Ex.No:8

Cascading Style Sheets, types of CSS, Selector forms

Write a program to apply different types (or levels of styles or style specification formats) - inline, internal, external styles to HTML elements. (identify selector, property and value).

<u>Description:</u>

Introduction to CSS:

- **CSS** stands for **C**ascading **S**tyle **S**heets
- Styles define **how to display** HTML elements
- Styles were added to HTML 4.0 **to solve a problem**
- **External Style Sheets** can save a lot of work
- External Style Sheets are stored in **CSS files**

Types of CSS:

- There are three ways of inserting a style sheet:

1. Inline styles
2. Internal style sheet
3. External style sheet

Inline CSS:

- **Inline sheets** can be used to format **only one tag** at a time
- The **inline** cascading style sheet is a kind of style sheet which the styles can be applied to **html tags** only.
- Using inline sheets, we can apply uniform style on tags for the whole document.
- **Disadvantage:** Inline sheet is not much suitable for web page designing because the actual contents of web page are mixed with the presentation.

  <u>Syntax:</u>
- <Tag style="property : value " >

Internal CSS:

- **Advantage** of Internal style sheet comparing with inline sheets, at a time **several** tags can be formatted with **internal** sheets, where as in **inline sheets only one tag** at a time can be formatted.
- **Disadvantage** : when we want to apply style to more than one document at a time then internal sheet of no use.

  Syntax:
- <head>
- <style type="text/css">
- Tagname{
- Tagproperties;
- }
- </style>
- </head>

External CSS:

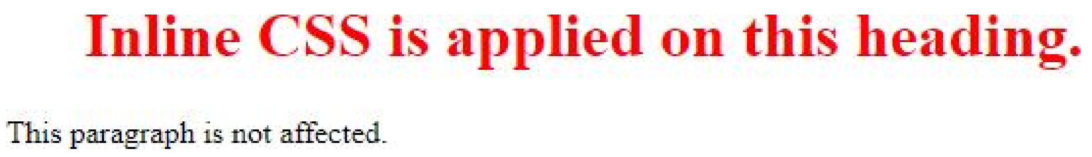
- When we want to apply style to more than one document at a time then external sheets are used.
- Total style elements are defined in a separate document and this document is added to required web page.
- By using this, we can use this style sheets in different web pages. So we can achieve **reusability** by using external sheets.
- The document where all the style formats are placed , should have extension **.css**
- This page can be called in the web page by using **LINK** tag.
  Syntax:
- <link rel=”stylesheet” type=”text/css’’ href=”sample.css”>
- **rel:** Specifies relationship between documents.
- **type:** indicates which type we are including.
- **href**: indicates style sheet document address.

Program:

Inline CSS:

```
<!DOCTYPE html>
<html>
<body>
<h1 style="color:red;margin-left:40px;">Inline CSS is applied on this heading.</h1>
<p>This paragraph is not affected.</p>
</body>
</html>
```

Output:



Inline CSS is applied on this heading.

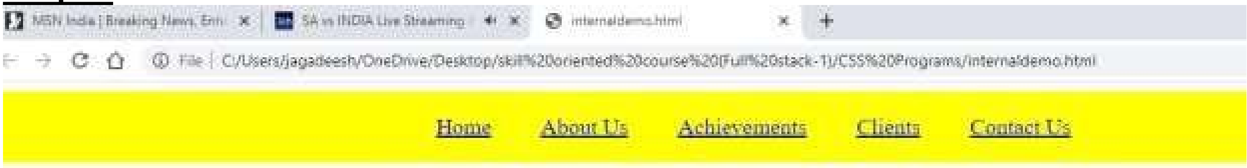
This paragraph is not affected.

Internal CSS:

```
<html>
<head>
<style>
.nav{
background-color:yellow;
list-syle-type:none;
text-align:center;
margin:0;
padding:0;
}
.nav li{
```

```
display:inline-block;
font-size:20px;
padding:20px;
}
</style>
</head>
<body>
<ul class="nav">
<li><a href="#home">Home</a></li>
<li><a href="#about us">About Us</a></li>
<li><a href="#achievements">Achievements</a></li>
<li><a href="#clients">Clients</a></li>
<li><a href="#contact us">Contact Us</a></li>
</ul>
</body>
</html>
```

Output:


MSN India | Breaking News, Ent
SA vs INDIA Live Streaming
internaldemo.html
File | C:/Users/jagadeesh/OneDrive/Desktop/skill%20oriented%20course%20(Full%20stack-1)/CSS%20Programs/internaldemo.html
Home
About Us
Achievements
Clients
Contact Us

External CSS:

Style.css

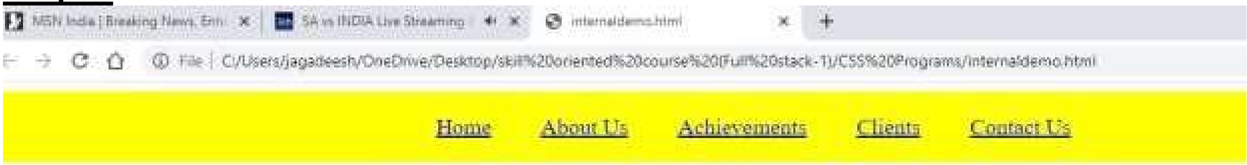
```
.nav{
background-color:yellow;
list-syle-type:none;
text-align:center;
margin:0;
padding:0;
}
.nav
li{ display:inline-
block;font-size:20px;
padding:20px;
}
```

