DOMS | Page No.
Date / /

CSS

4 Cascading Style Sheets

DOM

It stands for document object model. When a page is loaded, the browser creates a DOM of 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 other hand, when an HTML element is given a class, it now belongs to that class. More than one element can belong to a single class but every element must have a unique id.

We can add multiple classes to an element like this

```
<div id = "first" class = "c1 c2 c3" >...</div>
```

unique id — Multiple class followed by space

Three ways to add CSS to HTML

i) <style> tag : Adding <style> .... </style> to HTML.

ii) Inline CSS : Adding CSS using style attribute.

iii) External CSS : Adding a stylesheet (.css) to HTML using <link> tag.

* CSS selector

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

```
body {
    color: red
    background-color: pink }
```

i) Element selector

It is used to select an element based off the tag name

For Eg.
```
h2 {
    color: Blue; }
```

ii) id selector

It is used to select an element with a given id.

For Eg. `#first { color: White; }`

use to target by id.

iii) Class selector

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

For Eg. .red { background: red; }

↑ use dot to target class

Points to remember

⇒ We can group selectors like this

h1, h2, h3, div { color: blue }

Result: h1, h2, h3 & div will be blue

⇒ We can use element class as a selector like this:

p.red { color: red; }

Result: All paragraph of class red will get color red.

⇒ '*' It is used as universal selector

* { margin: 0;
padding: 0;
box sizing: border box; }

⇒ An inline style override external & internal style.

Colors and Background

1. Color property: Used to set the text color inside an element.

Types of Color value

1. RGB → rgba(200, 98, 70)
2. HEX Code → Specify color using hexcode
   Eg. #ff7180
3) HSL => Hue, Saturation, Lightness.
   Specify color using HSL values.
   Eg. hsl(8, 90%, 63%)

2. Background color → set background color of the container.

3. Background image
   ↳ Set an image as a background of container.

```
body { background-image: url("....jpg")
}
```

- Image (By default) is repeated in x and y direction.

4. Background-repeat property
- Repeat-x
- Repeat-y
- No Repeat
* More value from MDN.

5. Background-size property
⇒ Cover: fits and no empty space remain.
⇒ Contain: fits and image is fully visible.
⇒ auto ⇒ Display original size.
⇒ {{width}} ⇒ set width & height is set automatically.
⇒ {{width}} {{Height}} ⇒ set height & width

6.) Background-position property
Sets the starting position of a background image.
div { background-position: left Top; }

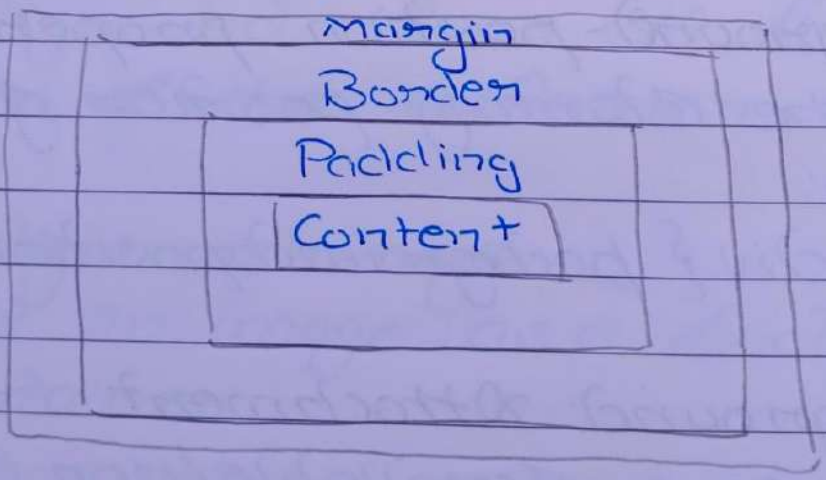
7) Background Attachment
↳ Defines a scrollable/non-scrollable character of background image
.div { background-attachment: Fixed; }
- Fixed / Local / Scroll ...

8) Background Shorthand
A single property to set multiple background properties.

color img.
div{ background: red url('img.png')
no-repeat fixed right top
repeat Attach position

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

Chapter-3 CSS Box model
The CSS box model looks at all the HTML element as boxes.



1. Setting Width & Height

```
#box {
    height: 70px;
    width: 70px; }
```

DOMS | Page No.
Date / /

- Total width/height is calculated as :-
  Total W/H = height + top/bottom padding + top/bottom borders + top/bottom margin.

2. Setting Margin & Padding

- .box{ margin: 3px;
  padding: 4px; }

⇒ It sets top, bottom, left & right values.

- .boxmain{ Top Right Bottom Left
  margin: 3px 7px 2px 11px

* top
  left ○ Right
  Bottom

or

Top & Bottom Left & Right

.box{ margin: 7px 3px }

⇒ We can set individual margin like this:
.box{margin-top: 70px}. . . .

