CSS

-->css stands for cascading style sheets
-->its a language to make our web-page presentable
-->designed to make style sheets for webpage

-->ACRONMY:
cascading :- falling of style
styles :- adding designs/styling our html tags
sheets :- writing our style in different documents

HISTORY OF CSS

--->1994 first proposed by HAKON WIUM LIE on 10TH october
--->1996 css was published on 17th november with influencer BERT BOS.
--->later he became co-author of css.
--->1996 css became offical with css was published in december

---

TYPES OF CSS OR DIFFERENT WAYS OF ADDING CSS

*INLINE STYLE
*INTERNAL STYLE
*EXTERNAL STYLE

1)INLINE STYLE:-

-->inline css is used to apply css on single line or element
-->before css this way the only way to apply styles
-->not an efficient way to write as it has lot of redundancy

eg:- hello

2)INTERNAL STYLE:-

-->internal css is used to apply css on a single document or page.
it can effect all the element of the page.
-->it is written inside the style tag within head section of html.

eg:-
```
<head>
  <style>
   p{
       color : blue;
     }
  </style>
  </head>
```

3)EXTERNAL STYLE:

-->External CSS contains separate CSS file which contains only style property with the help of tag attributes.
(For example class, id, heading, … etc).

-->CSS property written in a separate file with .
css extension and should be linked to the HTML document
using link tag.

eg:-
```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="style1.css"/>
</head>

<body>
   <h1 id="demo" > HELLO </h1>
</body>
</html>
```

Example: The file given below contains CSS property.
This file save with .css extension. For Ex: style1.css

```
#demo{
      color : "yellow";
      }
```

*PRIORITY ORDER:
INLINE->INTERNAL->EXTERNAL

---

2)COMBINATORS SELECTOR:-

A combinator is something that explains the relationship/combination between the selectors.

There are four different combinators in CSS:

descendant selector (space)
child selector (>)
adjacent sibling selector (+)
general sibling selector (~)

1.descendant selector (space)

-->it will take all the direct child and in-direct child

eg:
```
<div>
<h1>first child </h1>
<h1>secod child </h1>
 <span><h1>third child</h1></span>
</div>

 div h1{
     background-color : "yellow";
```

```
}
```

2.child selector (>)

-->it will take only direct child

eg:
```
<div>
  <h1>first child </h1>
  <h1>secod child </h1>
   <span><h1>third child</h1></span>
  </div>

   div > h1{
       background-color : "yellow";
         }
```

3.adjacent sibling selector (+)

-->it will select adjacent sibling or immediate sibling right after the first sibling.

eg:
```
<div>
    <h1>first child </h1>
    <h1>secod child </h1>
     <span><h1>third child</h1></span>
    </div>

    <div>
    <h1>first sibling </h1>
    <h1>second sibling </h1>
    </div>
     div + h1{
         background-color : "yellow";
           }
```

4.general sibling selector (~)

-->it will select all the siblings right after it.

eg:
```
<div>
  <h1>first child </h1>
  <h1>secod child </h1>
   <span><h1>third child</h1></span>
  </div>

  <div>
  <h1>first sibling </h1>
  <h1>second sibling </h1>
  </div>
   div ~ div h1 {
       background-color : "yellow";
         }
```

<----------->

\\PSEUDO ELEMENT\\

*pseudo element selector(::)

A CSS pseudo-element is a keyword added to a selector that lets you style a specific part of the selected elements.

Style the first letter or line of an element
Insert content before or after the content of element

Syntax:-

```
selector::pseudo-element{

    property:value;

}
```

1::first-line pseudo element :-

first-line is used to style only first-line of the content

```
 <p>

      Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit recusandae

      libero maxime voluptatibus saepe tempore, fugiat soluta ad sit fugit,

    </p>
```

```
   p::first-line {

  color: red;

  background-color: aquamarine;

}
```

2.::first-letter pseudo element:-

it is used to style only first-letter/characther in a content

```
     <p>

     Lorem ipsum dolor sit amet consectetur adipisicing elit.
     Velit recusandae
    </p>
   p::first-letter {

  color: red;

  background-color: aquamarine;

}
```

3.::before pseudo element:-

The ::before pseudo-element can be used to insert some content/img before the content of an element.

Syntax :

```
            selector::before {

                property:value;

            }
```

```
 <p>

      Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit recusandae

      libero maxime voluptatibus saepe tempore, fugiat soluta ad sit fugit,

    </p>
```

eg:-

```
p::before {

  content: url("https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTtZFJN1r
           G7233vqSCo-DF0xwsCo6MZaHohkg&usqp=CAU");

}
```

<--->

4.::after pseudo element :-

The ::after pseudo-element can be used to insert some content after the content of an element.

