



MODULE 1

Cascading Style Sheets (CSS3)

CSS3: Levels of style sheets; Style specification formats; Selector forms; Property value forms; Font properties; List properties; Color; Alignment of text; Background images, Conflict Resolution, CSS Box Model .CSS3 features: Box Shadow, Opacity, Rounded corners, Attribute selector

CSS: Cascading Style Sheets

Q) What is CSS?

- ◆ CSS stands for **Cascading Style Sheets**
- ◆ CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- ◆ CSS saves a lot of work. It can control the layout of multiple web pages all at once
- ◆ External style sheets are stored in CSS files

Q) Why Use CSS?

- ♦ CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS Syntax

- ◆ A CSS rule consists of a **selector** and a **declaration** block.
- ◆ In this example all `<p>` elements will be center-aligned, with a **red text color**:

Example:

```
p {  
color: red;  
text-align: center;  
}
```

Example Explained

`p` is a **selector** in CSS (it points to the HTML element you want to style: `<p>`).
`color` is a **property**, and `red` is the **property value**
`text-align` is a **property**, and `center` is the **property value**

□ CSS code within HTML Code

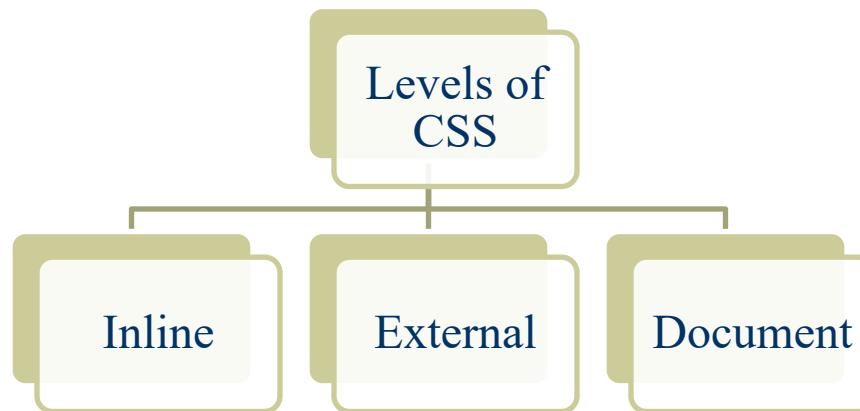
```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: lightblue;
}
h1 {
    color: white;
    text-align: center;
}
p {
    font-family: verdana;
    font-size: 20px;
}
</style>
</head>
<body>
<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

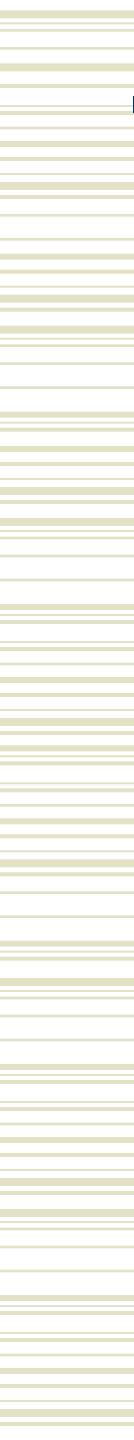
My First CSS Example

This is a paragraph.

□ Levels of style sheets

The *Cascading Style Sheets* is defined at **three** different levels to specify the style of a Document, from **lowest** level to **highest** level are :





What are the **three ways** of specifying styles?



- ◆ CSS can be added to HTML documents in 3 ways: **Inline - by using the style attribute inside HTML elements.**
- ◆ **Internal - by using a <style> element in the <head> section.**
- ◆ **External - by using a <link> element to link to an external CSS file.**

Inline: Inline style specifications appear within the **opening tag** and apply **only to the content** of that tag

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1 {color: blue;}
p {color: red;}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

CSS Introduction

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>
            CSSHelloWorld.html
        </title>
        <link rel="stylesheet" type="text/css" href="style1.css"
              title="Style 1" />
        <link rel="alternate stylesheet" type="text/css" href="style2.css"
              title="Style 2" />
    </head>
    <body>
        <p>
            Hello World!
        </p>
    </body>
</html>
```

Link element associates style sheet with doc.

External

External CSS

- ◆ An external style sheet is used to define the style for many HTML pages.
- ◆ To use an external style sheet, add a link to it in the <head> section of each HTML page:
- ◆ The external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension.
- ◆ Here is what the "styles.css" file looks like:

Example

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

styles.css:

```
body {
  backgroundcolor: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

OUTPUT(external)

This is a heading.

This is a paragraph.

CSS External

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>
            CSSHelloWorld.html
        </title>
        <link rel="stylesheet" type="text/css" href="style1.css"
              title="Style 1" />
        <link rel="alternate stylesheet" type="text/css" href="style2.css"
              title="Style 2" />
    </head>
    <body>
        <p>
            Hello World!
        </p>
    </body>
</html>
```

type attribute specifies style language used

CSS external

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>
            CSSHelloWorld.html      href attribute provides style sheet URL
        </title>
        <link rel="stylesheet" type="text/css" href="style1.css"
              title="Style 1" />
        <link rel="alternate stylesheet" type="text/css" href="style2.css"
              title="Style 2" />
    </head>
    <body>
        <p>
            Hello World!
        </p>
    </body>
</html>
```

CSS

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>
            CSSHelloWorld.html
        </title>
        <link rel="stylesheet" type="text/css" href="style1.css"
              title="Style 1" />
        <link rel="alternate stylesheet" type="text/css" href="style2.css"
              title="Style 2" />
    </head>
    <body>
        <p>
            Hello World!
        </p>
    </body>
</html>
```

title attribute provides style sheet name

CSS

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>
            CSSHelloWorld.html
        </title>
        <link rel="stylesheet" type="text/css" href="style1.css"
              title="Style 1" />
        <link rel="alternate stylesheet" type="text/css" href="style2.css"
              title="Style 2" /> Alternative, user selectable style sheets
    </head> can be specified
    <body>
        <p>
            Hello World!
        </p>
    </body>
</html>
```

Document level style sheets : apply to the whole body of a document. Document level style sheets apply to the whole body of a document.

