<body background='abc.png'
style='background-repeat: no-repeat;
background-attachment: fixed;
background-position = center'

Chapter 3
Cascading Style Sheets

Introduction and Levels of style sheets

features
flexibility
code Rendering
easy manage
Global change
time saving
easy maintenance

- □ CSS stands for "Cascading Style Sheets".
- □ Cascading Style Sheets is used to format the layout of web pages.
- □ CSS helps web developers to create uniform look across several pages on websites.
- □ Using CSS makes task easier to define common layout across the pages.
- □ CSS is specially used to style pages, including the design, layout and variations in display for different devices and screen sizes. .

There are three levels of style sheets:

(The style specification formats : The format of a style specification depends on the level of style sheet .)

Line level
page label
external
Superior to html
multi device compatible
W3C compatibility

1. **Inline** level <body><li style='color#0000FF'>welcome </li>
   - □ is specified for a specific occurrence of a tag and apply only to that tag.
   - □ appears in the *tag* itself.
   - □ is fine-grain style, which defeats the purpose of style sheets - uniform style.
2. **Document-level** style sheets(internal) embeded
   - □ apply to the whole document in which they appear
   - □ appear in the *head* of the document

   <head>
   <style type=text/css>
   div{ color;red;
   background-color:green;
   font-size:20PX;}
3. **External** style sheets
   - □ can be applied to any number of documents
   - □ are in separate files, potentially on any server on the Internet
   - □ use a <*link*> tag to let the browser fetch and use an external style sheet file
   - □ <link rel = "stylesheet" type = "text/css" href = "style.css"></link>
   - □ The file should not contain any html tags.
   - □ The style sheet file must be saved with a .css extension.

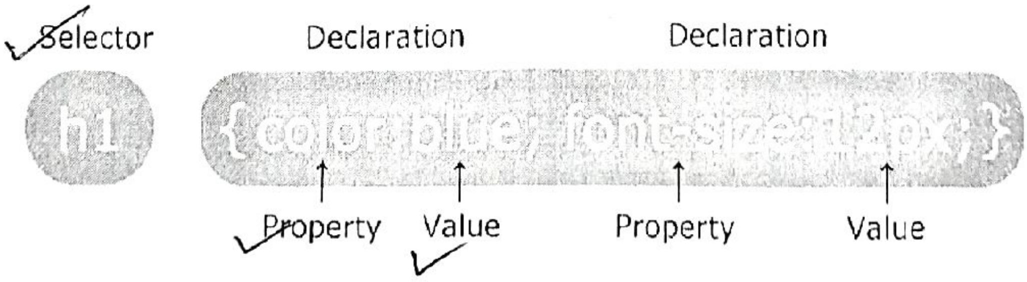
1-m
m-m

<body>
<div>welcome
</div>
<div> Hisptal </div>

Note: So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

CSS Syntax

- □ A CSS rule-set consists of a selector and a declaration block:



- □ The selector points to the HTML element you want to style.

- □ The declaration block contains one or more declarations separated by semicolons.
- □ Each declaration includes a CSS property name and a value, separated by a colon.
- □ A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.
- □ In the above example all <h1> elements will be of font color blue, with font size 12px.

Example:

```
<!DOCTYPE html>
<html>
<head>
//Internal level css
<style>
body {
                    background-color: linen;
}
h1 {
                    color: maroon;
                    margin-left: 40px;
}
</style>
//External level css
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
  <p style="color: red;">This is also paragraph</p>         //Inline Level css
</body>
</html>
/*style.css*/          /*This is an external file named style.css*/
  h1{ font-size: 25px;}
```

local path (same path

'C:\user\admin\desktop\ one.css'>

'file:/// " '>

Note:

Do not add a space between the property value and the unit (such as margin-left: 25 px;) . The correct way is : margin-left: 25px;

CSS Selectors

- □ CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

  Types of selectors: 0) tag selector

  1. **The element selectors**
     - □ The element selector selects elements based on the element name.
     - □ The simplest selector form is a single element name ,such as *h1*.
     - □ The property values in the rule apply to all occurrences of the named element.
     - □ The selector could be a list of element names separated by commas (*h1,h2,h3,……*).

