

Title

1. Introduction to CSS	2
2. Creating our first CSS Website	3-7
3. Color and Backgrounds • background-image / repeat / size / position	8-12
4. CSS Box Model • Margin, Padding, Border	13-20
5. Fonts and Display • display, text-decoration	21-24
6. Size, position and Lists • px, rem, vh/vw, % Position	25-31
7. Flex box • float, display	32-37
8. CSS Grid & media Queries	38-42
9. Transform, Transition, Animation	39-48

Introduction to CSS - 1

HTML is just an skeletal layout of a website. We need CSS to design a website add styles to it and make it look beautiful.

What is CSS?

CSS stands for cascading style sheets.
CSS is optional but it converts an off looking HTML page into a beautiful & responsive website.

Why learn CSS?

CSS is a very demanded skill in the world of web development. If you are successfully able to master CSS, you can customize your website as per your liking.

Your first line of CSS

```
<style>
```

```
body {
```

```
background-color: red;
```

```
3
```

```
</style>
```

Creating our first CSS Website - 2

SMTH at 222 bhn at appu giri

What is DOM?

Dom stands for document object model. When a page is loaded, the browser creates a DOM of the page which is constructed as a tree of objects.

HTML id and class attributes

When an HTML element is given an id, it serves as a unique identifier for that element.

On the other hand when an HTML element is given a class, it now belongs to that class, but every element must belong to one class but every element must have a unique id (if assigned).

We can add multiple class to an element

<div id="first" class="c1 c2"> like this

.....
</div>

→ multiple classes with space

Three ways to add CSS to HTML

1. `<style>` tag → adding style tag into head of

HTML code

2. Inline CSS → adding style attribute

→ `<div style="color:red">`

3. External CSS → adding a stylesheet(.css)

linking to HTML with link

`<link rel="stylesheet" href="Name.css">`

CSS Selectors

A CSS selector is used to select an HTML elements for styling.

With just one selector

```
body {
```

color: red;

background: pink;

```
}
```

Declaration
(property: value)

Element selector

It is used to select an element based off the tag name for example:-

```
h2 {
```

color: blue;

```
}
```

In an ~~each~~ triangle give me id :-

id selector

It is used to select an element with a given id for ex:-

```
#first {
```

is used to target by id

color: aqua;

class selector

It is used to select an element with a given class for style.

• use for class

• Name {

background: red;

color: blue;

Important Notes:

• We can group selectors like this.

h1, h2, div {

color: green;

background-color: red;

• We can use element class as a selector like this!

element-class-name { color: blue; }

color: blue;

:red { color: blue; }

all div element that have red class then styled [blue words]

- * can be used as a universal selector to select all the elements

* {

margin: 0;

padding: 0;

}

- An inline style will override external and internal style

Comments in CSS

/* comments ... */ same as C++

Colors and Backgrounds 3

~~Color~~ CSS rules are simple key-value pairs with a selector we can write CSS rules to change colors and set backgrounds.

The color property

The CSS color property can be used to set the text color inside an element.

eg

(color: pink;) Text color will be

(background-color: red) - background changed to red.

similarly we can set color for different elements

Types of color values

1. RGB → Specify color using Red, green, blue

:("purple") { color: values ex-rgb(200, 90, 70)}

2. HEX Code → Specify color using hex code.

background-color: eg. #ff7f00

with each value X and X in

Q3. HSL → specify the color using HSL values
 e.g. hsl(8, 90%, 63%)
 hue, saturation, lightness
 drawing

The background-color property

The CSS background-color property
 specifies the background color of a container

Brown class

body { background-color: brown; }

background-color: brown;

The background-image property
 Used to set an image as the background

body { background-image: url("Pic.png"); }
 3

The background is by default repeated
 in X and Y direction

The background - repeat property

Can be any of:

1. repeat - x repeat in horizontal direction

2. repeat - y repeat in vertical "

3. no-repeat, no-image, not repeat

See more possible values at MDN docs

The background-size property

spans background area

1. cover fits and no empty space remains

2. contain fits and image is fully visible

3. automatic Display in original size

4. {width} set width and height will be

5. {width}{height} set automatically

5. {{width}}{{height}} set width and height

The background-position property sets the starting position of a background image.

div{

" background-position: left top; }

left top, left bottom, right top/bottom

The background-attachment property

defines a scrollable/non scrollable character of a background image

background-attachment: fixed;

The background shorthand

A single property to set multiple background properties.

div.3 {

background: red url('img.png')

no-repeat fixed right top;

}

One of the properties can be missing given the others are in order.