```
<!DOCTYPE html>
<html>
<head>
<title> Sample CSS </title>
<style>
h1{
    font-family : Lucida Handwriting;
    font-size   : 50pt ;
    color       : red ;
}
</style>
</head>
<h1> Dayananda Sagar University</h1>
</html>
```

◆ Output

Dayananda Sagar University

□ Style Specification Formats

- ◆ Format depends on the level of the style sheet
 - ◆ Format for Inline Style sheets
 - ◆ Inline: Style sheet appears as the values of the style attribute of a tag
 - ◆ General form:

```
style = "property_1: value_1; property_2: value_2; ...
property_n: value_n; "
```

- ◆ Example:<h1 style="color:red;
font-family:sans-serif">
India is my country</h1>

Document Style Specification

A style is applied to the entire HTML file.

- Use it when you need to modify all instances of particular element (e.g., h1) in a web page.
- Style sheet appears as a list of rules that are the content of a document

General form:

```
<style type="text/css">  
rule list  
</style>
```

□ Style specification formats

▪ Document Style Specification

```
< Style type = "text/css" >  
Rule list  
</Style>
```

Each Style rule ,In a rule has two parts

A selector ,which indicates the tag or tags affected by the rule and a list has same form as the quoted list for inline style sheets, except that it is delimited by braces rather than double quotes , So the form of a style rule is as follows

```
Selector {Property1.value1,Property2.Value2 , Property3.Value3 ,  
.....Propertyn.Valuen;}
```

Selector forms

Simple Selector Forms

In case of Simple selector , a tag is used .

If the properties of the tag are **changed** ,then it reflects at all the places when used in the program.

The selector can be any tag .If the new properties for a tag are not mentioned within the rule list , then the browser uses default behavior of a tag.

SELECTOR FORMS

Simple Selector Forms

```
<!DOCTYPE html>
<html>
<head>
<title> Sample CSS </title>
<style type =“text/css”>
P
{ font-family : Lucida Handwriting;
  font-size    : 50pt;
  color       : ;
}
</style>
</head>
<body>
<p> Dayananda Sagar University</p>
<p> Computer Science and Engineering</p>
</body>
</html>
```

The CSS element Selector:

The element selector selects HTML elements based on the element name. Here, all `<p>` elements on the page will be center-aligned, with a red text color:

```
p {
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```

2. Class Selector : It is possible to give different properties for different elements.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>

</body>
</html>
```



output

This heading will not be affected

This paragraph will be red and center-aligned

- In this example the `<p>` element will be styled according to `class="center"` and to `class="large"`:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
    text-align: center;
    color: red;
}

p.large {
    font-size: 300%;
}
</style>
</head>
<body>

<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>
<p class="center large">This paragraph will be red, center-aligned, and in a large font-size.</p>

</body>
</html>
```

output

This heading will not be affected

This paragraph will be red and center-aligned

This paragraph will be red, center-aligned and in a large font-size

3.Generic selector: Generic selectors **can be applied to any HTML element.** Generic class selectors begin with a period (.).

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>

</body>
</html>
```

Output:

Red and center-aligned heading

Red and center-aligned paragraph.

Id Selector : An Id Selector allows the applications of a style to one specific element.

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>

</body>
</html>
```

Output:

```
Hello World!
This paragraph is not affected by style.
```

Universal Selector : The Universal selector ,denoted by an asterisk (*) ,applies its style to all elements in a document .

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>

<h1>Hello world!</h1>

<p>Every element on the page will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```

Output:

Every element on the page will be affected by the style.

Me too!

And me!

Style Specification Format:

Pseudo Classes : Pseudo class selectors are used if properties are to be changed dynamically.

Example : When mouse movement happens ,in other words hover happens or focus happens .

```
<!DOCTYPE html>
<html>
<head>
<title> Sample CSS </title>
<style type =“text/css”>
Input : focus
{ font-family : lucida Calligraphy ;
  color : purple ;
  font-size :100pt;
}
</style>
</head>

<body>
<form action=“”>
<p>
<label>
NAME:
<input type=“text”/>
</label>
</p>
</form>
</body>
</html>
```

Pseudo-classes:

Output:

NAME:

NAME: **Text**

*PROPERTY VALUE FORMS:

CSS includes 60 different properties in seven categories : fonts ,lists ,alignment of text , margins ,colors , backgrounds and borders .

Property value can appear in variety of forms

- . **Keyword** property values are used when there are only a few possible values and they are predefined.
- . A number value can be either an integer or a sequence of digits with a decimal point and can be preceded by a sign(+ or -)
- . Length values are specified as number values that are followed immediately by a two character abbreviation of a unit name. The possible unit names are px for pixels , in for inches ,cm for centimeters ,mm for millimeters ,pt for points.

