.
2) name attribute is also useful when servers are used and also it acts as a identifier.
3) There should be one to one mapping for labels that's why id is used.
4) select and option tag is used to add drop-down in our form.
5) At backend, value attribute of option is sent.
6) <input type = "radio" name = "rating" id = "rating1" value = "1">
<label for = "rating1" > 1 </label>
* name should be same if we don't want that multiple selections are done.
7) When we click on submit, then packet of the information is created & sent to the backend.

Scanned with Cam

10/01/2023

Gradients → (Transition in color)
CSS gradients let us display smooth transitions b/w two or more specified colors.
Types of gradients
1) Linear gradient goes down / up / left / right / diagonally
In this we must define atleast two color stops.

background-image : linear-gradient (red, blue);
By default top to down is direction
: linear-gradient (to right, red, blue);
left to right

to top → bottom to top
to left → right to left
to bottom right → from left corner to right corner (Diagonally)
How to make it at a particular angle?
linear-gradient (angle, red, blue);
45 deg, 210 deg etc.
Note → linear-gradient (45 deg, red, blue, green); → There can be more than 2 colors as well.

We can also use rgb & rgba format
To give transparency effect
2) Radial gradient
It is defined by its center. To create a radial gradient, we must define atleast 2 colors.

background-image : radial-gradient (red, blue);
: radial-gradient (circle, red, blue);
To make circular gradient

Scanned with Cam

Homework
1) Explore conic gradient
2) Add border via shadow
3) Explore spread radius of shadow

radial-gradient (yellow 5%, orange 10%, red);
% of area that will be occupied in the gradient

Note → % can be given in linear gradient as well.

3) Conic gradient
background-image : conic-gradient (red, green, blue);
conic-gradient (red 45deg, yellow);
conic-gradient (from 90 deg, red, yellow);
entire conic gradient is rotated by 90

CSS shadow effects
Shadow can be added to text & to elements

1) Text-shadow
text-shadow : 3px 3px;
text-shadow : 3px 3px red;
text-shadow : 3px 3px 3px red;
horizontal vertical Blur Color of shadow
Generally 2 px is used as good practice
text-shadow : 3px 3px 3px red, 5px 5px 5px green;
Multiple shadows

2) Box-shadow
box-shadow : 10px 10px;
Whatever will be text color inside box will be color of shadow.
horizontal, vertical, blur
box-shadow : 10px 10px 10px green; color of shadow
box-shadow : 10px 10px 10px red, 5px 5px 5px green;
multiple shadowns

Note → By using -ve values in horizontal & vertical value, we can change direxn of shadow.

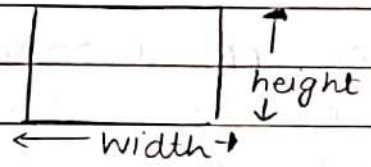
Scanned with Cam

4) Position text in corners and centre of an image.
5) matrix () → 2D Transform
6) display : inline - block ; → ? (Inline + h, w adjust)

Spread radius This is 4th value in box-shadow syntax
+ve value → Shadows expand & grows bigger
-ve value → Shadows shrink
Default value is 0.

CSS dimension properties

1) Width
2) Height



3) Max-height → Suppose we put max height 50px & if content could not come in 50px, then overflow occurs. max-height : 100px ;
box will resize itself till 100 px & then overflow will occur. This is basically a limit
4) Min-height → If content is able to fit, then height will be of min-height but if content is big, then block will automatically resize itself.

CSS position property
1) Static
position : static ; → By default it is setup. If not mentioned, then also behaviour is same.

<p>	p
<h1>	h1
<p>	p
<h2>	h2

2) Relative
Positioned relative to its normal position.
position : relative ; → we can change position by using left, top, right and bottom property. These won't work with position : static ;
Other content will not be adjusted to fit into any gap left by element.

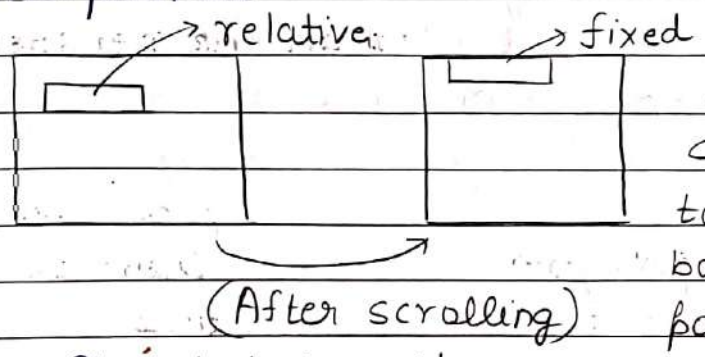
Scanned with Cam

7) Why no perspective is used when rotating w.r.t z axis?

3) fixed top, left, right, bottom → to change position
position : fixed; → If we scroll, then it remain intact at same place.

