CASCADING STYLE SHEETS

INTRODUCTION

What is CSS?

CSS stands for **cascading style sheets**. It is used to apply different styles and display out HTML elements in better way. We can also present our elements with different styles with attributes inside html tags but it is very limited styles only. So, to make our webpages more beautiful, we need to use CSS.

CSS Syntax

A CSS rule has two main parts: a selector, and one or more declarations:

The **selector** is normally the HTML element that you want to apply the style.

Each declaration consists of a **property** and a **value.**

The property is the style attribute you want to change. Each property has a value.

We will write different style properties as follows.

```
property: value; property: value;
```

Example:

```
h1{
    border: 1px solid green;
    color: red;
}
```

Implementing CSS

We can implement CSS in three ways

1) Inline Style Sheets
2) Internal Style Sheets
3) External Style Sheets

Inline Style Sheets:

Styles that can be applied to one line i.e., one tag are said to be inline style sheets. In this style can be written inside the tag with the help of style attribute.

Ex: < h1 style = “color: red; “>

Internal Style Sheets:

Styles that can be applicable to one entire web page is said to be Internal style sheet. Here a separate style tag will be included in head part that specifies selectors and the related styles.

```
<style type=”text/css”>
    Styles with selectors will be written here
</style>
```

External Style Sheets:

Styles that can be applicable to one or more web pages are said to be external style sheets. Here styles will be saved as separate file with the extension name .css and each html file will be linked with that external css file with the help of link tag.

INLINE STYLE SHEETS

The styles that are applicable to only one tag are said to be inline style sheets. To implement inline styles in a tag we will write **style** attribute with properties and values.

Syntax :

<tag style = "property:value;property:value; property:value; property:value;">

Example :

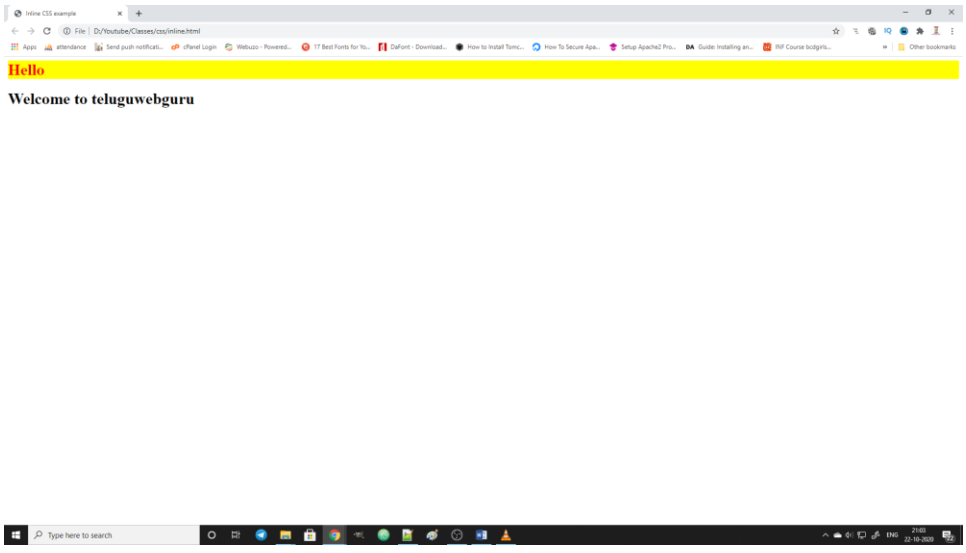
```
<h1 style="text-align:center;color:red;">hello</h1>
```

Styles applied in above h1 tag will only be applicable to this h1 tag only. These will not be applicable to any other tags including other h1 tags also which are available in the same webpage.

Example

```
<html>
        <head>
          <title>Inline CSS example</title>
        </head>

        <body>
          <h1 style="background-color:yellow;color:red;">Hello</h1>
          <h1>Welcome to teluguwebguru</h1>

        </body>
</html>
```

Output



Please observe in above output that style is applied to only one h1 tag where style attribute is written.

INTERNAL STYLE SHEETS

Styles that can be applicable to one entire web page is said to be Internal style sheet. Here a separate style tag will be included in head part that specifies selectors and the related styles.

```
<style type=”text/css”>
    Styles with selectors will be written here
</style>
```

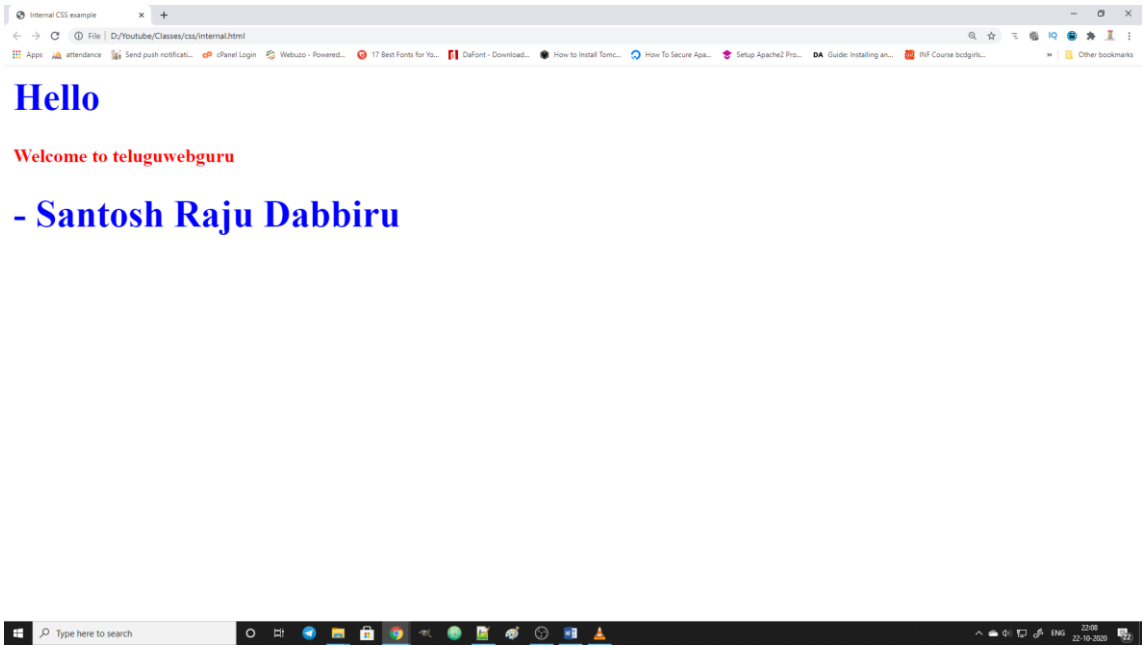
The styles with selectors will be written as

```
Selector {
    Property : Value;
    Property : Value;
    Property : Value;
}
```

Example:

```
<html>
  <head>
    <title>Internal CSS example</title>

    <style type="text/css">
      h1{
        color:blue;
       }

       b{
        color:red;
       }

    </style>
  </head>

  <body>
    <h1>Hello</h1>
    <b>Welcome to teluguwebguru</b>
    <h1>- Santosh Raju Dabbiru</h1>

  </body>
</html>
```

Output



Please observe that inline styles (h1) are applicable to the entire web page that matches the selector.

EXTERNAL STYLE SHEETS

Styles that can be applicable to one or more web pages are said to be external style sheets. Here styles will be saved as separate file with the extension name **.css** and each html file will be linked with that external css file with the help of **link** tag.

Link tag syntax is as follows

```
<link rel = "stylesheet"  href="css file name">
```

Example

external1.html

```
<html>
  <head>
        <title>External CSS example</title>
        <link rel="stylesheet" href="external.css">
      </head>

      <body>
        <h1>Welcome to external style sheet1</h1>
        <b>telugu web guru</b>
      </body>
</html>
```

