

ch-4.

INTRODUCTION to CSS

CSS → Cascading style Sheets.

- used to style HTML documents.
- describes how HTML elements should be displayed.
- It is used to define style for your webpage, including design, layout and variation in display for different device and screen size.

CSS selector :-

CSS selectors used to find the HTML.
It is of two types :-

- i) Id selector
- ii) Class selector .
- iii) ~~select~~ selector.
element

i) Element Selector :- (should be written inside head)

Syntax.

```
tag-name {  
    property1: value;  
    property2: value;  
}
```

ex. p {

~~text-align: center;~~
~~color: red;~~
}



ii) id selector :-

- uses id attribute to select a specific element.
- To select the element with a specific id, write hash (#) followed by the id of element.

ex <!DOCTYPE html>

<html>

<head>

<title> x </title>

<style>

#id1 {

~~text-align: center;~~
~~color: red;~~

}

</style>

</head>

<body id = "id1">

Hi:

</body>

</html>



Syntax

```
#nameofselector {  
    property1: value1;  
    property2: value2;  
    ...  
}
```

Note: An id name cannot start with a no.

iii) Class selector :-

- selects HTML elements with specific class attribute.
- To select the element with a specific id, write a "." character followed by class name.

ex. <!DOCTYPE html>

<html>

<head>

<title> X </title>

~~</head>~~

<style>

.sel1 {

text-align: center;
 color: red;

}

</style>

</head>

<body class = "sel1">

X

</body>

</html>

Here ~~these~~ all HTML elements will be red and center-aligned.

Note.

→ If you want a specific tag to be selected.

Syntax.

<Tagname>. <name of selector> {

property1: value1;
property2: value2;

}

Group Selectors :-

Selects all HTML elements with same style definition.

Ex.

h1 {

text-align: center;
color: red;

}

h2 {

text-align: center;
color: red;

}

p {

text-align: center;
color: red;

}



h1, h2, p {

text-align: center;
color: red;

}

Comments in CSS :-

Syntax: /*-----*/

↳ should be written inside style.

Methods To INTEGRATE CSS with ~~TO~~ HTML :-

CSS can be added to HTML in 3 ways :-

i) Inline : using style attribute inside HTML elements :

ex. `<body style=" " >`

`<p style=" " >`

etc.

ii) Internal : using style attribute inside `<head>`.

ex. `<head>`

`<title> --- </title>`

`<style>`,

,

,

`</style>`

`</head>`

iii) External :

using `<link>` element to link an external CSS file in HTML..

Step 1. Create a file , write the `<style>` --- `</style>` and then save the file in `.css` extension.

Step 2. Now, in HTML file inside `<head>` write:

`<link rel="stylesheet" href=_____>`

Note. Both the CSS file and HTML file should be in same folder.

CSS - colors :-

i) RGB :- R - red
G - green
B - Blue

Syntax. `rgb (— , — , —)`

value should be b/w
0 to 255.

ex. `<style="background-color: rgb(232, 3, 4);">`

ii) RGB A value :-

R = red

G = green

B = blue

A = alpha.

Syntax. `rgba(— , — , — , —)`

value should
be betn 0 to
255

value should be
betn 0 to 1.0.

iii) Hex value :-

Syntax : #rrggbb



— — — — —

value should be betn 0 to ~~255~~ ff.
(hexadecimal value)

ex. `< style = "background-color: #fa3ecc;" >`

iv) HSL :-

H = hue

S = Saturation

L = lightness

Syntax. `hel(—, —, —)`

0 - 360

value should be bet' 0 - 100.

ii) ex. `style="background-color: hsl(217, 28%, 57%);"`

Setting up Height and Width of element:

Syntax.