CSS Box Model - EP

13

Aniket

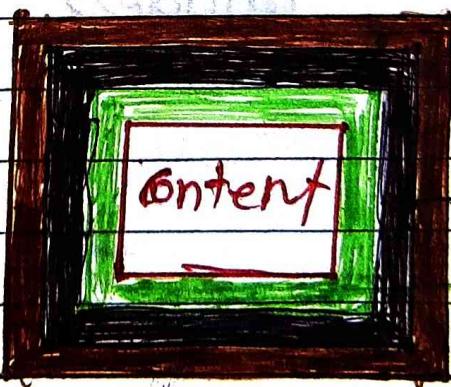
Date
Page

'15

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS model is essentially a box that around every HTML elements. It consists of margin, border, padding and actual content.

The image below illustrates the box model.



Margin

Border

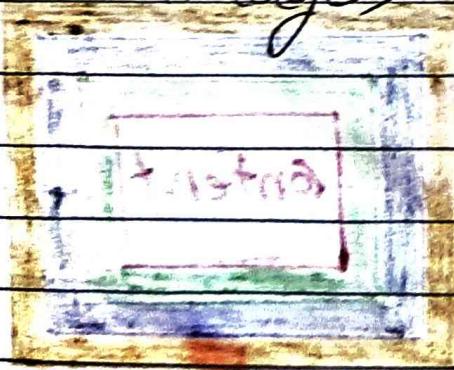
Margin
Padding

Margin - It clears an area outside the border. The margin by default is transparent.

Border - A border that goes around the padding and content.

Padding - Clears an area around the content. The padding is transparent.

content - The content of the box, where text and images appear.



Setting width and height

We can set width and height as follows

#box {

height: 70 px;

width: 50 px;

Note - the total width / height is calculated as follow :

Total height = height + top/bottom padding + top/bottom border + top/bottom margin

Total width = width + left/right padding + left/right border + left/right margin

Setting Margin and padding

box

margin: 3px;

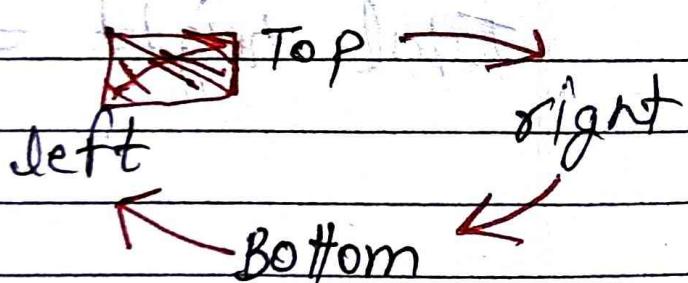
padding: 8px;

top, bottom, left,
right all

same as written

boxall

margin: 6px 7px 8px 9px;



box2

margin: 10px 6px;

left, right

top, bottom

We can set individual margin padding

margin-top: 6px;
margin-bottom: 12px; • use padding
margin-left: 11px; on place of
margin-right: 8px; margin

Setting Border

We can set the border like this

• boxborder {

border-width: 2px;
border-style: solid;
border-color: pink;

{

(OR)

• boxborder {

border: 2px solid pink;

{

Border Radius

We can set border-radius to make box rounded.

