CSS

Chapter - 0
Introduction to CSS

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
Create a CSS file inside your directory and add it to your HTML. Add the following line to your CSS.

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

This will make your page background as red.

Chapter - 1
Creating our first CSS website

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 and 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. More than one elements can belong to a single class but every element must have a unique id (if assigned).

We can add multiple classes to an element like this.

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

Unique Id

↳ multiple classes followed by spaces

Three ways to add CSS to HTML

1) <Style> tag → Adding <style> ... </style> to HTML
2) Inline CSS → Adding CSS using style attribute
3) External CSS → Adding a stylesheet (.css) to HTML using <link> tag.

CSS selectors

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

Selector

```
body {
    color : red ;-> declaration (Property : Value)
    background-color : pink ;-> declaration
}
```

Element Selector

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

For example :

```
h2 {
    Color : blue ;
}
```

Id selector

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

For example :

```
#first { -> # is used to target by id
    color : red;
}
```

Class selector

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

For example :

```
.red { -> . is used to target by class
    background-color : red;
}
```

Decendent Selector	Chain Selector
div P { . . . }	Selector Selector { . . . }
Parent child	used to select very very
Selects all P inside div	specific element.

Important Notes :

→ We can group selectors like this :

```
h1, h2, h3, div {
    Color : blue → h1, h2, h3, div will be grouped
}
```

→ We can use element.class as a selector like this :

```
p.red {
    color : red ; → all paragraphs of will get color of red
}
```

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

```
* {
    margin : 0 ;
    padding : 0 ;
}
```

Parent child direct descendant selector

Selector > Selector can be used to select the direct child of the given element

```
<div>
  └> <div>
```

This select this

→ An inline style will override external and internal CSS.

Comments in CSS

Comments in CSS is a text which is not parse and is thus ignored.

Syntax :-

```
/* This is comment */
```

Chapter - 2
Colors & Backgrounds

CSS rules are simply key-value pairs with a selector. We can write CSS rules to change color and set background.

The color Property

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

```
p {
    color : red ;
}
```
→ Text color will be changed to red

Similary we can set color for different elements.

Types of Color value

Following are the most commonly used color values in CSS.

1> RGB → Specifies color using Red, Green, Blue values.
Eg. rgb(200, 98, 75)

2> Hex code → Specifies color using Hex code.
Eg. #ff7180

3> HSL → Specify the color using hsl values
Eg. hsl(8, 90%, 63%) ↳ hue, saturation, lightness

The value of the color or background color is provided as any one of these values.

The background-color property

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

for eg.

```
.brown {
    background-color: brown;
}
```

The background-image property

Used to set an image as the background.

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

The image is by default repeated in x & y directions

The background-repeat property

Can be any of:

- repeat-x → repeat in horizontal direction
- repeat-y → repeat in Vertical direction
- no-repeat → image not repeat

See more Possible values at MDN docs.

The background-size Property

Can be following:

- Cover → fits & no empty space remains
- Contain → fits & image is fully visible
- auto → Display in original size.

-> {{width}} -> Set width & height will be set automatically
-> {{width}} {{height}} -> Set width & height

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

```
.div1 {
    background-position: left-top;
}
```

The background-attachment property
Defines a scrollable/non-scrollable character of a background image.

```
.div2 {
    background-attachment: fixed;
}
```

The background shorthand
A single property to set multiple background properties

```
.div3 {
    background: red url("...jpg") no-repeat fixed right-top;
}
```
color ↓ image ↓ repeat ↓ attachment ↓ position

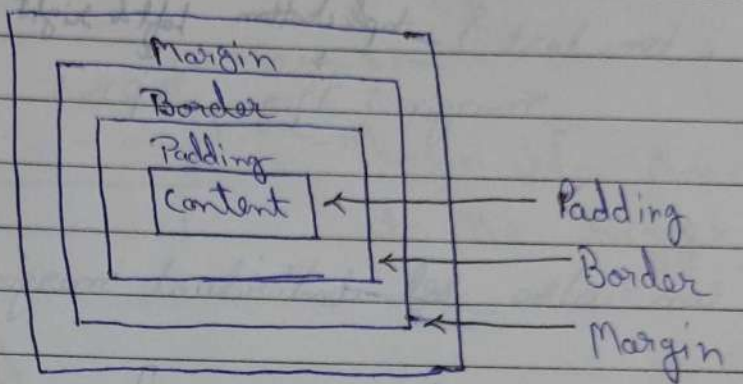
One of the properties can be missing given the others are in order.