`<style>`

selector {

`height: _____;`

`width: _____;`



either in cm

px

%

or

"auto"

Note : `max-width:`

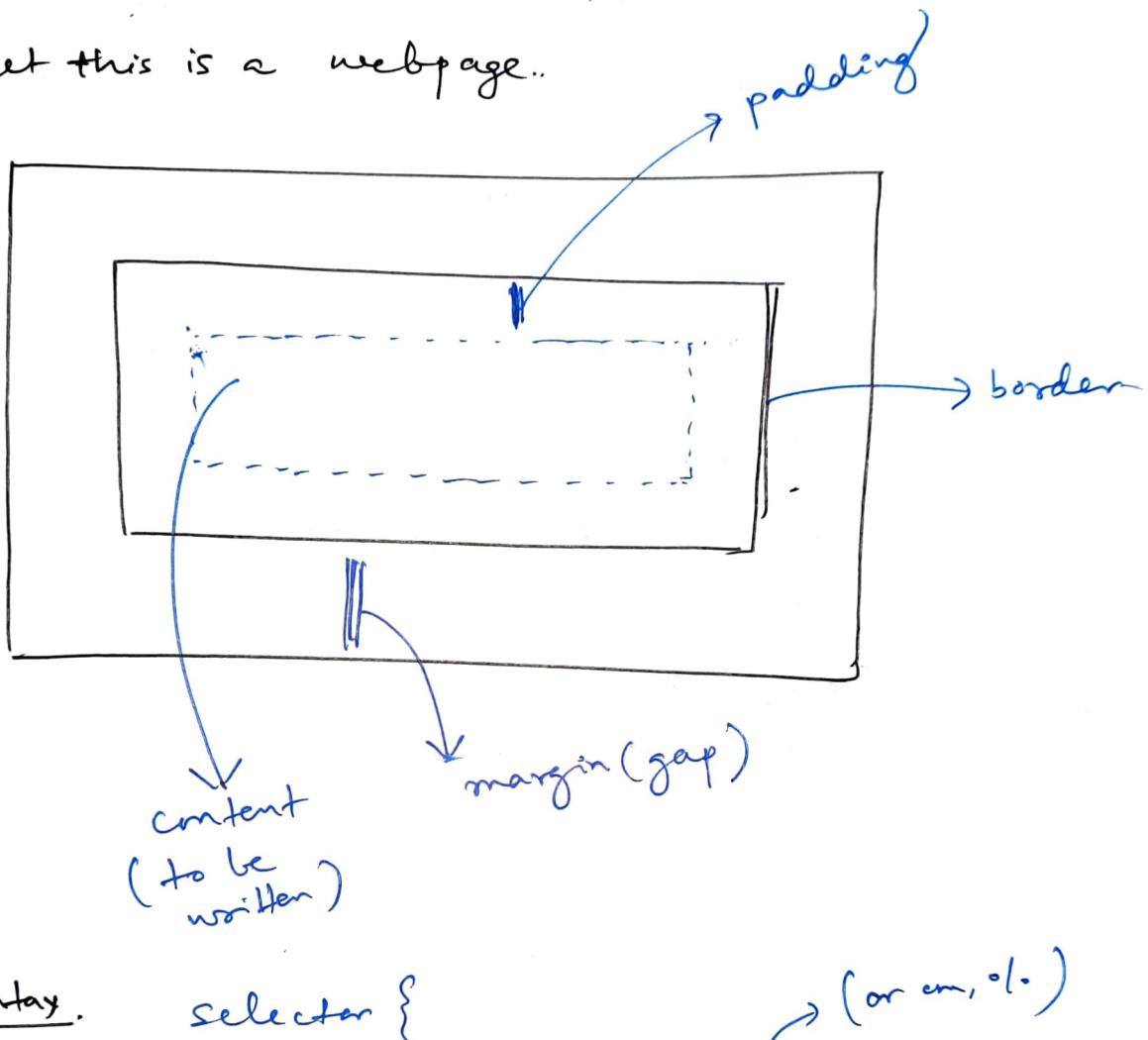
`max-width: _____;`

—;

Box Model:-

used to give layout, It consists of margin, Border, padding, content.

ex. Let this is a webpage..



Syntax.

selector {

 width: — px; → (or cm, %.)
 border: — px — color name;
 padding: — px;
 margin: — px;

}

CSS outline :-

It is a line drawn outside of element's border.