1. **Inline styles** (Presence of style in document). An inline style lives within your XHTML document. It is attached directly to the element to be styled. E.g. `<h1 style="color: #fff;">`
2. **IDs (# of ID selectors)** ID is an identifier for your page elements, such as `#div`.
3. **Classes, attributes and pseudo-classes (# of class selectors)**. This group includes `.classes`, `[attributes]` and pseudo-classes such as `:hover`, `:focus` etc.
4. **Elements and pseudo-elements (# of Element (type) selectors)**. Including for instance `:before` and `:after`.

PROPERTY VALUE FORMS:

- .Percentage values are used to provide a measure that is relative to the previously used measure for a property value. Percentage values are numbers that are followed immediately by a percentage sign(%). Percentage values can be signed if preceded by a plus sign .The percentage is added to the previous value. If negative ,the percentage is subtracted .
- .There can be no space between url and left parenthesis .
- .Color property values can be specified as color names , as six digit hexadecimal number or in RGB form. RGB form is just word rgb followed by a parenthesized list of three nos that specify the levels of red,green and blue respectively.

PROPERTY VALUE FORMS:

- .The RGB values can be given either as decimal numbers between 0 to 255 or percentage hexadecimal numbers must be preceded with pound sign (#) as in #43AF00.

FONT PROPERTIES:

.Font families :

- The font family property is used to specify a list of font names . The browser uses a list of font names.
- The browser uses the first font in the list that it supports.
- **Font-family : Arial ,Helvetica, Futura**
- It tells the browser to use font Arial ,if it supports that font , if not it will use Helvetica ,if it supports it. If the browser supports neither Arial nor Helvetica ,it will use Futura . If it can .If browser doesnot support any of the specified fonts ,it will use an alternative of its choosing.
- If a font name has more than one word ,the whole name should be delimited by single quote
- **Font-family : 'Times new roman'**

FONT PROPERTIES:

.Font Sizes :

- The font-size property does what it name implies , For example ,the following property specification sets the font size for text to 10 points
- **Font-size : 10pt**
- Many relative font size values are defined including **xx-small ,x-small, small, medium, large, ,x-large , xx-large.**
- In addition smaller or larger can be specified further more the values can be a percentage relative to the current font size.

.Font Styles :

- The font-style property is most commonly used to specify italic as in
- **Font-style : italic**

FONT PROPERTIES:

.Font Variants :

- The default value of font-variant property is **normal** , which specifies usual character font .This property can be set to **small-caps** to specify small capital characters .
- These characters are all uppercase ,but the letters that are normally uppercase are somewhat larger than those that are normally lowercase.

.Font weights :

- The font-weight property is used to specify the degree of boldness .
font-weight : bold
- Besides bold ,the values **normal, bolder,lighter** can be specified
- Specific number also can be given in multiples of 100 from 100 to 900 where 400 is same as normal and 700 is same as bold

FONT PROPERTIES:

. Font Shorthands :

- If more than one font property must be specified ,the values can be stated in a list .As the values of the font property.
- The order in which the property values are given in a font value, list is important .
- The order must be as follows
 - The font names must be last.
 - The font size must be second to last.
 - Font style , font variant , and font-weight ,when they are included they are in any order but must precede the font-size and font names .
- Font :bold 14pt Times New Roman'

Font Properties:

```
<!DOCTYPE html>
<html>
<head>
<title> Sample CSS </title>
<style type =“text/css”>
p.one
{
    font-family : ‘lucida Calligraphy’ ;
    font-weight : bold ;
    font-size : 75pt ;
    color : purple ;
}
h1.two
{
    font-family : ‘cambria’ ;
    color : purple ;
    font-style : italics;
}
p.three
{
    font :small-caps italic bold 50pt ‘Times New Roman’ ;}
```

Font Properties:

```
</style>
</head>
<body>
<p class="one"> CSE </p>
<h1 class="two" > ECE</h1>
<p class="three"> MECH </p>
</body>
</html>
```

Text Decoration:

The **text-decoration** property is used to specify some special features of text.

The available values are **line-through**, **overline**, **underline** and **none** which is the default.

```
<!DOCTYPE html>
<html>
<head>
<title>Text Decoration</title>
<style type = "text/css">
h1.one
{text-decoration: line-through;}
h1.two
{text-decoration: overline;}
h1.three
{text-decoration: underline;}
</style>
</head>
```

Text Decoration:

```
<body>
<h1 class = "one">Computer Science and Engineering</h1><p>[This is line-
through]</p><br/>
<h1 class = "two">Computer Science and Engineering</h1><p>[This is
overline]</p><br/>
<h1 class = "three">Computer Science and Engineering </h1><p>[This is
underline]</p><br/>
</body>
</html>
```

*LIST PROPERTIES

---List -style-image

'list-style-type'

Property name: 'list-style-type'

Value: disc | circle | square | decimal | lower-roman | upper-roman | lower-alpha | upper-alpha | none | [inherit](#)

Initial: disc

'list-style-image'

Property name: 'list-style-image'

Value: <uri> | none | [inherit](#)

Initial: none

Applies to: elements with the 'display' property set to 'list-item'

Inherited: yes

Percentage values: N/A

Media groups: [visual](#)

List Properties:

- Two presentation details of lists can be specified in XHTML documents
- :
- The shape of the bullets that precede the items in an unordered list and the sequencing values that precede the items in an ordered list.
- The list-style-property is used to specify both of these
- The list-style-property of an ordered list can be set to disc ,circle , square or none