External.html

```
<html>
<head>
<link rel=”stylesheet” type=”text/css” href=”style.css”>
</head>
<body>
<ul class="nav">
<li><a href="#home">Home</a></li>
<li><a href="#about us">About Us</a></li>
<li><a href="#achievements">Achievements</a></li>
```

```
<li><a href="#clients">Clients</a></li>
<li><a href="#contact us">Contact Us</a></li>
</ul>
</body>
</html>
```

Output:


MSN India | Breaking News, Ent
SA vs INDIA Live Streaming
internaldemo.html
File
C:/Users/jagadeesh/OneDrive/Desktop/skill%20oriented%20course%20(Full%20stack-1)/CSS%20Programs/internaldemo.html
Home
About Us
Achievements
Clients
Contact Us

Ex.No: 9

Selectors in CSS

Write a program to apply different types of selector forms

i. Simple selector (element, id, class, group, universal)
ii. Combinator selector (descendant, child, adjacentsibling, general sibling)
iii. Pseudo-class selector
iv. Pseudo-element selector
v. Attribute selector

Description:

Selectors are used to apply special effects.
Types of selectors are:
1. Simple selector
2. Class selector
3. Generic Selector
4. Universal selector
5. Id Selector

Simple Selector:

- ➢ The simple selector form is a single element to which the property and value is applied.
- ➢ Syntax:
- ➢ <head>
- ➢ <title>Simle selectors</title>
- ➢ <style type="text/css">
- ➢ tagname{
- ➢ properrties;
- ➢ }
- ➢ </style>
- ➢ </head>

Class Selector:

- ➢ Using class selector we can apply different styles to same element.
- ➢ Syntax: <head>
- ➢ <style type="text/css">
- ➢ Tagname.classname{
- ➢ Properties;
- ➢ }
- ➢ </style>
- ➢ </head>
- ➢ <tagname class="classname">….</tagname>

Generic Selector:

- ➢ The class can be defined in the generalized form.
- ➢ So that the particular class can be applied to any tag.
- ➢ Syntax: <head>
- ➢ <style type="text/css">

- .classname{
- Properties;
- }
- </style>
- </head>
- <tagname class=“classname”>….</tagname>

Universal Selector:

- This selector can be applied to all the elements in the document.
- This selector is denoted by * symbol.
- Syntax:<head>
-   <title>Universal selectors</title>
-     <style type="text/css">
-         *
-           {
-             properties;
-           }

Id Slector:

- The id selector is used to specify a style for a single, unique element.
- The id selector uses the id attribute of the HTML element, and is defined with a "#".
- Do **not** start an ID name with a number
- **Syntax:**
-         #para1
  ```
  {
   text-align:center;
   color:red;
  }
  ```
- <tagname id=“idname”> ……</tagname>

Attribute Slector:

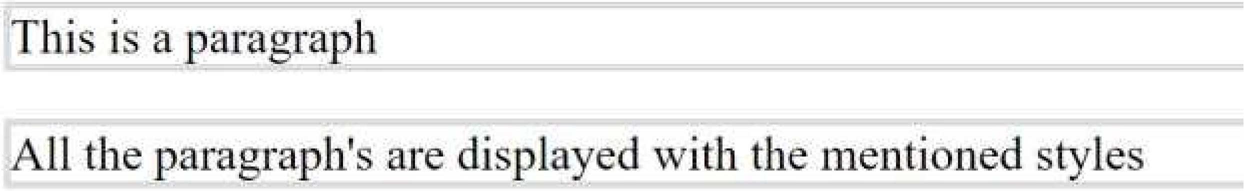
- The [attribute] selector is used to select elements with a specified attribute.
- The [attribute="value"] selector is used to select elements with a specified attribute and value.
- The [attribute~="value"] selector is used to select elements with an attribute value containing a specified word.
- The [attribute|="value"] selector is used to select elements with the specified attribute starting with the specified value.
- The [attribute^="value"] selector is used to select elements whose attribute value begins with a specified value.
- The [attribute$="value"] selector is used to select elements whose attribute value ends with a specified value.
- The [attribute*="value"] selector is used to select elements whose attribute value contains a specified value.

Program:

Simple Slector Demo:

```
<html>
<head>
<style type="text/css">
p{
font-style:Tahoma;
font-size:40px;
border:5px double #ccc;
}
</style>
<body>
<p>This is a paragraph</p<br>
<p>All the paragraph's are displayed with the mentioned styles</p><br>
</body>
</html>
```

Output:



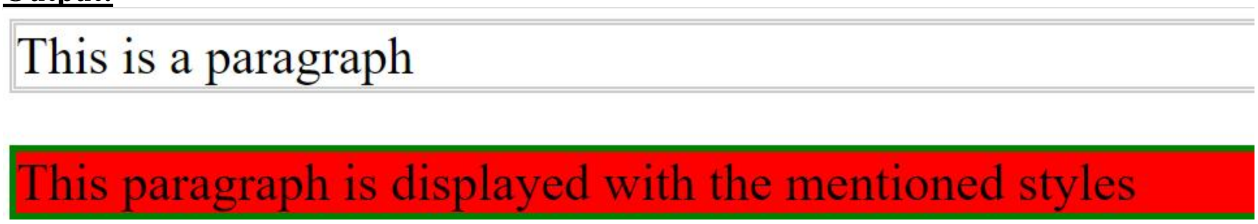
This is a paragraph

All the paragraph's are displayed with the mentioned styles

Class Selctor Demo:

```
<html>
<head>
<style type="text/css">
p{
font-style:Tahoma;
font-size:40px;
border:5px double #ccc;
}
p.redpara{ color:none;
background-color:red;
border:5px solid green;
}
</style>
<body>
<p>This is a paragraph</p<br>
<p class="redpara">This paragraph is displayed with the mentioned styles</p><br>
</body>
</html>
```

Output:



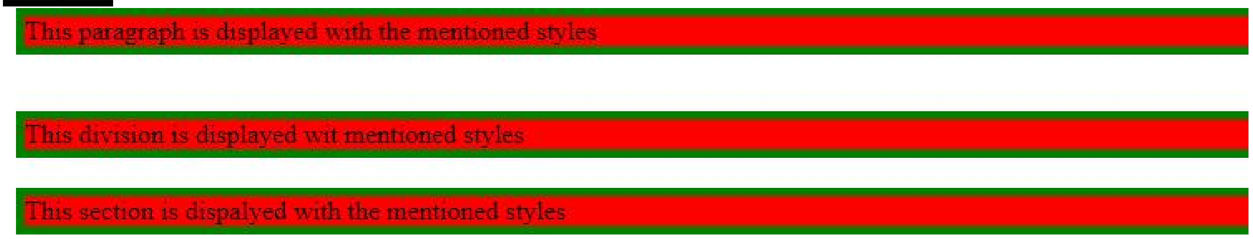
This is a paragraph

This paragraph is displayed with the mentioned styles

Id Selector Demo:

```
<html>
<head>
<style type="text/css">
#red{
color:none;
background-color:red;
border:5px solid green;
}
</style>
<body>
<p id="red">This paragraph is displayed with the mentioned styles</p><br>
<div id="red">This division is displayed wit mentioned styles</div><br>
<section id="red">This section is dispalyed with the mentioned styles</section><br>
</body>
</html>
```

Output:



This paragraph is displayed with the mentioned styles

This division is displayed wit mentioned styles

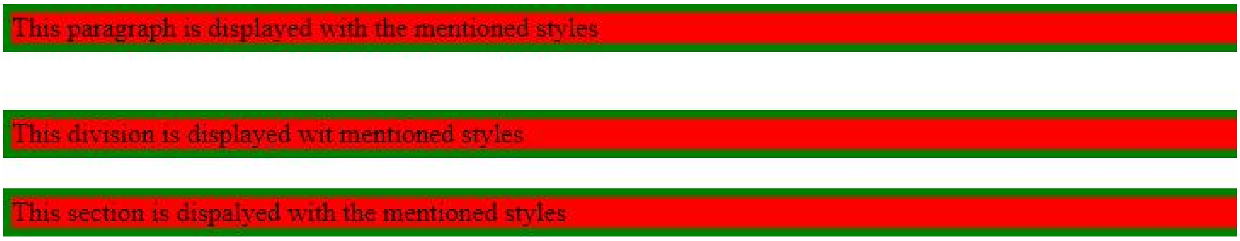
This section is dispalyed with the mentioned styles

Generic Slector Demo:

```
<html>
<head>
<style type="text/css">
.red{ color:
none;
background-color:red;
border:5px solid green;
}
</style>
<body>
<p class="red">This paragraph is displayed with the mentioned styles</p><br>
<div class="red">This division is displayed wit mentioned styles</div><br>
```

```
<section class="red">This section is dispalyed with the mentioned styles</section><br>
</body>
</html>
```

Output:



This paragraph is displayed with the mentioned styles

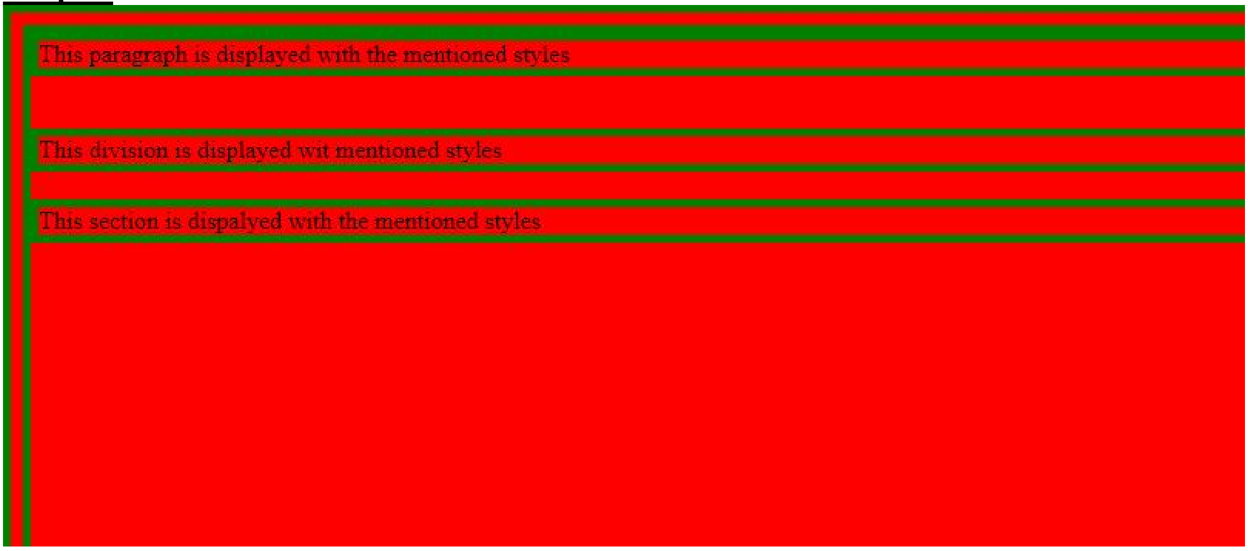
This division is displayed wit mentioned styles