It has following outline properties :

- i) outline-style
- ii) outline-color
- iii) outline-width
- iv) outline-offset
- v) outline-ε.

i) outline-style :-

It has following values :-

- 1) dotted
- 2) dashed
- 3) solid
- 4) double
- 5) groove }
6) ridge } 3D.
7) inset /
8) outset }
- 9) none → no outline
- 10) hidden -

ex. ~~#~~ # id {

outline-style : solid;
}

ii) Outline-color :-

used to specify the color of outline

It can be set by :-

- i) name : ex. red
- ii) hex : ex. "# ff 30 2a"
- iii) rgb : ex. rgb(2, 8, 251),
- iv) rgba: ex. rgba(52, 81, 99, 0.5),
- v) hsl: hsl(0, 100%, 50%),

Syntax: <selector>

outline-color: _____;

}

ex. red;
rgb(255, 255, 0);

iii) Outline-width :-

Specify the width of the outline
and can have either one of the
values;

- 1) thin (around 1px)
- 2) medium (around 3px)
- 3) thick (around 5px)
- 4) specified size (in cm, px or %)

Syntax: <selector>

outline-width: _____;

thin or
1px etc.

}

N) Outline offset :-

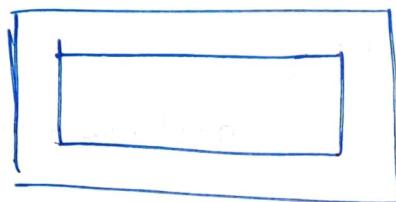
add space b/w outline and border of element.

Syntax: selector {

outline-offset: 15px;

}

O/P



Text in CSS :-

Text in css has following property:

- i) text color
- ii) text alignment
- iii) text decoration.
- iv) text transformation
- v) Text spacing
- vi) text shadow.

i) text-color:

specifies the color of text;

Syntax. <style="color: _____;">

at

selector {

 color: _____;

}

ii) text-alignment:

used to set the horizontal alignment of text. It can be:

- i) right —
- ii) center —
- iii) left —
- iv) justify —

Syntax:

<style="text-align: _____;">

↓

- left
- right
- center
- justify



iii) text-decoration:

It has following properties:

- i) overline → abc
- ii) line-through → ~~abc~~
- iii) underline → abc

v) Text transformation :-

used to specify uppercase and lowercase in a text.

It has 3 properties:-

- i) uppercase → turn every letter to uppercase capital letter.
- ii) Lowercase → turn every letter to small letter.
- iii) Capitalize → turn every first letter of each word to capital letter.

④ Syntax. i) `<style="text-transform: uppercase;">`
ii) `<style="text-transform: lowercase;">`
iii) `<style="text-transform: capitalize;">`

v) Text-spacing :-

It contains following properties:

- i) text-indent : specify the indentation/gap for first line of text.
- ii) letter-spacing : space b/w the letters.
- iii) line-height : Space b/w the lines.
- iv) word-spacing : space b/w the words.
- v) white-space :

Syntax.

- i) <style="text-indent: -px;">
- ii) <style="letter-spacing: -px;">
- iii) <style="line-height: -px;">
- iv) <style="word-spacing: -px;">

FONTS in CSS :-

- Fonts in CSS has some property. Some of these are :

- i) font-style
- ii) font-family:

i) font-style:

This has 3 property:

- i) normal : the text shown normally.
- ii) italic : text shown in italic
- iii) oblique : text shown in leaning (same as italic but less supported)

Syntax.

- i) <style="font-family: normal;">
- ii) <style="font-family: italic;">
- iii) <style="font-family: oblique;">

ii) Font-family:

used to specify the font of a text.

It has 5 generic font families.

i) Serif → Times New Roman

Georgia

Garamond etc...

ii) Sans-serif: Arial,

Verdana; etc...

iii) Monospace: Courier New;

Monaco etc.

iv) Cursive: Brush Script MT;

Lucia Handwriting etc.

v) Fantasy: papyrus;

Copperplate etc --

Syntax. <style = "font-family: _____, _____, _____,
_____ ; "> .

here you can add
as many many font
types, ~~the~~

Note. If the first font type is available in
the system, it will display that only else,
it will move to ~~the~~ next font type and
so on.