<Head>
<style = type = 'text/css'>
h1{
{ color : 0000FF; }

<head>
<body>
<h1> welcome <h1>
<h2> welcome <h2>

you can not declared same selector name with same

Grouping selector are those selector which are seperted by comma and its internal property will be the same for all.

<~~body~~> h1, h2, h3 { color: 0000FF; }

body में

<h1> — <h3> —

h2 —

.p{
color: 00FF00;
}

<h1 class="p">
hellooo </h1>

2. Class selectors

- □ The class selector selects elements with a specific class attribute.
- □ To select elements with a specific class, write a period (.) character, followed by the name of the class.
- □ Class selectors are used to allow different occurrence of the same tag to use different style specifications.

3. Id selectors

must not be duplicate

prefix [illegible]

<Style
[illegible] # h1{
color: 0000FF; }

[illegible] prefix
<h1 id="h1">hello</h1>
<p id="h1"> </P>

- □ The id selector uses the *id* attribute of an HTML element to select a specific element.
- □ The id of an element should be unique within a page, so the id selector is used to select one unique element!
- □ To select an element with a specific id, write a hash (#) character, followed by the id of the element.

[illegible] prefix [illegible]

prefix id =

duplicate [illegible]

4. Universal selector

- □ CSS Universal selector selects any type of elements in HTML page.
- □ CSS Universal selector is defined by using asterisk (i.e. *).
- □ CSS Universal selector is also followed by selector.
- □ Universal selector is useful to style all elements in HTML pages or used to style all elements with in element.

*{ margin: 0;
padding: 0;
}

5. Pseudo Classes

- □ Sometime we need to apply style when something happen , rather than because the target element simply exists, i.e. specify by pseudo classes.
- □

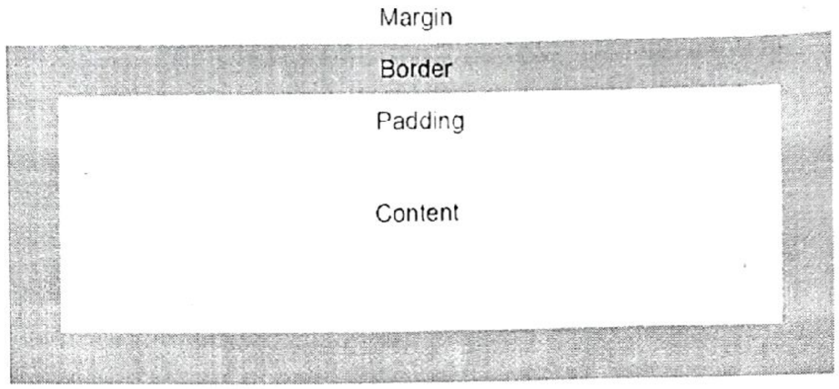
Example:

```
/* Single * is used to set up for all body elements */
*{
    color: blue; /*defines blue color for all font*/
   background-color: green; /*define green background color */
    font-size: 20px;  /*defines font size for all font*/
}
/*Following use of asterisk is used to style all elements of div element*/ div *{
    color: yellow; /*define yellow color to all fonts with in div elements*/
    font-size: 30px; /*define font size to all fonts with in div elements*/
}
```

00FF00	green
FF0000	Red
0.000FF	Blue

Box Model

- □ All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.
- □ The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:



Explanation of the different parts:

- □ **Content** - The content of the box, where text and images appear
- □ **Padding** - Clears an area around the content. The padding is transparent space inside border
- □ **Border** - A border that goes around the padding and content
- □ **Margin** - Clears an area outside the border. The margin is transparent . space outside border

The box model allows us to add a border around elements, and to define space between elements.

```
h1{
border: 1px solid black
padding: 10px;
color: white;
weidth: 30px;
height: 20px;
text-align: center
font-family: "cosmic sans ms";
background-color: Black;
```