This section is dispalyed with the mentioned styles

Universal Slector Demo:

```
<html>
<head>
<style type="text/css">
*{ color:no
ne;
background-color:red;
border:5px solid green;
}
</style>
<body>
<p >This paragraph is displayed with the mentioned styles</p><br>
<div >This division is displayed wit mentioned styles</div><br>
<section >This section is dispalyed with the mentioned styles</section><br>
</body>
</html>
```

Output:



This paragraph is displayed with the mentioned styles

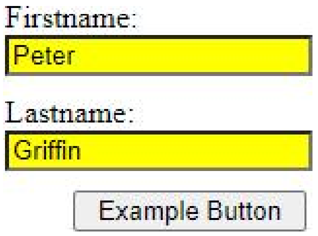
This division is displayed wit mentioned styles

This section is dispalyed with the mentioned styles

Attribute Selector Demo:

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=text]
{width: 150px;
display: block;
  margin-bottom: 10px;
  background-color: yellow;
}

input[type=button]
  {width: 120px;
  margin-left: 35px;
  display: block;
}
</style>
</head>
<body>

<form name="input" action="" method="get">
  Firstname:<input type="text" name="Name" value="Peter" size="20">
  Lastname:<input type="text" name="Name" value="Griffin" size="20">
  <input type="button" value="Example Button">
</form>

</body>
</html>
```

<u>Output:</u>



Firstname: Peter

Lastname: Griffin

Ex.No:10

Color, Background and CSS Box Model

a) Write a program to demonstrate the various ways you can reference a color in CSS.

Description:

Color Properties in CSS:

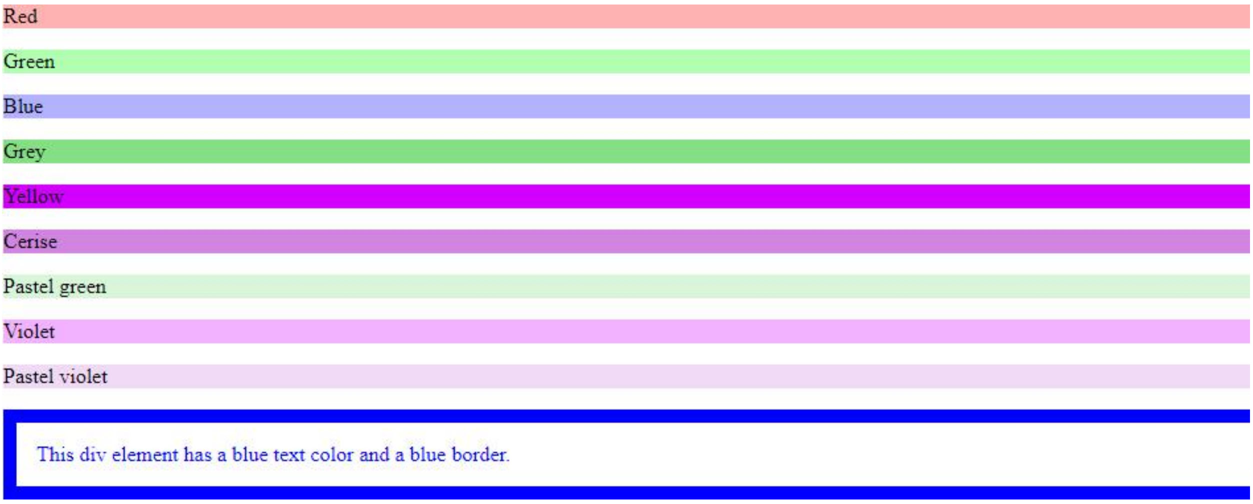
- Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.
- Background-color:colorname/rgb/hex;(applied for background)
- Color:colorname/rgb/hex;(applied for text)
- Border:width in pixels type colorname/rgb/hex (applied for borders)

Program:

```
<!DOCTYPE html>
<html>
<head>
<style>
#p1 {background-color:rgba(255,0,0,0.3);}
#p2 {background-color:rgba(0,255,0,0.3);}
#p3 {background-color:rgba(0,0,255,0.3);}
#p4 {background-color:hsl(120,60%,70%);}
#p5 {background-color:hsl(290,100%,50%);}
#p6 {background-color:hsl(290,60%,70%);}
#p7 {background-color:hsla(120,60%,70%,0.3);}
#p8 {background-color:hsla(290,100%,50%,0.3);}
#p9 {background-color:hsla(290,60%,70%,0.3);}
div {
  color: blue;
  border: 10px solid currentcolor;
  padding: 15px;
}
</style>
</head>
<body>