4) absolute
It can be applied only when its parent element (nearest) is positioned explicitly. If no such element is positioned, then body is treated as parent element
↳ document body
Parent element → non-Static

5) Sticky
Toggles b/w relative & fixed dependending upon scroll position.



position : sticky
top : 0px;
↳ we have to mention top, left, right or bottom to make it position relative

Ex → Chat bot in webpages

2D Transforms
CSS transform allow us to move, rotate, scale and skew elements

1) Translate () → This is movement
transform : translate (100px, 200px);
X-axis Y-axis

2) rotate () → Rotates the element
transform : rotate (45deg);
↳ Angle by which we want to rotate

3) scale () → Zooming

Scanned with Cam

→ In vertical direxn

transform : scale (2, 3) ;

↑ In horizontal direxn

Note → We can also use decimal values in scale().

4) Skew () → Used to tilt by some angle
transform : skew (20 deg);

3D Transforms

Here we can perform transforms in x, y and z axi
If we want to perform operations in 3rd axis, then explicitly we have to do something & i.e to turn perspective.

1) transform : translate Z (-10px);
perspective (10px); → To on the perspective

transform : perspective (10px) translate Z (-10px
Distance b/w eyes & z-axis plane is perspective.
goes far away from us in -z axis

2) transform : perspective (350px) scale Z (2);
scales in the direxn of z but we will see front view & won't be able to see the change & for that we need to rotate the figure.

11/01/2023

Depth in Conic gradients

1) conic-gradient (at 60% 45%, red, yellow);
center position

2) repeating-conic-gradient (red 10%, yellow 20%
Repeats itself

Scanned with Cam

Adding border with the help of shadow

box-shadow : 0 0 0 2px ;

Spread radius

matrix () in 2D transforms

The matrix () method combines all the 2D transforms method in one.

matrix (1, 0.3, 0, 1, 0, 0) ;

scaleX, SkewY, SkewX, scaleY, translateX, translateY

display : inline-block

Makes element inline & also if height & width are specified, then that is applied.

Why is no perspective required in rotate Z ?

It is because in rotation, there is nothing to do with the depth.

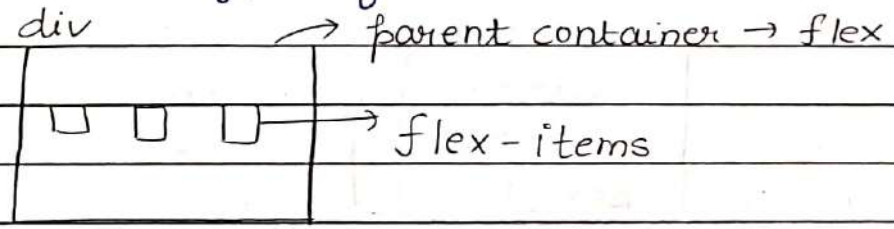
Scanned with Cam

12/01/2023

Flexbox → 1D layout model

This is used to get more flexibility in our layouts & to simplify responsive layout design. Space distribution can be done & also very powerful aligning capabilities are provided to us.

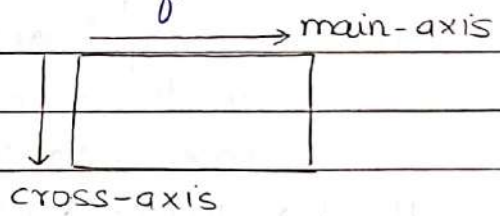
Flexbox is a type of container



The children elements are known as flex-items

Types of axis

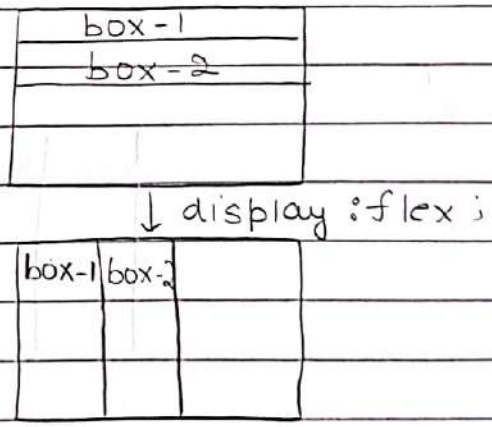
1) main-axis → horizontal
2) cross-axis → vertical



This is set by default.

Flex container : properties

1) display : flex ; → This will make our container flexbox & then the elements starts getting stacked on the main-axis.



To make the main-axis vertical, we have to use flex-direction.

2) flex-direction

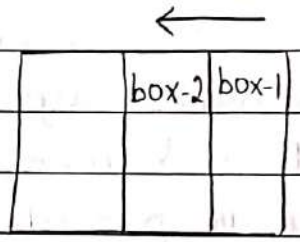
flex-direction : column ; → Makes the main-axis vertical

Note → By default, this property is set to row.