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Bullets</title>
<style type = "text/css">
li.one {list-style-type:disc}
li.two{list-style-type:square}
li.three{list-style-type:circle}
```

*List Properties:

```
</style>
</head>
<body>
<h3>Dayananda Sagar University</h3>
<ul>
<li class = "one"> Computer Science and Engineering</li>
<li class = "two"> Electronics Engineering and Technology</li>
<li class = "three"> Mechanical Engineering </li>
</ul>
</body>
</html>
```

output

Dayananda Sagar University

- Computer Science and Engineering
- Electronics Engineering and Technology
- Mechanical Engineering

List Properties:

- Bullets in unordered lists are not limited to discs, squares or circles .
- Any image can be used in a list item bullets .
- Such a bullet is specified with the list-style-image property, whose value is specified with the url form

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Bullets-Image</title>
<style type = "text/css">
li.image {list-style-image: url(arrow.png); font-size:25pt;}
</style>
</head>
```

List Properties:

```
<body>
<h1>Dayananda Sagar University</h1>
<ul>
<li class = "image"> Computer Science and Engineering</li>
<li class = "image"> Electronics and Telecommunication</li>
<li class = "image"> Mechanical Engineering</li>
</ul>
</body>
</html>
```

List Properties:

- The following example illustrates the use of different sequence value in nested lists .

```
<html>
<head>
<title> CSS nested lists </title>
<style type = "text/css">
ol {list-style-type:upper-roman;}
ol ol {list-style-type:upper-alpha;}
ol ol ol {list-style-type:decimal;}
</style>
</head>
<ol>
<li> Information Science </li>
<ol>
<li>OOMD</li>
<li>Java & J2ee</li>
```

List Properties:

```
<ol>
<li>classes and methods</li>
<li>exceptions</li>
<li>applets</li>
<li>servelets</li>
</ol>
<li>Computer Networks</li>
<ol>
<li>Part 1</li>
<li>Part 2</li>
</ol>
<li>DBMS</li>
<li>Operations Research</li>
</ol>
<li> Computer Science</li>
<ol>
<li>Compiler Design</li>
<li>FLAT</li>
```

List Properties:

```
<ol>
<li>NFA</li>
<li>DFA</li>
<li>CFG</li>
</ol>
<li>Computer Graphics</li>
<li>Artificial Intelligence</li>
</ol>
</ol>
</html>
```

Output

- I. Information Science
 - A. OOMD
 - B. Java & J2ee
 - 1. classes and methods
 - 2. exceptions
 - 3. applets
 - 4. servelets
 - C. Computer Networks
 - 1. Part 1
 - 2. Part 2
 - D. DBMS
 - E. Operations Research
- II. Computer Science
 - A. Compiler Design
 - B. FLAT
 - 1. NFA
 - 2. DFA
 - 3. CFG
 - C. Computer Graphics
 - D. Artificial Intelligence

Color:

Color Groups :

- .Three levels of collections of colors might be used by an XHTML document.
- .The smallest useful set of colors includes only those that have standard names and are guaranteed .
- .To be correctly displayable by all browsers on all color monitors .
- .This collection of 17 colors is called named colors.
- .Larger set of colors called the web palette , consist of 216 colors. The colors of the web palette can be viewed at
https://www.web-source.net/216_color_chart.htm

Color:

Name	Hexadecimal code	Name	Hexadecimal code
aqua	00FFFF	olive	808000
black	000000	orange	FFA500
Blue	0000FF	purple	800080
fuchsia	FF00FF	red	FF0000
Gray	808080	silver	C0C0C0
Green	008000	teal	008080
Lime	00FF00	white	FFFFFF
maroon	800000	yellow	FFFF00
navy	000080		

Color Properties:

The color property is used to specify the foreground color of XTMLelements

```
<!DOCTYPE html>

<html>

<head>

<title>Colours</title>

<style type = "text/css">

p.one

{color: pink; }

p.two

{color: # 9900FF; }
```

Color Properties:

The color property is used to specify the foreground color of XHTML elements

p.three

```
{background-color:#99FF00;}  
</style>  
</head>  
<body>  
<p class = "one">Dayananda Sagar University</p>  
<p class = "two">CSE</p>  
<p class = "three">ECE</p>  
</body>  
</html>
```

ALIGNMENT OF TEXT:

- The **text-indent** property can be used to indent the first line of a paragraph. This property takes either a length or a percentage value. The **text-align** property, for which the possible keyword values are left, center, right, and justify, is used to arrange text horizontally.
- The **float** property is used to specify that text should flow around some element, often an image or a table. The possible values for **float** are left, right, and none, which is the default.
-

ALIGNMENT OF TEXT:

```
<!DOCTYPE html>
<html>
<head>
<title>Text Alignment</title>
<style type = "text/css">
h1.one
{text-align: center}
p.two
{text-indent: 0.5in; text-align: justify;}
img{float:right}
</style>
</head>
<body>
```

