Unit-4 Introduction to CSS

CSS stands for **Cascading Style Sheets.**

- ✓ CSS is a language for specifying how documents are presented to users- How they are styled, laid out, etc.
- ✓ A document is usually a text file structured using a markup language. HTML is the most common markup language.
- ✓ Presenting a document to a user means converting it into a form usable by your them.
- ✓ Browsers, like Firefox, Chrome, or Edge, are designed to present documents visually, for example, on a computer screen, projector, or printer.

Initially, without adding a css, What you are seeing are the browser's default styles — very basic styles — that the browser applies to HTML to make sure that the page will be basically readable even if no explicit styling is specified by the author of the page.

The web would be a boring place if all websites looked like that. **Using CSS**, you can control exactly how HTML elements look in the browser, presenting your markup using whatever design you like.

Structure or Syntax of CSS

- ✓ The syntax of CSS is slightly different from that of an HTML. CSS uses **(curly braces { }), (colons :)** and **(semicolon ;)**.

Syntax:

```
selector
{
        property-name 1 : value;
        property-name 2 : value;
        |                       |
        property-name n : value;
}
```

- ✓ In above syntax “selector” can be an element(tag), id, class, attribute, child, universal, descendant etc which will be explained in detail in selector topic.
- ✓ The “property-name 1”, “property-name 2” and so on are the different properties like font-size, background, border, color etc which we want to assigned to different elements.
- ✓ Value is assigned to respective property.

Example:

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

- ✓ **p is a selector(element).** It styles all the <**p**> element of the document with color blue and aligned the content in the center of the screen.
- ✓ **color and text-align are properties**, and **blue and center are the respective property values**.

Types of CSS

1) Inline CSS

- ✓ An inline CSS is used to apply a unique style to a single HTML element.
- ✓ An inline CSS uses the **style attribute** of an element.

Syntax

```
<element style="property-name:value; property-name:value"></element>
```

Example:

```
<html><body>

<h1 style="color:green;font-size:30px;">A Green Heading</h1>

</body></html>
```

- ✓ Above example sets the text color of the <h1> element to green and font size to 30px.

Note : Add **style attribute** for inline css.

2) Internal CSS/Embedded CSS

- ✓ An internal CSS is used to define a style for a single HTML page.
- ✓ An internal CSS is defined in the <head> section of an HTML page, within a <style> element.

Syntax

```
<head>
<style>
Selector{
```

```
        property-name 1 : value;
        property-name 2 : value;
        |                       |
        property-name n : value;
}
</style>
</head>
```

Example:

```
<html>
<head>
<style>
    body {background-color: pink;}
    h1 {color: blue;}
    p {color: purple; font-size:18px;}
</style>
</head>
<body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
</body>
</html>
```

The above example applies the style as below.

- The text color of all the <h1> elements on page to blue,
- The text color of all the <p> elements to purple with 18px font size.
- The page will be displayed with a "pink" background color.

2) External CSS

- ✓ We can also add css file as an external link.
- ✓ Create a separate file with .css extension and should be linked to the HTML document using <link> tag.
- ✓ This means that for each element, style can be set only once and that will be applied across web pages.

```
<link rel="stylesheet" href="style.css" type=”text/css”>
```

- ✓ **<link>** tag is used to link the external style sheet with the html webpage.

- ✓ The rel stands for "relationship", and is one of the key features of the <link> element — the value denotes how the item being linked to is related to the containing document.
- ✓ **href attribute** is used to specify the location of the external style sheet file.

Why to use External CSS?

- It keeps your website design and content separate.
- It's much easier to reuse your CSS code if you have it in a separate file. Instead of typing the same CSS code on every web page you have, simply have many pages refer to a single CSS file with the "link" tag.
- You can make drastic changes to your web pages with just a few changes in a single CSS file.

Example

demo.html

```
<head>
<link rel="stylesheet" href="style.css">
</head>

<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
```

style.css

```
body {background-color: pink;}
h1 {color: blue;}
p {color: purple; font-size:18px;}
```

This is a heading

This is a paragraph.

There are four categories which define the specificity level of a selector(Priority):

1. **Inline styles** - Example: <h1 style="color: pink;"> (Highest Priority)
2. **IDs** - Example: #test
3. **Classes, pseudo-classes, attribute selectors** - Example: .test, :hover, input[type='text']
4. **Elements and pseudo-elements** - Example: h1, ::before

Various CSS Selectors

✓ CSS selectors are used to "find" (or select) the HTML elements you want to style.

1. CSS Element Selector

✓ The element selector selects HTML elements based on the element name.

Syntax:

element-name { property: value; property: value; }

Example:

Here, all <p> elements on the page will be center-aligned, with a red text color:

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

Output:

Test

Hello!

---

2. CSS ID Selector

✓ The id selector uses the **id attribute** of an HTML element to select a specific element.

✓ 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**.

Note: id attribute must begin with a letter and is case sensitive

Synatx:

#element-id{ property1: value; property2: value; }

Example:

```
<html>
<head>
<style>
#para1 {
        text-align: center;
        color: red;
}
</style>
</head>
<body>
        <p id="para1">Paragraph with ID</p>
        <p>Paragraph without ID</p>
</body>
</html>
```

Output:

Paragraph with ID

Paragraph without ID

---

3. CSS Class Selector

- ✓ The class selector selects HTML elements with a specific **class attribute**.
- ✓ To select elements with a specific class, write a **period (.)** character, followed by the **class name**.

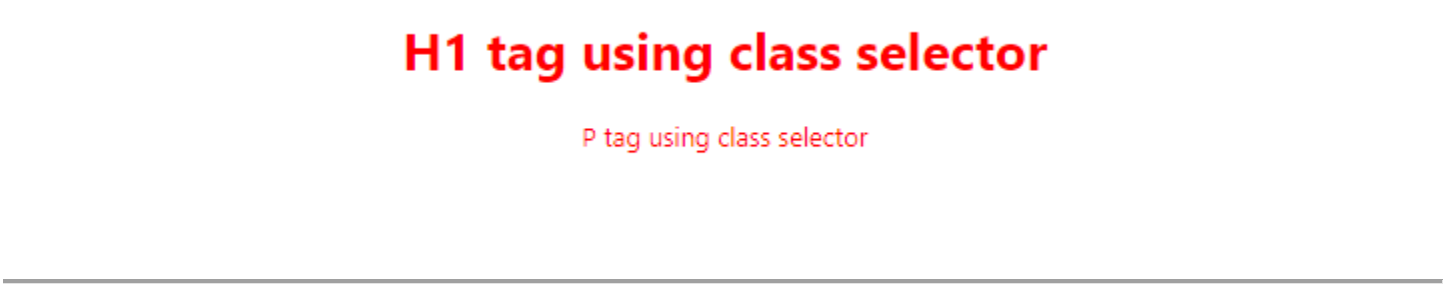
Syntax:

.element-classname { property: value; }

Example1:

```
<html>
<head>
<style>
        .center{
        text-align: center;
```

```
        color: red;
        }
</style>
</head>
<body>
        <h1 class="center">H1 tag using class selector</h1>
        <p class="center">P tag using class selector</p>
</body>
</html>
```



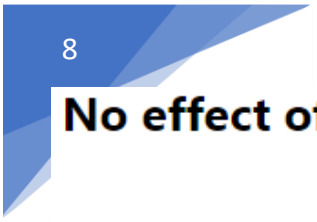
Example 2:

✓ **You can also specify that only specific HTML elements should be affected by a class.**

✓ In this example only <p> elements with class="center" will be affected.

```
<html>
<head>
<style>
        p.center {
        text-align: center;
        color: red;
        }
</style>
</head>
<body>
        <h1 class="center">No effect of center class</h1>
        <p class="center">Red and Center aligned</p>
        <p>No effect of center class </p>
</body>
</html>
```

Output:



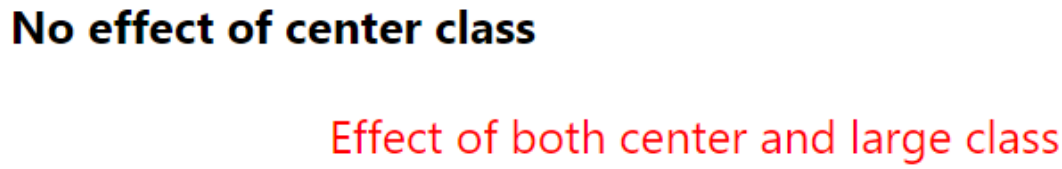
Example 3:

✓ **HTML elements can also refer to more than one class.**

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

```
<html>
<head>
<style>
        p.center {
        text-align: center;
        color: red;
        }
        .large {
        font-size: 200%;
        }
</style>
</head>
<body>
        <h1 class="center">No effect of center class</h1>
        <p class="center large">Effect of both center and large class</p>
</body>
</html>
```

Output:



4. Universal Selector

✓ The universal selector (*) selects all HTML elements on the page.