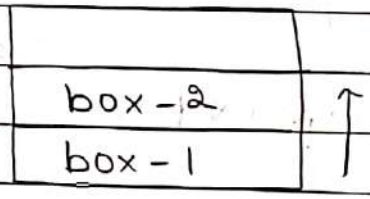
main-axis → horizontal

Scanned with Cam

flex-direction : row-reverse;



flex-direction : column-reverse;

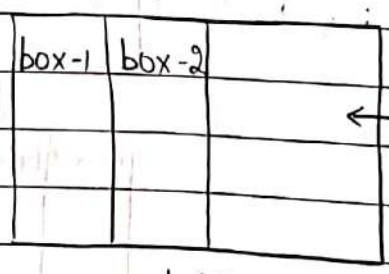


3) flex-wrap.

(i) flex-wrap : wrap; → When we have hardcoded the width & we want to make it intact until it is possible, then this property is used

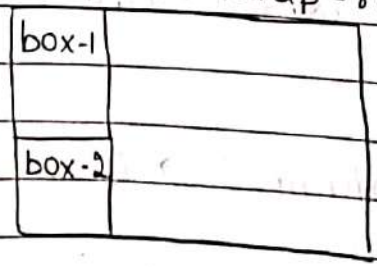
(ii) flex-wrap : no wrap; → This is the default value.
no space

(iii) flex-wrap : wrap-reverse; → Whenever we squeeze the screen, the element starts moving at top to preserve width



← Squeezing

↓ This is simply wrap, but in wrap-reverse box-2 will come on top.



Scanned with Cam

Homework

1) Code flex-flow yourself.

4) flex-flow

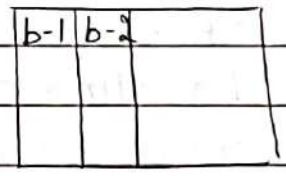
This is short hand notation in which flex-direction & flex-wrap can be written in same line.

flex-flow : row wrap ;

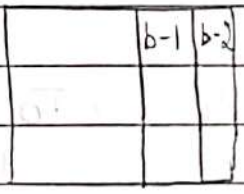
5) justify-content

This will align items according to the main-axis.

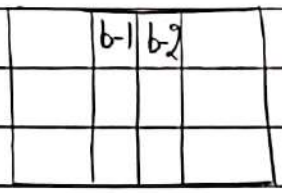
(i) justify-content : flex-start ; → All the elements gets aligned on the left end.



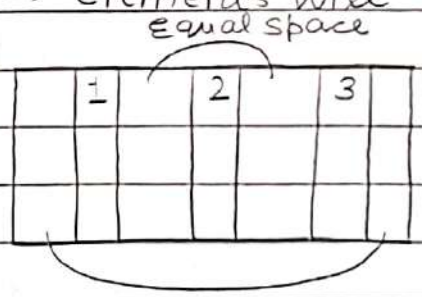
(ii) justify-content : flex-end ; → All the elements get aligned to right end



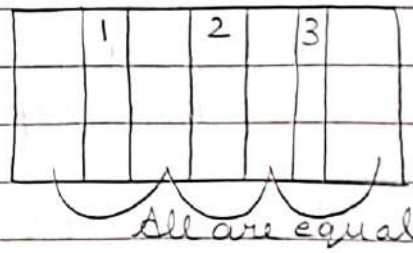
(iii) justify-content : center ; → The elements will get aligned to the center



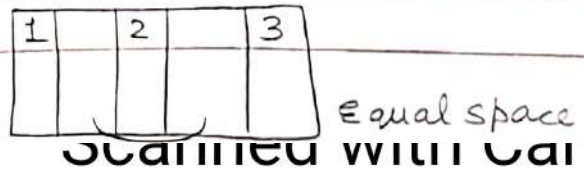
(iv) justify-content : space-around ; → Elements will have same space in between them



(v) justify-content : space-evenly ; → Everywhere there is equal space.



(vi) justify-content : space-between ; → The elements will have only between them



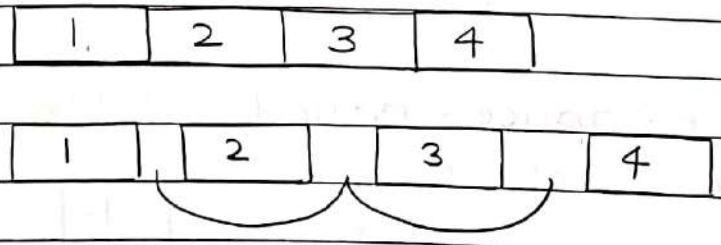
Scanned with Cam

6) align-item
This will align items according to the vertical axis i.e cross-axis.

(i) flex-start → On top of container
(ii) flex-end → Bottom of container
(iii) center → Center w.r.t the cross axis
(iv) stretch → This is the default property & by this the element's height will become equal to height of container
(v) baseline → If content size is different, then also they will be placed with the same base line & are horizontally placed.

