CSS (Cascading Style Sheet)

* It is language that is used to describe the style of a document.
* Basic syntax:- h1 {
* Color : red;

}

* Level 1

# Including style

* There are two types of style

1. Inline:-

Syntax:- <h1 style= “color : red”>Apna college</h1>

1. <style> tag:-

Syntax:<style>

h1{

color:red;

}

</style>

* External stylesheet

Writing CSS in a separate document &linking it with HTML file.

For example:-

#style.css

h1{

    color:red;

}

#index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>CSS level 1</title>

    <link rel="stylesheet" href="style.css"> <!—-rel=relation-->

</head>

<body>

    <h1>this is my page</h1>

</body>

</html>

# Color property

* Used to set the color of foreground.
* For example:- text,element,link.

## Background color property

* Used to set the color background.
* For example: background-color:red;
* Color systems
* RGB
* Color:rgb(255,0,0) (rgb=r🡪red,b🡪blue,g🡪green)(0-255)
* Hex (hexadecimal)

Color:#ff0000; (red color)

# Selectors

* Universal selector:- \*{}
* Element selector:- h1{}
* Id selector-🡪unique name:- #myId{}
* Class selector🡪 Multiple element ko same id set karna hota tab use karte hai.

Syntax:- .myclass{}

# Text properties

* Text-align

Syntax:- text-align: left/right/center

* Text decoration:-

Syntax:- text-decoration: underline/overline/line-through/none

* Font-weight🡪 it show the text color such that text is light or dark.

Syntax:- font-weight:normal/bold/bolder/lighter

Font-weight:100-900

* Font-family🡪style

Syntax: font-family : arial;

## Units in css

### Absolute

Pixels(px)

96px= 1 inch;

Font-size:2px;

* Line-height:- it check the height of text.

For example:- line-height:2px.

Another example:- line-height:normal.

* Text-transform:-it change the case of content.

Syntax:- text-transform: uppercase/lowercase/capitalize/none.

* Level 2

# Box model in css

1. Height:- By default, it sets the content area height of the element.

For example:- div{

Height: 50px;

}

1. Width:- By default , it sets the content area width of the element.

For example:-

div{

width: 50px;

}

1. Border:- Used to set an element’s border.

For example:

Border-width: 2px;

Border-style: solid/dotted/dashed

Border-color : black;

* Border(shorthand):-

Syntax: - border: width style color;

For example:- border: 2px solid black;

* Border-radius:- used to round the corners of an element’s outh=er border edge.

For example:- border-radius: 10px;

Another example:- border-radius: 50%;

1. Padding:- it is space between content or border is known as padding.

* Padding-left:25px;
* Padding-right
* Padding-top
* Padding-bottom
* Padding(shorthand)

Padding: 50px;

Or

Padding: top | right | bottom | left->clockwise.

Padding: 1px 2px 3px 4px;

1. Margin:- Margins are used to create space around elements, outside of any defined borders.

* Margin-right
* Margin-left
* Margin-top
* Margin-bottom
* Margin(shorthand):- margin: 50px;

Margin: 1px 2px 3px 4px (top |right| bottom| left->clockwise)

# Display property:-

Display: inline/block/inline-block/none

* Inline:- Takes only the space required by the element.(no margin & padding)
* Block:- Takes full space available in width.
* Inline-block:- Similar to inline but we can set margin & padding.
* None:- To remove element from document flow.
* Visibility :- visibility : hidden;
* Note:- When visibility is set to none , space for the element is reserved.

But for display set to none, no space is reserved or blocked for the element.

* Alpha channel:- Opacity(0-not visible to 1-complete)

For example:- color: rgba(255,0,0,0.25) -🡪it show light red.

* Level 3

### Relative

* Percentage(%):- It is often used to define a size as relative to an element’s parent object.

for example:- Width:33%;

Margin-left:50%;

* Em:- Font size of the parent, in the case of typographical properties like font-size, and font-size of the element itself, in the case of other properties like width.
* Vh : releative to 1% viewport height.
* Vw: reative to 1% viewport width.
* Position:- The position css property sets how an element is positioned in a document.