Syntax:

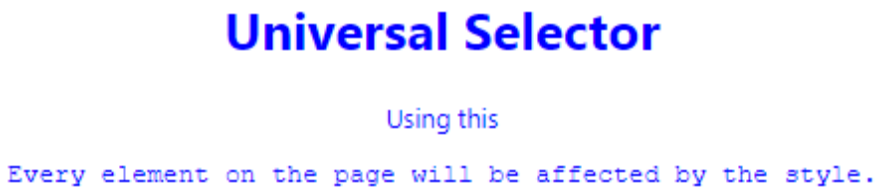
```
* { property: value; }
```

Example:

- ✓ The CSS rule below will affect every HTML element on the page:

```
<html>
<head>
<style>
* {
text-align: center;
color: blue;
}
</style>
</head>
<body>
<h1>Universal Selector</h1>
<p>Using this</p>
<pre>Every element on the page will be affected by the style.</pre>
</body>
</html>
```

Output:



By adding a css to particular element as shown below. It will override the color of p element(s).

```
<style>
* {
text-align: center;
color: blue;
}
p{
  color: red;
}
</style>
```

Universal Selector

Using this

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

---

5. Attribute selector:

- ✓ It is possible to style HTML elements that have specific attributes or attribute values.
- ✓ The [attribute] selector is used to select elements with a specified attribute.
- ✓ The attribute selectors can be useful for styling forms or any other elements using their attribute.

Syntax:

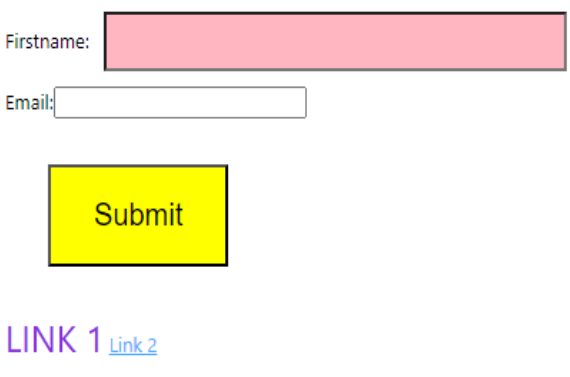
```
element-name[type=”text”]
{
property: value;
}
```

Example:

```
<head>
<style>
input[type=text] {
        width: 300px;
        margin-bottom: 10px;
        background-color: lightpink;
        color:rgb(125, 73, 247);
        padding:10px;
        margin: 10px;
}
input[type=button] {
        margin:30px;
        background-color: yellow;
        padding:20px 30px;
        font-size:20px;
}
a[target=_blank] {
        color: blueviolet;
        font-size: x-large;
        text-decoration: none;
        text-transform: uppercase;
}
</style>
</head>

<body>
<form name="input" action=""
method="get">
Firstname:<input type="text"
name="Name"> <br>
Email:<input type="email"
name="Name"><br>
<input type="button" value="Submit">
</form>
<a href="#" target="_blank">Link 1</a>
<a href="#">Link 2</a>
</body>
```

Output:



CSS Combinators

✓ A combinator is something that explains the relationship between the selectors.
✓ A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

Combinators in CSS:

- **descendant selector (space)**
- **child selector (>)**
- **adjacent sibling selector (+)**

6. Descendant Selector

✓ If some tag is nested in the other tag then nested tag is called as descendant of parent tag.
✓ The Descendant Selectors can be any selector having the white-space in between the elements without using any combinators. Descendant is a manner to nested anywhere within the DOM tree. This selector is used to select all the child elements of the specified tag.

Syntax:

```
element1 element2
{
    property: value;
}
```

Example 1:

```
<head>
<style>
.d1 p {
  background-color: yellow;
 }
 </style>
 </head>
```

```
<body>
 <h2>The descendant selector matches all elements that are descendants of a specified element.</h2>
 <div>
  <p>Paragraph 1 in the div.</p>
  <section><p>Paragraph 2 in the div.</p></section>
 </div>
 <div class="d1"><p>Paragraph 3. is in a div with class d1</p></div>
```

Output:

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

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3. is in a div with class d1

Example 2:

```
<head>
<style>
div.d1 p {
 background-color: lightblue;
}
</style>
</head>
<body>
<div class="d1">
 <p>Paragraph 1 in the div.</p>
  <section>
        <p>Paragraph 2 in the div.</p>
  </section>
  <p>Paragraph 3 in the div.</p>
</div>

<p>Paragraph 4 After a div.</p>
</body>
```

Output:

It applies background color to the all nested **p** elements of the **div** element. (child, grandchild, great grandchild …..)

7. Child Selector >

- ✓ The child selector selects all elements that are the children of a specified element.
- ✓ To apply CSS, nested tag must be a direct child of previous tag.
- ✓ The following example selects all <p> elements that are children of a element:

Syntax:

```
element1 > element2
{
property:value;
}
```

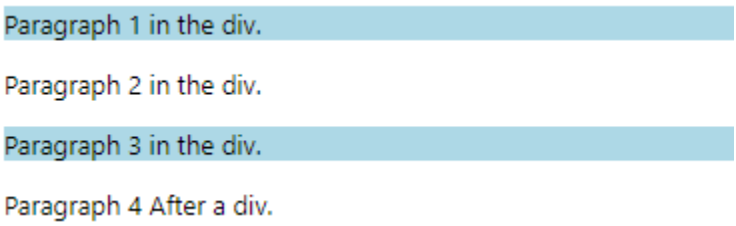
Example 1:

```
<head>
  <style>
  div.d1>p {
    background-color: lightblue;
  }
  </style>
  </head>
  <body>

  <div class="d1">
    <p>Paragraph 1 in the div.</p>
     <section>
       <p>Paragraph 2 in the div.</p>
     </section>
     <p>Paragraph 3 in the div.</p>
  </div>

  <p>Paragraph 4 After a div.</p>
  </body>
```

Output:



- ✓ It applies background color to only direct child p element/s of div element.
- ✓ Grandchild , great grandchild etc will not get affected in child selector.

8. Adjacent Sibling Selector (+)

- ✓ The adjacent sibling selector is used to select an element that is directly after another specific element.
- ✓ Sibling elements must have the same parent element, and "adjacent" means "immediately following".
- ✓ The following example selects the first <p> element that are placed immediately after elements:

Syntax:

```
element1 + element2 { property:value; }
```

Example:

```
<head>
  <style>
  div.d1+p {
   background-color: lightblue;
  }
  </style>
  </head>
  <body>

  <div class="d1">
   <p>Paragraph 1 in the div.</p>
    <section>
      <p>Paragraph 2 in the div.</p>
    </section>
    <p>Paragraph 3 in the div.</p>
```

```
</div>

<p>Paragraph 4 After a div.</p>
</body>
```

Output:

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4 After a div.

Example:

Write CSS to perform the tasks as asked below.

1. **Add unordered list with 3 list items.**
2. **If direct child of the ul element is <b> then apply font color blue and font size should be 20px.**
3. **If direct child of the ul element is <li> then item should be displayed in red color and in smaller font size.**

```
<head>
  <style>
  ul>b {
     font-size:20px;
     color:blue;
  }
  ul>li {
     font-size:smaller;
     color:rgb(255, 0, 43);
  }
  </style>
</head>
<body>
  <ul> <li><b>abc</b></li>
     <li><b>xyz</b></li>
     <b><li>pqr</li></b>
  </ul>
</body>
```

- abc
- xyz
- pqr

CSS colors

- ✓ Colors are specified using predefined color names, or RGB, HSL, RGBA, HSLA values.
- ✓ Colors are specified using HEX code #rr gg bb or Simple #rgb.

  Suppose, hex code for the color is #00ff00 then it can be written as #0f0.
- ✓ Colors are specified using RGB function: rgb(RRR, GGG, BBB)
- ✓ In CSS, a color can be specified by using a predefined color name: Orange, Red, White, Gray etc.