Note → To center the content, we can set both align-item and justify-content to center.

gap property

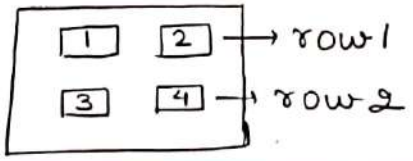


gap will be generated only at these 3 places

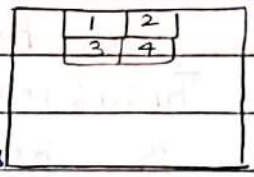
gap : 10 px ;
↳ Short hand notation
row-gap : 10px ;
column-gap : 10px ;

7) align-content
This is used to handle the space b/w the multiple rows.

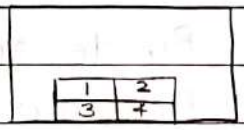
Scanned with Cam



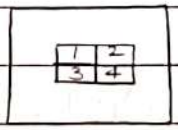
(i) flex-start → This will move the boxes to the top without any space



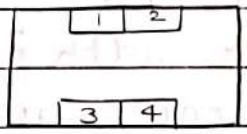
(ii) flex-end → This will move the boxes to the bottom without any space



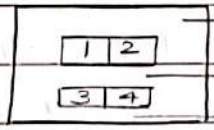
(iii) center → This will moves the boxes to the center of container without any space.



(iv) space-between →



(v) space-around → Only the end spaces are equal.

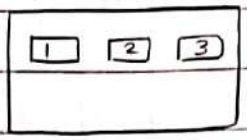


} Not equal spaces.

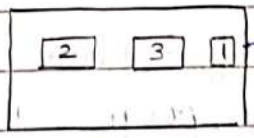
(vi) space-evenly → This is used to make all the spaces equal. End, between, start all the spaces will be equal.

Flex-items : properties

1) order → This is used to change the order of the flex-items. By default order is 0. As the order is increased, then the items shifts to the end. Order can not be put the decimal values.



#box1 { order : 1 } →

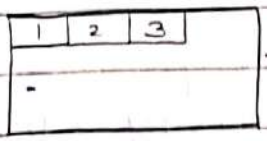


→ Shifts to the end.

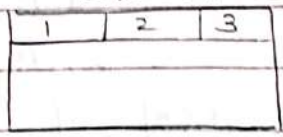
If order is same, then that item just comes before that particular item.

2) flex-grow → This is used to increase width of the element

flex-grow : 1 ;



→



Equal width & covers full container

By default, it has value 0.

Scanned with Cam

We can also use this on a particular item.

3) flex-shrink

This is used to decrease the width of element & also speed of shrink can be set

flex-shrink : 2 ;

By default, it has value 1.

4) flex-basis

This is used to give width to the element.

flex-basis : 100px ;

Note → Difference b/w width : 100px & flex-basis : 100p is that if content is not able to fit, then in width it will be trimmed but in flex-basis, the size of box will increase.

5) flex

It is shorthand notation by which we can manage the above 4 properties.

flex : 3 4 2 120px ;

3 → Order, 4 → grow, 2 → Shrink, 120px → flex-basis

6) align-self

This is used to align only single item ;

(i) align-self : stretch ; → takes height of parent container

(ii) align-self : flex-start ; → Sends that element to top of page

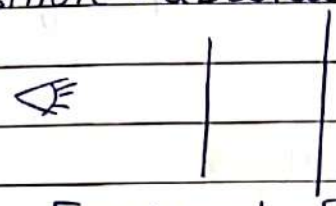
(iii) align-self : flex-end ; → Sends that element to the bottom of that page.

Note → flex-basis : 50% ; → One item will occupy 50% of width of parent container. Keep in mind concept of box-sizing : border-box ;

Scanned with Cam

Parallax effect

This is created by 2 images i.e background & foreground image & they will have some speed difference when we scroll. One moves fast & other moves slow.
Now we need to overlap the images & for this position : absolute ; is used.



Foreground Background-image

We need to use 3D axis.

Note → It is best practice to make a div and then code inside it.

Tag used in parallax website

1) transform-style : preserve-3d → This specifies that child elements will preserve its 3D position
2) z-index → This specifies stack order of element. If stack order is high, then it will be infront.
3) object-fit : cover ; → Image keeps its aspect ratio & fills the given dimensions. The image will be clipped to fit.