<p id="p1">Red</p>
<p id="p2">Green</p>
<p id="p3">Blue</p>
<p id="p4">Grey</p>
<p id="p5">Yellow</p>
<p id="p6">Cerise</p>
<p id="p7">Pastel green</p>
<p id="p8">Violet</p>
<p id="p9">Pastel violet</p>
<div>
This div element has a blue text color and a blue border.
</div>
```

```
</body>
</html>
```

Output:



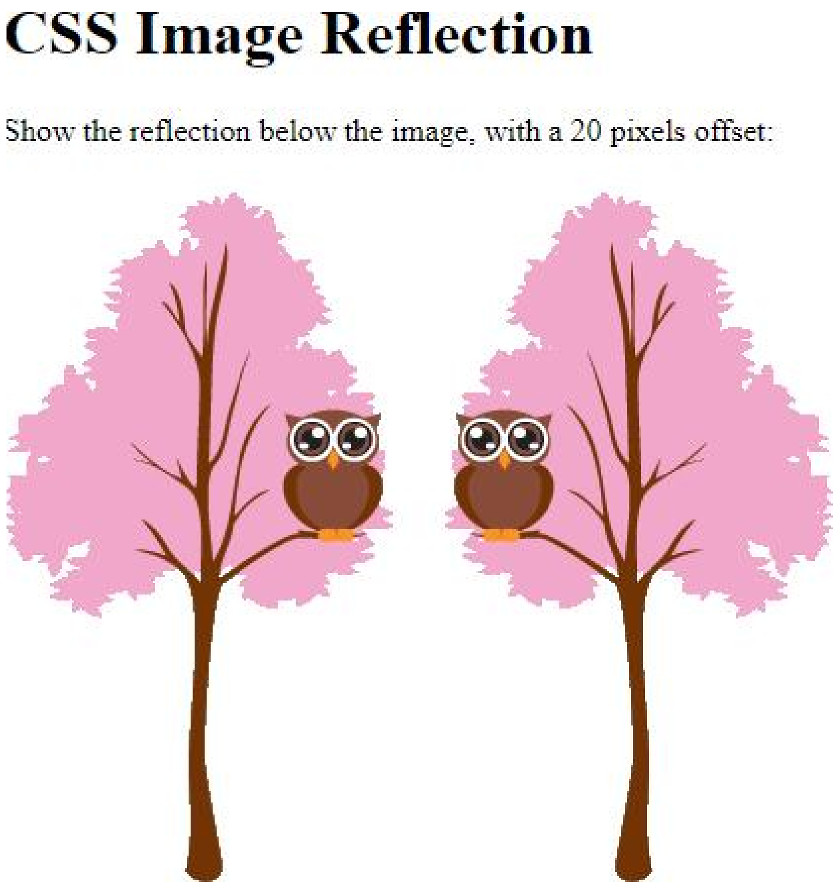
```
Red
Green
Blue
Grey
Yellow
Cerise
Pastel green
Violet
Pastel violet
This div element has a blue text color and a blue border.
```

b) Write a CSS rule that places a background image halfway down the page, tilting it horizontally. The image should remain in place when the user scrolls up or down.

Program:

```
!DOCTYPE html>
<html>
<head>
<style>
img {
  -webkit-box-reflect: right 20px;
}
</style>
</head>
<body>

<h1>CSS Image Reflection</h1>
<p>Show the reflection below the image, with a 20 pixels offset:</p>
<img src="./images/img_tree.gif">

</body>
</html>
```

Output:



CSS Image Reflection

Show the reflection below the image, with a 20 pixels offset:

Ex.No:11

CSS Box Model, CSS Font and Image Proerpties

a) Write a program, to explain the importance of CSS Box model using

i. Content

ii. Border

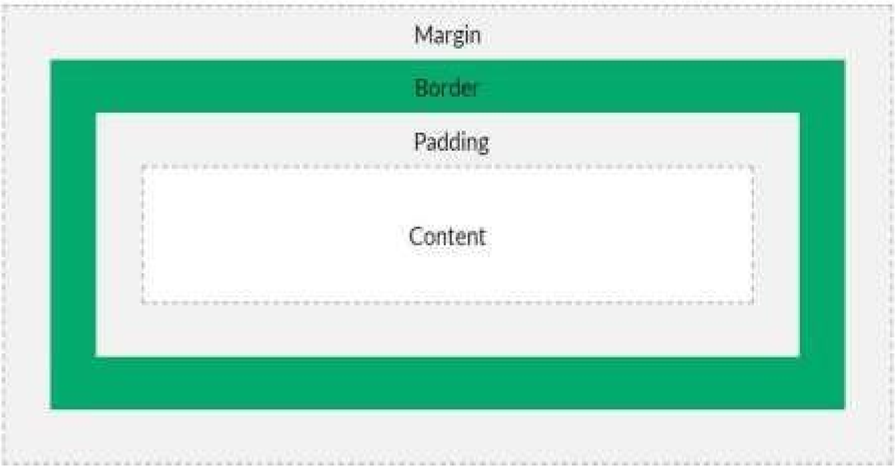
iii. Margin

iv. padding

Description:

Box Model:

- 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:



- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent
- The box model allows us to add a border around elements, and to define space between elements.
- div {

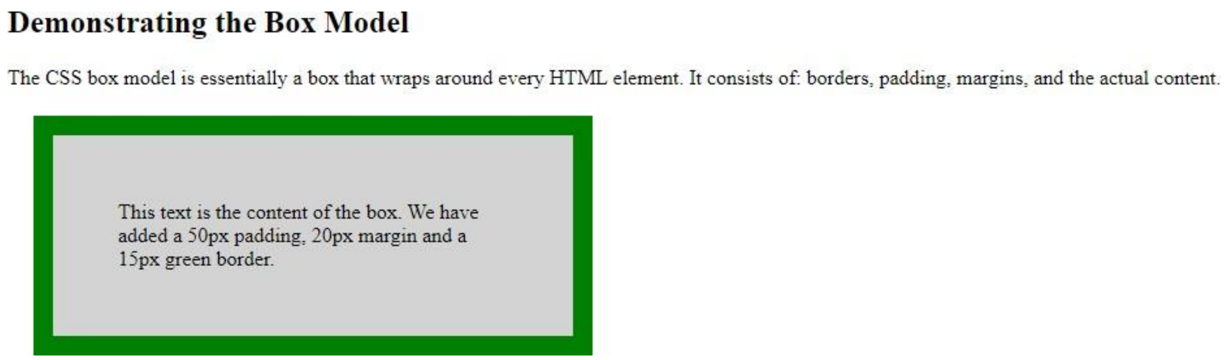
```
  width: 300px;
  border: 15px solid green;
```