CSS Margin:

- □ The CSS *margin* properties are used to create space around elements, outside of any defined borders.
- □ CSS has properties for specifying the margin for each side of an element.
  - margin-top
  - margin-right
  - margin-bottom
  - margin-left

All the margin properties can have the following values:

- □ *auto* - the browser calculates the margin
- □ *length* - specifies a margin in px, pt, cm, etc.
- □ *%* - specifies a margin in % of the width of the containing element
- □ *inherit* - specifies that the margin should be inherited from the parent element

Example:

If the margin property has four values:

margin: 25px 50px 75px 100px;

- □ top margin is 25px

- □ right margin is 50px
- □ bottom margin is 75px
- □ left margin is 100px

If the margin property has three values:

margin: 25px 50px 75px;

- □ top margin is 25px
- □ right and left margins are 50px
- □ bottom margin is 75px

If the margin property has two values:

margin: 25px 50px;

- □ top and bottom margins are 25px
- □ right and left margins are 50px

If the margin property has one value:

margin: 25px;

- □ all four margins are 25px

Note:

- □ The *auto* value – when the auto value is set to margin property , specified width took up first and then remaining space will be split equally between left and right margin from centre of container .
- □ The *inherit* value – lets the left margin of the elements be inherited from parent element.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
    div {
                border: 1px solid red;
                margin-left: 100px;
    }
    p.demo {
                margin-left: inherit;
    }
    div.demo1 {
                width:300px;
                margin: auto;
                border: 1px solid red;
    }
</style>
</head>
<body>
    <p>You can set margin horizontal< p>
    <div class="demo1">
```

you can set margin horizontal

This div will horizontal centred

Use of the inherit value

Let the left

This paragraph has

```
            This div will be horizontally centered because it has margin: auto;
    </div>
    <h2>Use of the inherit value</h2>
            <p>Let the left margin be inherited from the parent element:</p>
    <div>
            <p class="demo">This paragraph has an inherited left margin (from the div
            element).</p>
    </div>
</body>
</html>
```

CSS Padding:

□ The CSS padding generates space around the elements content, inside of any defined border.

□ CSS has properties for specifying the padding for each side of an element.

- padding-top
- padding-right
- padding-bottom
- padding-left

All the padding properties can have the following values:

□ *auto* - the browser calculates the margin

□ *length* - specifies a padding in px, pt, cm, etc.

% - specifies a padding in % of the width of the containing element

□ *inherit* - specifies that the padding should be inherited from the parent element

Example:

If the padding property has four values:

padding: 25px 50px 75px 100px;

□ top padding is 25px

□ right padding is 50px

□ bottom padding is 75px

□ left padding is 100px

he

t

If the padding property has three values:

padding: 25px 50px 75px;

□ top padding is 25px

□ right and left paddings are 50px

□ bottom padding is 75px

If the padding property has two values:

padding: 25px 50px;

□ top and bottom paddings are 25px

tr : nth-child(even){
backgrnd-color : #   }

□ right and left paddings are 50px

If the padding property has one value:

padding: 25px;

□ all four paddings are 25px

Note:

□ The width property specifies the width of the element's content area. The content area is the portion inside the padding, border, and margin of an element.

□ So, if an element has a specified width, the padding added to that element will be added to the total width of the element.

CSS Border

□ The CSS border properties allow you to specify the style, width, and color of an element's border.

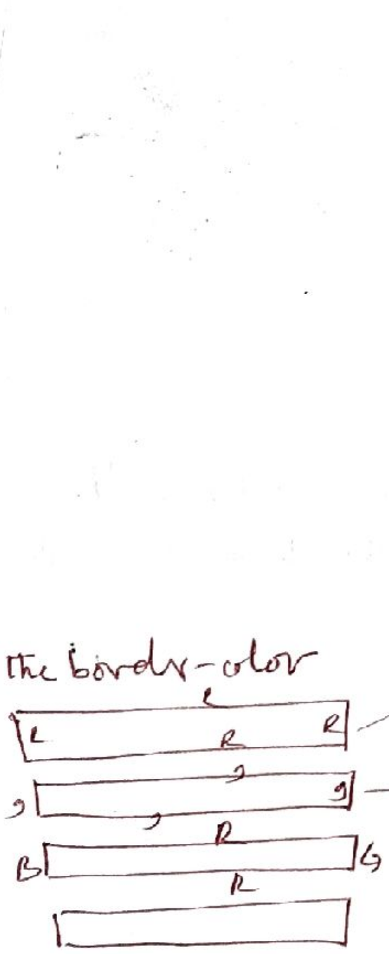
Some border related properties are given below:

- **border-style** – specify what kind of border to display (values are*(dotted, dashed, solid, double, groove, ridge, inset, outset, none, hidden)*. Also can have one to four value(*border-top-style, border-right-style, border-bottom-style, border-left-style*).
- **border-width** – specify width of the four borders.
- **border-color** – specify color of the four borders. Also can have one to four value (*top, right, bottom, left*).
- **border** – this property is used as shorthand property of *border-width, border-style and border-color.*
- **border-radius** – is used to add rounded borders to an element.

Example:

```
<!DOCTYPE html>
<html><head>
<style>
        p.one {
                border-style: solid;
                border-color: red; //This not support if not border-style
        }
        p.two {
                border-style: solid;
                border-color: green;
                border-radius: 5px;
        }
        p.three {
                border-style: solid;
                border-color: red green blue yellow;
        }
        p.four{
```

The border-color



border-radius : 10px;

scrollbars

visible
hidden
scroll
auto

overflow

p{ width: 200px
height: 100 px;
border: 1px solid red;
overflow: auto; overflow: scroll
overflow-x, }

```
                border: 1px dashed blue;
                }
        </style></head>
        <body>
                <h2>The border-color Property</h2>
                <p class="one">A solid red border</p>
                <p class="two">A solid green rounded border</p>
                <p class="three">A solid multicolor border</p>
                <p class="four">This is shorthand property:</p>
        </body>
        </html>
```

CSS Visibility

- □ Visibility property effect on the appearance of an element.
- □ Visibility property specifies whether or not an element visible.
- □ When you use value hidden , the hidden element take space on the page.
- □ Visibility property values are
  - **visible** – default value , element is visible
  - **hidden** – the element is hidden but still take space **collapse** – only used in
  - table , collapse the borders
  - **inherit** – specify that a property should inherit its value from its parent element
  - **initial** – sets this property to its default value

Example:

```
<!DOCTYPE html>
<html>
<head>
  <style>
            h2.a {
                        visibility: visible;
                        }
            h2.b {
                              visibility: hidden;