ALIGNMENT OF TEXT:

```
<h1 class = "one">Dayananda Sagar University</h1>
```

```
<p>
```

```
<img src = "DSU.jpg" alt="error"/>
```

```
</p>
```

```
<p class = "two">Dayananda Sagar Institutions founded in the 60s by one such  
visionary, late Sri Dayananda Sagar committed to take knowledge to the people,  
transforms today's students into responsible citizens and professional leaders of  
tomorrow. Dayananda Sagar University created by an Act of the Karnataka State  
in 2014, built on this adorable legacy and inspired by its own milestones,  
meeting the needs of quality higher education in this part of the world.
```

```
</p>
```

```
</body>
```

```
</html>
```

output

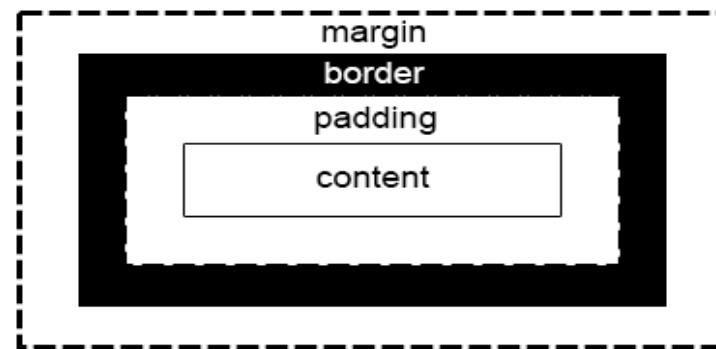
Dayananda Sagar University

Dayananda Sagar Institutions founded in the 60s by one such visionary, late Sri Dayananda Sagar committed to take knowledge to the people, transforms today's students into responsible citizens and professional leaders of tomorrow. Dayananda Sagar University created by an Act of the Karnataka State in 2014, built on this adorable legacy and inspired by its own milestones, meeting the needs of quality higher education in this part of the world.



THE BOX MODEL:

- On a given web page or document ,all the elements can have borders.
- The borders have various styles ,color and width .
- The amount of space between the content of the element and its border is known as padding .
- The space between border and adjacent element is known as margin.



Borders:

- **Border-style.**
- **It can be dotted ,dashed ,double**
 - Border-top-style.
 - Border-bottom-style
 - Border-left-style
 - Border-right-style
- **Border-width.**
- **It can be thin ,medium ,thick or any length value**
 - Border-top-width.
 - Border-bottom-width
 - Border-left-width
 - Border-right-width

Borders:

.Border-color.

- **Border-top-color**
- **Border-bottom-color**
- **Border-left-color**
- **Border-right-color**

Example :

```
<html>
<head>
<title> Table with border effects </title>
<style type = "text/css">
table
{
border-width:thick;
border-top-color:red;
border-left-color:orange;
border-bottom-color:violet;
border-right-color:green;
border-top-style:dashed;
border-bottom-style:double;
border-right-style:dotted;
}
```

```
</style>
</head>
<body>
<table border = "border">
<caption>Department of CSE, DSU </caption>
<tr>
<td> Prof. Sanjay Chitnis </td>
<td><img src = "sanjaychitnis.jpg" alt = "cant display"/></td>
</tr>
</table>
</body>
</html>
```

```
p.one {  
    border-style: solid;  
    border-width: 5px 20px; /* 5px top and bottom,  
20px on the sides */  
}
```

```
p.two {  
    border-style: solid;  
    border-width: 20px 5px; /* 20px top and bottom,  
5px on the sides */  
}
```

```
p.three {  
    border-style: solid;  
    border-width: 25px 10px 4px 35px; /* 25px top,  
10px right, 4px bottom and 35px left */  
}
```

You can also specify all the individual border properties for just one side:

Left Border

```
p {  
    border-left: 6px solid red;  
    background-color: lightgrey;  
}
```

Box Shadow:

The box-shadow property attaches one or more shadows to an element .

Value	Description
None	Default value, No shadow is displayed
H offset	The horizontal offset of the shadow. A positive value puts the shadow on the right side of a Box , A negative value puts the shadow on left side of the box
V-offset	The verticle offset of the shadow .A positive value puts the shadow below the box , A negative value puts The shadow above the box
blur	Optional .The blur radius ,the higher the number ,the more blurred the shadow will be
color	The color of the shadow ,the default value is text color

Box-Shadow:

```
<!DOCTYPE html>  
  
<html>  
  <head>  
    <style>  
      #example1 {  
        border: 1px solid;  
        padding: 10px;  
        box-shadow: 5px 10px 8px #888888;  
      }  
  
      #example2 {  
        border: 1px solid;  
        padding: 10px;  
        box-shadow: 5px 10px 18px #888888;  
      }  
    </style>  
  </head>  
  <body>  
    <div id="example1">This is example 1</div>  
    <div id="example2">This is example 2</div>  
  </body>  
</html>
```

Box-Shadow:

```
#example3 {  
    border: 1px solid;  
    padding: 10px;  
    box-shadow: 5px 10px 18px red;  
}  
</style>  
</head>  
<body>  
<h2>box-shadow: 5px 10px 8px #888888:</h2>  
<div id="example1">  
    <p>The optional third value adds a blur effect to the shadow.</p>
```

Box-Shadow:

```
</div>

<h2>box-shadow: 5px 10px 18px #888888:</h2>
<div id="example2">
    <p>More blurred.</p>
</div>

<h2>box-shadow: 5px 10px 18px red:</h2>
<div id="example3">
    <p>More blurred and red.</p>
</div>
</body>
</html>
```

Box Shadow: output

box-shadow: 5px 10px 8px #888888:

The optional third value adds a blur effect to the shadow.

box-shadow: 5px 10px 18px #888888:

More blurred.

box-shadow: 5px 10px 18px red:

More blurred and red.