border:

div {

border-radius: 7px;

border-radius

with other values like this we can give

circle: 50px solid red

triangle: 50px solid red

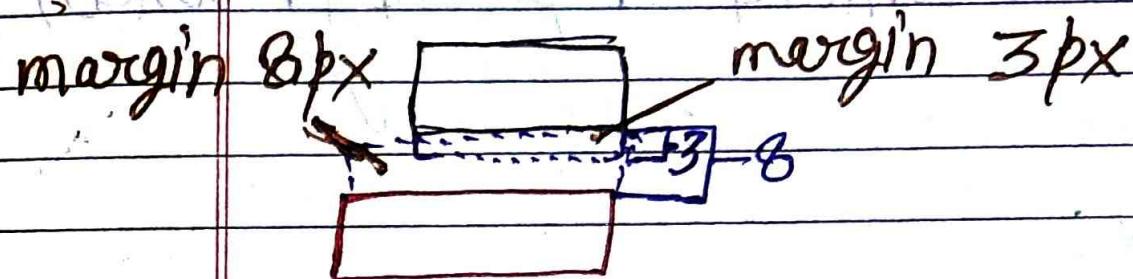
square: 50px solid red

circle

square like this

Margin Collapse [Note]

When two margins from different elements overlap, the equivalent margin will be greater of the two, called margin collapse



Box sizing

Determines what out of padding and border is included in element's width and height.

content - box

Include only content in width/height

border - box

The content width and height includes content + padding + border

.div {

box-sizing : border-box;

box width to two times exaggerated
padding of child of header
child has other size

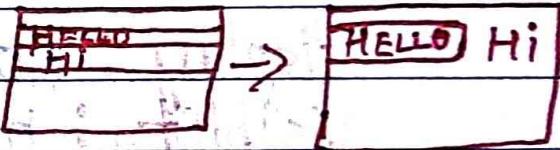
FONTS And Display - 5

display : property

The CSS display property is used to determine whether an element is treated as a block/inline element and the layout used for it children

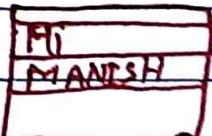
- flexbox / grid etc.

display : inline;



Takes only required space,
No line break, can't set
width/height, margin/padding

display : block;



Takes whole width and make
new-line

H - Height
W - width
M - margin
P - padding

22

Aniket

Date _____

Page _____

54

display : inline-block;

inline

block

→ require place → set H, M, P

→ takes less space → W, P = inline-block

inline-block

Takes require place and we can
set H, W, M, P

display : none;

visibility : hidden;

remove from document can't see but
place is reserved



→ should be empty

show the hidden element easiest

→ will move

Fonte

letter-spacing: 6px; cool boy man
→ n v

word-spacing: 6px; cool-boy-mannu

text-align : ↓

center left right



text-decoration:

overline line-through underline none

~~MANISH~~

~~MAALISH~~

CANISH MANISH

text-transform:

uppercase

lowercase

capitalize

HI, HEELLO

hi, hello

Hi, Hello

~~strength~~



text-shadow: 2px 4px red;

(2px 4px)

shadow
color

(-4) Top

-2 Hi : 2 Right
left

(4)

Bottom



position of text

text shadow position - styling

Font: serif, sans-serif, monospace

background-color - gray

background, background-color, background-image

width, height

offset-top, offset-left

outline, outline-width

Size, position & lists - 6

There are more units for describing size other than 'px'.

There are → rem, em, vh, vw, %

Why not pixels?

Not suitable for all kind of devices like we make website in 1080 / 1920 px, when we open website into 1080 / 2340 px (phone) it will look different, not as you make.

Relative lengths

There are units relative to the other length property

Let us about →
root, parent

HTML

body — Parent of div

div — Parent of p

p

Note → These units make website responsive

26

Date _____
Page 60

following are some of the most commonly used relative lengths

em → relative to the parent font size

PE it become

font-size: 3em; → 30px

3x of parent

rem → relative to the root font size

Note → root always a HTML tag

root → font-size = 3px

P → 11 11 - 4rem → 3x4
12px

vh → relative to 1% viewport height

your device width \geq 1000 px

device P_E width: 20

device P_E height: 20 vh;

P have $1000 \times \frac{20}{100} = 200$ px height.

vw → relative to 1% viewport width.