14/01/2023

Grid

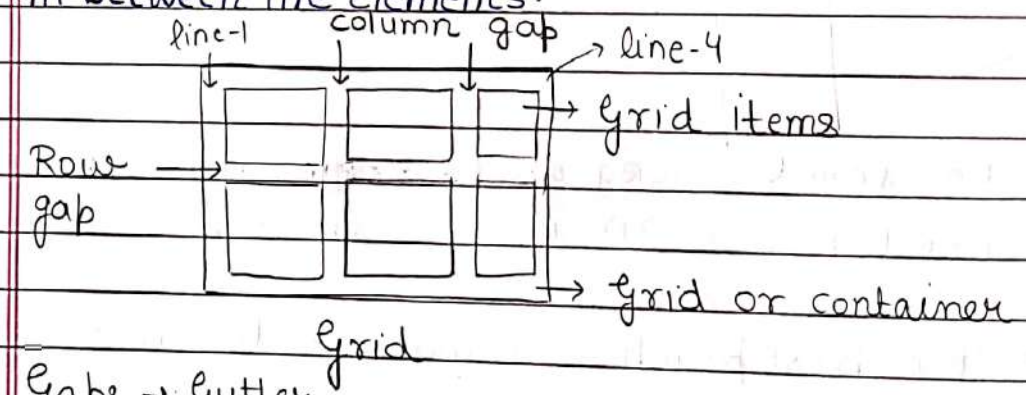
This is a 2D layout model. Ex → 2D array, matrix, table etc. This has rows and columns.
A grid will have rows, columns and gaps and these gaps are known as gutter. There will be row-gap and column-gap. Flexbox is how the content flows & grid is simple layout.

Scanned with Cam

Flexible grid

To create a grid, we need to do display : grid ; In this there will be a container / grid & the elements inside this container are known as grid items. Not

When we do display : grid ; , then spacing occurs in between the elements.



1)

Gaps → Gutter

Note → Flex → Relative to content positioning
Grid → Relative to layout creation

Properties of grid

1) display : grid ; → This is used to make the contai a grid. The elements will have some spacing in betwee
2) grid-template-colums ; → This is used to introduce columns in our grid.

grid-template-columns : 100px 210px 200px ;

width of 1st column — 2nd column width — 3rd column width

3) grid-template-rows ; → This is used to introduce rows in our grid.

Scanned with Cam

grid-template-rows : 90 px 90 px 90 px ;
3 rows with height 90px.

Note → This is not a good practice i.e to mention in pixels. There is a unit named fr which means fraction.

fr unit

1) grid-template-columns : 1fr 1fr 1fr ;
3 columns of equal width & will cover full grid.

This can also be written as
repeat (3, 1fr) ;
1fr 1fr 1fr

Combination of pixels & fr can also be written

100px repeat (3, 1fr) 200px ;
width of 1st column — 1fr 1fr 1fr — width of last column

This is used to make flexible grid

The same fr unit can also be used in the grid-template-rows

gap
gap : 15 px ; → This is used to introduce gaps or gutter in grid.

4) grid-column-start : 1 ;
grid-column-end : 3 ;
} This is grid-item property & thus will span from line-1 to line-3

This is used to span in the column-direction

Scanned with Cam

5) grid-row-start : 1 ; grid-row-end : 3 ; } This is also grid-item property and this spans item in vertical direxn from line-1 to line-3.

This is used to span in the row-direction.

Shorthand notations

1) grid-row : 1 /span 3 ;
This means that start on row1 & span 3 rows.

2) grid-column : 1 /span 3 ;
This means that start on column1 & span 3 columns.

Some more properties

1) justify-content
2) align-content
3) justify-self
4) justify items
5) align-items
6) align-self
7) place-items
8) place-self

} Explore them

Scanned with Cam

Homework (14/01/2023)

Letter-spacing
This sets the horizontal spacing b/w the text characters.

letter-spacing : 0.2rem;
Hi much not
increased spacing b/w the letters

img vs background-image
1) We will be using the <img> tag when we want people / or people want to print it.
2) We use background-image when the image is not the part of our content.
3) img tag will dramatically improve performance of animations over a background.

background-attachement
This sets whether a background image's position is fixed within the viewport or scrolls within its containing block.
(1) scroll This is default value & the background-image will scroll with page.
(2) fixed Background image will not scroll with page
(3) local Background image will scroll with its element's content

background-size
This sets the size of element's background image.
(1) contain Scales the image as large as possible within its container without cropping or stretching the image.
(2) cover Scales the image while preserving the ratio

Scanned with Cam

Homework
1) Explore grid area from MDN
2) grid-auto-column is there or not?

to smallest possible size to fill container.

17/01/2023

Homework (15/01/2023)

Filters in CSS
This defines visual effects like blur to an elemen
(i) blur filter : blur (2 px); → mentioned in px
(ii) brightness filter : brightness (200%); → mentioned i
(iii) grayscale filter : grayscale (100%); → mentioned in
→ Class-notes
Grid Area
This is applied to grid items itself along with CSS.
Grid area is a shorthand notation of grid-row-start, grid-column-start, grid-row-end and grid-column-end.

Syntax
grid-area : 1/1/3/4; → column-end
column-start
row-end
row-start

grid-row-start : 1;
grid-row-end : 3;
grid-column-start : 1;
grid-colum-end : 4;
} grid-area : 1/1/3/4;

Grid-template-area
This is used to name rows & columns of a grid & set its layout.