Margins and Padding:

The Margin properties are named margin ,which applies to all four sides of an element : margin-left ,margin-right, margin-top and margin bottom. The padding properties are named padding ,which applies to all four sides:

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

Example :

```
<html>
<head>
<title> Margins and Padding </title>
<style type = "text/css">
p.one
{
margin:0.1in;
padding:0.5in;
background-color:#FF33FF;
border-style:dotted;
}
p.two
{
margin:0.5in;
padding:0.1in;
background-color:#00FF33;
border-style:dashed;
}
```

Example :

```
p.three  
{margin:0.3in;  
background-color:#FFFF00;  
}  
p.four  
{  
padding:0.3in;  
background-color:#FF9900;  
}  
</style>  
</head>  
<body>
```

Example :

```
<p class = "one"> Dr.Sanjay Chitnis is the Chairman of Computer Science  
and Engg ,Department ,DSU<br/>  
[margin=0.1in, padding=0.5in]</p>  
<p class = "two"> Dr.Sanjay Chitnis is the Chairman of Computer Science  
and Engg ,Department ,DSU<br/>  
[margin=0.5in, padding=0.1in]</p>  
<p class = "three"> Dr.Sanjay Chitnis is the Chairman of Computer Science  
and Engg ,Department ,DSU<br/>  
[margin=0.3in, no padding, no border]</p>  
<p class = "four"> Dr.Sanjay Chitnis is the Chairman of Computer Science  
and Engg ,Department ,DSU<br/>  
[no margin, padding=0.3in, no border]</p>  
</body>  
</html>
```

Margins and Padding: output

Dr.Sanjay Chitnis is the Chairman of Computer Science and Engg ,Department ,DSU
[margin=0.1in, padding=0.5in]

Dr.Sanjay Chitnis is the Chairman of Computer Science and Engg ,Department ,DSU
[margin=0.5in, padding=0.1in]

Dr.Sanjay Chitnis is the Chairman of Computer Science and Engg ,Department ,DSU
[margin=0.3in, no padding, no border]

Dr.Sanjay Chitnis is the Chairman of Computer Science and Engg ,Department ,DSU
[no margin, padding=0.3in, no border]

Back Ground Image:

- The background-image property is used to place an image in the background of an element.
- Example code to display background image

```
<html>
  <head>
    <title>Background Image</title>
    <style type = "text/css">
      body {background-image:url(DSU.jpg);}
      p
        {text-align: justify; color:blue;font-size:20pt;}
    </style>
  </head>
  <body>
```

Back Ground Image:

<p>WELCOME TO DAYANANDA SAGAR UNIVERSIT

Dayananda Sagar Institutions founded in the 60s by one such visionary, late Sri Dayananda Sagar committed to take knowledge to the people, transforms today's students into responsible citizens and professional leaders of tomorrow. Dayananda Sagar University created by an Act of the Karnataka State in 2014, built on this adorable legacy and inspired by its own milestones, meeting the needs of quality higher education in this part of the world. Universities of great legacy across the world are the invaluable contribution of certain visionaries to the world. Universities don't manufacture products with specific use and determined life cycle. They share & impart multitudes of streams of knowledge and create wonderful human beings – learned practitioners & Disseminators of knowledge to make the world a better place to be. These Universities of great significance have lived through the centuries building centers of knowledge and great alumni of such Universities.</p>

</body>

</html>

Back Ground - Output:



Back Ground Image:

- .In the example , notice that the background image is replicated as necessary to fill the area of the element .
- .This replication is called Tiling .
- .Tiling can be controlled with the background –repeat property ,which can take the value repeat (the default) , no-repeat ,repeat-x, or repeat-y.
- .The no-repeat value specifies that just one copy of the image to be displayed .
- .The repeat-x value means that the image is to be repeated horizontally; repeat-y means that the image is to be repeated vertically. In addition, the position of a non repeated background image can be specified with the background-position property, which can take a large number of different values.

Back Ground Image:

- The keyword values are top, center ,bottom ,left and right .all of which can be used in many different combinations.

The and <div> tags:

- .In many situations , we want to apply special font properties to less than a whole paragraph of text.
- .The tag is designed for just this purpose.

```
<html>
  <head> <title>span</title>
  <style type = "text/css">
    .spanviolet {font-size:25pt;font-family:'lucida calligraphy';color:violet;}
  </style>
  </head>
  <body>
```

The and <div> tags:

<p> Dayananda Sagar Institutions founded in the 60s by one such visionary, late Sri Dayananda Sagar committed to take knowledge to the people, transforms today's students into responsible citizens and professional leaders of tomorrow</p>

</body>

</html>

The and <div> tags:

.It is more convenient ,however ,to be able to apply a style to a section of document ~~rather than to each paragraph~~. This can be done with the <div>tag. As with , there is no implied layout for the content of the <div> tag , so its primary use is to specify presentation details for a section or division of a document.