                  }
  </style>
</head>
<body>
  <h1>The visibility Property</h1>
  <h2 class="a">This heading is visible</h2>
  <h2 class="b">This heading is hidden</h2>
<p>Notice that the hidden heading still takes up space on the page.</p> </body> </html>
```

The visibility property
This heading is visible

CSS Display

- □ The display property is an important property that is used for controlling layout.
- □ The display property specifies if/how an element is displayed.
- □ As you saw that every HTML element has a default display value depending on what types of element it is.
- □ The default display value for most elements is either block or inline.

□ A block-level element always starts on a new line and takes up the full width available(*eg: <div>, <h1>....<h6>, <header>, <footer>, and <section>*).

□ An inline element does not start on a new line and only takes up as much width as necessary(*eg: <span>, <img>, and <a>*).

□ Some values of display property are as follows:

- **display**: *none* – is used to hide and remove an element from document layout.
- **display**: inline – is used to display in inline format.
- **display**: block – is used to display in block format.

CSS Backgrounds

height: 100vh;
width: 100%.
background-color: green;
body{ backgrnd-imge :url(" .jpg");
backgrod-repet: repet;
backgrod-size: 100% 100%

□ The CSS backgrounds are specified for background effects on elements.

□ The CSS background properties are:

- **background-color** – specifies background color of an element.
- **background-image** – specifies background image of an element(*background-image*: url("karnali.jpg"). ***Note** – if necessary give path.*
- **background-attachment** - property sets whether a background image scrolls with the rest of the page, or is fixed. (Values are: *scroll, fixed, inherit, initial and local*)
- **background-position** - sets the starting position of a background image.(Values are: *left top, left center, left bottom, right top, right center, right bottom, center top, center bottom, and center center*)
- **background-repeat** - sets if/how a background image will be repeated. (Values are: *no-repeat, repeat-y, repeat-x, space, and round*)
- **background-size** - specifies the size of the background images.(Values are: *auto, cover, contain, length, percentage, initial, and inherit*)
- **background-origin** - specifies the origin position (the background positioning area) of a background image.(Values are: *padding-box, content-box, border-box, inherit and initial*)
- **background-clip** - defines how far the background (color or image) should extend within an element. (Values are: *padding-box, content-box, border-box, inherit and initial*)
- **background-blend-mode** - defines the blending mode of each background layer (color and/or image).(Values are: *normal, multiply, screen, overlay, darken, lighten, color-dodge, saturation, color, luminosity;*)

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
#myDIV {
```

```
                width: 200px;
                height: 200px;
                background-repeat: no-repeat, repeat;
                background-image: url("bas2.jpg"), url("karnali.jpg");
                background-blend-mode: lighten;
        }
        </style>
        </head>
        <body>
                <div id="myDIV"></div>
                <p><b>Note:</b> Edge/Internet Explorer do not support the background-blend-
                mode property.</p>
        </body>
        </html>
```

CSS Colors

ackground-color: red;
"color: red;"
order: 2px solid red;"
background-color: rgb(---)
: #------
: rgba(255, 99, 71, 0.5)
#000000 black
FFFFFF white

- □ Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.
- □ You can add color to text, border and background.
- □ There are different method to assign colors. Which are given below:
  - **RGB Value** - In HTML, a color can be specified as an RGB value, using this formula: **rgb(*red, green, blue*).** Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.
  - **HEX Value** - In HTML, a color can be specified using a hexadecimal value in the form: ***#rrggbb*** Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).
  - **HSL Value** - In HTML, a color can be specified using hue, saturation, and lightness (HSL) in the form: **hsl(*hue, saturation, lightness*)** Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue. Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color. Lightness is also a percentage, 0% is black, 50% is neither light or
  - dark, 100% is white.

    RGBA – RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity for a color. An RGBA color value is specified with: **rgba (*red, green, blue, alpha*)** The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):
  - **HSLA** - are an extension of HSL color values with an alpha channel - which specifies the opacity for a color.

: hsla(9, 100%, 64%, 0.4)

CSS Text Properties

- □ **color** – is defined to set color for text.
- □ **text-decoration** – is used to set or remove decorations from text.(Values are: none, underline, overline and line-through)
- □ **text-align** – is used to set horizontal text alignment.(Values are: left, center, right and justify)

- □ **text-transform** – is used to specify uppercase and lowercase letters in text.(Values are: uppercase, lowercase and capitalize)
- □ **text-indent** – is used to specify indentation of the first line of a text.
- □ **line-height** – is used to define space between lines.
- □ **direction** – is used to change the text direction of an element. (Value is *rtl* i.e. right to left)
- □ **word-spacing** – is used to specify the space between the words in a text.
- □ **text-shadow** – add shadow to text.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
        .changeColor{
                color: green;
        }
        .txtAlignment{        right
                text-align: center;
        }                     left
        a{
                text-decoration: none;  /*remove decoration to text*/
        }
        .uppercase{
                text-transform: uppercase;
        }
        .txtIndent{
                text-indent: 50px;
        }
        h2 {    word-spacing : 5PX ;
                letter-spacing: 7px;  /*Also can use -ve number.*/
        }
        .txtShadow{
                text-shadow: 3px 2px red;
        }
</style>
</head>
<body>
        <h1 class="changeColor">This is heading 1</h1>
        <h1 class="txtAlignment">You can align as you like</h1>
     <a href="https://www.google.com">Hello</a>
        <p class="uppercase">You Can use other Transform.</p>
        <p class="txtIndent"> It start text line after 50px indentation.</p> <h2>This is heading 2</h2>
        <h1 class="txtShadow">Text-shadow effect</h1>
</body>
</html>
```

direction : rtl ;
text-transform : upperca

This is body

You can align

YOU CAN USE

it starts

This is heading 2

Text with Shadow

CSS max-width property

- □ The max-width is used to set the maximum width of an element.

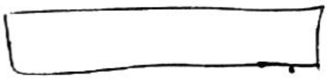
□ The max-width can be specified in *length values*, like px, cm, etc., or in percent (%) of the containing block, or set to none (this is default. Means that there is no maximum width).

□ The max-width property handle the browser if the content of element is larger then browser.

Example:

```
<html>
<head>
<style>
div {
  max-width: 500px;
                         height: 100px;
                       background-color: powderblue;
}
</style>
</head>
<body>
<h2>Set the max-width of an element</h2>
<div></div>
<p>Resize the browser window to see the effect.</p>
</body>
</html>
```

set the max



Resize

CSS Layout – The position Property

□ specifies the type of positioning method used for an element (*static*, *relative*, *fixed*, *absolute* or *sticky*).

□ position ***static*** – this is default value .

□ position ***relative*** – relative to its normal position.

□ position ***fixed*** – is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled.

□ position ***absolute*** – is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

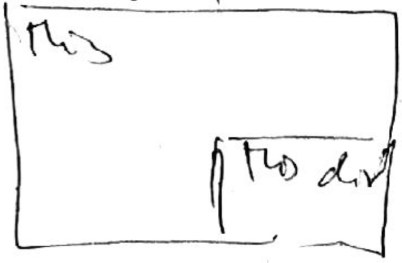
□ position ***sticky*** – is positioned based on the user's scroll position .A sticky element toggles between relative and fixed , depending on the scroll position.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.shift{
                        top: 100px;
                        padding-top: 25px;
}
div.relative {
                        position: relative;
                        width: 400px;
                        height: 200px;
                        border: 3px solid #73AD21;
}
div.absolute {
```

This is body
Because this page lay

position: absolute
An element



List

ul{ list-style-type: square;
}
ol{ list-style-type:
lower-alpha
}

```
                    position: absolute;
                    top: 80px;
                    right: 0px;
                    width: 200px;
                    height: 100px;
                    border: 3px solid #73AD21;
}
img {
                    position: absolute;
                    left: 0px;
                    top: 0px;
                    z-index: -1;
}
</style>
</head>
<body>
        <h1>This is a heading</h1>
        <img src="bas2.jpg" width="300" height="140">
        <p>Because the image has a z-index of -1, it will be placed behind the text.</p> <div
        class="shift"><h2>position: absolute;</h2>
        <p>An element with position: absolute; is positioned relative to the nearest positioned
        ancestor (instead of positioned relative to the viewport, like fixed):</p>
        <div class="relative">This div element has position: relative;
        <div class="absolute">This div element has position: absolute;</div>
</div>
</div>
</body>
</html>
```

Note: Do yourself remaining properties , including <*div*> and <*span*>.

Font

font family is used to change the face of font
font style is used to make font italic or oblique
font variant is used to create a small caps effects
font size is used to set the size of text
font weight is used to set the weight of font.
google font style use गर्न link गर्नु पर्छ।

Link
link: unvisited hyperlink
visted: visited hyperlink
hover: user mouse pointed over
active: currently clicking

<body>
<a href="demo.php"
target="_blank">
This is page </a>

a:link{color: red;}
a:visited{color: green}
a:hover{color: yellow;}
a:active{color blue;}

#font{ font-family: Notable, sans-serif;
font-variant: small-caps;
font-size: 2em (1em = 2px)
font-style: italic;
font-weight bold; (100-900)
500