Syntax :

```
selector::after {

    property:value;

}
```

Marker :-

---->it is used to style only items in a list.

```
 <ul>

      <li>HTML</li>

      <li>CSS</li>

      <li>JS</li>

    </ul>


li::marker {

  color: blueviolet;

}
```

```
<ul>

      <li>HTML</li>

      <li>CSS</li>

      <li>JS</li>

    </ul>


li::marker {

  content: "□";

}
```

5.::Selection pseudo-element:-

The ::selection pseudo-element matches the portion of an element that is selected by a user.
The part will selecting will reflect changes

Syntax:

```
selector::selection{

    property:value;

}
```

```
<p>

  Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit recusandae

</p>
```

```
p ::selection {

color: blueviolet;

background-color: aquamarine;

}
```

<--->

PSEUDO-CLASS SELECTOR

A CSS pseudo-class is a keyword added to a selector that specify a special state of the selected element.

Syntax:-

```
selector: pseudo-class {

    property:value;

}
```

1.Anchor pseudo class -

The anchor pseudo-classes represent the state of links as visited, unvisited,active/currently selected

Visited - purple
Active / currently selected
Unvisited
Focus

Anchor pseudo-classes also enables you to activate
the HTML elements or apply a specified style to an
element when the mouse pointer is kept over it

THE ANCHOR PSEUDO-CLASSES INCLUDE THE FOLLOWING :

:link (Applies styles to non visited links)
:visited (Applies styles to visited links)
:hover (Applies styles to an element over which the mouse-pointer moves)
:active (Applies styles to an active element)

Syntax:-

I) Link : once we visit the color will change to mentioned one.

```
   a:link {
color: darkgoldenrod;
}
```

<----------------------->

Ii) Visited

```
<body>
 <a href="https://www.google.co.in/">click here</a>
</body>
/* visited */
a:visited {

  color: red;
}
```

<------------------------------->

3)**Hover :** when u place cursor color will change

```
a:hover{

    color: brown;

}
```

Iv) Active :

when user clicks on the link

```
a:active{

    color: black;

}
```

<------------------------------------>

V) Focus : when its in active state we can add style property
once user clicks on any input fields

```
input:focus {

  color: gray;

}
```

v1)checked : when user checks on radio or check-box input field.

```
input:checked{
    height : 100px;
    width :100px;
    accent-color : blue;

 }
```

<----------------------------------->

PSEUDO CHILD

1.First-child:-

```
<div>
      <p>First child1</p>

</div>
<div>
 <p>First child2</p>

    <p>First child3</p>

</div
<p>First child4</p>
p:first-child {

  color: rgb(11, 245, 11);

}
```

<----------------------------------->

2.Last-child

```
<p>First child1</p>

<div>

    <p>First child2</p>

    <p>First child3</p>

</div>

    <p>First child4</p>


p:last-child {

  color: rgb(11, 245, 11);

}
```

<--->

3.Nth-child

1.Odd

```
<p>First child1</p>

    <p>First child2</p>
```

```
    <p>First child3</p>

    <p>First child4</p>

p:nth-child(odd) {

  color: rgb(11, 245, 11);

}
```

<--->

2.Even

```
  <p>First child1</p>

    <p>First child2</p>

    <p>First child3</p>

    <p>First child4</p>


p:nth-child(even) {

  color: rgb(11, 245, 11);

}
```

<----------------------------->

4.first of type:-

Specify a background color for the first <p> element of its parent:

eg:

```
<h1>Welcome to My gallerypage</h1>

<p>This paragraph is the first child of its parent (body).</p>


 <h1>Welcome to My Homepage</h1>
<p>This paragraph is not the first child of its parent.</p>

<div>
  <p>This paragraph is the first child of its parent (div).</p>
  <p>This paragraph is not the first child of its parent.</p>
</div>


 p:first-of-type {
    background-color: #ff0000;
}
```

<-->

5.last-of-type:- it will select only particular tag from the bottom of the body

```
<body>
  eg:- <div>
        <p> first child </p>
        <p> second child </p>
        <p> third child </p>
        </div>
        <h1> h1 from body  </h1>
         <p> p from body </p>  </body>
 p:last-of-type{
color : red;
}
```

<----------->

TEXT PROPERTY

color : color_name
Text-align : right , left , center
Text-transform : capatilize , uppercase , lowercase
Text-shadow : x-axis , y-axis , blur colorname
Text-decoration : underline , linethrough , overline
Letter-spacing : provides space between each letters
Word-spacing : provides space between each word
Text-indention : provides space at the starting of phara
line-height : provides space between each line

FONT PROPERTY

Font-size : large/small/medium
Font-weight : bold/bolder/lighter/normal
Font-style : italic
Font-family : font styles
Font-varient : normal/small-caps

BACKGROUND PROPERTY

Background-image : url ("image.jpg")
Background-repeat : repeat/no-repeat/repeat-x/Y
Background-size : cover/100%
Background-Position : right/left/center/top/bottom
Background-attachment : scroll/fixed

<----------->

COLOR/BACKGROUND-COLOR

```
Color : (color name)
Color :  #efefef;
Color : rgb(255,255,0)
Color : rgba(red, green, blue, alpha(0  - 1.0))
Color :  hsl(hue(deg), saturation(%), Lightening(%))
Color : hsla(hue(deg), saturation(%), lightening(%), alpha)

Background-color : (use any color property)
```

Hue:
Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, 240 is blue.