The and <div> tags:

```
<html>
<head>
<title>div</title>
<style type = "text/css">
.one
{font-size:20pt;font-family:'lucida calligraphy';color:violet;}
.two
{font-size:25pt;font-family:'comic sans ms';color:green;}
</style>
</head>
<body>
```

The and <div> tags:

```
<div class = "one">  
  <p>Paragragh 1 under division 1</p>  
  <p>Paragragh 2 under division 1</p>  
  <p>Paragragh 3 under division 1</p>  
</div>  
  
<div class = "two">  
  <p>Paragragh 1 under division 2</p>  
  <p>Paragragh 2 under division 2</p>  
  <p>Paragragh 3 under division 2</p>  
</div>  
  
</body>  
</html>
```

Conflict Resolution:

- There can be one or more type of conflict:
- This conflict is resolved by providing priority to the different levels of style sheets.
- The inline level gets the highest priority over the document level.
- The document level gets the higher priority over the external level.
- Check the following example:

sheet applies, the inner style sheet overrides it to specify the font size.

```
/* cstyle.css - an external style sheet
   for use with cascade.html
*/
p {font-size: 0.8em;}

<!DOCTYPE html>
<!-- cascade.html
      An example to illustrate the three levels
      of style sheets
-->
<html lang = "en">
  <head>
    <title> Style sheet levels </title>
    <meta charset = "utf-8" />
    <link rel = "stylesheet" type = "text/css"
          href = "cstyle.css" />
    <style type = "text/css">
      p.docstyle {font-size: 1.2em; }
    </style>
  </head>
  <body>
    <p>
      Now is the time
    </p>
    <p class = "docstyle">
      for all good men
    </p>
    <p class = "docstyle" style = "font-size: 1.6em">
      to come to the aid
    </p>
  </body>
</html>
```

Property-value conflicts can occur in several other ways.

1) For example, a conflict may occur within a single style sheet. Consider the following style specifications, which are next to each other in the same document-level style sheet:

- h3 {color: blue;}
 - body h3 {color: red;}
- Both these specifications apply to all h3 elements in the body of the document.
The browser has to resolve this conflict

2) Another source of conflict can arise from the fact that there can be several different origins of the specification of property values. For example, they may come from a style sheet written by the author of the document itself but they may come from the browser user and from the browser.

For example, an FX user can set a minimum font size in the Tools-Options-Advanced window. Furthermore, browsers allow their users to write and use their own style sheets

Conflict Resolution:

There can be several different origins of the specification of property values

- One of the value may come from a style sheet created by the author or it can be specified by the user using the options provided by the browser.
- The property values with different origin have different precedence
- The precedence can also be set for a property by making it as important
 - The process of conflict resolution is a multi-stage sorting process
 - The first step is to gather information about levels of style sheet.
 - Next, all the origins and weights are sorted. The following rules are considered: The precedence is followed as below ,
 - 1. Important declarations with user origin**
 - 2. Important declarations with author origin**
 - 3. Normal declarations with author origin**
 - 4. Normal declarations with user origin**
 - 5. Any declarations with browser (or other user agent) origin.**

- If there are other conflicts even after sorting ,The next step is sorting by specificity
- Rules are
 - 1. id selectors
 - 2. Class and pseudo class selectors
 - 3. Contextual selectors (more element type names means that they are more specific)
 - 4. Universal selectors
- If there still conflicts, they are resolved by giving precedence to most recently seen specification
- The whole sorting process that is used to resolve style specification conflicts is called the cascade

2) Inheritance is another source of property-value conflicts; although as we already know, the inherited property value is always overridden by the property value given to the descendant element.

A child element always inherits the styles of the parent element, unless declared otherwise separately. Even here, if the declarations are not in conflict, they simply merge

```
<head>
  <style>
    .parent{
      color: blue;
      background-color: green;
      width: 500px;
      height: 500px;
    }
    .box{
      font-size: 3em;
      background-color: red;
      border: 1px solid black;
      margin: 10px;
    }
    .box4{
      color: yellow;
      font-size: 1em;
    }
  </style>
</head>
<body>
  <div class="parent">
    <div class="box">I am a box</div>
    <div class="box">I am a box</div>
    <div class="box">I am a box</div>
    <div class="box box4">I am a box</div>
  </div>
</body>
```



As we can see, the text colour is inherited from the parent element. The background colour is received from the styles of the box. There are 2 conflicting declarations for “`box4`”, both of which have the same specificity (we will come to specificity in the next paragraph). So, the bottom declaration wins, and the font-size becomes `1em`.

3) **every property-value specification has a particular specificity**, depending on the particular kind of selector that is used to set it, and those specificities have different levels of precedence. These different levels are used to resolve conflicts among different specifications.

4) property-value specifications can be marked as being important by including **`!important`** in the specification

- `p.special {font-style :italic !important ; font-size:14 }`
- **This means that `font-style:italic` is important [this is known as weight of specification]**

CSS3: Introduction

.CSS3 version introduced new features that allowed developers to style ~~HTML elements with less CSS code.~~

.CSS3 is most notorious for modules that speed up the specification process.

.At first , browsers did not support CSS3 features and took a while for them to become fully Compatible.

Difference between CSS1 and CSS3

.CSS3 does not deprecate older CSS Code because it is only an addition to the features offered by CSS1. This list provides the main arguments in the CSS3 Vs CSS Debate.

- .CSS3 allows developers to style HTML elements easier
- .They are less dependent on image files and can complete CSS Styling with fewer lines of code
- .The aim of CSS1 was for appearance formatting , and it did not allow responsive designs.