LINKS in CSS :

It is used to style the links in many ways.

ex Link : Link etc.

- Links can be styled in any CSS property ex. color, font-family etc.
- Links can be styled depending upon on what states they are in:
 - i) link: normal, unvisited link
 - ii) visited: visited link / used one.
 - iii) hover: when mouse is kept over the link.
 - iv) active: At the time of click.

Syntax.

i) `<style> a:link:color: —, etc. -- >`

or

`<style>`
`a:link {`
 `color: — ;`
 `;`
 `}`
`</style>`

any other design

ii) a:visited {

color: _____;

}

iii) a:hover {

color: _____;

:

}

iv) a:active {

color: _____;

}

:

LIST in CSS :

There are two types of Lists.

i) Unordered list: ~~start~~ bullets. ex

circle ; ○

square ; ■

ii) Ordered list: Alphabet. Nos; Roman nos.

~~ex.~~ ex. lower-alpha a

upper-alpha A

lower-roman i

upper-roman I

Syntax:

<style="list-style-type: — ;" >

square
circle
lower-alpha etc.
numbers

TABLE in CSS :

It has following properties :

- i) Table Borders
- ii) Table size
- iii) Table Alignment
- iv) Table style
- v) Table Responsive.

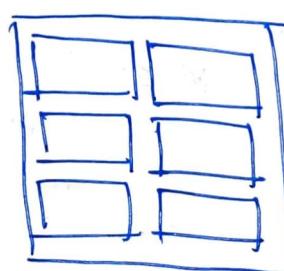
i) Table Borders :

Syntax:

table, th, td {

border: — px —
↓
solid
dotted

o/p:



Note: The o/p has double border.

* Full-width Table:

To span the entire screen (full-width);

Syntax: table {
 width: ____%;
}

* Collapse Table Border:

To remove the double border and convert to a single border.

Syntax: table {
 border-collapse: collapse;
}

o/p,

ii) Table Size:

Table width and height:

used to set the height, width of table, td, or th.

* For table:

Syntax. table {
 width: ____%;
 height: ____%;
}

* For table header:

`th {
height: ____ %;
width: ____ %;
}`

* For table data:

`td {
height: ____ %;
width: ____ %;
}`

* For any two:

ex. ~~table th, td;~~

`th, td {
height: ____ %;
width: ____ %;
}`

iii) Table Alignment:

It is divided to two types.

- i) Horizontal alignment
- ii) Vertical Alignment.

i) Horizontal Alignment:

sets the horizontal alignment
of content in <th> or <td>.

By default: <th> → center aligned
<td> → left aligned.

Syntax :

```
<tag-name> {  
    ↓  
    <td>  
    or  
    <th>  
    }  
    text-align: _____;  
    ↓  
    center  
    or  
    left  
    or  
    right
```

ii) Vertical Alignment:

sets the vertical alignment of the content in `<td>` or `<th>`.

By default, the vertical alignment of `<td>` and `<th>` is middle.

Syntax .

```
<tag-name> {  
    ↓  
    text  
    vertical-align: _____;  
    ↓  
    top  
    or  
    middle  
    or  
    bottom
```

iv) Table style:

* Table padding:

used to control space between border and content. on <td> and <th> tag.

Syntax: th, td {
padding: -px;
}

v) Responsive table:

A responsive table will display a horizontal scroll bar if the screen is too small to display the full content.

To view it, resize the window to see the effect.

Syntax:

```
<div style="overflow-x: auto;">  
  <table>  
    <tr>  
      <th>  
        |  
        |  
        |  
        |  
    </th>  
  </tr>  
</table>  
</div>
```

Position Property in CSS:

Also called CSS layout.

It defines the type of positioning method used for an element.

There are 5 different positioning values:

- i) static
- ii) relative
- iii) fixed
- iv) absolute
- v) sticky

* Static:

HTML elements are positioned static by default.

- this is not affected by properties: top, bottom, left, right.
- The elements are positioned according to normal flow of page.