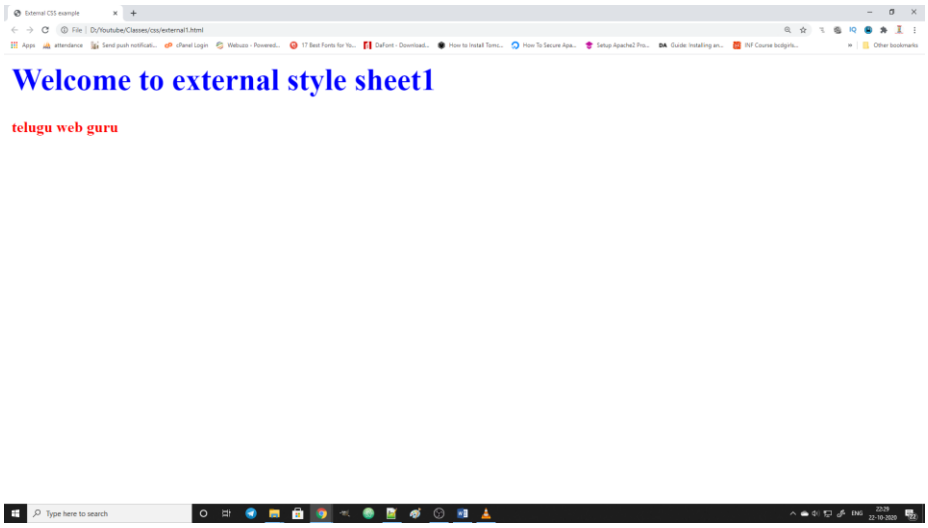
external2.html

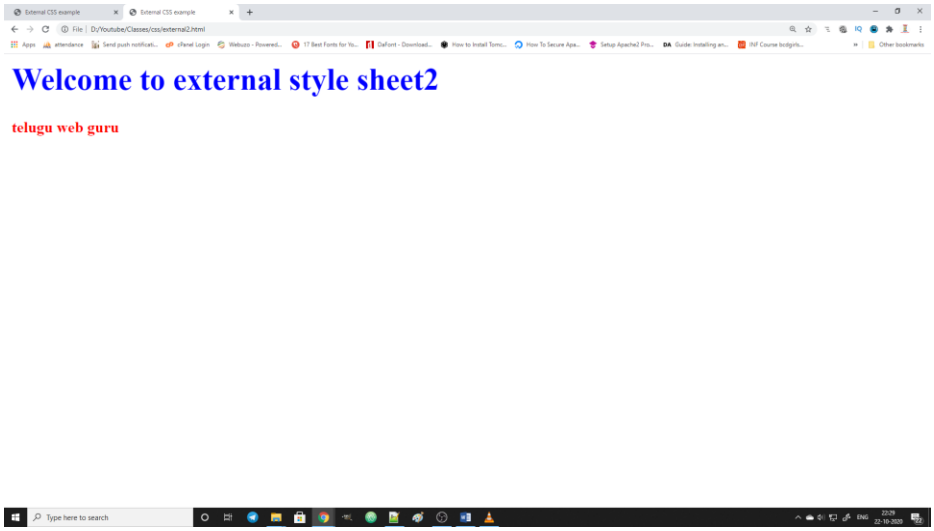
```
<html>
  <head>
        <title>External CSS example</title>
        <link rel="stylesheet" href="external.css">
      </head>

      <body>
        <h1>Welcome to external style sheet1</h1>
        <b>telugu web guru</b>
      </body>
</html>
```

external.css

```
h1{
  color:blue;
}

b{
  color:red;
 }
```

Output:





Please observe the both outputs of external1.html, external2.html.

Styles that are defined in external.css is linked with both the files through link tag so styles are applied to both the files at a time.

By using external style sheet, it is very easy to maintain the projects because in case of any changes required in styles we just require to change the style in just one file that is external.css so that the updates will be affected to all the web pages that are linked with this css file.

SELECTORS IN STYLE SHEETS

CSS selectors are used to select the tags/elements that we want to apply the styles.

Generally, there are 5 different CSS selectors:

Simple selectors: In this, we select elements based on tag name, id, class

Combinator selectors: We can select elements based on a relationship between them

Pseudo-class selectors: We can select elements based on their state

Pseudo-elements selectors: We can select & apply styles to part of an element.

Attribute selectors: We can select elements based on attributes and their values.

Simple Selectors (tag based, id based, class based)

Tag based selector:

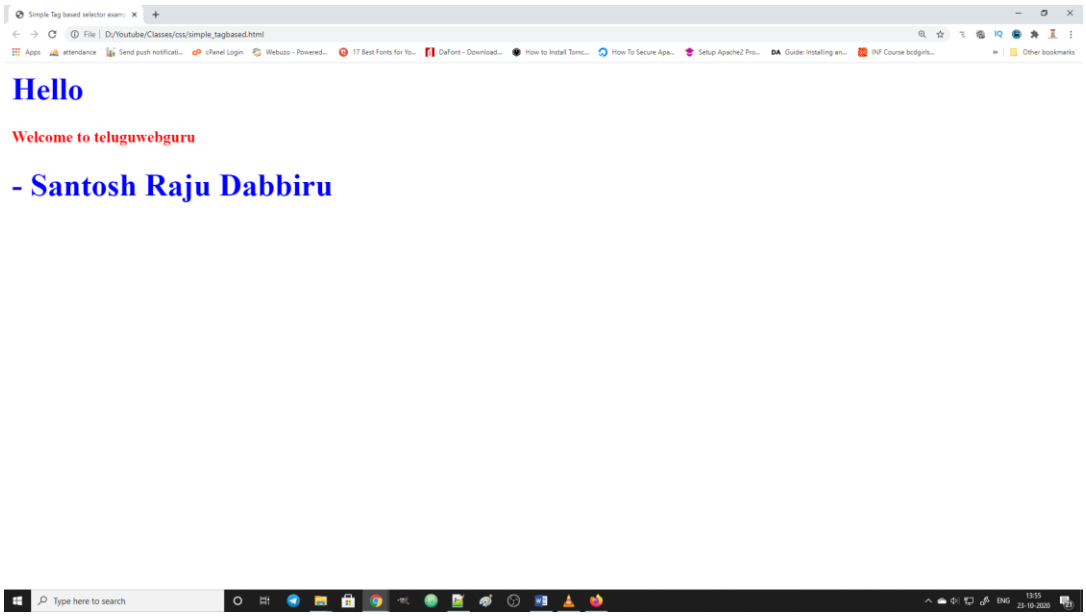
If we apply styles based on the element name then it is said to be tag based selector.

Example

```
<html>
    <head>
        <title>Simple Tag based selector example</title>

        <style type="text/css">
           h1{
                color:blue;
             }

             b{
                color:red;
             }

        </style>
     </head>

     <body>
        <h1>Hello</h1>
        <b>Welcome to teluguwebguru</b>
        <h1>- Santosh Raju Dabbiru</h1>

     </body>
</html>
```

In the above example please observe that styles are applied to the element h1, b1. So, these styles are applicable to all the h1, b1 elements in the entire web page.

Output:



The CSS id Selector:

Here the selector is selected based on id attribute of an HTML element and please remember that we can give the same id to different tags but as per programming standards it is not recommended.

If you want to group set of elements then you may go with class instead of id.

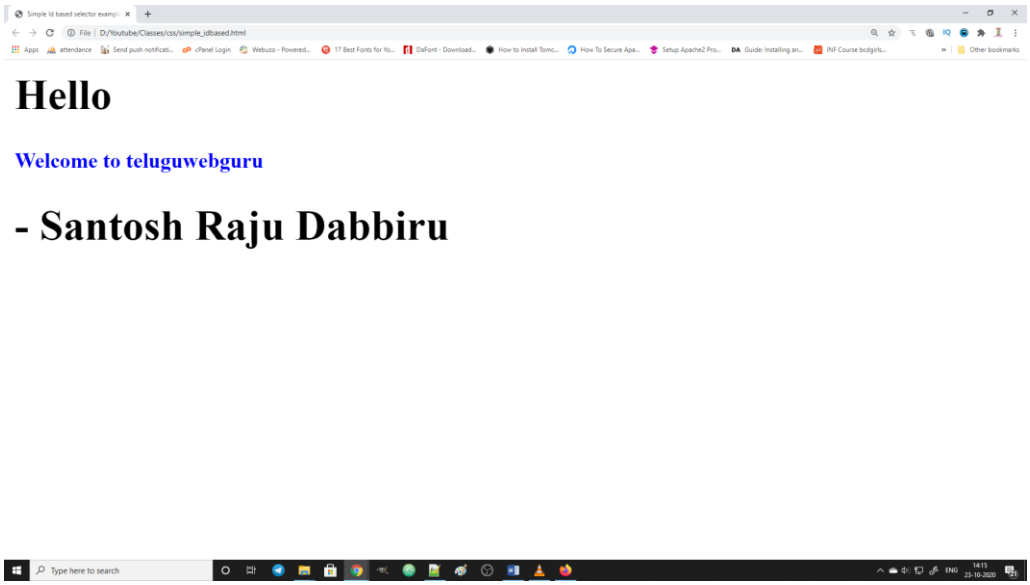
The id of an element is unique within a page, so the id selector is used to select one unique element!

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

Example

```
<html>
  <head>
        <title>Simple Tag based selector example</title>

        <style type="text/css">
          #special{
                color:blue;
              }
        </style>
      </head>

      <body>
        <h1>Hello</h1>
        <b id="special">Welcome to teluguwebguru</b>
        <h1>- Santosh Raju Dabbiru</h1>

      </body>
</html>
```

Output:



Please observe that style is applied to the only element that contains id value “special”.

The CSS class Selector

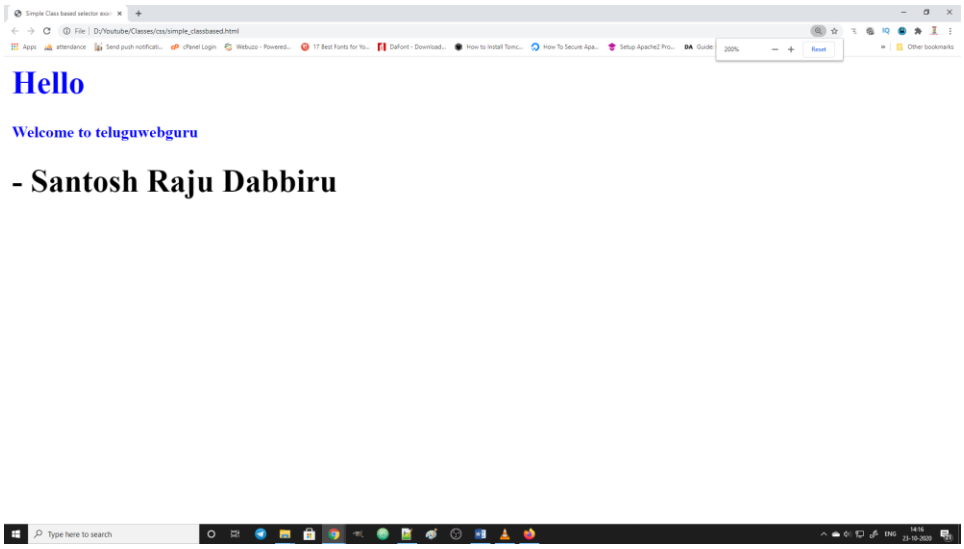
Using this method, we can apply the same style to set of tags having the same class name in class attribute.

To select elements with a specific class, write a period (.) character, followed by the class name.