* These all rules same goes with padding.

3) Setting Border
We can set border as:

```
.box{ border width: 2px;
      border style: solid;
      border color: red; }
```

* Shorthand

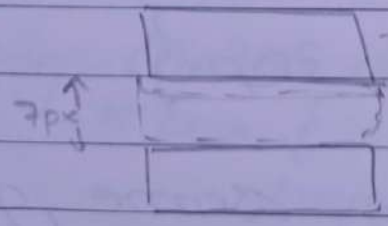
```
.box{border: 2px solid red}
```

(2 → width, style → solid, color → red)

4. Border Radius
We can set border radius to create rounded border

```
.div{
    border radius: 7px
    }
```

* Margin collapse
When two margin from different element overlap, the equivalent margin is greater of two. This is called margin collapse.



* 7px margin will be applied

- Margin b/w them is collapsed to bigger margin

Box sizing
Determines what out of padding & border is included in elements width and height.

- Can be content box or border-box
  - ↳ Include only content in width & height

```
div{
    box-sizing: border-box
}
```

The content width & height includes
⇒ Content + padding + border

Chapter - 4 Fonts & display

* The display property
The CSS display property is used to determine whether an element is treated as a block/inline element & the layout used for its children
Flexbox/grid/etc.

1. Display inline
Takes only the space required by element. No linebreak before &

often. Setting width/height not allowed (on margin/padding)

2) Display: block
Takes full space available in width and leaves a new line before & after the element.

3. Display: inline-block
Similar to inline but setting height, width, margin and padding is allowed. Element can sit next to eachother.

4. Display: none vs Visibility: hidden
With 'display: none', the element is removed from document flow. Its space is not blocked.
With 'visibility: hidden', the element is hidden but its space is reserved.

5. Text-align property
Used to set horizontal alignment of a text.

```
div{ text-align: center};
```

6.) Text decoration
- Used to decorate the text
- Can be overline, line-through, none, Underline.

7) Text transform
Used to specify uppercase and lowercase letter in a text.

p.uppercase { text-transform: uppercase; }

(p. → para, uppercase → class)

8. Line height
Used to specify the space between lines.

.line { line height: 0.7; }

Font
Font plays a very important role in the look and feel of a website.

i) font-family
- It specifies the font of a text
- Can hold multiple values as a 'fallback' system.

p{ font family: "Times new", monospace;

⇒ Web safe font
These fonts are universally installed across browsers.

How to add Google fonts
In order to use custom google font go to google font the select a style and finally paste it to the style.css of your page.

Other font properties
- font size - set size of font
- font style → set style
- font variant → set if text is display in caps
- font weight → set the weight of font

Generic families
Broad class of similar font Eg. Serif
Just like when we say fruit, it can be any fruit, similar if we say serif it mean any serif font
font family → specific
Generic family → Generic

DOMS | Page No.
Date / /

Chapter-5

There are more units for discussing size other than px. i.e rem, em, vw, vh, % etc.

Whats wrong with px?

- px are relative to the viewing device. For a device with size 1920×1080, 1px is 1 unit out of 1080/1920.

Relative length

1. em : Unit relative to parent font size.

ii) rem : Unit relative to root font size (<Html> tag)

iii) Vw : Unit relative to 1% viewport width

iv) Vh → Unit relative to 1% viewport height

DOMS | Page No.
Date / /

5.) % → Unit relative to parent element.

min/max-height/width property
CSS has this property, if the content is smaller than minimum height, min height will be applied.

* Position property
Used to manipulate the location of an element, following are the possible values.

↳ Static: The default position. Top/bottom/left/right/z-index has no effect.

↳ Relative: The top/bottom/left/right/z-index will now work. Otherwise the element is in flow of document like static.

↳ Absolute: The element is removed from the flow & is relatively positioned to its first non-static ancestor. top/bottom etc work.

DOMS | Page No.
Date / /

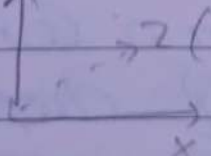
- Fixed: Just like absolute except the element is positioned relative to browser window. (Don't move while scroll)

- Sticky: The element is positioned based on user's scroll position.

List style property
The list style property is a short hand for type, position & image.

```
ul{
    list-style: square inside url( );
}
```
(style type, position, img.)

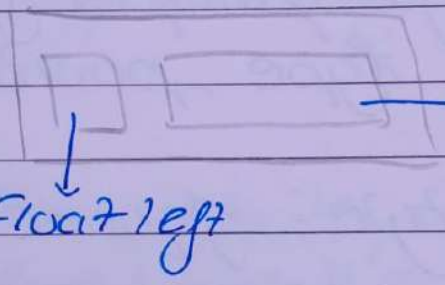
Z-index property
It specifies the stack order of an element.
It define which layer will be above which in case of overlapping elements.