Syntax: <style="position: static;">

* relative:

- It is positioned relative to its normal position.
- We can use the property:

- i) right: —px;
- ii) left: —px;
- iii) top: —px;
- iv) bottom: —px;

Syntax: <style>

<^④selector>{

 position: relative;

 <property>: —px;

}

* fixed:

- It is positioned relative to viewport meaning it always stays in the same place even if the page is scrolled.

- we can use the properties:

- i) top: —px;
- ii) bottom: —px;
- iii) left: —px;
- iv) right: —px;

* Absolute:

- it is positioned to the nearest position ancestor (instead of positioned relative to viewport like fixed)
- If an absolute positioned elements has no ancestors, it will use the document body and move along page scrolling.

Note: absolute elements are removed from normal flow and can overlap elements.

- we can also use the property of top, bottom, left, right.

Syntax:

<style>
<selector> {

position: absolute;

<property>: —px;

- * Sticky:
 - it is positioned on user's scroll position.
 - It is positioned relative to until a given offset position in viewport, then it sticks at that place.
 - we can use property: top, bottom, left, right.

Note: At least one of the ~~top~~ above properties must be used.

Syntax:

```
<style>
  (selector) {
    position: sticky;
    (property): -px;
  }
}
```

Navigation Bar in CSS:

There are two types of Navigation Bar.

- i) Vertical
- ii) Horizontal.

- Navigation Bar is a list of links.
- It needs standard HTML as base.

ex.

Vertical :



Horizontal:



Vertical Navigation Bar :

It is a list of links.

ex.

```

<ul>
  <li> <a href="#home">Home </a> </li>
  <li> <a href="#news">News </a> </li>
  <li> <a href="#contact">Contact </a> </li>
  <li> <a href="#about">About </a> </li>
</ul>
  
```

O/P :

- Home
- News
- Contact
- About

Note: To remove bullets, margin, padding;

```
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
}
```

O/P: Home
News
Contact
About

* <style>

```
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    width: 200px;  
    background-color: #f1f1f1;  
}  
  
li a {  
    display: block;  
    color: #000;  
    padding: 8px 16px;  
    text-decoration: none;  
}  
  
li a:hover {  
    background-color: #555;  
    color: white;  
}
```

<style>

a proper vertical Navigation Bar.

Arranging the
blocks of long
vertically

Designing the
blocks for
links

what when we
take the mouse
there

Horizontal Navigation Bar:

there are two ways to create horizontal navigation bar.

- i) inline
- ii) floating.

i) Inline List items:

ex. `ul { list-style-type: none; margin: 0; padding: 0; }` General code
~~for all~~

`li { display: inline; }` ***

o/p: Home News Content About.

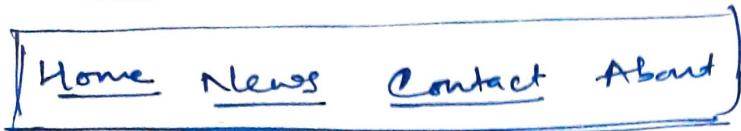
ii) Floating list items:

ex. `ul { list-style-type: none; margin: 0; padding: 0; overflow: hidden; }
li { float: left; } .wp`

`li a { display: block; padding: 8px; }`

```
background-color: #cccccc;  
}
```

O/P:



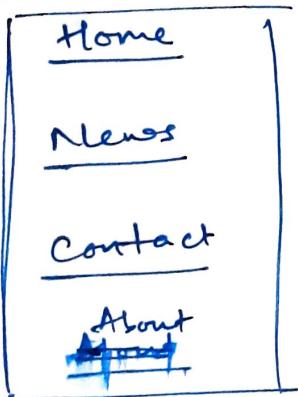
* If float: right;

O/P:

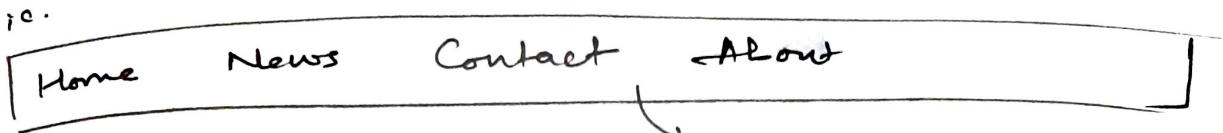


* If float: center;