```
            padding: 50px;
            margin: 20px;
          }
```

Program:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>

<h2>Demonstrating the Box Model</h2>

<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>

<div>This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. </div>

</body>
</html>
```

Output:



Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border.

b) Write a program using the following terms related to CSS font and text:
i. font-size
ii. font-weight
iii. font-style
iv. text-decoration
v. text-transformation
vi. text-alignment

Description:

Font Properties in CSS:

- Using a font that is easy to read is important.
- The font adds value to your text. It is also important to choose the correct color and text size for the font.
- In CSS there are five generic font families:
- **Serif** fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
- **Sans-serif** fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
- **Monospace** fonts - here all the letters have the same fixed width. They create a mechanical look.
- **Cursive** fonts imitate human handwriting.
- **Fantasy** fonts are decorative/playful fonts.
- All the different font names belong to one of the generic font families
- In CSS, we use the font-family property to specify the font of a text.
- The font-family property should hold several font names as a "fallback" system, to ensure maximum compatibility between browsers/operating systems.
- Start with the font you want, and end with a generic family (to let the browser pick a similar font in the generic family, if no other fonts are available).
- The font names should be separated with comma.
- font-family: *family-name*|*generic-family*|initial|inherit;
- Font-family:Georgia|Palatino Linotype|Book Antiqua|Times New Roman|Arial|Helvetica|Arial Black|Impact|Lucida Sans Unicode|Tahoma|Verdana|Courier New|Lucida Console|initial
- In a small-caps font, all lowercase letters are converted to uppercase letters.
- However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text.
- The font-variant property specifies whether or not a text should be displayed in a small-caps font.
- font-variant: normal|small-caps|initial|inherit;
- The font-variant-caps property controls the usage of alternate glyphs for capital letters.
- font-variant-caps: normal|small-caps|all-small-caps|petite-caps|all-petite-caps|unicase|titling-caps|initial|inherit|unset;
- The font-size property sets the size of a font.
- font-size:medium|xx-small|x-small|small|large|x-large|xx-large|smaller|larger|*length*|%|initial|inherit;
- The font-style property specifies the font style for a text.
- font-style: normal|italic|oblique|initial|inherit;
- The font-weight property sets how thick or thin characters in text should be displayed.
- font-weight: normal|bold|bolder|lighter|*number*|initial|inherit;

Text Properties in CSS:

- CSS has a lot of properties for formatting text
- The color property is used to set the color of the text. The color is specified by:
- a color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"
- Look at CSS Color Values for a complete list of possible color values.
- The default text color for a page is defined in the body selector.
- Text-color/color:colorname/rgb/hex/hsl|initial|inherit;
- The text-align property is used to set the horizontal alignment of a text.
- A text can be left or right aligned, centered, or justified.
- When the text-align property is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers)
- Text-align: left|right|center|justify|initial|inherit;
- The text-decoration property is used to set or remove decorations from text.
- The value text-decoration: none; is often used to remove underlines from links
- The other text-decoration values are used to decorate text
- Text-decoration: *text-decoration-line text-decoration-color text-decoration-style*|initial|inherit;
- The text-decoration-line property sets the kind of text decoration to use (like underline, overline, line-through).
- Text-decoration-line:none|underline|overline|line-through|initial|inherit;
- The text-decoration-style property sets the style of the text decoration (like solid, wavy, dotted, dashed, double).

text-decoration-style: solid|double|dotted|dashed|wavy|initial|inherit;