Chapter-6 : Flexbox

Before we look into CSS flexbox, we will look into float & clear property.

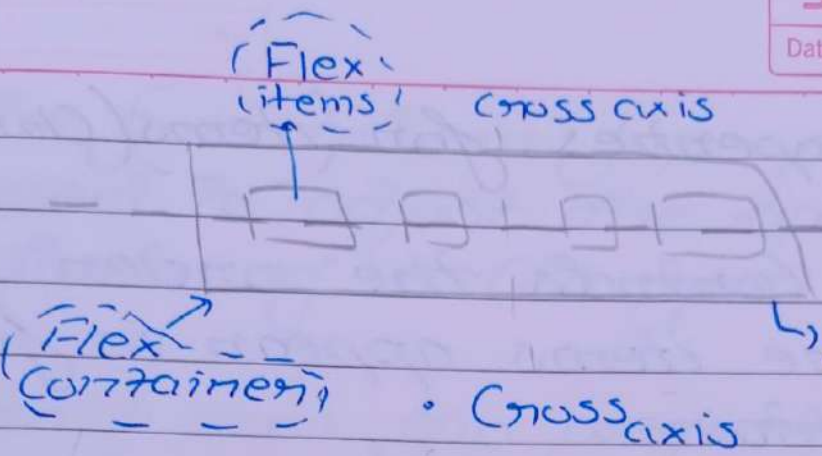
The float property
It just follows the element towards left/right.



Clear property
- Used to clear the float. It specifies what elements can float beside a given element.

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

```
Container { display: flex; }
```
↳ Initialise a flexbox



Flex direction property
Defines the direction towards which items are laid can be row, row-reverse, column, column reverse → default

Flex properties for parent (container):

i) Flex-wrap: Can be wrap, nowrap, wrap-reverse. Wraps items as needed with this property.

ii) Justify content: Define alignment along main axis (Horizontally)

iii) Align-item: Define alignment along cross axis (Vertically)

iv) Align-content: Aligns flex container's lines when there is extra space in cross axis.

Flex properties for items (children)

i) Order: Controls the order in which the items appear in flex container.

ii) Align-self: Allows default alignment to be overridden for the individual flex items.

iii) Flex-grow: Defines the ability for a flex item to grow.

iv) Flex-shrink: Specifies how much a flex item will shrink relative to the rest of flex items. (Use while overflowing)

Chapter-7 CSS Grid & Media Queries

A css grid can be initialized using:

```
Container {
    display: grid; }
```

All direct children automatically become grid items.

1. The grid-column gap property used to adjust the space between columns of a CSS grid.

2. Grid-row gap property used to adjust the space between the rows of a CSS grid.

3. Grid-gap property.
Shorthand property for grid-row-gap & grid-column-gap.

```
Container{ display: grid;
           grid-gap: 40px 100px;
         }
```

Note: For a single value of grid-gap, both row and column gaps can be set in one value.

Properties for grid Container

i) The grid-template-columns property can be used to specify the width of column

```
.container{ display grid
        grid-template-columns: 80px
                               120px auto;
}
```

ii) The grid template rows property can be used to specify the height of each row.

```
.container{ display: grid;
        grid-template-row: 70px 150px;
}
```

1st row, 2nd row

iii) The Justify content property is used to align the whole grid inside the container.

iv) The align-content property is used to vertically align the whole grid inside container.

Properties for grid items

i) The Grid-column property defines how many column an item will span.

```
.grid-item{ grid-column: 1/5;
}
```

ii) The Grid row property defines how many rows an item will span.

iii) We can make an item to start on column 1 & span 3 columns like this:

```
.items { grid-column: 1/span 3; }
```

CSS Media Queries
Used to apply CSS only when certain condition are true.
Syntax:

```
@media only screen and(max-width: 800px)
    body{ background: red; }
}
```

Chapter-8 Transforms, Transition & Animation

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

i) Transform property
Used to apply a 2D or 3D transform to an element.

ii) Transform-origin property
Allows to change the position of transformed elements.
2D transform: Change X-Y axis
3D transform: Can change Z axis as well.

CSS transform (2D) method

1.) Translate()
ii) rotate()
iii) Scale()
iv) ScaleY()
v.) Skew()
vi) matrix()
vii) ScaleX()

CSS 3D transform method

1. rotateX()
2. rotateY()
3. rotateZ()

DOMS | Page No.
Date / /

CSS Transition
used to change property values smoothly, over a given duration.