Scanned with Cam

3) Find difference b/w grid & inline grid.
4) Do explore time on page 8.


```
grid-template-area : "hd hd" ;
                     "side main" ;
                     "ft ft" ;
```

	col1	col2
row1	hd	hd
row2	side	main
row3	ft	ft

```
grid-area : hd ;
grid-area : side ;
grid-area : ft ;
grid-area : main ;
```

→

header	
side	main
footer	

Note → If we have defined say 2 rows & we try to insert 3 rows, we can do that but it will get squeezed.

Advanced grid concept

1) fr unit
It represents fraction.
2) repeat function
repeat (3, 1fr); → 1fr 1fr 1fr
3) grid-auto-rows: minmax ( ); → When we have to use rows but we don't know how many rows to take.

↱ automatically selects maximum

minmax (100px, auto);

↳ minimum height of the row

Scanned with Cam

Grid properties

1) Justify content This is same as flex box & hence the alignment can be done w.r.t main axis.
2) Align item This is same as flex box & hence the alignment can be done w.r.t cross axis.
(i) start To top of the cell in grid
(ii) end To bottom of the cell in grid
(iii) stretch This will occupy the full cell size space. This is by default property.
(iv) baseline Same as start but o/p will vary when content size will differ.
3) Justify self This is applied on the grid items & is used to align items along main-axis within a given cell
(i) start Aligns to start of cell
(ii) end Aligns to end of cell
(iii) center Aligns to center of cell
4) align self This is applied on the grid item & is used to align items w.r.t cross axis.
5) align content In this, rows spacing is considered i.e spacing b/w the rows can be managed by this property.

6) Place-items Shorthand notation for justify item & align item.

place-items : center;

7) place-content Shorthand notation for justify content & align content

place-content : center;

Scanned with Cam

8) place-self Shorthand notation for justify self & align self.

justify content vs justify item

justify content	justify item
↓	↓
Position of the box changes.	Width changes according to content

Media Queries → Good practice to add them at end of CSS file.

1) Viewport The area of window in which web content can be seen. We use dimensions of viewport usually width & sometimes height as the basis of our media queries.

Media queries are used to set different style rules for different devices or sized screens.

Logic ↓

@media (feature : value)

Essentially, media query breakpoints are pixel values that a developer /designer can define in CSS. When a responsive website reaches those pixel values, a transformation occurs.

```
@media (max-width : 400 px) {
        // CSS properties
}
```

→ As soon as this breaches, The below CSS properties will be applied. i.e (<= 400 px is the condition)

Multiple Breakpoints

```
@media (min-width : 600 px) and (max-width : 900 px) {
        /* CSS */
}
```

>= 600 and <= 900

[600, 900]

Scanned with Cam

Nested Grids → Grid within grid is nested grid.
This is simple & can be done simply by applying display: grid; for both parent & child.

```
.container {
    display : grid;
    //--
}
#one {
    display : grid;
}
```

18/01/2023

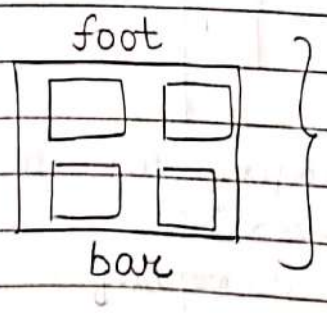
Homework of 17th Jan 2023

1) grid-auto-columns
This property sets the size for columns in a grid container.

grid-auto-columns: minmax (100px, auto);

Working of this is same as that of grid-auto-row

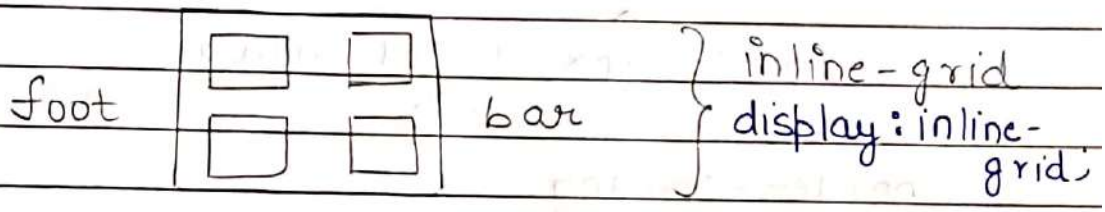
2) Grid vs Inline Grid
In grid, the element made is a block level element



Grid
display : grid;

Scanned with Cam

In inline-grid, the element made is an inline element.



CSS cheatsheet Notes

Font variant
It specifies whether or not the text is displayed in small caps font or not.

font-variant : small-caps ;
Text decoration
It specifies the decoration added to text.
text-decoration : underline ; → Hi
: overline ; → Hi
: wavy underline ; → Hi
Letter spacing
Used to specify space b/w the letters/characters
letter-spacing : 5px ;
Text transform
Controls capitalization of text
text-transform : uppercase ;

Note → word-spacing : 2px ; → Controls spacing b/w words.
text-indent
Specifies indendation of first line in a text-block.
Negative values are also allowed.
Line height
Used to control the spacing b/w the lines.