Example

```
<html>
<body>
<h3 style="color:rgb(255, 0, 170);">'उठो, जागो और तब तक मत रुको जब तक मंजिल मिल न जाए'</h3>
<h4 style="color:rgb(25,140,230);">Live life to the fullest, and focus on the positive.</h4>
<h5 style="color: #dfa80f;">Keep Smiling!</h5>
<h2 style="color: #0f0;">Learn something new today</h2>
</body>
</html>
```

Output:

'उठो, जागो और तब तक मत रुको जब तक मंजिल मिल न जाए'

Live life to the fullest, and focus on the positive.

Keep Smiling!

Learn something new today

CSS background properties

The CSS background properties allow you to control various background effects on HTML elements. Here's a breakdown of commonly used background properties and how they work:

Background Rules: background-color

The background-color property specifies the background color of an element.

Example

```
<html>
<head>
<style>
h1 {background-color: rgb(25, 128, 134);}
div {background-color: #FFC0CB;}
p {background-color: #bcd;}
pre {background-color: aqua;}
</style>
</head>
<body>
<h1>CSS background-color</h1>
<div>
        This is a text inside a div element.
        <p>This paragraph has its own background color.</p>
        We are still in the div element.
        <pre>Preformatted text
        with its own
        background color
        </pre>
</div>
</body>
</html>
```

Output:

CSS background-color

This is a text inside a div element.

This paragraph has its own background color.

We are still in the div element.

Preformatted text
with its own
background color

Background Rules: background-image

- ✓ Sets the background image for an element
- ✓ The background-image property specifies an image to use as the background of an
- ✓ element.
- ✓ By default, the image is repeated so it covers the entire element.
- ✓ The background image can also be set for specific elements, like the <p> element.

Note: When using a background image, use an image that does not disturb the text.

Example:

```
<html>
<head>
<style>
body { background-image: url("scenary.jfif"); }
</style>
</head>
<body>
        <h1>BACKGROUND IMAGE</h1>
        <p>By default, the image is repeated so it covers the entire element.</p>
</body>
</html>
```

Output:

Background Rules: background-repeat

- ✓ By default, the background-image property repeats an image both horizontally and vertically.
- ✓ Some images should be repeated only horizontally or vertically, or they will look strange.
- ✓ To repeat an image vertically, set background-repeat: repeat-y;
- ✓ To repeat an image horizontally, set background-repeat: repeat-x;
- ✓ To repeat an image only once, set background-repeat: no-repeat;

Example (repeat-x):

```
<head>
  <style>
  body {
        background-image: url("nature.jfif");
        background-repeat: repeat-X;
        }
  </style>
  </head>
  <body>
        <h1>BACKGROUND IMAGE(REPEAT-X)</h1>
         <p>THE IMAGE IS REPEATED HORIZONATLLY</p>
  </body>
  </html>
```

Output:

Example (repeat-y):

```
<html>
<head>
<style>
body {
        background-image: url("scenary.jfif");
```

```
        background-repeat: repeat-y;
        }
</style>
</head>
<body>
        <h1>BACKGROUND IMAGE(REPEAT-Y)</h1>
        <p>THE IMAGE IS REPEATED VERTICALLY</p>
</body>
</html>
```

Output:

Example (no-repeat)

```
<html>
<head>
<style>
body {
        background-image: url("scenary.jfif");
        background-repeat: no-repeat;
}
</style>
</head>
<body>
        <h1>BACKGROUND IMAGE(no-repeat)</h1>
        <p>THE IMAGE IS REPEATED ONLY ONCE</p>
</body>
</html>
```

Output:

Shorthand Syntax

CSS also allows you to combine multiple background properties into a single shorthand declaration. This shorthand format is useful for specifying background color, image, repeat, position, and more in one line.

Syntax:

background: [color] [image] [repeat];

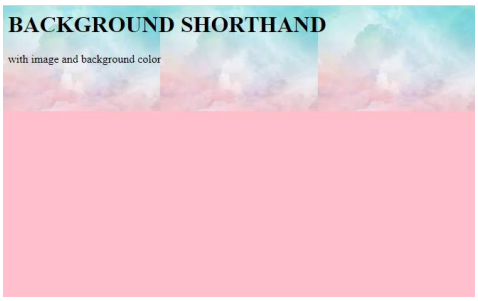
Example:

```
body {
   background: red url("path/to/image.jpg") repeat-x;
}
```

Example (Shorthand):

```
<html>
<head>
<style>
body {
        background: url("scenary.jfif") pink repeat-x;
        }
</style>
</head>
<body>
        <h1>BACKGROUND SHORTHAND</h1>
        <p>with image and background color</p>
</body>
</html>
```

Output:



Text Manipulation properties

1. text-indent

- ✓ The text-indent property specifies the indentation of the first line in a text-block. Defines a fixed indentation in px, pt, cm, em, % etc. Default value is 0.

Note: Negative values are allowed. The first line will be indented to the left if the value is negative.

Example:

```
<html>
<head>
<style>
.a {  text-indent: 50px;  }
.b {  text-indent: -2em; }
.c {  text-indent: 30%; }
</style>
</head>
<body>

<h2 class="a">Hello! how are you? (text-indent: 50px)</h2>
<h3>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Error minima, dolores non aperiam voluptatem sed id magni ipsam nihil perspiciatis, dolore neque quis, ipsa excepturi ducimus eos. Perferendis, repellendus magni. (default)</h3>
<h3 class="b">Everybody Lorem ipsum dolor sit amet consectetur adipisicing elit. (text-indent: -2em) </h3>
<h3 class="c">Good Morning! Lorem ipsum dolor sit amet consectetur adipisicing elit. (text-indent: 30%)</h3>

</body>
</html>
```

Output:

Hello! how are you? (text-indent: 50px)

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Error minima, dolores non aperiam voluptatem sed id magni ipsam nihil perspiciatis, dolore neque quis, ipsa excepturi ducimus eos. Perferendis, repellendus magni. (default)

verybody Lorem ipsum dolor sit amet consectetur adipisicing elit. (text-indent: -2em)

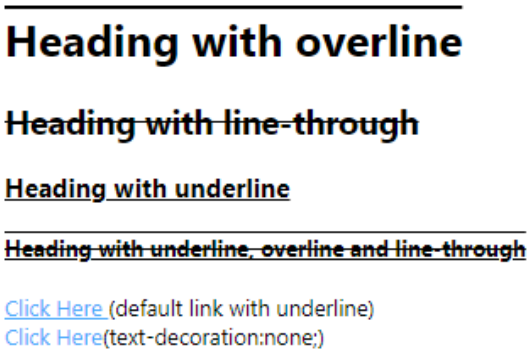
Good Morning! Lorem ipsum dolor sit amet consectetur adipisicing elit. (text-indent: 30%)

2. text-decoration

- ✓ The **text-decoration** property is used to "decorate" the content of the text.
- ✓ It is essentially decorating the text with different kinds of lines.

Example:

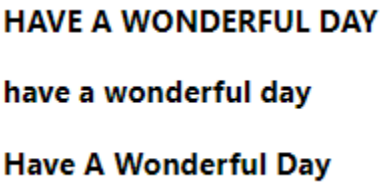
```
<html>
<head>
<style>
h1 { text-decoration: overline; }
h2 { text-decoration: line-through; }
h3 { text-decoration: underline; }
h4 { text-decoration: underline overline; }
h4 { text-decoration: overline underline line-through; }
.a1{ text-decoration: none;}
</style>
</head>
<body>
<h1>Heading with overline</h1>
<h2>Heading with line-through</h2>
<h3>Heading with underline</h3>
<h4>Heading with underline, overline and line-through</h4>
<a href="#">Click Here </a>(default link with underline) <br>
<a href="#" class="a1">Click Here</a>(text-decoration:none;)
</body>
</html>
```



3. text-transform

- ✓ This CSS property allows us to change the case of the text.
- ✓ It is used to control the text capitalization.
- ✓ This CSS property can be used to make the appearance of text in all-lowercase or all-uppercase or can convert the first character of each word to uppercase.
- ✓ Default value is **none**.

```
<html>
<head>
<style>
.a { text-transform: uppercase; }
.b { text-transform: lowercase; }
.c { text-transform: capitalize; }
</style>
</head>
<body>
<h3 class="a">Have a wonderful day</h3>
<h3 class="b">Have a wonderful day</h3>
<h3 class="c">Have a wonderful day</h3>
</body>
</html>
```



4. text-align

- ✓ The **text-align** property in CSS is used to specify the horizontal alignment of text in an element.
- ✓ it is used to set the alignment of the content horizontally, inside a block element.