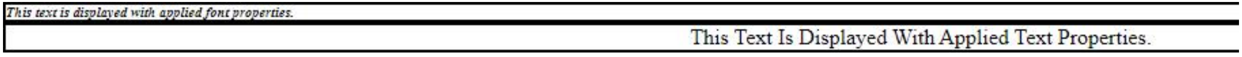
- The text-transform property is used to specify uppercase and lowercase letters in a text.
- It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word
- Text-transform:none|capitalize|uppercase|lowercase|initial|inherit;
- The text-indent property is used to specify the indentation of the first line of a text
- Text-indent:px|pts|cm|em|initial|inherit;
- The direction property specifies the text direction/writing direction within a block-level element.
- direction: ltr|rtl|initial|inherit;
- The text-overflow property specifies how overflowed content that is not displayed should be signaled to the user. It can be clipped, display an ellipsis (...), or display a custom string.
- Both of the following properties are required for text-overflow:
- text-overflow: clip|ellipsis|*string*|initial|inherit;
- The letter-spacing property is used to specify the space between the characters in a text.
- Letter spacing can be given either a positive or –ve value.
- Letter-spacing:+-normal|px|pts|cm|em|initial|inherit;
- The line-height property is used to specify the space between lines as floating point value
- Line-height:normal|value|length|initial|inherit;
- The word-spacing property is used to specify the space between the words in a text
- word-spacing: normal|px|pts|cm|em|inherit;
- The white-space property specifies how white-space inside an element is handled.
- White-space: normal|nowrap|pre|pre-line|pre-wrap|initial|inherit;
- The unicode-bidi property is used together with the direction property to set or return whether the text should be overridden to support multiple languages in the same document.
- unicode-bidi: normal|embed|bidi-override|initial|inherit;

- The vertical-align property sets the vertical alignment of an element.
- vertical-align: baseline|*length*|sub|super|top|text-top|middle|bottom|text-bottom|initial|inherit;
- The text-shadow property adds shadow to text.
- In its simplest use, you only specify the horizontal shadow (2px) and the vertical shadow (2px)
- text-shadow: *h-shadow v-shadow blur-radius color*|none|initial|inherit;

Program:

```
<html>
<head>
<style>
.font{
font-family:Times;
font-style:oblique;
font-weight:900;
font-size:65%;
border:2px solid;
}
.text{
text-align:center;
text-decoration:overline;
text-transform:capitalize;
border:2px solid;
}
</style>
</head>
<body>
<section class="font">
This text is displayed with applied font properties.
</section>
<div class="text">
This text is displayed with applied text properties.
</div>
</body>
</html>
```

Output:



This text is displayed with applied font properties.

This Text Is Displayed With Applied Text Properties.

Animations

Expr:12

a) Write a CSS program, to apply 2D transformations in a web page.

Description:

2D Transforms:

The CSS 2D transforms are used to re-change the structure of the element as translate, rotate, scale and skew etc.

Following is a list of 2D transforms methods:

- **translate(x,y):** It is used to transform the element along X-axis and Y-axis.
- **translateX(n):** It is used to transform the element along X-axis.
- **translateY(n):** It is used to transform the element along Y-axis.
- **rotate():** It is used to rotate the element on the basis of an angle.
- **scale(x,y):** It is used to change the width and height of an element.
- **scaleX(n):** It is used to change the width of an element.
- **scaleY(n):** It is used to change the height of an element.
- **skewX():** It specifies the skew transforms along with X-axis.
- **skewY():**It specifies the skew transforms along with Y-axis.
- **matrix():** It specifies matrix transforms.

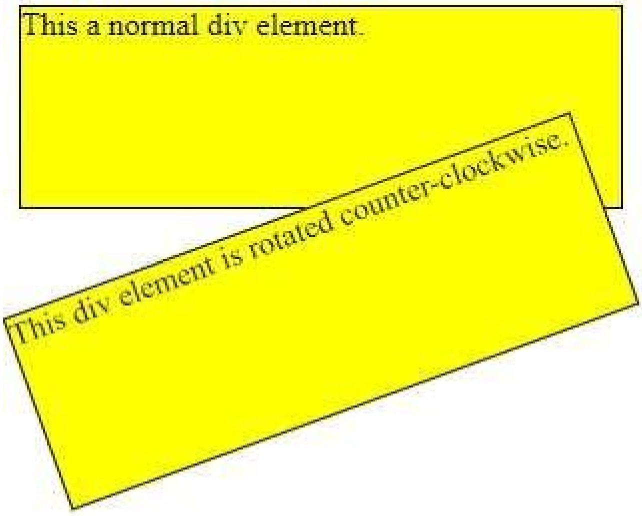
Program1

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 300px;
  height: 100px;
  background-color: yellow;
  border: 1px solid black;
}

div#myDiv
  { transform:
  scaleX(1);transform:
  scaleY(1);transform:
  skewX(1);transform:
  skewY(1);
```