Scanned with Cam

border
Shorthand notation

width

border : 20px dotted blue;

style ↳ color of border

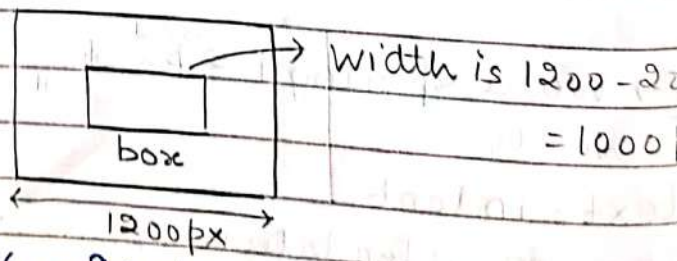
border-spacing
Sets distance b/w borders of adjacent cells.
caret-color
It is used to specify the color of cursor in inputs, textarea or any other element that is editable.

19/01/2023

Animation
Gradual change from one style to another style in our webpage is animation.

Functions in CSS

i/p → Function → o/p
Function takes some i/p & gives some output.
scale(), repeat(), translate() etc. are some of the functions in CSS.
Mathematical functions in CSS
1) calc( )



width : calc(100% - 200px);
of parent container

Scanned with Cam

Homework
1) Difference in :root & * for global var?
2) Make Hire me button

2) min ()
height : min (50%, 400px);
whichever is small is assigned to height.

3) max ()
height : max (60%, 140px);
whichever is bigger is assigned to height

Variables
This concept is similar to variables of programming language.

```
.box {
    --dark-red : blue; → To assign value to variable
color: var(--dark-red); → To use variable
}
```

The scope of this variable will be only till this box.

```
:root {  → This is used to create the variable.
    --dark-red : #981a2c;
}
```

global variable

To have global scope i.e it can be accessed in whole CSS file

Ques → What is the difference b/w these two?

```
:root {
    --dark-red : blue;
}
```

```
* {
    --dark-red : blue;
}
```

There is no difference as o/p will be same but it is better if we use :root to declare global variables.

Scanned with Cam

Using variables is considered as best practice.

Animations

We can use transition and animation, keyframes to animate our object

(1) Transitions

(i) transition-property

To apply which transition property to apply. On wh property, transition should be applied.

transition-property : all; → bad practice as it takes more power & hence not good when performance is taken under consideration.

(ii) transition-delay

After how much time, transition will be applied or transition will start.

transition-delay : 1s; → After 1s, transition will be applied.

(iii) transition

This is a shorthand notation.

Transition-

transition duration

transition : background-color 2s ease-in 4s;

transition-property

transition-func

(iv) transition-timing-function

This specifies the speed curve of the transition effect.

- ✓ ease → slow start, then fast, then end slowly.
- ✓ linear → same speed from start to end
- ✓ ease-in → slow start
- ✓ ease-out → slow end

Scanned with Cam

✓ ease-in-out → slow start and end

Note → Suppose that we make to make more than 1 changes, then we have set transition: all; -property.

(v) transition-duration
How much time a transition takes to complete.
transition-duration : 2s; → transition will happen for 2s.

Note → cubic-bezier (n,n,n,n) can be used to define our own transition-timing-function & this can be done via inspect element (purple box).

(2) Animations and keyframes
(i) animation-name
To set name of animation
animation-name : bhavya;
(ii) an-imation-duration
For how much time animation runs
(iii) animation-iteration-count
How many times animation will run.
(iv) animation-timing-function
Same as that of transition
(v) animation-delay
After how much time, animation will run.

Keyframes To define starting, end or between state of the animation, for this purpose we need keyframes.

Scanned with Cam

name of animation

```
@ keyframes bhavya {
    from {
        width : 212 px ;
    }
    to {
        width : 800px ;
    }
}
```

name of animation

```
@ keyframes bhavya {
    0% {
        width : 212 px ;
    }
    30% {
        width : 35 % ;
    }
    65% {
        width : 70 % ;
    }
}
```

0% animation

30% animation

→ Other properties can also be changed

Note → animation-duration : infinite ; → The animation will run forever.

(vi) animation - direction

This changes the direx^n of animation.

animation - direction : reverse ; → If original the animation moves from left to right, then by applying this, animation will move from right to left

animation-direction : alternate ; → Animation is played forward first & then backwards.
animation-direction : alternate-reverse ; → First backwards & then forward
animation-direction : normal ; → This is the default value

(vii) animation-fill-mode
This property specifies a style for element when animation is not playing

✓ forwards → Element retains style value that is set by last keyframe
✓ backwards → Element retains style value that is set by first keyframe
✓ none → this is the default value.
✓ both → animation follow rules of both forward & backward. (Pending, will be done in next class)
22/01/2023
Homework of 19/01/2023

::before and ::after
This is a pseudoselector & we can place the content before the actual content. Using ::after we can place content after actual content. → class

Hire me | btn-pink

```
.btn-pink ::before {
        content : "abc";
}
```

↳ This can be anything

abc Hire me
↳ content placed before actual content

```
.btn-pink::after {
        content : "123";
}
```

Hire me 123

Scanned with Cam

transform-origin
This will set the origin for an element's transform

transform-origin : left ; → This is used to start the transition from left side.