CSS Opacity

.The opacity property specifies the opacity/transparency of an element

.Opacity value can take a value from 0.0 -1.0 .The lower value ,the more transparent

CSS Opacity

.Image opacity with 20% transparency

```
<html>
<head>
<style>
img {
  opacity: 0.2;
}
</style>
</head>
<body>
```

CSS Opacity

```
<h1>Image Transparency</h1>
```

```
<p>The opacity property specifies the transparency of an element. The  
lower the value, the more transparent:</p>
```

```
<p>Image with 20% opacity:</p>
```

```

```

```
</body>
```

```
</html>
```

CSS Opacity

Output :

Image Transparency

The opacity property specifies the transparency of an element. The lower the value, the more transparent:

Image with 20% opacity:



CSS Opacity

.Image opacity with 50% transparency

```
<html>
<head>
<style>
img {
  opacity: 0.5;
}
</style>
</head>
<body>
```

CSS Opacity

```
<h1>Image Transparency</h1>
```

```
<p>The opacity property specifies the transparency of an element. The  
lower the value, the more transparent:</p>
```

```
<p>Image with 50% opacity:</p>
```

```

```

```
</body>
```

```
</html>
```

CSS Opacity

Output :

Image Transparency

The opacity property specifies the transparency of an element. The lower the value, the more transparent:

Image with 50% opacity:



CSS Opacity

.Image opacity with 50% transparency

```
<html>
<head>
<style>
img {
  opacity: 1.0;
}
</style>
</head>
<body>
```

CSS Opacity

```
<h1>Image Transparency</h1>
```

```
<p>The opacity property specifies the transparency of an element. The  
lower the value, the more transparent:</p>
```

```
<p>Image with 100% opacity:</p>
```

```

```

```
</body>
```

```
</html>
```

CSS Opacity

Output :

Image Transparency

The opacity property specifies the transparency of an element. The lower the value, the more transparent:

Image with 100% opacity:



CSS Rounded Corners

.With the CSS border-radius property ,you can give any element “Rounded Corners”

.CSS border-radius Property: The CSS border-radius property defines ,the radius of an elements corners.

The border-radius Property

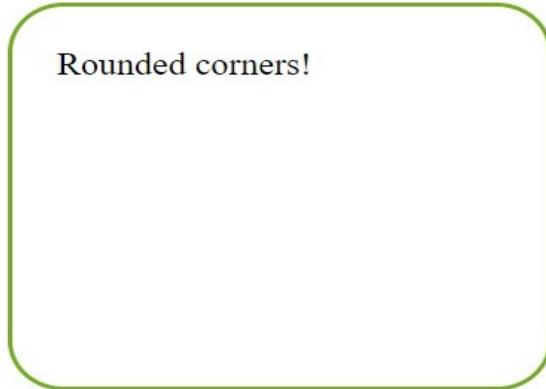
Rounded corners for an element with a specified background color:



Rounded corners!

CSS Rounded Corners

Rounded corners for an element with a border:



Rounded corners for an element with a background image:



CSS Rounded Corners

Code example :

```
<!DOCTYPE html>
<html>
<head>
<style>
#rcorners1 {
    border-radius: 25px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;
}
```

CSS Rounded Corners

Code example :

```
#rcorners2 {  
    border-radius: 25px;  
    border: 2px solid #73AD21;  
    padding: 20px;  
    width: 200px;  
    height: 150px;  
}
```

CSS Rounded Corners

Code example :

```
#rcorners3 {  
    border-radius: 25px;  
    background: url(klematis.jpg);  
    background-position: left top;  
    background-repeat: repeat;  
    padding: 20px;  
    width: 200px;  
    height: 150px;  
}  
</style>  
</head>  
<body>
```

CSS Rounded Corners

Code example :

```
<h1>The border-radius Property</h1>
<p>Rounded corners for an element with a specified background color:</p>
<p id="rcorners1">Rounded corners!</p>
<p>Rounded corners for an element with a border:</p>
<p id="rcorners2">Rounded corners!</p>
<p>Rounded corners for an element with a background image:</p>
<p id="rcorners3">Rounded corners!</p>
</body>
</html>
```

CSS Attribute Selector

- Style HTML elements with specific Attributes

It is possible to Style HTML elements that have specific Attributes or Attribute values .

- CSS [Attribute] selector

The [attribute] selector is used to select elements with a specified attribute.

- The following example selects all <a> elements with a target attribute

```
a[target]{  
    background-color :yellow;  
}
```

CSS Attribute Selector

```
<!DOCTYPE html>  
  
<html>  
  <head>  
    <style>  
      a[target] {  
        background-color: yellow;  
      }  
    </style>  
  </head>  
  <body>  
    <h2>CSS [attribute] Selector</h2>  
    <p>The links with a target attribute gets a yellow background:</p>
```

CSS Attribute Selector

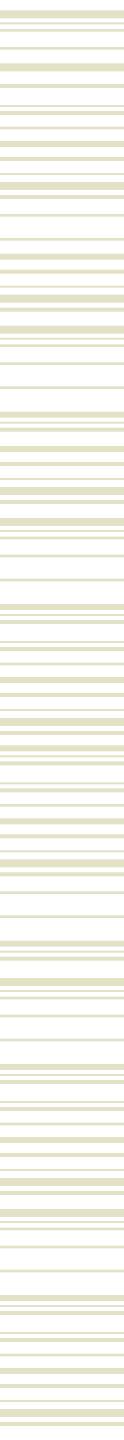
```
<a href="https://www.w3schools.com">w3schools.com</a>  
<a href="http://www.disney.com" target="_blank">disney.com</a>  
<a href="http://www.wikipedia.org" target="_top">wikipedia.org</a>  
  
</body>  
</html>
```

CSS Attribute Selector

CSS [attribute] Selector

The links with a target attribute gets a yellow background:

[w3schools.com](#) [disney.com](#) [wikipedia.org](#)



THANK YOU