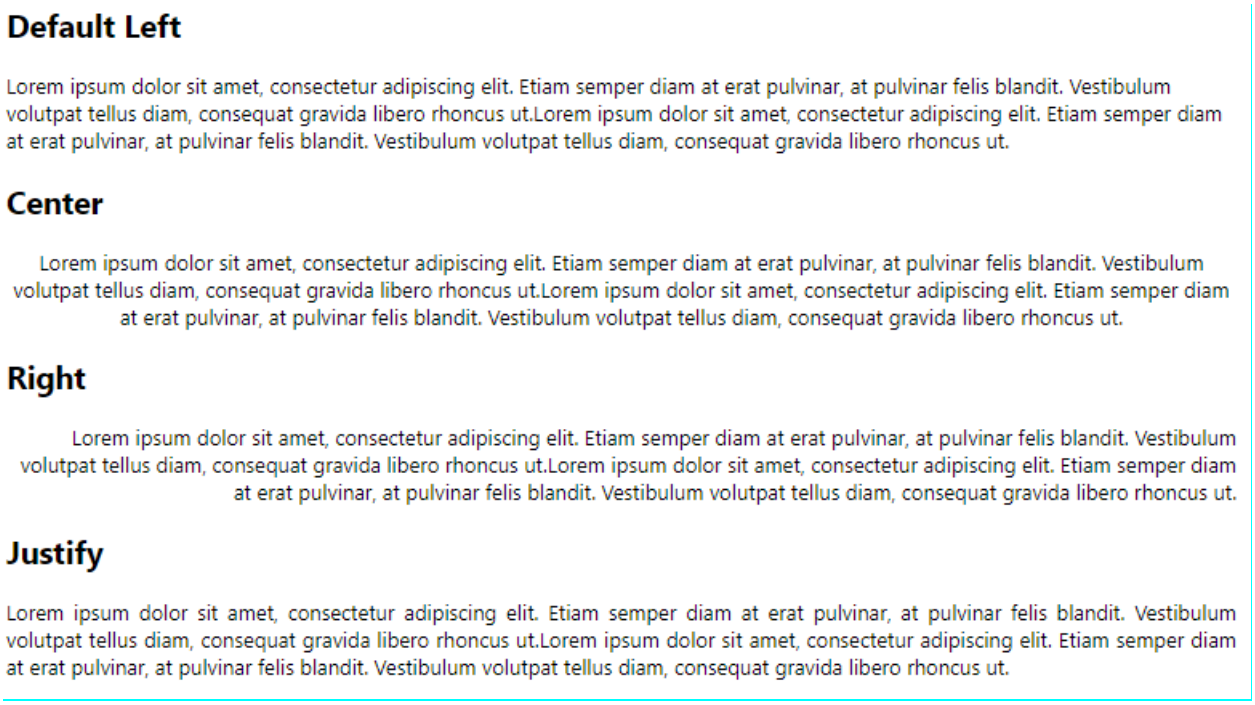
text-align: left|right|center|justify;

- **left:** It is used to set the text-alignment into left. This is the default property.
- **right:** It is used to set the text-alignment to right.
- **center:** It is used to set the text-alignment into the center.
- **justify:** It is used to spreads the words into the complete line i.e., by stretching the content of an element.

Example:

```
<html>
<head>
<style>
.b { text-align: center; }
.c { text-align: right; }
.d { text-align: justify; }
</style>
</head>
<body>
<h2>Default Left</h2>
<p class="a">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p>
<h2>Center</h2>
<p class="b">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p>
<h2>Right</h2>
<p class="c">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p>
<h2>Justify</h2>
<p class="d">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.Lorem ipsum dolor sit amet, consectetur adipiscing
```

```
elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p>
</body>
</html>
```



CSS Fonts

- ✓ CSS Font property is used to control the look of texts. By the use of CSS font property you can change the text size, color, style and more.
- ✓ Choosing the right font has a huge impact on how the readers experience a website.
- ✓ The right font can create a strong identity for your brand.
- ✓ 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.

These are some important font attributes:

- **CSS Font color**: This property is used to change the color of the text. (Already studied before color property to change color of the font)
- **CSS Font family**: This property is used to change the face of the font.
- **CSS Font size**: This property is used to increase or decrease the size of the font.
- **CSS Font style**: This property is used to make the font bold, italic or oblique.
- **CSS Font variant**: This property creates a small-caps effect.

- **CSS Font weight**: This property is used to increase or decrease the boldness and lightness of the font.

❖ font family

CSS font family can be divided in two types:

- **Generic family**: It includes Serif, Sans-serif, and Monospace.
- **Font family**: It specifies the font family name like Arial, New Times Roman etc.

1. **Serif** fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
2. **Sans-serif** fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
3. **Monospace** fonts - here all the letters have the same fixed width. They create a mechanical look.
4. **Cursive** fonts imitate human handwriting.
5. **Fantasy** fonts are decorative/playful fonts.

Generic Family	Font Family
Serif	Times New Roman Georgia Garamond
Sans-serif	Arial Verdana Helvetica
Monospace	Courier New Lucida Console Monaco
Cursive	*Brush Script MT* Lucida Handwriting
Fantasy	Copperplate Papyrus

Note: If the font name is more than one word, it must be in quotation marks, like: "Times New Roman".

Serif: Serif fonts include small lines at the end of characters. Example of serif: Times new roman, Georgia etc.

Sans-serif: A sans-serif font doesn't include the small lines at the end of characters. Example of Sans-serif: Arial, Verdana etc.

Example:

```
<html>
<head>
<style>
.a { font-family: 'Times New Roman', serif; }
.b { font-family: Arial, sans-serif; }
.c { font-family: 'Courier New', monospace; }
.d { font-family: cursive; }
</style>
</head>
<body>
<h2>Serif(Times New Roman)</h2>
<h3 class="a">Lorem ipsum dolor sit amet, consectetur adipiscing elit.</h3>
<h2>sans-serif(Arial)</h2>
<h3 class="b">Lorem ipsum dolor sit amet, consectetur adipiscing elit. </h3>
<h2>monospace(Courier New)</h2>
<h3 class="c">Lorem ipsum dolor sit amet, consectetur adipiscing elit. </h3>
<h2>cursive</h2>
<h3 class="d">Lorem ipsum dolor sit amet, consectetur adipiscing elit. </h3>
</body>
</html>
```

Serif(Times New Roman)

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

sans-serif(Arial)

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

monospace(Courier New)

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

cursive

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

❖ font Size

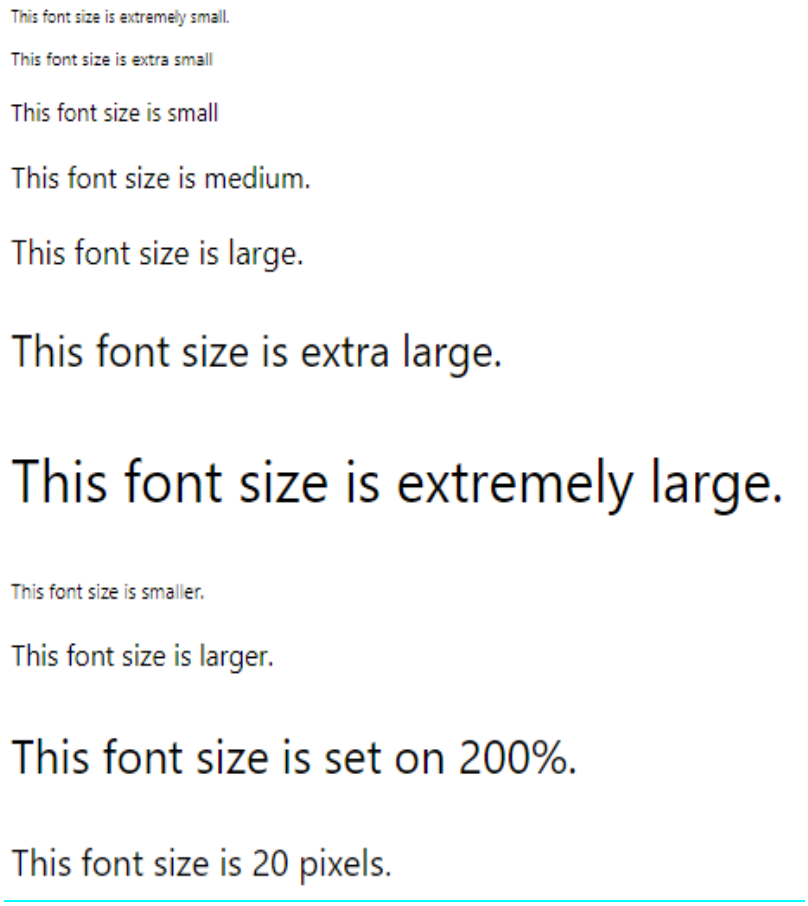
- ✓ CSS font size property is used to change the size of the font.
- ✓ These are the possible values that can be used to set the font size are as under:

Font Size Value	Description
xx-small	used to display the extremely small text size.
x-small	used to display the extra small text size.
small	used to display small text size.
medium	used to display medium text size.
large	used to display large text size.
x-large	used to display extra large text size.
xx-large	used to display extremely large text size.
smaller	used to display comparatively smaller text size.
larger	used to display comparatively larger text size.
size in pixels or %	used to set value in percentage or in pixels.