22/01/2023

Adjacent sibling selector (+)

```
<div>
    <p> Hi </p>
    <p> Hi1 </p>
</div>
    <p> Hi2 </p>
    <p> Hi3 </p>
```

```
div + p {
    background-color : r
}
```

The property will be applied to that paragraph only which just comes after the ending of div.

If b/w and <p> , say we added h1, then property won't be applied

General sibling selector (~)

```
div ~ p {
    background -color : red;
}
```

This property will get applied to all the paragraphs which comes after .

Scanned with Cam

23/01/2023

text-align
(i) left → aligns the text to left.
(ii) right → aligns the text to right.
(iii) center → centers the text.
(iv) justify → stretches the lines so that each line has equal width.

24/01/2023

Tailwind CSS
This is a utility based CSS framework by which we can style our webpage. We will be styling our webpage without leaving our HTML webpage.

Setup tailwind
Go to official website & select play CDN & copy <script> tag code in head tag. By this method, we won't get any suggestions. Suggestions don't come.

Classes in tailwind

(1) class = "flex"
This is used to make our container flexbox.

(2) class = "justify-evenly" Same as justify-content space-evenly.

(3) class = "mx-4" This is used to give margin into the x-direction.

(4) class = "bg-red-400" Setup the background color.

Scanned with Cam

Setup of tailwind CSS

1) Install node.js in your computer and then keep on click next until we reach the finish button.

2) If VS code was open, then just restart the system / VS code. Now open terminal in the VS code.

3) Run command

```
npm install -D tailwindcss postcss
autoprefixer vite
```

Wait for the command to execute fully.

4) After the above command is fully executed, run another command

```
npx tailwindcss init -p
```

Wait for the command to get fully executed.

5) Now we have to check whether extension of tailwind CSS intellisense has been install or not. If not, then install the extension.

6) Now on the left side, go in tailwind.conf.js, add the * in content.

```
content : ["*"],
```

7) Now go inside the file named package.json & add the script

Scanned with Cam

```
"scripts" : {
        "start" : "vite"
  },
```

8) Create css file say style.css & link it with the html file. Now write 3 lines in this css file

```
@tailwind base;
@ tailwind components;
@ tailwind utilities;
```

9) Now in the terminal, write the following command

```
npm run start
```

We will see a link under local : ——. Go on the link & this will act as the live server now.

Razorpay clone website classes used

(1) class = "bg-deepBlue" To make the background color deepBlue. We have mentioned the color in the tailwind.config.js file.

(2) class = "relative" This is used to set the position property as relative.

(3) class = "w-[1080 px]" To explicitly give width to container & the width give is 1080 px.

Scanned with Cam

(4) mx-auto This is similar to margin : 0 auto in CSS & is used to horizontally center the element i.e equal margin from left and right side.

(5) flex To make the container flexbox.

(6) items-center To align the items according to vertical axis & is same as align-items : center;

(7) justify-between This is used to provide equal space between the elements & it is same as justify-content : space-between;

(8) cursor-pointer This is similar to the cursor: pointer; i.e. our cursor becomes like hand when hovered over the element.

(9) py-7 To give padding from top & bottom 7 px.

(10) pr-7 To give padding from right equal to 7 px.

(11) text-white To make the text color as white.

(12) font-mullish To apply font-family as mullish

(13) hover:text-lightBlue Whenever hovered

Scanned with Cam

over element, the text color will change to lightBlue.

(14) transition-all The transition property will be set for everyone

(15) duration-200 The transition-duration will be 200 ms.

(16) group When we want to style an element based onstate of some parent element, mark the parent with group class

hover
Payments
hover

Both side hover, property should change

(17) w-full Take full width of its parent.

(18) h-1 This is equivalent to height : 0.25 rem.

(19) hidden This is to make display hidden

(20) group-hover : block Whenever we hover over the group/parent, make the display block.

(21) bottom-0 This is equivalent to bottom : 0px;

(22) space-x-6 This means margin-left is 1.5 rem.

(23) gap-6 This means gap : 1.5 rem;

Scanned with Cam

(24) border - lightBlue To give color to border lightBlue.

(25) border - 2 To set border width as 2 px.

(26) rounded - sm Used to give border - radius as 0.125 rem

(27) text - sm To make text size as 0.875 rem.

(28) font - bold font - style / font - weight : 700;

Scanned with Cam