Example

```
<html>
  <head>
        <title>Simple Class based selector example</title>

        <style type="text/css">
          .special{
                color:blue;
              }


        </style>
      </head>

      <body>
        <h1 class="special">Hello</h1>
        <b class="special">Welcome to teluguwebguru</b>
        <h1>- Santosh Raju Dabbiru</h1>

      </body>
</html>
```

Output :



CSS Combinators

A combinator will allow us to apply styles based on relationship between the selectors.

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

There are four different combinators in CSS:

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

Descendant Selector (space)

The descendant selector matches all elements that are descendants of a specified element.

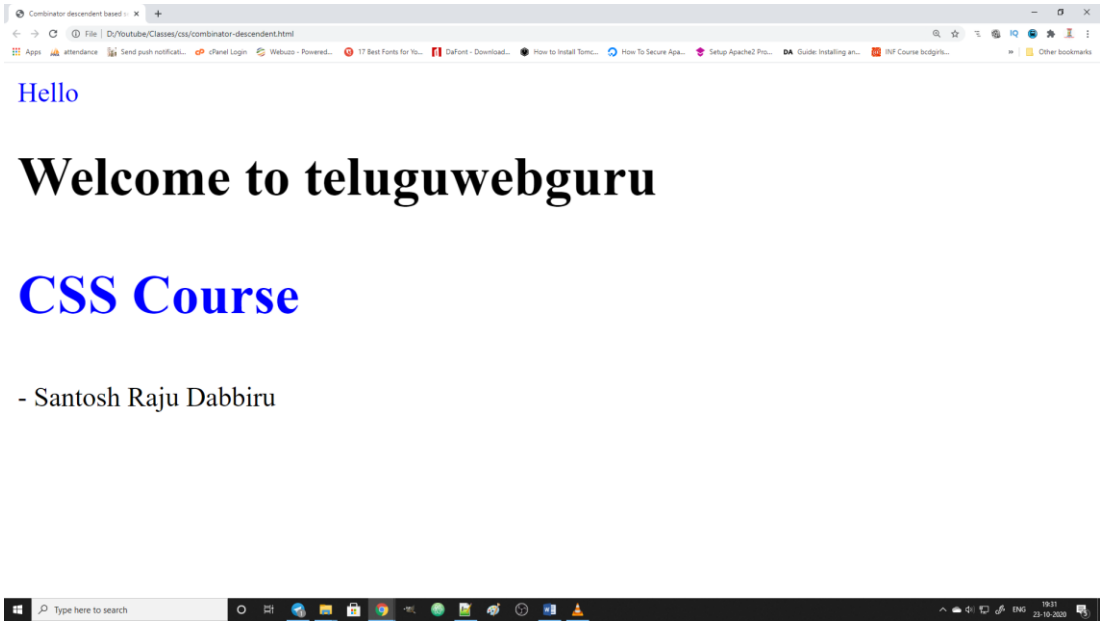
The following example selects all <p> elements inside elements:

Example

```
<html>
    <head>
        <title>Combinator descendent based selector example</title>

        <style type="text/css">
           div p{
                color:blue;
             }
```

```
        </style>
    </head>

    <body>
        <div>
        <p>Hello</p>
        <h1>Welcome to teluguwebguru</h1>
        <h1><p>CSS Course</p></h1>
        </div>


        <p>- Santosh Raju Dabbiru</p>

    </body>
</html>
```

Output:



Please observe in the above example that style is only applied to the p elements that exists inside div element(not only direct Childs, it is applicable to all p elements that are available in child of div also). That is the reason that style is not applied to second h1 tag which Is not inside div.

Child Selector (>)

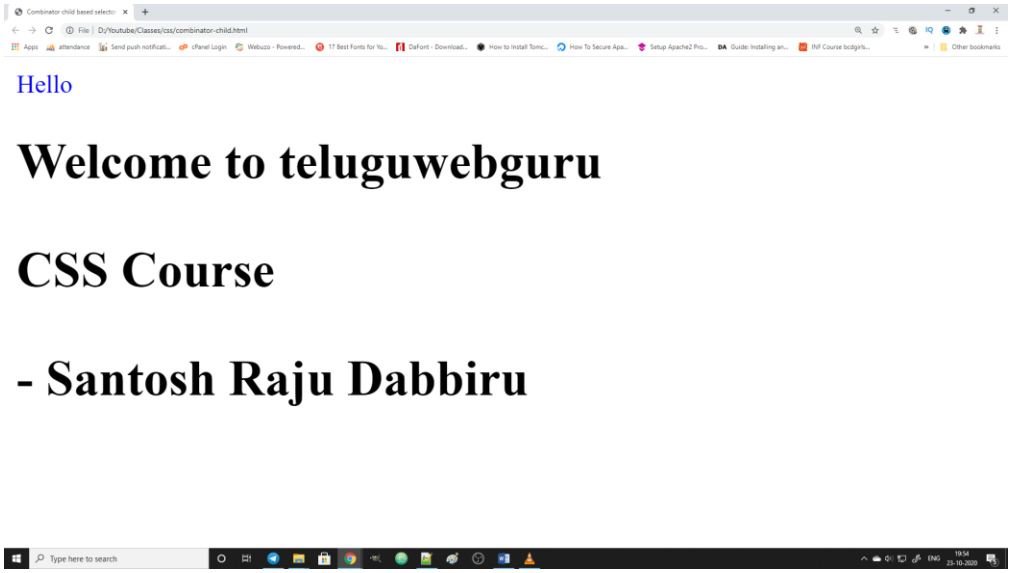
The child selector selects all elements that are direct children of the particular element.

The following example selects all <p> elements that are children of a element:

Example

```
<html>
  <head>
        <title>Combinator child based selector example</title>

        <style type="text/css">
          div > p{
                color:blue;
              }
        </style>
      </head>

      <body>
        <div>
        <p>Hello</p>
        <h1>Welcome to teluguwebguru</h1>
        <h1><p>CSS Course</p></h1>
        </div>

        <h1>- Santosh Raju Dabbiru</h1>

      </body>
</html>
```

Output



Here styles will be applied to direct Childs only. That's why "CSS Course" is not applied with the style even through it is in <p> element.

Adjacent Sibling Selector (+)

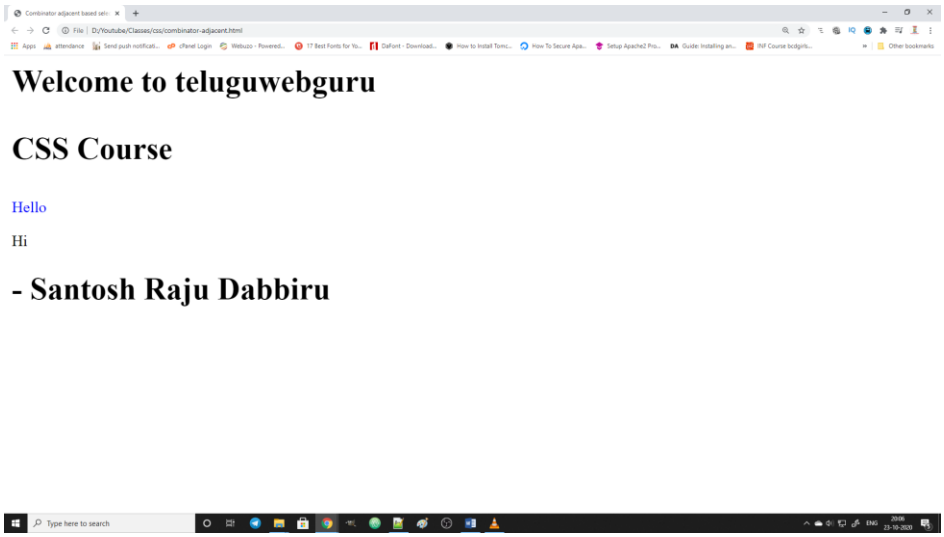
The adjacent sibling selector selects all elements that are the adjacent siblings of a specified element.Sibling elements must have the same parent element, and "adjacent" means "immediately following".

The following example selects all <p> elements that are placed immediately after elements:

Example

```
<html>
  <head>
          <title>Combinator adjacent based selector example</title>
          <style type="text/css">
            div + p{
                    color:blue;
                  }
          </style>
        </head>

        <body>
          <div>
          <h1>Welcome to teluguwebguru</h1>
          <h1><p>CSS Course</p></h1>
          </div>
          <p>Hello</p>
          <p>Hi</p>
          <h1>- Santosh Raju Dabbiru</h1>

        </body>
</html>
```

Output:



Here style is applied to adjacent p of div element (immediate next element) only. That's why ' hi ' is displayed normally.

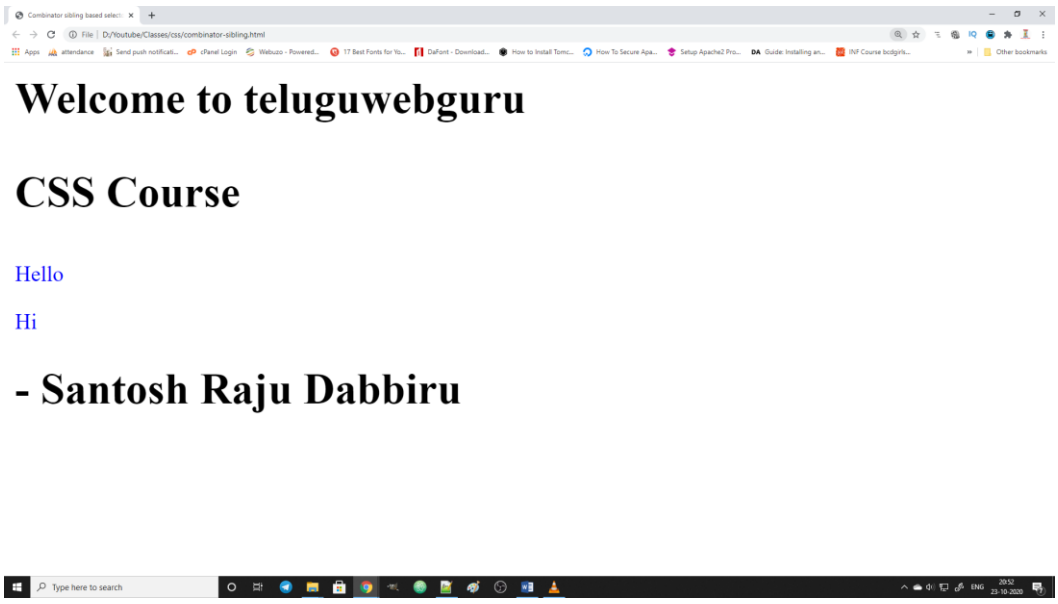
General Sibling Selector (~)

The general sibling selector selects all elements that are siblings of the particular element.