Pages
Date

Chapter - 3

CSS Box Model

The CSS box model looks at all the HTML elements as box



Setting width & height

We can set width and height in CSS as follows :-

```
#box {
    height : 70Px ;
    width : 80Px ;
}
```

Note that the total width/height is calculated as follows :-

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

Setting Margin & Padding

We can set margin & padding as follow :-

```
.box {
    margin : 3Px ;
    padding : 4Px ;
}
```

} set top, bottom, left & right values

3

```
.boxmain {          Top   right   bottom   left
    margin : 7Px    0px   2 px    11 px ;
}

.boxlast { top& bottom   left& right
    margin : 7px    3px ;
}
```

We can also set individual margins / padding like this:

```
margin-top : 70 Px ;
margin-bottom : 10 Px ;
margin-left : 8 Px ;
margin-right : 9 px ;
```
} Same goes with padding

Setting border

We can set the border as follows

```
.bx {
    border-width : 2px ;
    border-style : solid ;
    border-color : red ;
}
```
} Or just set, border : 2 Px solid red ;

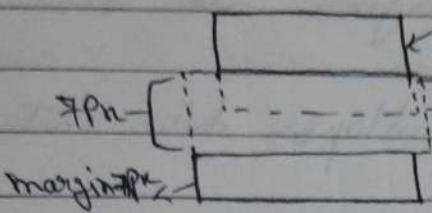
Border Radius

We can set border radius to create rounded borders

```
.div2 {
    border-radius : 7 Px ;
}
```

Margin Collapse

When two margins from different elements overlap, the equivalent margin is the greater of the two. This is called margin collapse.



→ Margin b/w them is collapsed to the bigger margin

Box Sizing

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

Can be Content - box or border - box

↳ Include only content in width / height

```
.div1 {
    box-sizing: border-box;
}
```

↳ The content width and 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

display : inline
Takes only the space required by the element. No linebreaks before and after. Setting width / height / margin / padding is not allowed.

display : block
Takes full space available in width and leaves a newline before and after the element.

display : inline-block
Similar to inline but setting height width, margin and padding is allowed. Elements can sit next to each other.

display : none vs visibility : hidden

With display : none; the element is removed from the document flow. Its space is not blocked.

With display visibility : hidden, the element is hidden but its space is reserved.

text-align property
Used to set the horizontal alignment of a text

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

text-decoration property
Used to decorate the text.
Can be overline, line-through, underline, none

text-Transform property
Used to specify uppercase and lowercase letters in a text.

```
p.uppercase {
    text-transform: uppercase;
}
```

line-height property
Used to specify the space between lines.

```
.small {
    line-height: 0.7;
}
```

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

font - family

font family specifies the font of a text.
Can hold multiple values as "a "fallback" system.

```
p {
    font-family: "Times new Roman", monospace;
}
```

↓

Always do this to ensure the correct font of your choice is rendered

Web Safe fonts

These fonts are universally installed across browsers

How to add google fonts

In order to use custom google fonts, go to google fonts then select a style and finally paste it to the style.css of your page.

Other font properties

Some of the other font properties are listed below:

font-size → sets size of the font
font-style → sets the font style
font-variant → sets whether text is displayed in small-caps.
font-weight → sets the weight of the font.

Date

Chapter - 5
Size, position & lists

There are more units for describing size other the 'px'. There are rem, em, vw, vh, percentages etc.

Whats wrong with pixels?
Pixels are relative to the viewing device. For a device with size 1920 x 1080, 1px is 1 unit out of 1080 / 1920.

Relative lengths
These units are relative to the other length property. Following are some of the most commonly used relative lengths.

1) em -> Unit relative to the parents font size.

2) rem -> Unit relative to the root font size (<HTML>)

3) Vw -> Unit relative to 1% viewport width.

4) Vh -> Unit relative to 1% viewport height.

5) % -> Unit relative to the parent element.

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

Similar is the case with other related properties.

The 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 work now. Otherwise the element is in the flow of document like static.

↪ absolute: The element is removed from the flow & is absolutely positioned to its first non-static ancestor. top / bottom etc. works.

↪ fixed: Just like absolute except the element is positioned relative to the browser window.

↪ sticky: The element is positioned based on user's scroll position

list-style property
The list style property is a shorthand for type, position & image.

```
ul {
    list-style: square inside url('ire.jpg')
}
```

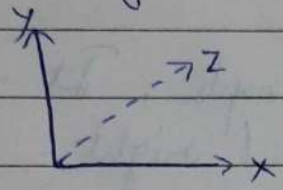
list-style-type
list-style-Position
list-style-image

Pages
Date

Z-index Property

The z-index property specifies the stack order of an element.

It defines which layer will be above which in case of overlapping elements



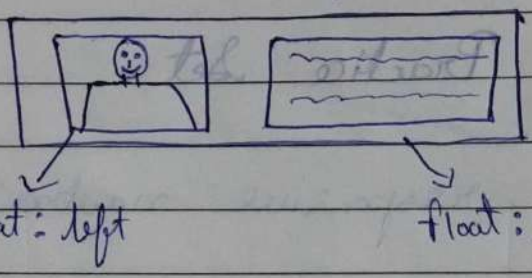
⇒ Z is the third dimension

Chapter - 6
Flexbox

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

The float property

Float property is simple. It just flows the element towards left / right.