```
transform: rotate(-20deg);
```

```
</style>
</head>
<body>


<div>
This a normal div element.
</div>

<div id="myDiv">
This div element is rotated counter-clockwise.
</div>

</body>
</html>
```

<u>Output:</u>



This a normal div element.

This div element is rotated counter-clockwise.

b) Write a CSS program, to apply 3D transformations in a web page.

<u>Descrription:</u>

3D Tranforms:

The CSS 3D transforms facilitates you to move an element to X-axis, Y-axis and Z-axis. Following is a list of 3D transforms methods:

Function	Description
matrix3D(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)	It specifies a 3D transformation, using a 4x4 matrix of 16 values.
translate3D(x,y,z)	It specifies a 3D translation.
translateX(x)	It specifies 3D translation, using only the value for the X-axis.

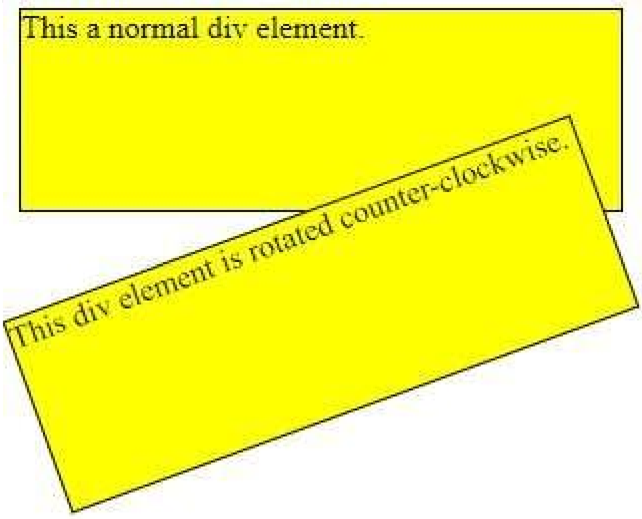
translateY(y)	It specifies 3D translation, using only the value for the Y-axis.
translateZ(z)	It specifies 3D translation, using only the value for the Z-axis.
scale3D(x,y,z)	It specifies 3D scale transformation
scaleX(x)	It specifies 3D scale transformation by giving a value for the X-axis.
scaley(y)	It specifies 3D scale transformation by giving a value for the Y-axis.
scaleZ(z)	It specifies 3D scale transformation by giving a value for the Z-axis.
rotate3D(X,Y,Z,angle)	It specifies 3D rotation along with X-axis, Y-axis and Z-axis.
rotateX(angle)	It specifies 3D rotation along with X-axis.
rotateY(angle)	It specifies 3D rotation along with Y-axis.
rotateZ(angle)	It specifies 3D rotation along with Z-axis.
perspective(n)	It specifies a perspective view for a 3D transformed element.

Program2

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 300px;
  height: 100px;
  background-color: yellow;
  border: 1px solid black;
}

div#myDiv
  { transform:
  scaleX(1);transform:
  scaleY(1);transform:
  scaleZ(1);
  transform: rotateZ(-20deg);
}
</style>
</head>
```

```
<body>
<div>
This a normal div element.
</div>

<div id="myDiv">
This div element is rotated counter-clockwise.
</div>

</body>
</html>
```

Output:



```
This a normal div element.
This div element is rotated counter-clockwise.
```

c) Write a CSS program, to apply Animations in a web page.

Description:

Animations:

An animation makes an element change *gradually* from one style to another. You can add as many as properties you want to add. You can also specify the changes in percentage.0% specify the start of the animation and 100% specify its completion.

How CSS animation works

When the animation is created in the @keyframe rule, it must be bound with selector; otherwise the animation will have no effect.

The animation could be bound to the selector by specifying at least these two properties:

- The name of the animation
- The duration of the animation

CSS animation properties

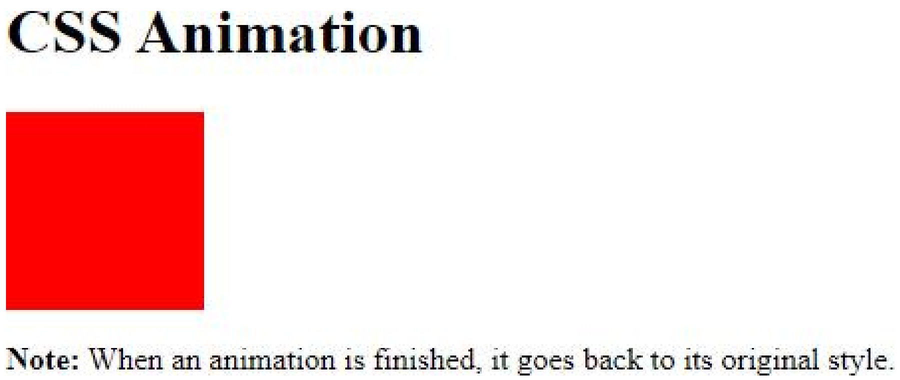
Property	Description
@keyframes	It is used to specify the animation.
Animation	This is a shorthand property, used for setting all the properties, except the animation-play-state and the animation-fill- mode property.
animation-delay	It specifies when the animation will start.
animation-direction	It specifies if or not the animation should play in reserve on alternate cycle.
animation-duration	It specifies the time duration taken by the animation to complete one cycle.
animation-fill-mode	it specifies the static style of the element. (when the animation is not playing)
animation-iteration-count	It specifies the number of times the animation should be played.
animation-play-state	It specifies if the animation is running or paused.
animation-name	It specifies the name of @keyframes animation.
animation-timing-function	It specifies the speed curve of the animation.

Program3

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}

@keyframes example {
  0%  {background-color: red;}
  25%  {background-color: yellow;}
  50%  {background-color: blue;}
  100% {background-color: green;}
}
</style>
</head>
<body>
```

```
<h1>CSS Animation</h1>

<div></div>

<p><b>Note:</b> When an animation is finished, it goes back to its original style.</p>

</body>
</html>
```

Output



CSS Animation

Note: When an animation is finished, it goes back to its original style.