The following example selects all <p> elements that are siblings of elements:

Example

```
<html>
  <head>
        <title>Combinator sibling based selector example</title>
        <style type="text/css">
          div ~ p{
                color:blue;
              }
        </style>
      </head>

      <body>
        <div>
        <h1>Welcome to teluguwebguru</h1>
        <h1><p>CSS Course</p></h1>
        </div>

        <p>Hello</p>
        <p>Hi</p>

        <h1>- Santosh Raju Dabbiru</h1>

      </body>
</html>
```



Here style is applied to all the p elements that are next to div element (not only immediate next element but all).

CSS Pseudo-classes

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouse over it
- Style visited and unvisited links differently
- Style an element when it gets focus

Syntax

```
selector:pseudo-class {
                        property: value;
                      }
```

Anchor Pseudo-classes

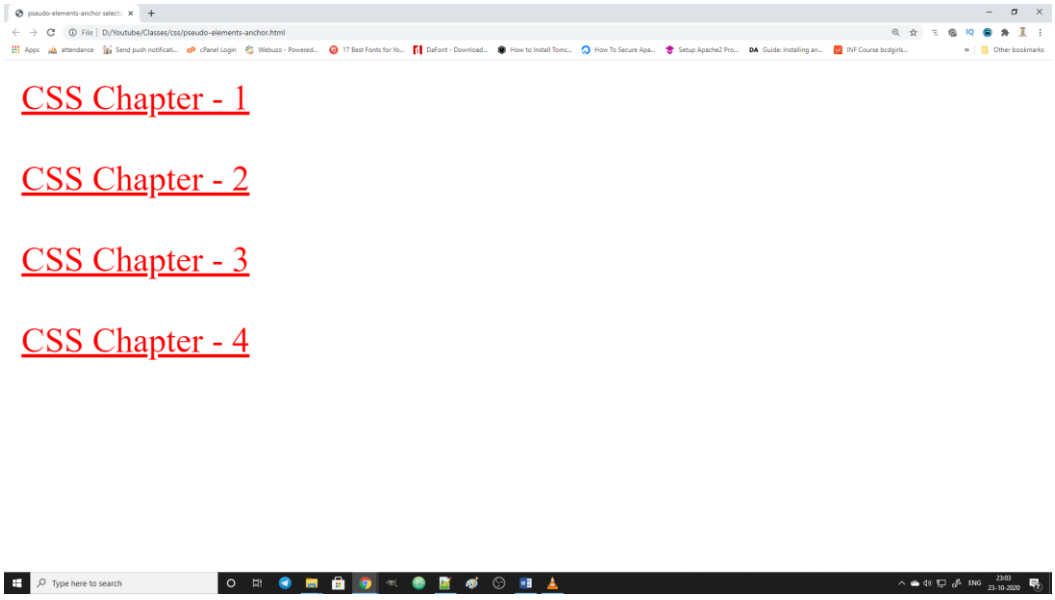
Links can be displayed in different ways:

Example

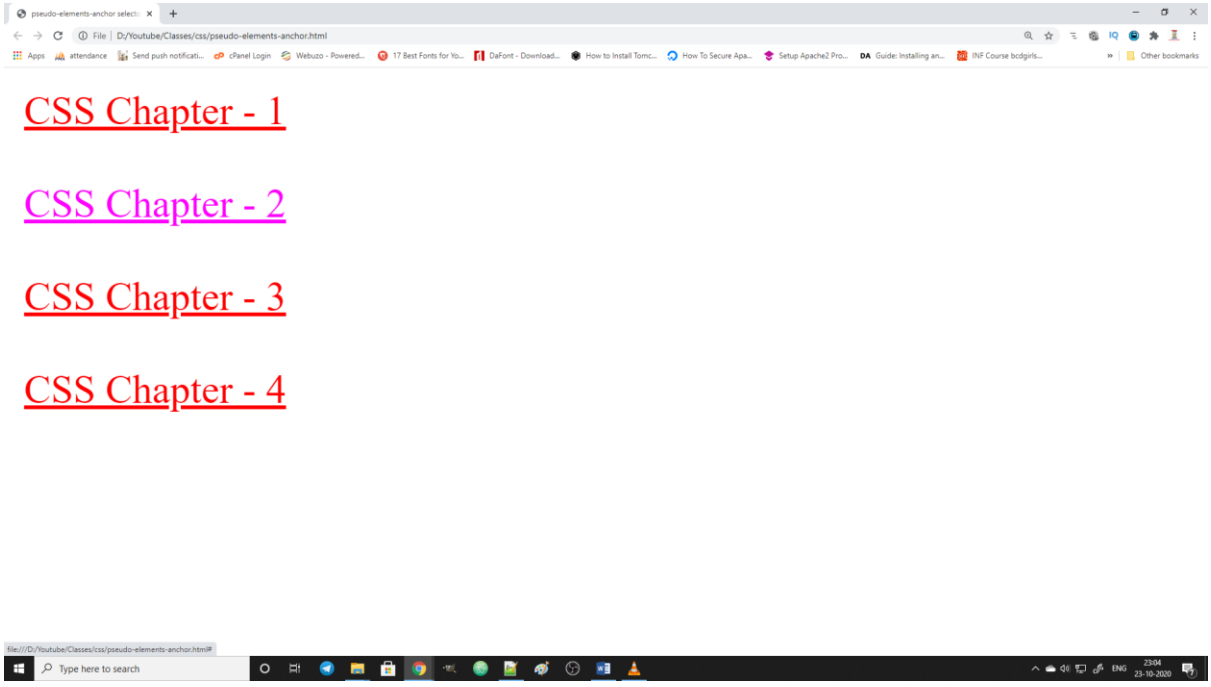
```
<html>
  <head>
        <title>pseudo-elements-anchor selector example</title>

        <style type="text/css">
         a:link {
                color: #FF0000;
                }

                a:visited {
                color: #00FF00;
                }

                a:hover {
                color: #FF00FF;
                }

                a:active {
                color: #0000FF;
                }

        </style>
      </head>

      <body>

        <a href="#">CSS Chapter - 1</a><br/><br/>
        <a href="#">CSS Chapter - 2</a><br/><br/>
        <a href="#">CSS Chapter - 3</a><br/><br/>
        <a href="#">CSS Chapter - 4</a><br/><br/>
```

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

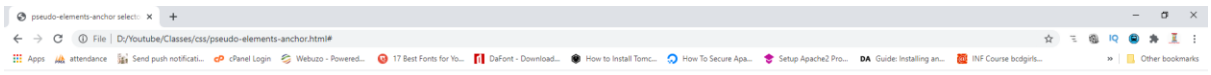
Output:



In the above screen no link is clicked so these are in link state now



In the above screen I placed the mouse pointer on css chapter-2 link (just placed the pointer on it…not clicked) then this particular state is called hover state



CSS Chapter - 1

CSS Chapter - 2

CSS Chapter - 3

CSS Chapter - 4

This indicates chapter-2 is in active state because it is just clicked. Generally we can observe the active state for fraction of seconds only and then it will be converted to visited state. All the remaining 3 links that are displayed in green are said to be visited state which indicates we already clicked and visited these links.

Pseudo-classes and CSS Classes

Pseudo-classes can be combined with CSS classes:

When you hover over the link in the example, it will change color:

Example

```
a.highlight:hover {
                    color: #ff0000;
                  }
```

Hover on

An example of using the :hover pseudo-class on a element:

Example

```
div:hover {
            background-color: blue;
          }
```

CSS Pseudo-element selector

A CSS pseudo-element is used to style specified parts of an element.
For example, it can be used to:

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

Syntax

```
selector::pseudo-element {
                property: value;
                }
```

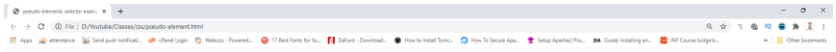
The ::first-letter Pseudo-element

The ::first-letter pseudo-element is used to add a special style to the first letter of a text.
The following example formats the first letter of the text in all <p> elements:

```
p::first-letter {
                color: #ff0000;
                font-size: xx-large;
            }
```

Example :

```
<html>
  <head>
        <title>pseudo-elements selector example</title>
        <style type="text/css">
          h1::first-letter{
                color:red;
                }
        </style>
      </head>
      <body>
        <h1>Telugu web guru welcomes you</h1>
      </body>
</html>
```



Telugu web guru welcomes you

In the above example only first character is presented with different style.