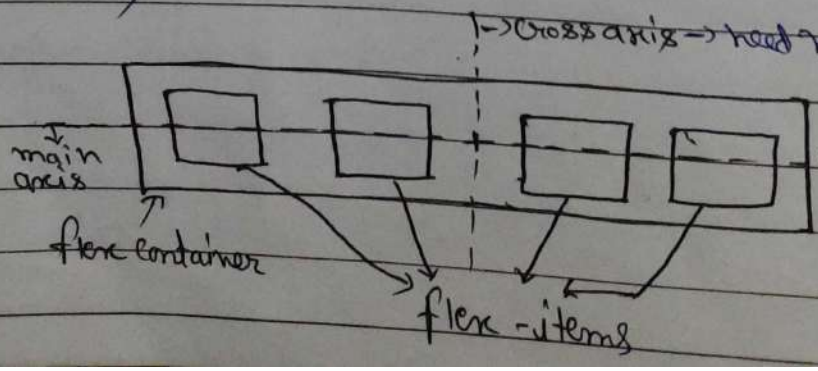
The clear property

used to clear the float. It specifies what elements can float beside a given element.

The CSS Flexbox

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

```
.container {
    display: flex ; => Initialize a flexbox
}
```



flex - direction property
Defines the direction forwards which items are laid
Can be row, row-reverse, column, column-reverse.
↓
default

Flex properties for parent (flex container)
Following are the properties for the flex parent:

1. flex-wrap: Can be wrap, nowrap, wrap-reverse. Wrap items as needed with this property.
2. justify-content: Defines alignment along main axis.
3. align-items: Defines alignment along cross axis.
4. align-content: Aligns a flex container's lines when there is extra space in the cross axis.

Flex properties for the children (flex items)
Following are the properties for the flex children.

1. Order: Controls the order in which the items appear in the flex container.
2. align-self: Allows default alignment to be overridden for the individual flex items.
3. flex-grow: Define the ability for a flex item to grow.

Chapter - 7
CSS Grid & Media Queries

A CSS grid can be initialized using :

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

All direct children automatically becomes grid items

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

The grid - row - gap property
Used to adjust the space between the rows of a CSS grid.

The grid - gap property
Sorthand property for grid - row - gap & grid-column-gap.

```
.Container {
    display : grid;
    grid-gap : 40px 100px;
}
```
(40px → row, 100px → column)

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

Following are the properties for grid container.

1. The grid-template-columns property can be used to specify the width of columns.

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

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

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

3. The justify-content property is used to align the whole grid inside the container.

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

Following are the properties for grid item:

1. The grid-column property defines how many columns an item will span.

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

2. The grid-row property defines how many rows an item will span.

3. We can make an item to start on column 1 and span 3 column like this:

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

CSS Media Queries

Used to apply CSS only when a certain condition is true.

Syntax:

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

Chapter - 8

Transforms, Transitions & Animations

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

The transform property
used to apply a 2D or 3D transformation to an elements. ~~They are used to create~~

The transform-origin property
Allows to change the position of tranformed elements
2D transforms → Can change X & Y axis
3D transforms → Can change Z axis as well

CSS 2D transform methods

You can use the following 2D transforms in CSS:

1) translate()
2) rotate()
3) scaleX()
4) scaleY()
5) skew()
6) matrix()
7) scale()

CSS 3D transform methods

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

CSS Transitions

Used to change property values smoothly, over a given duration

The transition property

The transition property is used to add transition in CSS.

Following are the properties used for CSS transition.

1. transition-property -> The property you want to transition
2. transition-duration -> Time for which you want transition to apply
3. transition-timing-function -> How you want the property to transition
4. transition-delay -> Specifies the delay for the transition

All these properties can be set using a single shorthand property.

transition: width 3s ease-in 2s;

- ① Property
- ② duration
- ③ timing function
- ④ delay

Transitioning multiple properties

We can transition multiple properties as follows;

transition: opacity 1s ease-out 1s, transform 2s ease-in;

yes you can skip transition delay here!

CSS Animations

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

```
@keyframes harry {
    from { width: 20px; } → Can change multiple property
    to { width: 31 px; }
}
```

Properties to add Animations

Following are the properties used to set animation in CSS:

1. animation-name → name of the animation
2. animation-duration → How long does the animation run?
3. animation-timing-function → Determines speed curve of animation
4. animation-delay → Delay for the start of an animation
5. animation-iteration-count → Number of times an animation should run.
6. animation-direction → Specifies the direction of the animation

The animation shorthand

All the animation properties from 1-6 can be applied like this:

```
animation: harry 6s linear 1s infinite reverse;
           (1)   (2)  (3)  (4)   (5)
```

Using percentage value states with animation

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

```
@ keyframes harry {
    0% {
        width: 20px;
    }
    50% {
        width: 80px;
    }
    100 % {
        width: 200px;
    }
}
```

⇒ Can add as many intermediate properties as possible.