% → relative to parent's width / height

max-height & height for property
min-width

max-width : 20vw ; initial, result

max width must be 20vw
you can not increase 21vw \textcircled{X}
you can keep small 15vw

min-width : 50vw ; initial, result

minimum width must be 50vw

min / max : same as width
height

The position property

Notes

Used to manipulate the location of an element. Following are the possible values:

`position: static;`

The HTML elements are positioned static by default.

It is always positioned according to the normal flow of the page.

`position: relative;`

We can set now

`top: 4px;` the box is shifted

`right: 10px;` in a small

`bottom: 11px;` downward

`left: 12px;` in a small

Everything will move with you screen size.

Note 3) We can say in the position, where element move, display will hidden.

Position: fixed;

An element with position : fixed is positional relative to the viewport, which means it always stays in the same place even if the page is scrolled.

top, right, bottom and left allowed

Note - A fixed element does not leave a gap in the page where it would normally have been located.

position: absolute;

An element with position: absolute;

is position relative to the nearest positioned ancestor (instead of positional relative to the viewport, like fixed.)

position: sticky;

position based on the user's scroll position.

The sticky element sticks at the top of the page (top: 0px) when you reach its scroll position.

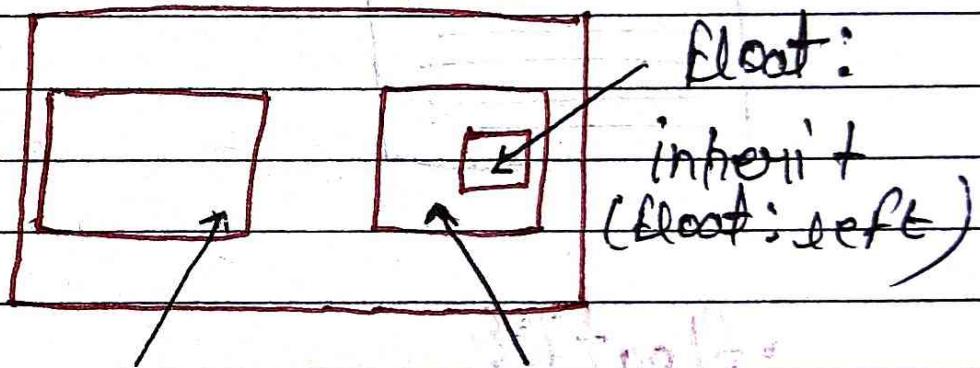
position: sticky;
top: 0px;

Flex box - 7

Before we look into the CSS flex box, we will look into float and clear properties.

The float Properties

The float property is used to positioning and layout on web pages.



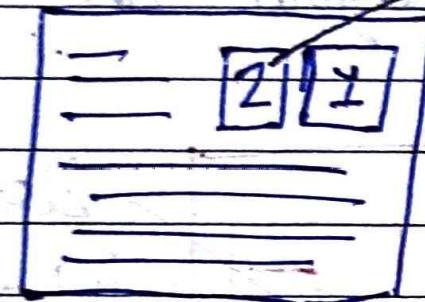
float: right float: left

The clear properties

The clear property specifies what elements can float beside the cleared element or

~~float~~~~float~~float: right; ~~float: left;~~~~float~~

float: right;

one ~~I~~ is floating my ~~right~~~~float~~

float: right;

~~float~~float: right;
clear: right;No one can float
my right

The Flex box

Aims at providing a better way to layout, align and distributes space among items in a container.

container ↳

`display: flex;`

Initialize a flex box

general size ↳

Flex box

cross size ↳

flex items

Main axis

Cross axis



main size ↳



height ↳

Flex-direction :

flow of items direction . which items are laid can be

`Flex-direction: row;` ← default

|| || : row-reverse ;

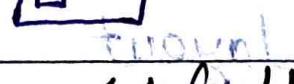
|| || : column ;

|| || : column-reverse ;

flex property for parent [flex container]