The ::first-line Pseudo-element

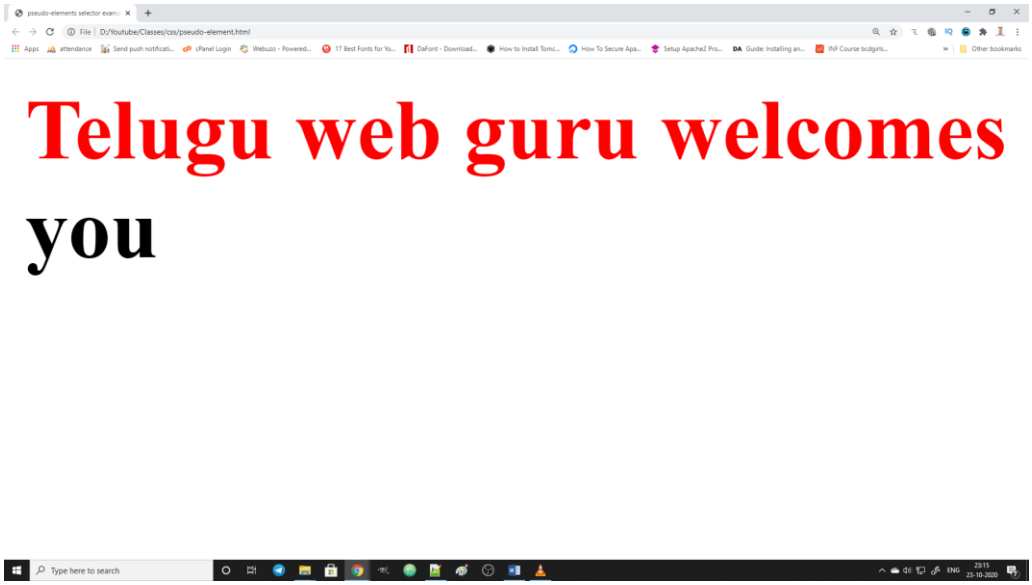
The ::first-line pseudo-element is used to add a special style to the first line of a text.

The following example formats the first line of the text in all <p> elements:

```
p::first-line {
    color: #ff0000;
    font-variant: small-caps;
}
```

Example:

```
<html>
  <head>
        <title>pseudo-elements selector example</title>

        <style type="text/css">
          h1::first-line {
              color:red;
              }

        </style>
      </head>

      <body>

        <h1>Telugu web guru welcomes you</h1>


      </body>
</html>
```



In the above output style is applied to only first line if it is displayed in multiple lines.

CSS Attribute Selectors

We can apply style to HTML Elements With Specific Attributes only.

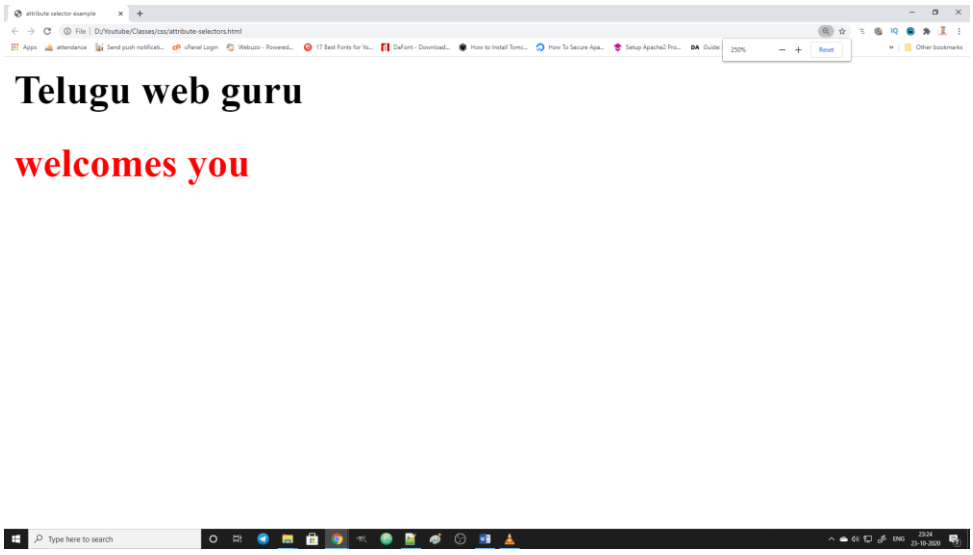
CSS [attribute] Selector

The [attribute] selector is used to select elements with a specified attribute.The following example selects all <a> elements with a target attribute:

Example

```
<html>
  <head>
        <title>attribute selector example</title>

        <style type="text/css">
          h1[id]{
                color:red;
                }

          </style>
        </head>

        <body>

          <h1>Telugu web guru</h1>
          <h1 id="spl">welcomes you</h1>

        </body>
</html>
```

Output



In the above example style is applied to h1 tag that contains id attribute.

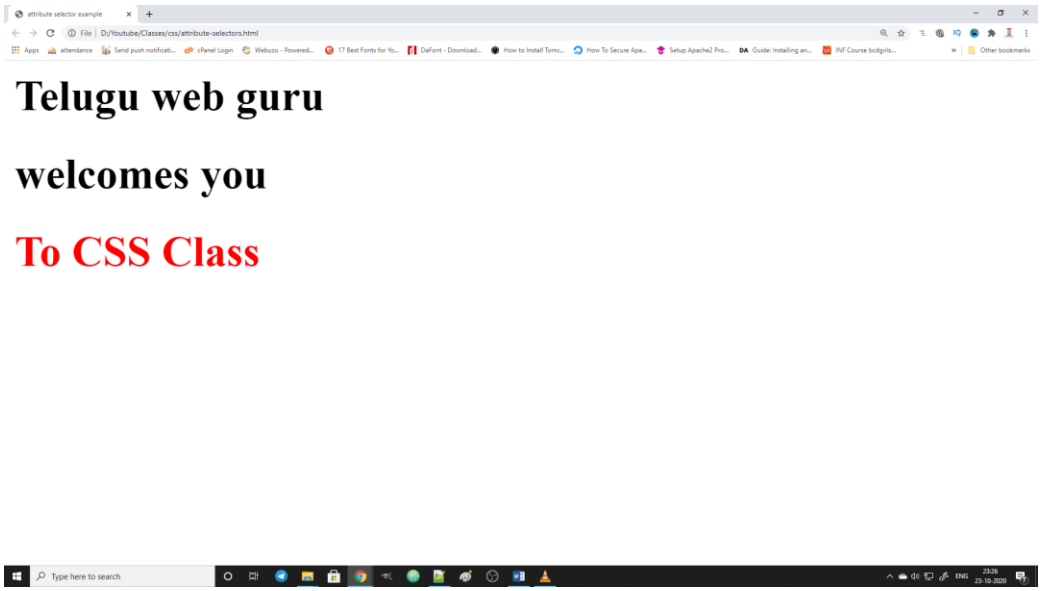
CSS [attribute="value"] Selector

The [attribute="value"] selector is used to select elements with a specified attribute and value.

The following example selects all <a> elements with a target="_blank" attribute:

Example

```
<html>
  <head>
        <title>attribute selector example</title>

        <style type="text/css">
          h1[id=normal]
                {
                color:red;
                }

          </style>
        </head>

        <body>

          <h1>Telugu web guru</h1>
          <h1 id="spl">welcomes you</h1>
          <h1 id="normal">To CSS Class</h1>
        </body>
</html>
```



Here style is applied to the h1 that is having id value “normal”.

Color style Properties in CSS

We can apply color to text in our webpage by using the property “color”.

color : color name / hex code / rgb(r value, g value, b value)

We can apply background color to any element by using background-color property.

In color property value we can give color names directly like red / green etc.,

In the place of color names red, yellow we can give either hexcodes or rgb values.

Hex codes are prefixed with # and 6characters. In that 6 characters first two(00-FF) represents red, second two represents green, last two represents blue.

We can provide rgb values also as

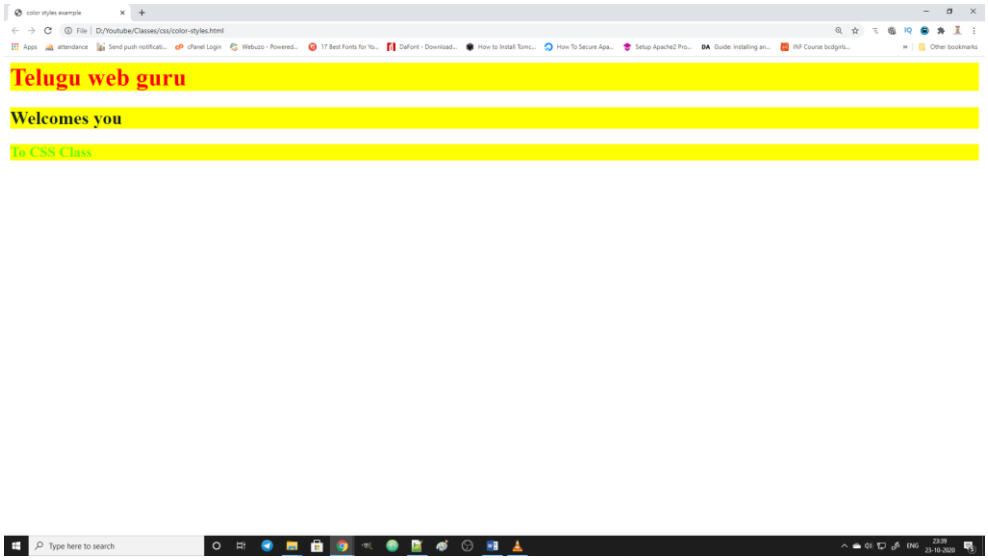
Rgb(red value, green value, blue value)

Where each value ranges from 0 to 255

Example

```
<html>
  <head>
        <title>color styles example</title>
        <style type="text/css">
              h1{
               color:red;
               background-color:yellow;
               }

               h2{
               color:#0F1F2F;
               background-color:yellow;
               }

               h3{
               color:rgb(100,250,10);
               background-color:yellow;
               }
        </style>
      </head>

      <body>

        <h1>Telugu web guru</h1>
        <h2>Welcomes you</h2>
        <h3>To CSS Class</h3>

      </body>
</html>
```

Output:



In the above example we applied three different type of color property methods to h1, h2, h3.