O/P:



* General structure for horizontal navigation bar:



with background color.

```
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    overflow: hidden;  
    background-color: #333;  
}  
    display in edge of page
```

(used to remove bullet s.)

that area will be occupied by the bar.

```
li {  
    float: left;  
}
```

```
li: a {  
    display: block;  
    text-decoration: none;  
    color: white;  
    text-align: center;  
    padding: 14px 16px;  
}
```

(display in block)

(used to remove underline)

Dropdown in CSS:

Generally, the dropdown is ~~gen~~ divided into 3 basic structure / ~~do~~ selectors.

- i). dropdown
- ii) dropdown-content
- iii) dropdown:hover; dropdown-content

ex. position for dropdown

```
<style>    → position for dropdown
    .dropdown {           position: relative;
        display: inline-block;
    }
```

```
    .dropdown-content {           display: none;
        position: absolute;
        background-color: ;
        min-width: ;
        padding: ;
    }
```

```
    .dropdown:hover .dropdown-content {           display: block;
    }
```

shows
as a block.

In Body :

```
<body>
  <div class="dropdown">
    <p> Mouse over me </p>
    <div class="dropdown-content">
      <p> Hello world </p>
    </div>
  </div>
</body>
```

Note:

for making Dropdown Link.

step i) style the dropdown button

step ii) position the dropdown content

step iii) dropdown content (that should be hidden initially)

step iv) Create link inside dropdown.

step v) Change the color of dropdown links on hover

step vi) Change the color of background of dropdown

button when content is shown.

ex.

.dropdown {

 background-color:

 color:

 padding:

 border: none;

}

• .dropdown {

 position: relative;

 display: inline-block;

}

• .dropdown-content {

 display: none;

 position: absolute;

 background-color:

 min-width:

}

• .dropdown-content a {

 color:

 padding:

 text-decoration: none;

 display: block;

}

• .dropdown-content a:hover {

 background-color:

}

```
.dropdown:hover .dropdown-content{  
    display: block;  
}
```

```
.dropdown:hover .dropbtn{  
    background-color: # ;  
}
```

In body:

```
<body>  
    <div class = "dropdown">  
        <button class = "dropbtn"> Drop down </button>  
        <div class = "dropdown-content">  
            <a href = "#> link 1 </a>  
            <a href = "#> link 2 </a>  
        </div>  
    </div>  
</body>
```

FORMS in CSS:

Forms: ex. First Name:

Last Name:

ex.

```
<!DOCTYPE html>
<html>
  <head>
    <title> form </title>
    <style>
      input[type=text], select {
        width:
        padding:
        margin:
        display:
        border:
      }
      input[type=number], select
      {
        width:
        padding:
        margin:
        display:
        border:
      }
      input[type=submit]
      {
        width:
        background-color:
        color:
        padding:
        margin:
        border:
        border-radius:
      }
    </style>
  </head>
  <body>
    <form>
      <input type="text" name="text">
      <input type="number" name="number">
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

```
input[type=submit]:hover
{
    background-color: green;
}

div
{
    border:
    background-color:
    padding:
}

</style>
</head>

<body>
<div>
    <form>
        <label for="name">Name </label>
        <input type="text" id="Name"
               name="Name" placeholder="Your name..."/>

        <label for="Regno"> Regno. </label>
        <input type="text" id="regd"
               name="regno" placeholder="Your
               registration no.">

        <label for="age"> Age </label>
        <input type="number" id="age"
               name="age" placeholder="Your age--"/>
```

```
<label for="city"> city </label>
<select id="city" name="city">
    <option value="Bhubaneswar"> BBSR </option>
    <option value="Puri"> Puri </option>
    <option value="Cuttack"> Cuttack </option>
</select>

<input type="Submit" value="Submit" />
<input type="Submit" value="Cancel" />

</form>
</div>
</body>
</html>
```