Transition property
↳ used to add transition
* Properties

i) Transition-property - The property you want to transition

ii) Transition duration - Time for which you want transition to apply.

iii) Transition-timing-function → How you want the property to transition

iv) Transition delay → Specifies the delay for transition

Shorthand
Transition: width 3s ease-in 2s;
(property ↓, duration ↑, timing function ↓, delay ↓)

Transitioning multiple properties

- Transition: opacity 1s ease-out 1s, transform 2s ease-in;
  ↳ we can skip transition delay here

CSS Animation

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

Name

```
@keyframes hanny{
    from{width: 20px;}
    to{width: 31px px;}}
```

- Can change Multiple properties

Properties to add animation

i) Animation name: Name of animation

ii) animation duration: How long does the animation run?

iii) animation-timing function → Determine speed curve of animation

iv) animation-delay: Delay for the start of an animation

v) animation-iteration-count: No. of times an animation should run.

vi) animation-direction: Specifies the direction of animation.

Shorthand)

Animation: Harry 6s linear 1s infinite reverse;
(1 2 3 4 5 → 6.)

Using % value states with animation
We can use % values to indicate what should happen when a certain % of animation is complete.

```
@keyframe harry {
    0% { width: 20px }
    50% { width: 80px }
    100% { width: 100px; }
}
```

⇒ Can add as many intermediate properties as possible.

3-Jun-21

CSS Cheat Sheet

"You will not become a Programmer if You won't Practice."

Background Properties

1. Background : Background-image
Background-position
Background-size
Background-repeat
Background-attachment
Background-origin
Background-clip
Background-color

Values

2. Background-attachment : Scroll, Local, Fixed

3. Background-clip : Border-box, padding-box, Content-Box, initial, inherit

4) Background-image : Url, None

5) Background-color : Color (Select/Write Name), Transparent

6.) Background origin: Border box, Padding box, Content box

7.) Background position: top left, top center, top right, center left, center, bottom left.

8.) Background repeat: No repeat, repeat x / y.

9. Background-size → Auto, Center, Contain, (Height/Wid can be set)

Border and Box

i) Border-width → Select value. Eg. 12vh...
ii) Border-color → Select color
iii) Border-style → Solid, dashed, dotted, double, groove, ridge.
iv) Border radius: Select value Eg. 3em, 12px...
v.) Opacity: Any no. b/w 0 & 1
↳ Make whole box semi-transparent
vi) Box shadow - h-offset, v-offset,
↳ horizontal
blur, spread, color.

Table

i) Border-collapse - Collapse, separate
ii) Border-spacing - set length
iii) Caption side - Top, Bottom, Left, Right
iv) Empty cells - show hide

Flex

For Container

i) Display: flex, inline-flex

ii) Flex direction: row, row-reverse, column, column reverse

iii) Flex-Wrap: wrap, wrap-reverse

iv) Justify content: Flex-start, Flex-end, center, space between, space-around, space evenly.

v) Align item: Flex-start, Flex-end, center, baseline, stretch

vi) Align Content: Flex-start, Flex-end, Center, Space-between, Stretch

For Children

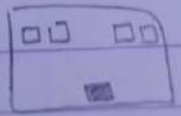
i) Order: (Set order in which they appear)
↳ Value: integer

ii) Flex-grow → 1 (applied to all) or (1, 2 and 3)

iii) Flex-Basis: First item 20%, second item 40%.

iv) Flex-Shrink: 2 (Number), unset, revert, initial

v) Align self:



3rd item has align self: flex-end.

CSS Grid

For Container

i) Display : Grid , Inline Grid, Subgrid

ii) Grid-template-columns : Value Eg. 12px 12px;

iii) Grid-template-rows : Values like - 8px 8px;
or repeat(3, 12px)
or 8px auto 12px etc.

iv) Grid Gap : 1px 9px ;
↳ For row → For column

v) Justify-items : Start, end, center, Stretch

vi) Align-items : Start , End , Center, Stretch.

vii) Justify Content : Start... , Space-around, Space-between etc.

viii) Align Content : Start, Center . . . etc.

ix) * Grid-auto-flow : Row, Column, Dense.

For Children

i) Grid Row : 1 / span 3
Start ↳ End (or 3)

ii) Grid-Column : 1 / span 2
Start End

iii) Justify Self : Start , End , Center

iv) Align Self : Start , End , Center

Transition

i) Transition-property : Width, Height, all.
ii) Transition-Duration : Eg. 2s
iii) Transition-Delay : 1s ;
iv) Transition-timing-function: Linear, Ease, Ease-in, Ease-out, Ease-in-out.

Animation

i) Animation-name : Anything
ii) Animation-duration : 25s
iii) Animation-iteration-count : 1... infinite ;
iv) Animation-play-state : Paused, running, initial, inherit.
v) Animation-delay : 0s ..... 10s ... etc.
vi) Animation-direction : Normal
vii) Animation-Fill-mode : Both
viii) Animation-timing-function : Ease in, Ease-out, Ease-in-out, Linear.

You don't need to memorise them, You will automatically do that by practicing.