Background style Properties in CSS

We have different styles related to background that can be applied to any element like applying background color, background image so on.

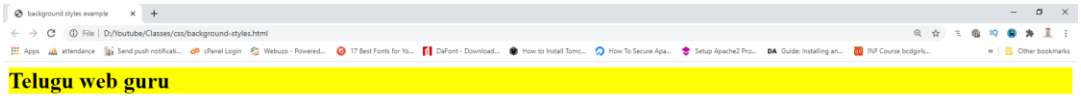
Following are the properties that are related to background properties

1) background-color : we can set the background color to any element
   syntax :
   background-color: color name / hex value / rgb(r,g,b)

   Example:

```
<html>
        <head>
            <title>background styles example</title>

            <style type="text/css">
              h1{
                  background-color:yellow;
                  }
            </style>
          </head>

          <body>
            <h1>Telugu web guru</h1>
          </body>
</html>
```

Output:



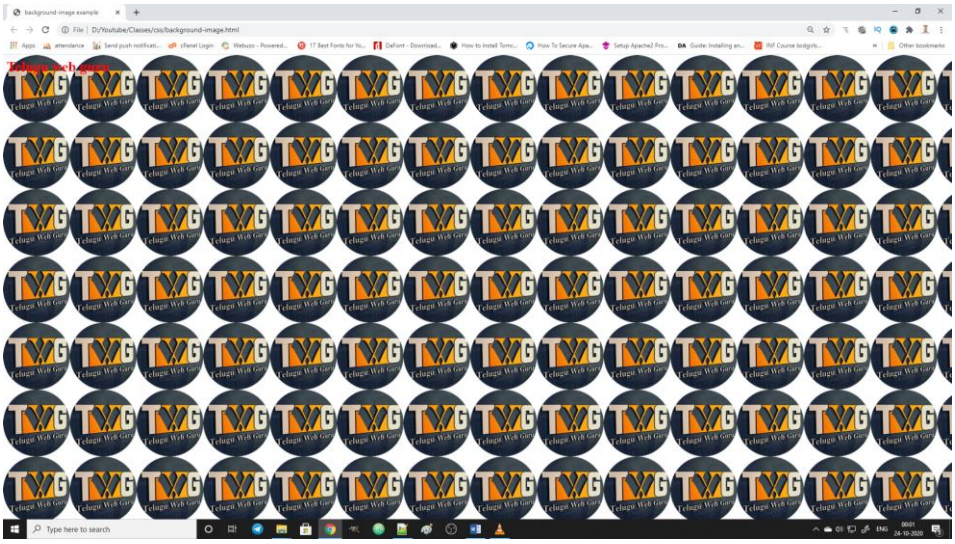
2) background-image : We can set image as background to any element

syntax :

background-image : url (' image path ')

Example:

```
<html>
  <head>
    <title>background-image example</title>

    <style type="text/css">
      body{
        background-image:url('small_logo.png');
        }
      h1{
        color:red;

       }
      </style>
    </head>

    <body>
      <h1>Telugu web guru</h1>
    </body>
</html>
```



As background-image is smaller than screen resolution, image is repeated by default until screen space is filled. We can control this by using background-repeat property

3) background-repeat: We can decide while setting image as background, if the image is too smaller than screen resolution then whether to repeat the image and fill screen or not

syntax:

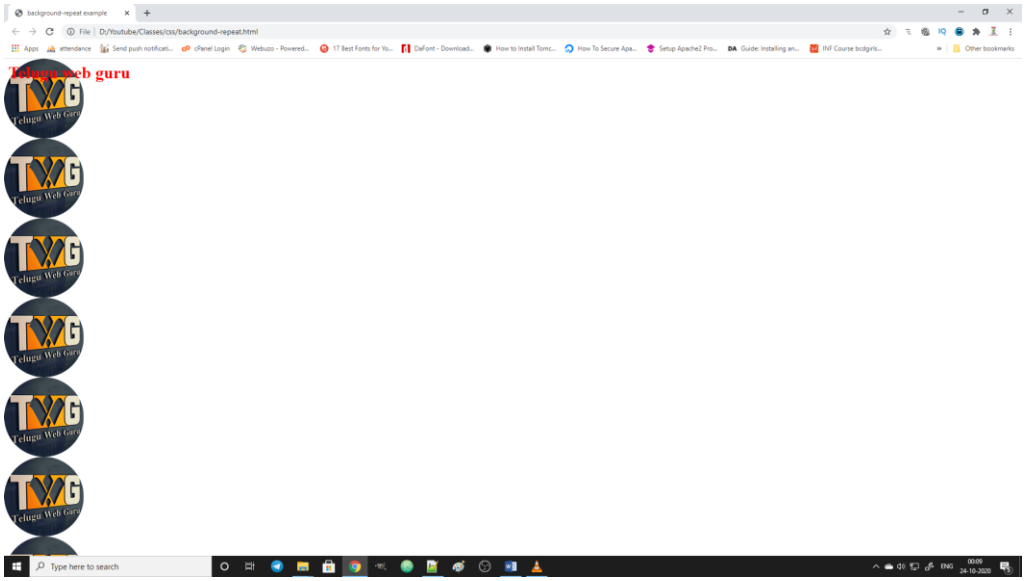
background-repeat: repeat / repeat-x / repeat-y / no-repeat

default one is repeat which indicates repeat the image in both x and y directions. Repeat-x, repeat-y are used to set repeat in only x direction or in only y direction. no-repeat is used to set not to repeat the image.

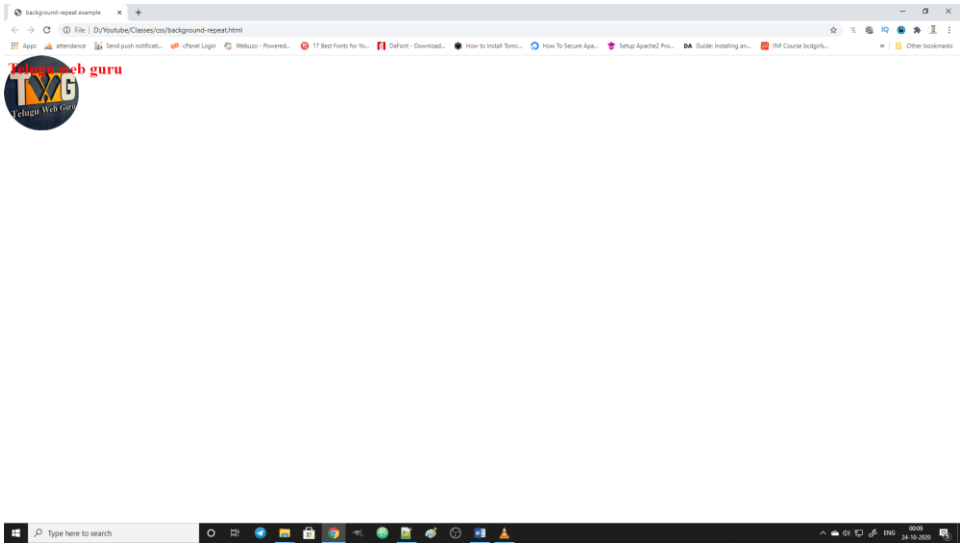
Example

```
<html>
  <head>
        <title>background-repeat example</title>

        <style type="text/css">
          body{
                background-image:url('small_logo.png');
                background-repeat:repeat-x;

              }

              h1  {
          color:red;
        }

          </style>
        </head>

        <body>
          <h1>Telugu web guru</h1>
        </body>
</html>
```

background-repeat:repeat-y outout is:



background-repeat:no-repeat outout is:



observe here that even though image is too smaller than background it is not repeated. Now we can set the image position to center, right , bottom etc by using background-position property

4) background-position : we can decide where you need to set your background image
syntax:

background-position : right top

In right value position you can specify left/right/center and in second value position you can specify top/bottom/center

Example

```
<html>
  <head>
    <title>background-position example</title>

    <style type="text/css">
      body{
        background-image:url('small_logo.png');
        background-repeat:no-repeat;
        background-position:right top;
      }

      h1  {
      color:red;
    }

    </style>
  </head>

  <body>
    <h1>Telugu web guru</h1>
  </body>
</html>
```






Above outputs represents right top, center center, left bottom, center bottom respectively.

5) background-attachment: We can specify whether background image to be scrolled with content or not

syntax :

background-attachment : fixed / scroll

Example

```
<html>
  <head>
        <title>background-position example</title>

        <style type="text/css">
          body{
                background-image:url('small_logo.png');
                        background-repeat:no-repeat;
                        background-position:center center;
                        background-attachment:scroll;
                    }

                h1  {
          color:red;
        }

          </style>
        </head>

        <body>
          <h1>
          What is CSS?
                CSS stands for cascading style sheets. It is used to apply different styles and
display out HTML elements in better way. We can also present our elements with different
styles with attributes inside html tags but it is very limited styles only. So, to make our
webpages more beautiful, we need to use CSS.
CSS Syntax
A CSS rule has two main parts: a selector, and one or more declarations:
The selector is normally the HTML element that you want to apply the style.
Each declaration consists of a property and a value.
The property is the style attribute you want to change. Each property has a value.


Inline Style Sheets:
        Styles that can be applied to one line i.e., one tag are said to be inline style sheets. In
this style can be written inside the tag with the help of style attribute.




Internal Style Sheets:
        Styles that can be applicable to one entire web page is said to be Internal style
```

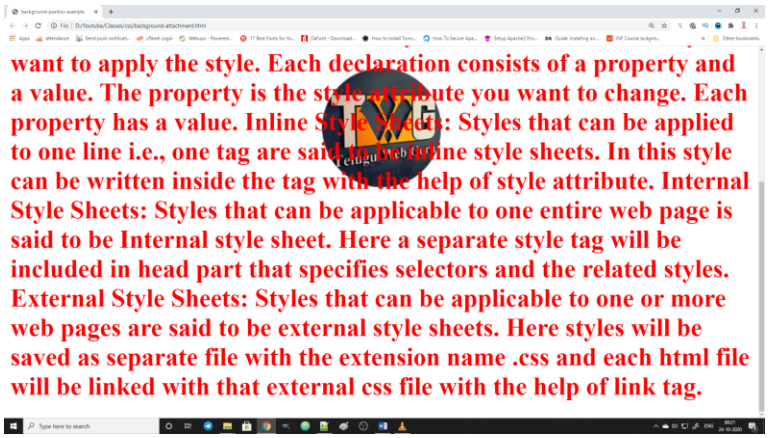
sheet. Here a separate style tag will be included in head part that specifies selectors and the related styles.