Syntax:- position: static/relative/absolute/fixed.

1. Static – Default position(The top, right, bottom , left, and z-index properties have no effect).
2. Relative- Element is relative to itself.(The top, right, bottom, left, and z-index will work).
3. Absolute- Positioned relative to its closest positioned ancestor.(remove from flow)
4. Fixed – Positioned relative to browser.(removed from flow)
5. Sticky - Positioned based on user’s scroll position.

### Background image

* Used to set an image as background.

Syntax:- background-image: url(“image.jpeg);

* Background-size:cover/ contain /auto
* Level 4

### Flexbox

* Flexible-Box layout:- It is a one-dimensional layout for arranging items in rows or columns.
* Flexbox direction:- It sets how flex are placed in the flex container, along which axis and direction.
* Flex-direction :row;(default) 🡪( main axis =row🡪left to right)
* Flex-direction : row-reverse; 🡪( main axis =row🡪right to left)
* Flex-direction: column; 🡪( main axis = top to bottom)
* Flex-direction : column reverse; 🡪( main axis =top to bottom)

### Flexbox property(for used in flex container)

* Justify-content:- Alignment along the main axis.

Flex-start/flex-end/center/space-evenly

* Flex-wrap:- nowrap/wrap/wrap-browser
* Align-item: alignment along the cross axis.
* Align-content: alignment of space between & around the content.

(for used in flex item)

* Algin-self:- Alignment of individual along the cross axis.
* Flex-grow:- How much a flex item will grow relative to the rest of the flex items if space is available.
* Flex-shrink:- How much a flex item will shrink relative to the rest of the flex items if space is available.

# Media queries

* It help create a responsive website.(like mobile display, laptop display, tablet, ipad etc.
* For example:-
* @media(width:600px)

{

div{

background-color: red;

}

}

* @media(min-width:600px)

{

div{

background-color: red;

}

}

* @media(min-width:200px) and (max-width:300px)

{

div{

background-color: green;

}

}

@media(min-width:200px) and (max-width:300px){

}

* Level 5

# Transitions

* Transitions enable you to define the transition between two states of an element.
* Transition-property: Property you want to transition(font-size,width etc.)
* Transition-duration : 2s/4ms…
* Transition-timing-function: ease-in/ease-out/linear/steps..
* Transition-delay: 2s/4ms…
* Transition shorthand: property name| duration| timing-function| delay.

For example:- transition: font-size 2s ease-in-out 0.2s;

## CSS transform

* Used to apply 2D & 3D transformations to an element.
* Rotate

Transform: rotate(45deg);

* Scale: (2D)

Transform: scale(2);

Transform: scale(0.5);

Transform: scale(1,2);

* Translate:-

Transform: translate(20px);

Transform: translate(20px,50px);

* Skew:-

Transform: skew(30 deg);

## Animation

* To animate CSS elements.

@keyframe myName

{

From {font-size: 20px;}

To{font-size: 40px}

}

### Animation properties

* Animation-name
* Animation-duration
* Animation-timing-function
* Animation-delay
* Animation-iteration-count
* Animation-direction: normal(circle)| reverse(to-from) |alternate(to-form and from-to) | alternate-reverse
* Animation shorthand:

Syntax:- animation: name duration fx delay iteration duration

Animation: myName 2s linear 3s infinite normal.

### % in animation

@keyframe myName{

(From)🡪0%{font-size: 20px;}

50%{font-size: 30px;}

(to)🡪 100%{ font-size: 40px;}

* for example:-

@keyframe colorAnimate

{

0%{

Left: 0px;

Background-color: yellow;

}

50%{

Background-color: blue;

}

100%{

Left: 200px;

Background-color: red;

Color: white;

}

}