Saturation:
It takes value in percentage; 0% means a shade of gray and 100% is the full color.

Lightness:
Lightness is also a percentage; 0% is black, 100% is white.

POSITION PROPERTY

The position property specifies the type of positioning method used for an element.

Note: There are five different position values Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

PROPERTY VALUES:-

- static - Default value. Elements render in order, as they appear in the document flow
- absolute - The element is positioned relative to its first positioned (not static) ancestor element
- fixed - The element is positioned relative to the browser window
- relative - The element is positioned relative to its normal position.
- sticky - The element is positioned based on the user's scroll position.

Note: A sticky element toggles between relative and fixed, depending on the scroll position.

CSS BOX MODEL

- The CSS box model is a container that contains multiple properties including borders, margin, padding, and the content itself.

- content: This property is used to displays the text, images, etc, that can be sized using the width & height property.
- padding: This property is used to create space around the element, inside any defined border. The padding is transparent
- border: This property is used to cover the content & any padding, & also allows to set the style, color, and width of the border. A border that goes around the padding and content
- margin: This property is used to create space around the element ie., around the border area.

<----------->

FLEXBOX

Flexlayout is based on flex flow directions:-

-->Block , for sections in a webpage
-->Inline for text
-->The main idea behind the flex layout is to give the container the ability
to alter its items width/height (and order) to best fill the available space
(mostly to accommodate kind of display devices and screen size).

<----------->

PROPERTIES

Display : flex | inline-flex ;

Flex-direction : row |column | row-reverse | col-reverse

Flex-wrap : wrap | nowrap |wrap-reverse

Justify-content : flex-start | flex-end |center| space-around | space-between | space-evenly

Align-items : flex-start | flex-end | center

Align content : flex-start | flex-end | center

<----------->

Z-Index:

---->Z Index (z-index) is a CSS property that defines the order of overlapping HTML elements.
    Elements with a higher index will be placed on top of elements with a lower index.
----> it will have a default value of auto

Opacity Property:

--->it is used to provide blurness(transparency) for element
---->The opacity property takes values from 0.0 to 1.0, with 1 being the default value for all elements.
     The lower the value, the more transparent. So if an element is given an opacity of 0, it would be invisible.
---->An opacity value of 1 is the default

filter Property:

--->The filter CSS property applies graphical effects like blur or color shift to an element. Filters are commonly
   used to adjust the rendering of images, backgrounds, and borders.

--->some of filter property are:
    filter: blur(pixels)
    filter: brightness(percentage)
    filter: contrast(percentage)
    filter: drop-shadow(x-axis y-axis blur color_name)
    filter: grayscale(percentage)
    filter: hue-rotate(degree)
    filter: invert(percentage)
    filter: opacity(percentage)
    filter: saturate(percentage)

<--->

Gradients

The Gradient in CSS is a special type of image that is made up of progressive & smooth transition between two or more colors.

There are 2 types of gradient

Linear-gradient (goes down/up/left/right/diagonally)

Background-image: linear-gradient(direction, color-stop1, color-stop2,…);
Background - image: linear-gradient(yellow, green)
It includes the smooth color transitions to going up, down, left, right, and diagonally.
The minimum two-color required to create a linear gradient.
More that two color elements can be possible in linear gradients.

The staring point and the direction are needed for the gradient effect.

Radial Gradient (defined by their center):

Background - image: radial- gradient(shape size at position, start-color,…., last-color);

The radial-gradient () function is an inbuilt function in css which is to set a radial gradient as the background image,

It starts at a single point and emanates outward.

By default, the first color starts at the center position of the element and then fade to the end color towards the edge of
The element.

Fade happens at an equal rate until specified.

By default, shape is ellipse, size is farthest-corner, and position is center.

Background-image: radial-gradient(circle, orange, yellow, green)

Uses of different size keywords in gradient

The size parameter defines the size of the gradient. It can take four values:

Closest-side
Farthest-side
Closest-corner
Farthest-corner

<----------→

Position

It sets how an element is positioned in a document. The top, right,bottom, and left properties determine the final location of positioned elements.

There are five values the position property can take. They are:

Static-(by default)it will not take any properity like top,left,bottom,right
Relative - fixed- we can fix top right or bottom(it will take whitespace)
Absolute- for element , we can change
Fixed - position will be fixed cannot move to other position
Sticky - its a combination of fixed and relative

Note:

If the position property is set to relative, absolute or fixed you can set the top, right,bottom and left properties
If the position property is set to static top, right bottom and left properties.

Description

static

Normal position for the element(where top ,right,bottom, and left have no effect)

```
Div { position: static;}
```

---

relative

Position the element absolutely relative to its conatiner

```
Div { position: absolute; top:10px;left:15px; }
```

---

absolute

Position the element absolutely relative to its container.

---

fixed

Positiion the element relative to the screen's viewport and stay fixed on screen when scrolling

```
Div { position : fixed; top:10px;left:15px
```