External Style Sheets:

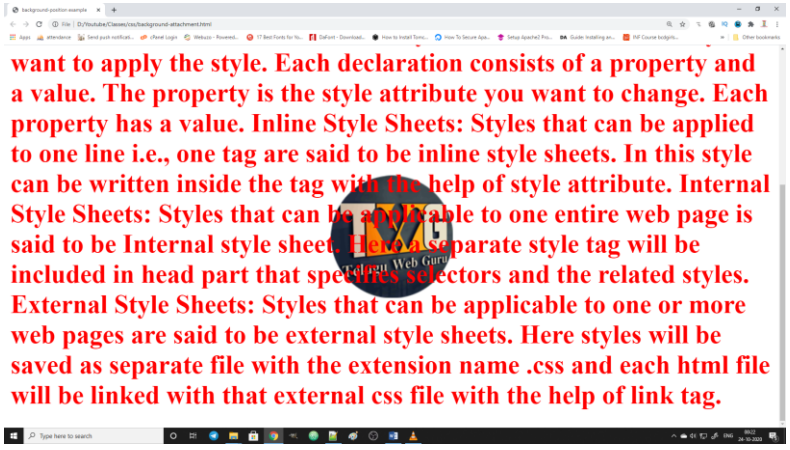
Styles that can be applicable to one or more web pages are said to be external style sheets. Here styles will be saved as separate file with the extension name .css and each html file will be linked with that external css file with the help of link tag.

```
    </h1>
  </body>
</html>
```

output:



observe here the background image also scrolled with content. (with background-attachment:scroll)



observe here the background image also scrolled with content. (with background-attachment:fixed)

Border Properties in CSS

We can set borders to any element in our webpage by using border properties.

We have the following styles related to borders.

1) border-style : with this we can specify the required border style that whether we want a solid border /dashed border etc.,

   syntax :

   border-style : solid / dashed / dotted / double

2) border-width : We can specify the required width of our borders by using this property.
   syntax:
   border-width: pixels value

3) border-color : We can set the color of the borders by using this property
   syntax :
   border-color:color name/ hex value / rgb
4) border : this is a shortcut property where we can set all the above three properties at a time.

   syntax :

   border : border-width border-style border-color ;

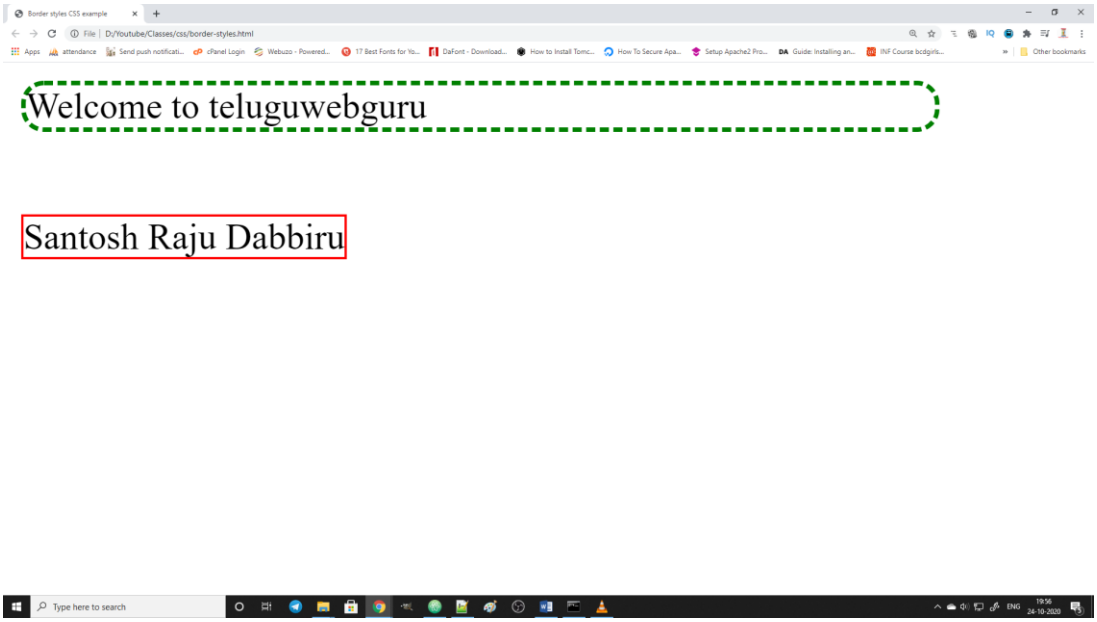
5) border-radius: We will get rounded corners by using this border-radius property.

   Here we can set each border side with different style by using border-top, border-left, border-bottom, border-right properties.

   Example:

```
<html>
  <head>
        <title>Border styles CSS example</title>
        <style type="text/css">
                div{
                    border: 2px dashed green;
                    width:400px;
                    border-radius:10px;
                }
                span{
                    border : 1px solid red;
                        width:400px;
                }
        </style>
```

```
        </head>

        <body>

          <div>Welcome to teluguwebguru</div><br/><br/>
          <span> Santosh Raju Dabbiru </span>
        </body>
</html>
```



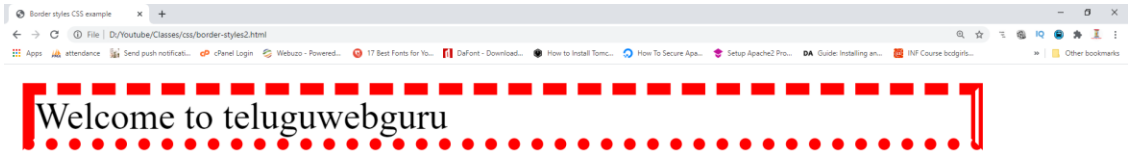
We can set separate border styles to each side of the container by using border-top-style, border-left-style,border-right-style,border-bottom-style

Example

```
<html>
  <head>
          <title>Border styles CSS example</title>
          <style type="text/css">
        div{
                border: 5px solid red;
                width:400px;

                border-top-style:dashed;
                border-left-style:solid;
                border-right-style:double;
                border-bottom-style:dotted;
        }

          </style>
        </head>
```

```
        <body>

          <div>Welcome to teluguwebguru</div><br/><br/>

        </body>
</html>
```

Output



Margin Properties in CSS

Space between screen border and element is said to be margin.

We can set the margin by using margin-left, margin-top,margin-right,margin-bottom.

We can alternatively set all the above 4 properties by using a single shorthand property " margin ".

In place of property value we can pass pixels, percentages, and values 'auto' & 'inherit' also.

Syntax :

margin : margin-top margin-right margin-bottom margin-left

We can set the margin with 3 values also

margin : margin-top margin-left&margin-left margin-bottom

We can set any element to center of the screen by using margin property as follows.

margin : 0 auto auto

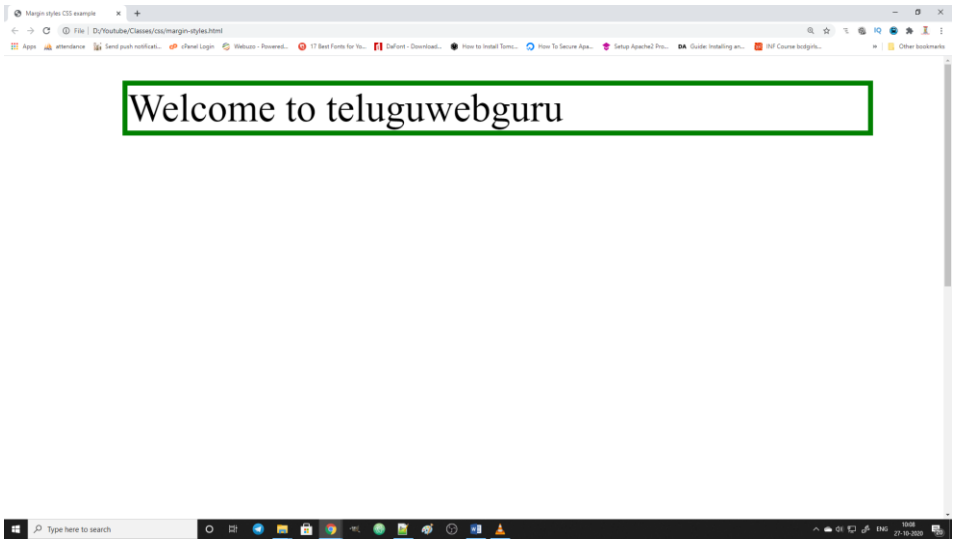
auto will arrange the element to the center of the screen

inherit will copy and apply the same styles that the current element's parent (container) has.