→ when child is overflowed - the scroll



flex-wrap: wrap - 

→ default
no-wrap

wrap-reverse - 

→ align-items

Flex-direction + flex-wrap

flex-direction

flex-wrap

flex-flow: dir. wrap;

flex-direction

flex-wrap

justify-content: flex-start



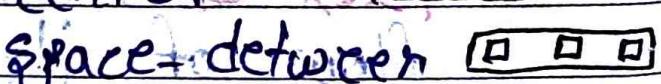
flex-end



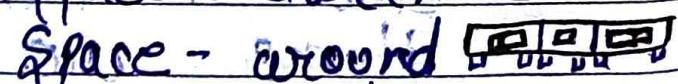
center



← initial, auto, space-between



← and final, space-around



space-evenly



flex-grow: 1 (grow): not growth reqd

flex-grow: 0



flex-grow: 1 (grow)

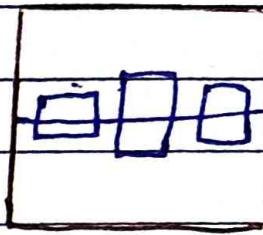
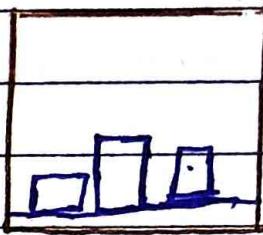
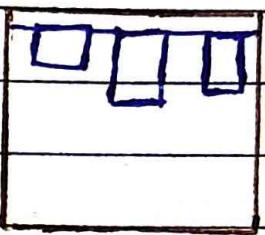


flex-grow: 1 (grow)

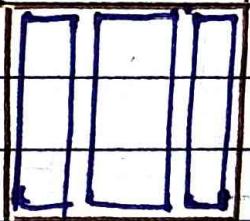


align-items:

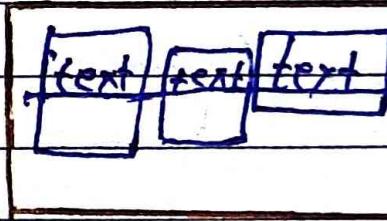
flex-start flex-end center



stretch



baseline



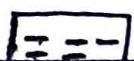
align-content : flex-start ;



flex-end ;



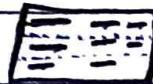
center ;



stretch ;



space-between ;



space-around ;



gap : 3px ;

row-gap : 4px ;

column-gap : 8px ;

Properties for the Children [Flex Items]

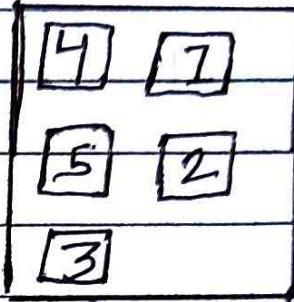
Order : +4; ¹

2; ²

3; ³

-4; ⁴

0; ⁵ default



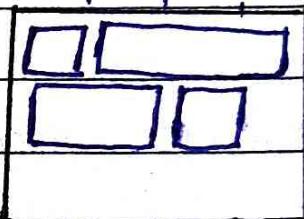
2, 4, 3, 1, 5

Flex-grow : 1;

3;

2;

1;



default

Flex-shrink : 4; default

Flex-basis : auto; default

Align-self : auto;

: flex-start;

flex-end;

center;

baseline;

CSS Grid & Media Queries

CSS Grid is used to make tabular like web page layout

ACSS grid can be initialized using

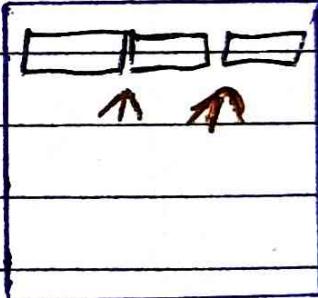
~~• containers~~

display: grid;

All direct children automatically become columns of a CSS grid.

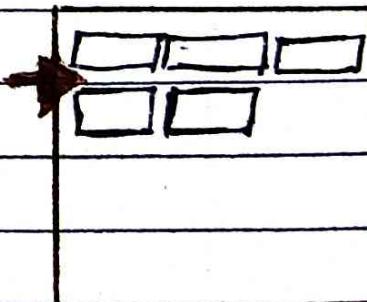
`grid-column-gap: 4px;` ↓↓ columns

Used to adjust the space b/w the column of a CSS grid



grid-gap: 10px;

adjust the space b/w grid-items



grid-gap:

Shorthand property for grid-row-gap and grid-column-gap

contained

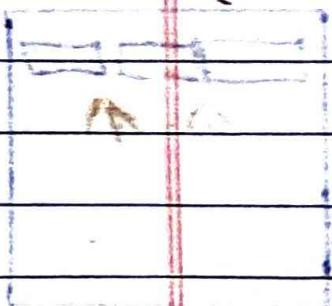
display: grid;

grid-gap: 10px 23px;

row

column

for a single value of grid-gap, both row and column gaps can be set in single : value



grid-gap: 10px; gap between all the grid

grid-gap: 20px; gap between all the grid

grid-gap: 20px;

grid-gap: 10px 20px;



grid-gap: 10px 20px;

grid-gap: 20px 10px;

grid-gap: 20px 10px;

Following are the properties for grid container:

grid-template-columns:

2px 5px auto;

To specifies width of column and number of columns.

grid-template-rows: 2px auto;

height of row and number of rows

justify-content ← whole

align-content ← vertically

Following are the properties for grid item:

grid-column: 1 / span 2;

How many columns spanned by a ~~column~~ item

grid-row: 3 / 6;

grid-row-start: grid-row-end:

CSS Media Queries

Media queries used to make a responsive website, when a certain width reach a extra css will launch. that css written in Media queries.

max-width: 800px

@media only screen and (↓) {
body {
background-color: red;
} }

When size of display reach 800px
body css will be apply.

Transforms, Transitions & Animations

Transforms are used to rotate, move, skew, or scale elements. They are used to create a 3-D effect.

transform : for 2D (x, y)

transform-origin : for 3D (x, y, z)

↓ position of transformation

transform: translate(x-axis 5px, y-axis 7px)

rotate(20deg)

scaleX(10)

scaleY(5)

skew(.10deg 20deg)

angle in x

angle in y

matrix();

scaleX(), skewX(), skewX(), scaleY(),

translateX(), translateY()

all properties (methods at a sometime)

scale;

CSS 3D transform methods

are transform: rotateX()
rotateY()
rotateZ()

CSS Transitions!

Used to change property values smoothly, over a given duration.

transition-property: width;
color;
opacity;

transition will apply on these properties

transition-duration: 6s;

(With timing functions)
from one state to another, it takes 6s.

transition-timing-function: ease

linear

ease-in

ease-out

ease-in-out

transition-delay: 1s;

set the delay before the transition start.

Shorthand property for transition.

transition: width 4s ease-in 2s;

property duration timing fun. delay

Transition multiple properties

transition: width 3s ease, height 2s ease;

using commas

CSS Animations

Used to animate CSS properties with more control. We can use **@keyframes** to change the animation from a given style to a new style.

@Keyframes manish {

 from { width: 20px; }

 to { width: 40px; }

} *Important : Every class has its own keyframes*

Properties to add animations

animation-name: Manish;

animation-duration: 9s

:initial

:inherit

animation-time-function: linear

: ease

: ease-in

: ease-out

: ease-in-out

iteration

forw - forward

backw - backward

animation-delay: It specifies the time

interval between the initial state and the start of the animation.

initial, inherit, 500s

animation-iteration-count: 4

infinite

initial

inherit

animation-direction: normal

reverse

alternate, forw.+backw

alternate-reverse

backw + forw

all, alternate, normal

initial

Shorthand

duration

delay

a break

animation: manish 4s ease-in 3s alternate

2s ease-in

Name

timefunction

it-count

0.5s - 3s ease

from 0.5s

Animation with percentage values

We can use % values to indicate what should happen when a certain percent of animation-duration is completed.

like animation-duration = 10s

① Keyframes manish {
0% {

width: 20px;

}

50% {

width: 80px;

{

100% {

width: 200px;

{

3

0s → 20px width

5s → 80px width

10s → 200px width