Example:

```
<html>
  <head>
        <title>Margin styles CSS example</title>
        <style type="text/css">
                    div{
                        border: 2px solid green;
                        width:300px;

                         margin:10px 20px 300px 40px;
                    }
        </style>
      </head>

      <body>

        <div>Welcome to teluguwebguru</div><br/><br/>

      </body>
</html>
```

Output:



In the above example element div that contains "welcome to teluguwebguru" is set with top margin 10 and right margin 20 and bottom margin 300 and left margin 40 pixels

Padding Properties in CSS

Padding is the space between element border and inside content. Padding refers to the space that is internal to the element where as margin refers to the space that is external to the element.

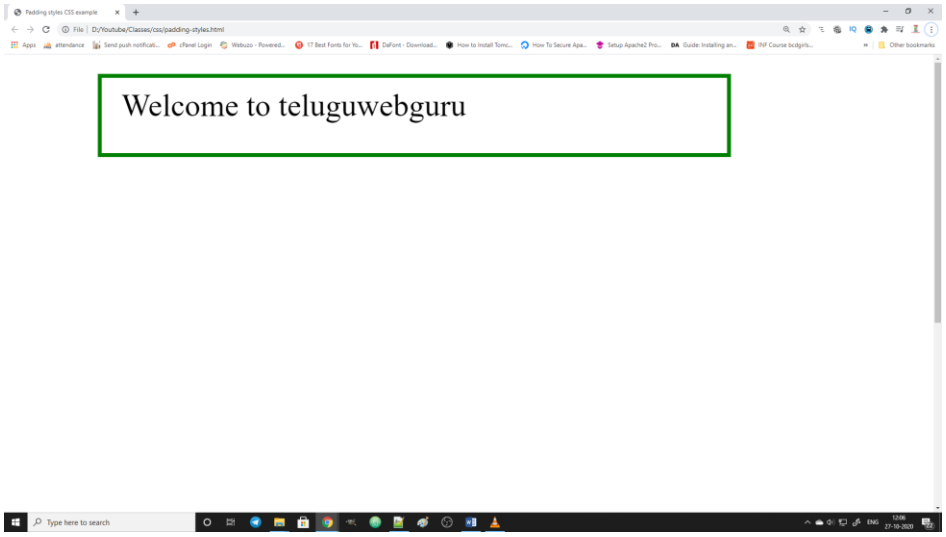
we can set padding (same as margin) by using padding-top,padding-right,padding-bottom,padding-left properties We can set all the 4 at a time by using shorthand property 'padding'.

Syntax:

padding : top right bottom left

Example

```
<html>
  <head>
        <title>Padding styles CSS example</title>
        <style type="text/css">
              div{
                    border: 2px solid green;
                    width:300px;
                    margin:10px 20px 300px 40px;
                    padding : 5px 10px 15px 10px;
                }
        </style>
      </head>
      <body>
        <div>Welcome to teluguwebguru</div><br/><br/>
      </body>
</html>
```



In the above example please observer the space between element border(green border) and content (First letter 'w'). This is called as padding.

Text Properties in CSS

We can present the text in our webpages in different styles.

The following are the important styles related to text styling

1) color : by using this property we can present text in different color in our webpage
2) text-align : We can set alignment of our text using this property with the values left, right, center, justify
3) text-decoration: We can decorate our text as underlined, over lined etc ., by using this property with the possible values underline, overline, line-through, none
4) text-transform: We can transform the text case to uppercase or lowercase by using the possible values uppercase, lowercase, capitalize
5) text-indent: We can display the text with indentation by using this property.
6) letter-spacing: By using this property we can set the space between letters in our webpage.
7) line-height: By using this property we can set the height of each line that is displayed in our webpage
8) word-spacing: We can set space between the words through this property
9) text-shadow: We can set and display shadowed text in our web pages by using this property.

syntax:

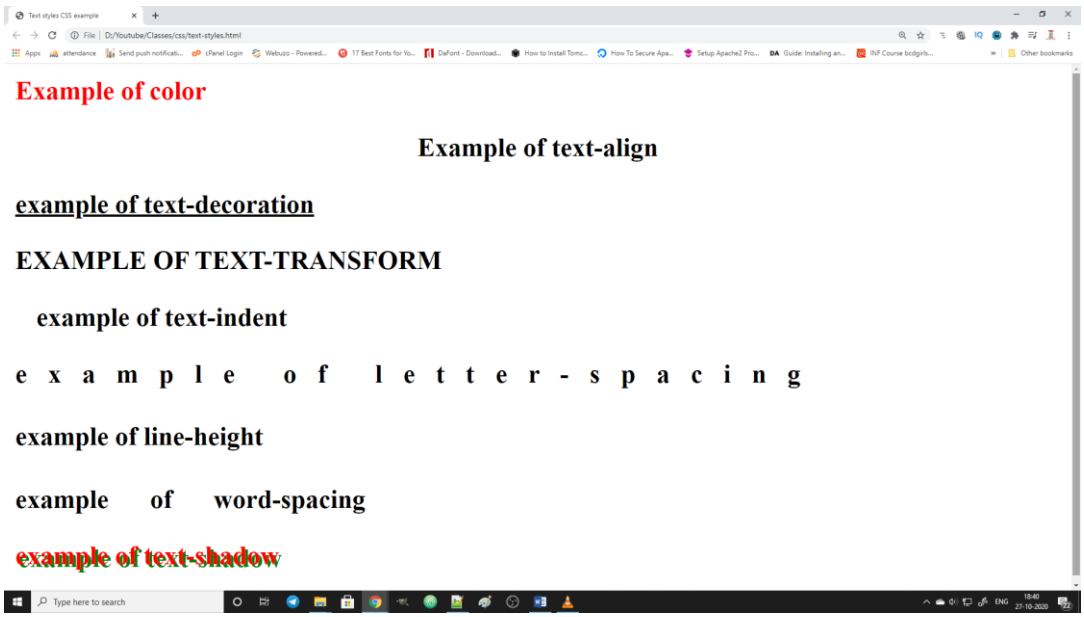
text-shadow:horizontal-space vertical-space color

Example:

```
<html>
  <head>
        <title>Text styles CSS example</title>
        <style type="text/css">
              .color{color:red}
    .textalign{text-align:center}
              .textdecoration{text-decoration:underline}
              .texttransform{text-transform:uppercase}
              .textindent{text-indent:15px}
              .letterspacing{letter-spacing:15px}
              .lineheight{line-height:30px}
              .wordspacing{word-spacing:25px}
              .textshadow{text-shadow:2px 2px green;color:red}
        </style>
      </head>

      <body>

        <h3 class="color">Example of color</h3>
        <h3 class="textalign">Example of text-align</h3>
        <h3 class="textdecoration">example of text-decoration</h3>
        <h3 class="texttransform">example of text-transform</h3>
        <h3 class="textindent">example of text-indent</h3>
        <h3 class="letterspacing">example of letter-spacing</h3>
```

```
            <h3 class="lineheight">example of line-height</h3>
            <h3 class="wordspacing">example of word-spacing</h3>
            <h3 class="textshadow">example of text-shadow</h3>

        </body>
</html>
```

Output



In the above example, we used color property in the first line where it displays the text in red color.

second line text alignment is set to center so it is displayed with center alignment.

third line content is transformed to uppercase so even though the text in html is lower case it is transformed to upper case automatically.

forth line content is set it's indent as 15px so it is displayed after 15px in that line. generally, while inserting new paragraphs in our web page we used to utilize this property and maintain some indent(space) in the first line of the paragraph.

fifth line displayed by setting letter spacing property so each letter is maintained 15px space from the next letter.

sixth line sets the line height as 30px; so 30 pixels height is allotted to this line and in that space the content is displayed

seventh line is displayed with word spacing of 25px so each word will maintain the space of 25px from the next word.

eighth line is displayed with text-shadow. Normal text is displayed in red color and shaded text is displayed in green color with vertical and horizontal space of 2px.

Font Properties in CSS

We can display font in different styles in our webpages by using the following properties

1) font-family : we can select our required font-family like arial, times new roman etc.,
   Syntax
   font-family : "Times New Roman", Times , Serif
   As shown in above syntax we can specify more than one font family with single property. if first font family specified is not available or supported in client system then the next mentioned font family will be considered.

2) font-style: we can display font in italic by using this property.
   syntax :
   font-style:italic/oblique/normal

3) font-size: we can set the font size by using this property

   syntax:
   font-size: value
4) font-weight: we can display font in bold by using this property.

   syntax:

   font-weight:bold/normal

Example:

```
<html>
  <head>
        <title>Font styles CSS example</title>
        <style type="text/css">
              .fontfamily{font-family:"Times New Roman",Times,Serif}
              .fontstyle{font-style:italic}
              .fontsize{font-size:35px;}
              .fontweight{font-weight:bold}

        </style>
      </head>

      <body>

        <h3 class="fontfamily">Example of font-family</h3>
        <h3 class="fontstyle">Example of font-style</h3>
        <h3 class="fontsize">example of font-size</h3>
        <h3 class="fontweight">example of font-weight</h3>


      </body>
</html>
```

Output

The above example sets the font-family to the first line, italic font-style to the second line, font-size applied to third line and forth statement is displayed in bold

Link Properties in CSS

Apart from applying color, fonts, backgrounds to the links, based on link's state we can apply the styles also.

The following are the 4 states related to links

1) a:link – When the link is still unvisited then it is called link state
2) a:visited - If user already visited the link, then it is called visited state.
3) a:hover – If user placed mouse pointer on a link then it is called hover state
4) a:active - a link the moment it is clicked. When user click on a link it will be in active for few seconds and then it will be immediately shifted to active state.

Note :Example program of above states is already explained in selectors concept. please go through it.