Example:

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

```
<p style="font-size:xx-small;"> This font size is extremely small.</p>
<p style="font-size:x-small;"> This font size is extra small</p>
<p style="font-size:small;"> This font size is small</p>
<p style="font-size:medium;"> This font size is medium. </p>
<p style="font-size:large;"> This font size is large. </p>
<p style="font-size:x-large;"> This font size is extra large. </p>
<p style="font-size:xx-large;"> This font size is extremely large. </p>
<p style="font-size:smaller;"> This font size is smaller. </p>
<p style="font-size:larger;"> This font size is larger. </p>
<p style="font-size:200%;"> This font size is set on 200%. </p>
<p style="font-size:20px;"> This font size is 20 pixels. </p>
</body>
</html>
```



❖ font Style

- ✓ CSS Font style property defines what type of font you want to display. It may be italic or normal.
- ✓ Default value is normal.

```
<html>
<head>
<style>
h2 { font-style: italic; }
</style>
</head>
<body>
<h2>This heading is shown in italic font.</h2>
<h4>This heading is shown in normal font.</h4>
</body>
</html>
```

This heading is shown in italic font.

This heading is shown in normal font.

❖ font Variant

- ✓ CSS font variant property specifies how to set font variant of an element. It may be normal and small-caps.
- ✓ Default value is normal.
- ✓ In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appear in a smaller font size than the original uppercase letters in the text.

Example:

```
<html>
<head>
<style>
h1 { font-variant: small-caps; }
</style>
</head>
<body>
<h1>This Paragraph Is shown In Small Caps.</h1>
</body>
</html>
```

Output:

THIS PARAGRAPH IS SHOWN IN SMALL CAPS.

❖ font Weight

- ✓ CSS font weight property defines the weight of the font and specify that how bold a font is. The possible values of font weight may be normal, bold, bolder, lighter or number (100, 200..... upto 900).
- ✓ Here we concentrate on bold,bolder and normal values only.
- ✓ Default value is normal.

Example:

```
<html>
<body>
<p style="font-weight:bold;">This font is bold.</p>
<p style="font-weight:bolder;">This font is bolder.</p>
</body>
</html>
```

Output:

This font is bold.

This font is bolder.

Text controls

Letter spacing

- ✓ This CSS property used to control the space between every letter inside an element or the block of text. It modifies the space between the adjacent characters.
- ✓ Using this property, we can increase or decrease the space between the characters of the text.

letter-spacing: normal | length

normal: It is the default value that does not provide any space between the characters. It does not change the default spacing between the letters. It is similar to set the value to 0.

length: It is used to specify an additional space between the characters. It allows the negative values also. The greater length implies the maximum space between the letters. This value supports the font-relative values (em, rem), absolute values (px).

Example:

```
<html>
<body>
<p style= "letter-spacing: normal;">
This paragraph has letter-spacing: normal;
</p>
<p style= "letter-spacing: 7px;">
This paragraph has letter-spacing: 7px;
</p>
<p style= "letter-spacing: 0.7em;">
This paragraph has letter-spacing: 0.7em;
</p>
<p style= "letter-spacing: -1px;">
This paragraph has letter-spacing: -1px;
</p>
</body>
</html>
```

Output:

This paragraph has letter-spacing: normal;

T h i s p a r a g r a p h h a s l e t t e r - s p a c i n g : 7 p x ;

T h i s p a r a g r a p h h a s l e t t e r - s p a c i n g : 0 . 7 e m ;

This paragraph has letter-spacing: -1px;

Word spacing

- ✓ This CSS property is used to control the space between the words.
- ✓ Using this property, we can increase or decrease the space between the words.
- ✓ It is similar to the **letter-spacing** property, but instead of specifying the space between the individual characters, this CSS property defines the space between the words in the piece of text.

A large negative or positive value of **word-spacing** will make the word unreadable. A very large negative value will overlap the word to each other, which makes the word unrecognizable.

word-spacing: normal | length

normal: It is the default value that defines the normal space (0.25em) between the words. It is used to specify the space which is defined by the browser.

length: It specifies an extra space between the words in terms of length (in **pt, px, em, cm,** etc.). It allows negative values.

Example:

```
<html>
<head>
<style>
#space{
word-spacing: 40px;
font-size: 20px;
color: red;
}
h2{  word-spacing: -2px; }
</style>
</head>
<body>
   <p id= "space">
     Lorem ipsum dolor, sit amet consectetur adipisicing elit. Pariatur ducimus nobis consequuntur expedita amet accusantium, perferendis aliquid voluptate vel laudantium assumenda, odit fuga cupiditate excepturi. Dolorem facere vel consectetur deserunt?
   </p>
   <h2>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Dignissimos eligendi eveniet dolorem libero quas ducimus ipsum velit. Non, aspernatur id! Non quos ut magnam dolor ea quisquam labore praesentium officiis?</h2>
</body>
</html>
```

Output:

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Pariatur ducimus nobis consequuntur expedita amet accusantium, perferendis aliquid voluptate vel laudantium assumenda, odit fuga cupiditate excepturi. Dolorem facere vel consectetur deserunt?

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Dignissimos eligendi eveniet dolorem libero quas ducimus ipsum velit. Non, aspernatur id! Non quos ut magnam dolor ea quisquam labore praesentium officiis?

CSS Borders

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

CSS Border Style

The **border-style** property specifies what kind of border to display.
The following values are allowed:

- ✓ dotted - Defines a dotted border
- ✓ dashed - Defines a dashed border
- ✓ solid - Defines a solid border
- ✓ double - Defines a double border
- ✓ none – No border

CSS Border Width

- ✓ The **border-width** property specifies the width of the four borders.
- ✓ The width can be set as a specific size (**in px, pt, cm, em, etc**) or by using one of the three pre-defined values: **thin, medium, or thick**.

CSS Border Color

The **border-color** property is used to set the color of the four borders.
The color can be set by:

- ✓ name - specify a color name, like "red"
- ✓ HEX - specify a HEX value, like "#ff0000"
- ✓ RGB - specify a RGB value, like "rgb(255,0,0)"

CSS Border - Shorthand Property

To shorten the code, it is also possible to specify all the individual border properties in one property.
The border property is a shorthand property for the following individual border properties:

- ✓ **border-width**
- ✓ **border-style (required)**
- ✓ **border-color**

Syntax

```
border: 1px solid red
```

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

- ✓ **border-left:** 6px solid red;
- ✓ **border-bottom:** 6px solid red;
- ✓ **border-top:** 6px solid red;
- ✓ **border-right:** 6px solid red;

Note:

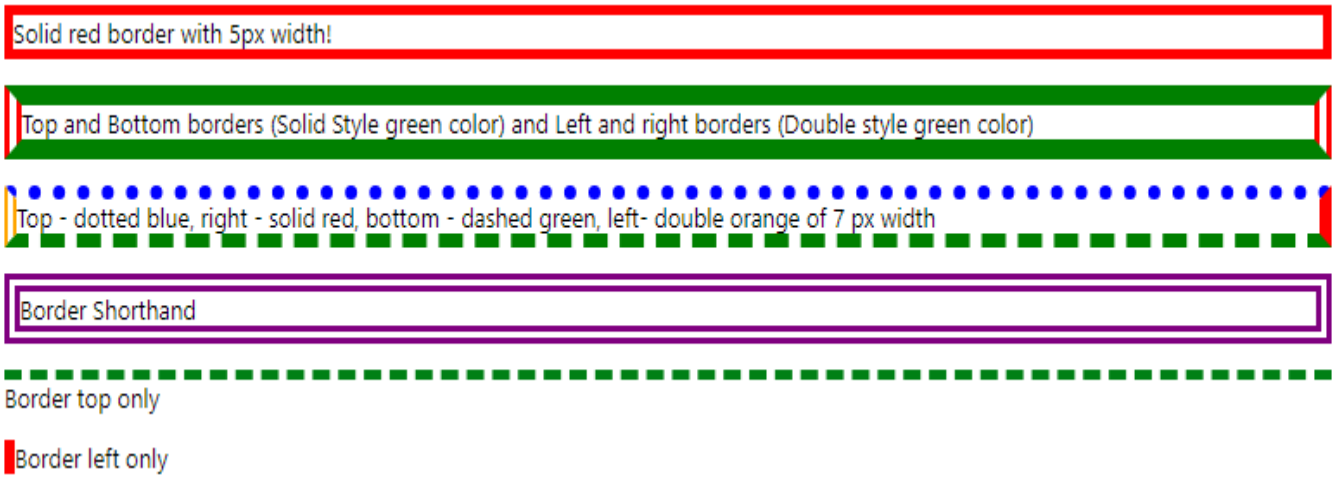
- ✓ The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.
- ✓ If border-color is not set, it inherits the color of the element.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {
  border-style: solid;
  border-color: red;
  border-width:5px;
}
p.two {
  border-style: solid double;
  border-color: green red;
  border-width:10px;
}
p.three {
  border-style: dotted solid dashed double;
  border-color: blue red green orange;
  border-width:7px;
}
p.four {
  border:9px double purple;
}
p.five {
  border-top:5px dashed rgb(0, 128, 21);
}
p.six {
  border-left:6px solid red;
}
</style>
```

```
</head>
<body>
<p class="one">Solid red border with 5px width!</p>
<p class="two">Top and Bottom borders (Solid Style green color) and Left and right borders (Double style green color)</p>
<p class="three">Top - dotted blue, right - solid red, bottom - dashed green, left- double orange of 7 px width</p>
<p class="four">Border Shorthand </p>
<p class="five">Border top only </p>
<p class="six">Border left only </p>
<p><b>Note:</b> The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.</p>
</body>
</html>
```

Output:



Pseudo classes

- ✓ A Pseudo class in CSS is used to define the special state of an element.
- ✓ It can be combined with a CSS selector to add an effect to existing elements based on their states.
- ✓ For Example, changing the style of an element when the user hovers over it, or when a link is visited. All of these can be done using Pseudo Classes in CSS.

Note that pseudo-class names are not case-sensitive.

Syntax:

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

There are many Pseudo-classes in CSS but the ones that are most commonly used are as follows:

:link Pseudo-class:

- ✓ The **:link** CSS pseudo-class represents an element that has not yet been visited.
- ✓ It matches every unvisited <a> or <area> element that has an href attribute.

:hover Pseudo-class:

- ✓ This pseudo-class is used to add a special effect to an element when our mouse pointer is over it.
- ✓ The following example demonstrates that when you hover on the box, its background color changes.

```
<html>
<head>
   <style>
   .box{
      background-color: yellow;
      width: 300px;
      height: 200px;
      font-size: 40px;
      text-align: center;
   }
   .box:hover{
      background-color: orange;
```

```
  }
  </style>
</head>
<body>
  <div class="box">
    My color changes if you hover over me!
  </div>
</body>
</html>
```

:active Pseudo-class:

- ✓ This pseudo-class is used to select an element that is activated when the user clicks on it.
- ✓ The following example demonstrates that when you click on the box, its background color changes for a moment.

```
<html>
<head>
  <style>
  .box{
    background-color: yellow;
    width: 300px;
    height: 200px;
    font-size: 40px;
    text-align: center;
  }
  .box:active{
    background-color: orange;
  }
  </style>
</head>
<body>
  <div class="box">
    My color changes for a moment if you click me!
  </div>
</body>
</html>
```

:focus Pseudo-class:

- ✓ This pseudo-class is used to select an element that is currently focused by the user.
- ✓ It works on user input elements used in forms and is triggered as soon as the user clicks on it.
- ✓ In the following example, the background color of the input field which is currently focused changes.

```
<!DOCTYPE html>
<html>
<head>
  <style>
    input:focus{
      background-color: rgb(38, 184, 25);
    }
  </style>
</head>
<body>
  <form>
    <label for="username">Username:</label>
    <input type="text" name="username" />
  </form>
</body>
</html>
```

:visited Pseudo-class:

- ✓ This pseudo-class is used to select the links which have been already visited by the user.
- ✓ In the following example, the color of the link changes once it is visited.

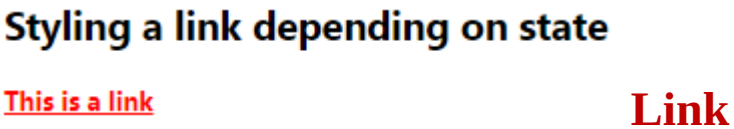
```
<html>
<head>
  <style>
    a:visited{
      color: red;
    }
  </style>
</head>
<body>
   <a href="https://www.google.com/" target="_blank">
     My color changes once you visit this link
   </a>
  </p>
</body>
</html>
```

Example

```
<html>
<head>
<style>
/* unvisited link */
a:link {
color: red;
}
/* visited link */
a:visited {
color: green;
}
/* mouse over link */
a:hover {
color: pink;
}
/* selected link */
a:active {
color: blue;
}
</style>
</head>
<body>
<h2>Styling a link depending on state</h2>
<p><b><a href="https://www.google.com" target="_blank">This is a link</a></b></p>
</body>
</html>
```

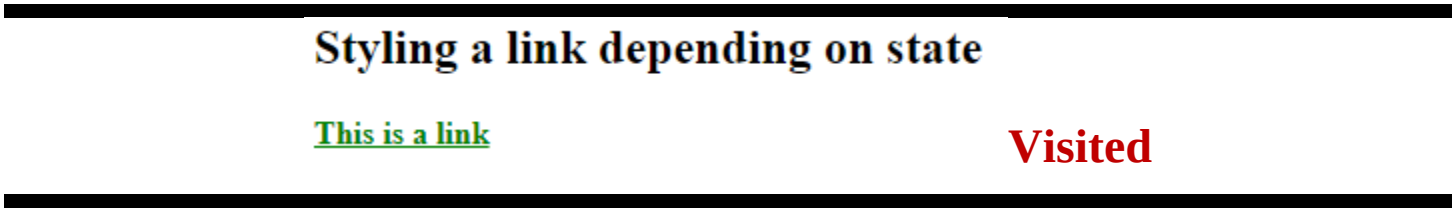
a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.
a:active MUST come after a:hover in the CSS definition in order to be effective.

Output:



Link

Hover



Example

Add one div tag of 100px width and 100px height with color purple. Write css code to change its color to red while you hovering over it.

```
<!DOCTYPE html>
<html>
<head><style>
div{
  height: 100px;
  width: 100px;
  background-color: purple;
}
div:hover {
background-color: red;
}
</style></head>
<body>
<div></div>
</body>
</html>
```

Pseudo Elements

A CSS pseudo-element is used to style specified parts of an element. It can be used to:

- ✓ A pseudo-element can be used to style the first letter or the first line of an element.
- ✓ The pseudo-elements can also be used to insert the content after or before an element.

Syntax

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

We have used the **double colon notation (::pseudo-element)** in the syntax.

In CSS3, the double colon replaced the single colon notation for pseudo-elements. It was an attempt from W3C to differentiate between the pseudo-elements and pseudo-classes. So, it is recommended to use **double colon notation (::pseudo-element)** instead of using single-colon notation **(:)**.

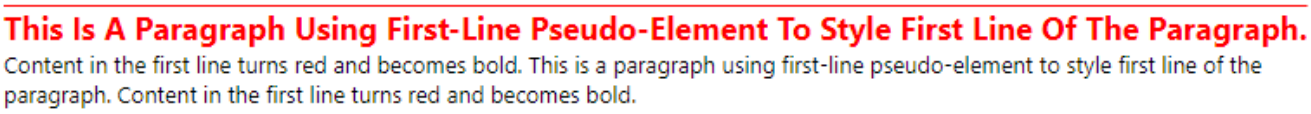
1. **::first-line Pseudo-element**
   - ✓ It applies styles to the first line of a block-level element.
   - ✓ Only a few properties are applied for first-line pseudo-element like font properties, color properties, background properties, word-spacing, letter-spacing, text-decoration, text-transform etc.

   Example:

```
<html>
<head>
   <style>
      p::first-line {
         color: Red;
         font-weight: bold;
         text-decoration: overline;
         text-transform: capitalize;
         font-size: 18px;
      }
   </style>
</head>
<body>
      <p>
      This is a paragraph using first-line pseudo-element to style first line of the paragraph.
Content in the first line turns red and becomes bold. This is a paragraph using first-line
```

```
pseudo-element to style first line of the paragraph. Content in the first line turns red and
becomes bold.
   </p>
</body>
</html>
```

Output:



2. ::first-letter Pseudo-element

- ✓ It applies styles to the first letter of the first line of a block-level element, but only when not preceded by other content (such as images or inline tables).
- ✓ Only a few properties are applied for first-line pseudo-element like font properties, color properties, background properties, word-spacing, letter-spacing, text-decoration, text-transform etc.

Example:

```
<html>
<head>
   <style>
      p::first-letter{
         color: Red;
         font-weight: bold;
         text-decoration: underline;
         text-transform: capitalize;
         font-size: 50px;
      }
   </style>
</head>
<body>
      <p>
      This is a paragraph using first-letter pseudo-element to style first letter of the
paragraph.    </p>
</body>
</html>
```

Output:

This is a paragraph using first-letter pseudo-element to style first letter of the paragraph.

3. **::before Pseudo-element**
   - ✓ It creates a pseudo-element that is the first child of the selected element.
   - ✓ It is often used to add cosmetic content to an element with the **content property.**
   - ✓ It is inline by default.

Example:

```
<html>
<head>
<style>
h1::before
{
content:url("smile.png");
}
</style>
</head>
<body>
<h1>Spread smile wherever you go.....</h1>
</body>
</html>
```

Spread smile wherever you go.....

4. **::after Pseudo-element**
   - ✓ It creates a pseudo-element that is the last child of the selected element.
   - ✓ It is often used to add cosmetic content to an element with the **content property.**
   - ✓ It is inline by default.

Example:

```
<html>
<head>
<style>
```

```
h1::after
{
content:url("smile.png");
}
</style>
</head>
<body>
<h1>Spread smile wherever you go.....</h1>
</body>
</html>
```

Spread smile wherever you go.....

Example:

```
<html>
<head>
   <style>
      p::before{
         content: '"Added before an element"';
         color: red;
         font-size: 30px;
      }
      p::after{
         content: '"Added after an element"';
         color: red;
         font-size: 30px;
      }
   </style>
</head>
<body>
   <p>
      This is a paragraph to which we added red color content using ::before and ::after pseudo elements.
   </p>
</body>
</html>
```

Output:

"Added before an element" This is a paragraph to which we added red color content using ::before and ::after pseudo elements. "Added after an element"

5. **::marker Pseudo-element**
   - ✓ It selects the marker box of a list item, which typically contains a bullet or number.
   - ✓ It works on any element or pseudo-element set to display: list-item, such as the <li> and <summary> elements.

```
<html lang="en">
<head>
   <title>marker Demo</title>
   <style>
      .u1 li::marker {
         color: red;
         font-size: 30px;
      }
      .u2 li::marker {
        content:"#";
        color: rgb(201, 27, 192);
        font-size: 30px;
      }
   </style>
</head>
<body>

   <ul class="u1">
      <li>HTML</li>
      <li>CSS</li>
      <li>JavaScript</li>
   </ul>
   <ul class="u2">
     <li>HTML</li>
     <li>CSS</li>
     <li>JavaScript</li>
  </ul>
</body>
</html>
```

- HTML
- CSS
- JavaScript

#HTML
#CSS
#JavaScript

6. **::selection Pseudo-element**
   - ✓ It applies styles to the part of a document that has been highlighted by the user such as clicking and dragging the mouse across the text.

Example:

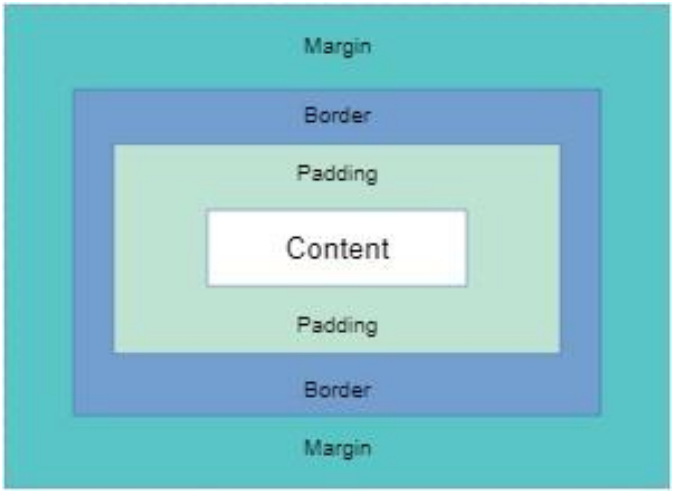
```
<html lang="en">
<head>
  <title>marker Demo</title>
  <style>
    ::selection{
      color: aqua;
      background-color: blueviolet;
    }
    .u2 li::selection{
      background-color: yellowgreen;
      color: blue;
    }
  </style>
</head>
<body>
  <ul class="u1">
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>
  </ul>
  <ul class="u2">
   <li>HTML</li>
   <li>CSS</li>
   <li>JavaScript</li>
```

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

- HTML
- CSS
- JavaScript

- HTML
- CSS
- JavaScript

CSS Box Model



- ✓ The **CSS box model** is a container that contains multiple properties including borders, margins, padding, and the content itself.
- ✓ It is used to create the design and layout of web pages.

Box-Model has multiple properties in CSS. Some of them are given below:

- **content**: This contains the actual data in the form of text, images, or other media forms and it can be sized using the width & height property.
- **padding**: This property is used to create space around the element, inside any defined border.
- **border**: This property is used to cover the content & any padding, & also allows setting the style, color, and width of the border.
- **margin**: This property is used to create space around the element ie., around the border area.

CSS Padding

- ✓ The CSS padding properties are used to generate space around an element's content, inside of any defined borders.
- ✓ There are properties for setting the padding for each side of an element (top, right, bottom, and left).

Padding - Individual Sides

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

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

All the padding properties can have the following values:

- *length* - specifies a padding in px, pt, cm, etc.
- *%* - specifies a padding in % of the width of the containing element

Padding - Shorthand Property

- ✓ To shorten the code, it is possible to specify all the padding properties in one property.

✓ The padding property is a shorthand property for all the individual padding properties.

1. **If the padding property has four values:**

padding: 25px 50px 75px 100px;

- top padding is 25px
- right padding is 50px
- bottom padding is 75px
- left padding is 100px

2. **If the padding property has three values:**

padding: 25px 50px 75px;

- top padding is 25px
- right and left paddings are 50px
- bottom padding is 75px

3. **If the padding property has two values:**

padding: 25px 50px;

- top and bottom paddings are 25px
- right and left paddings are 50px

4. If the padding property has one value:

padding: 25px;

- all four paddings are 25px

CSS Margins

Margins are used to create space around elements, outside of any defined borders.

Margin - Individual Sides
CSS has properties for specifying the margin for each side of an element:

- ✓ margin-top
- ✓ margin-right
- ✓ margin-bottom
- ✓ margin-left

All the margin properties can have the following values:

- auto - the browser calculates the margin
- *length* - specifies a margin in px, pt, cm, etc.
- *%* - specifies a margin in % of the width of the containing element

Note: Negative values are allowed.

Margin - Shorthand Property

- ✓ To shorten the code, it is possible to specify all the margin properties in one property.
- ✓ The margin property is a shorthand property for all the individual margin properties.

1. If the margin property has four values:

margin: 25px 50px 75px 100px;

- top margin is 25px
- right margin is 50px
- bottom margin is 75px
- left margin is 100px

2. If the margin property has three values:

margin: 25px 50px 75px;

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

3. If the margin property has two values:

margin: 25px 50px;

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

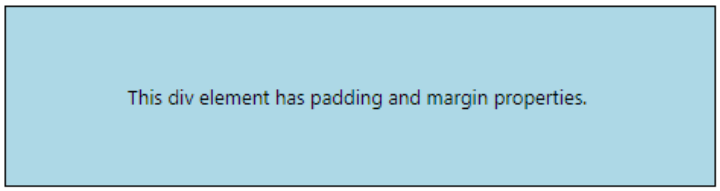
4. If the margin property has one value:

margin: 25px;

- all four margins are 25px

Example:

```
<html>
<head>
<style>
div {
  border: 1px solid black;
  padding: 50px 80px;
  background-color: lightblue;
  margin: 50px 200px 20px 100px;
}
</style>
</head>
<body>
<div>This div element has padding and margin properties.</div>
</body>
</html>
```



CSS Position

- ✓ The **CSS position property** is used *to set position for an element*. it is also used to place an element behind another and also useful for scripted animation effect.
- ✓ You can position an element using the top, bottom, left and right properties.
- ✓ These properties can be used only after position property is set first. A position element's computed position property is relative, absolute, fixed or sticky.

CSS positioning:

- ✓ CSS Static Positioning
- ✓ CSS Fixed Positioning
- ✓ CSS Relative Positioning
- ✓ CSS Absolute Positioning

1) CSS Static Positioning

- ✓ This is a by default position for HTML elements. It always positions an element according to the normal flow of the page.
- ✓ It is not affected by the top, bottom, left and right properties.

2) CSS Fixed Positioning

- ✓ The fixed positioning property helps to put the text fixed on the browser.
- ✓ This fixed test is positioned relative to the browser window, and doesn't move even you scroll the window.

3) CSS Relative Positioning

- ✓ The relative positioning property is used to set the element relative to its normal position.

4) CSS Absolute Positioning

- ✓ The absolute positioning is used to position an element relative to the first parent element that has a position other than static.
- ✓ If no such element is found, the containing block is HTML.
- ✓ With the absolute positioning, you can place an element anywhere on a page.

sticky position

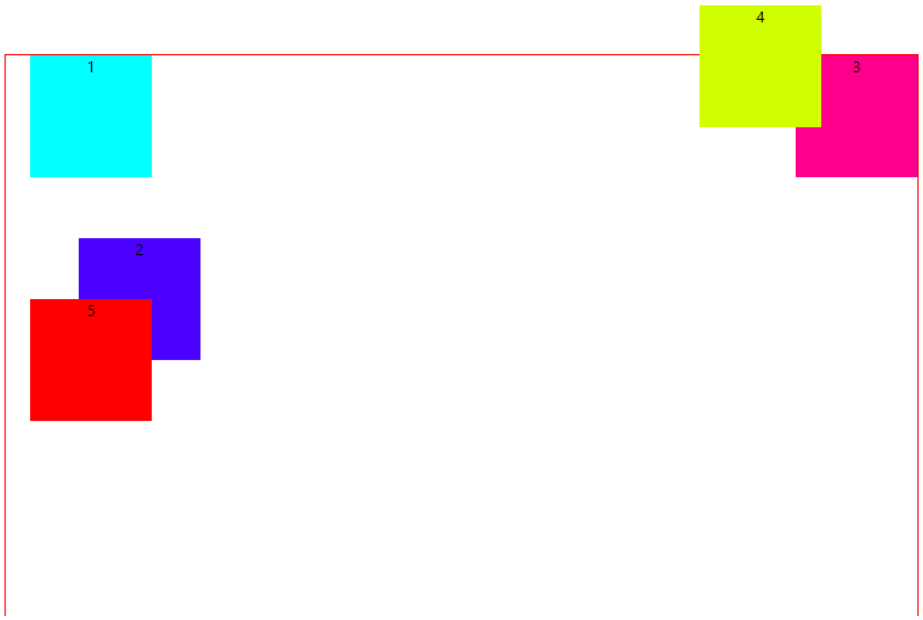
The element is positioned based on the user's scroll position

A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

Example to understand position

```
<html>
<head>
   <style>
.container{
   margin-top: 50px;
   height: 800px;
   border: 1px solid red;
   position: relative;
}
.container1{
   margin-top: 30px;
   height: 1000px;
   border: 2px solid black;
}
.test {
   margin-left: 20px;
   height: 100px;
   width : 100px;
   text-align: center;
}
#test1{
   background-color: aqua;
}
#test2{
   background-color: rgb(76, 0, 255);
   position: relative;
   top: 50px;
   left:40px;
   z-index: 1;
}
#test3{
   background-color: rgb(255, 0, 140);
   position: absolute;
   top: 100px;
```

```
    right: 0px;
}
#test4{
    background-color: rgb(208, 255, 0);
    position: fixed;
    top: 50px;
    right: 0;
}
#test5{
    background-color: rgb(255, 0, 0);
    position: sticky;
    top: 0px;
}
    </style>
</head>
<body>
    <div class="container">
        <div class="test" id="test1">1</div>
        <div class="test" id="test2">2</div>
        <div class="test" id="test3">3</div>
        <div class="test" id="test4">4</div>
        <div class="test" id="test5">5</div>
    </div>
    <div class="container1"></div>
</body>
</html>
```



Z-index property

- ✓ Z Index (z-index) is a CSS property that defines the order of overlapping HTML elements.
- ✓ Elements with a higher index will be placed on top of elements with a lower index.
- ✓ The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).
- ✓ Z index only works on **positioned** elements. **(position:absolute, position:relative, or position:fixed).**
- ✓ An element can have a positive or negative stack order:

Syntax:
z-index: auto|number|initial|inherit;

Property values:

- ✓ **number:** It means that the element's stack level is set to the given value. It allows negative values.
- ✓ **initial:** Sets the property to its default value.
- ✓ **inherit:** Inherits the property from the parent element.
- ✓ **auto:** It means that the order of the stack is equivalent to the parent, i.e., default.

Example

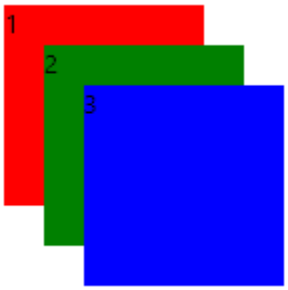
```
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>z-index Example</title>
 <style>
  .box {
   width: 100px;
   height: 100px;
   position: absolute;
  }

  .box1 {
   background-color: red;
   left: 30px;
   top: 30px;
   z-index: 1;
  }

  .box2 {
   background-color: green;
   left: 50px;
```

```
    top: 50px;
    z-index: 2;
   }

   .box3 {
    background-color: blue;
    left: 70px;
    top: 70px;
    z-index: 3;
   }
  </style>
</head>
<body>
  <div class="box box1">1</div>
  <div class="box box2">2</div>
  <div class="box box3">3</div>
</body>
```



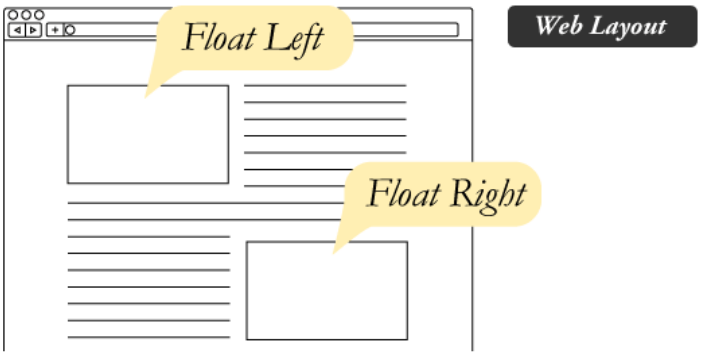
Example:

```
<head>
<style>
h1{
position:relative;
color:blue;
top: 60px;
z-index: 1;
}
</style> </head>
<body>
<h1>Hello</h1>
<img src="smiley.png" width="200" height="200"/>
</body>
</html>
```

In this example image is placed below the heading. If we move heading 60px from the top by giving a relative position it will move behind the image. To make it visible we have to add z-index property.

CSS Float

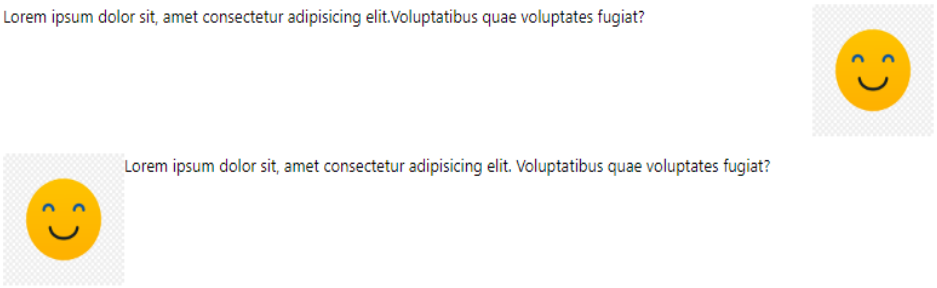
- ✓ The **CSS float property** is *a positioning property*. It is used *to push an element to the left or right*, allowing another element to wrap around it.
- ✓ It is generally used with images and layouts.
- ✓ Elements are floated only horizontally.
- ✓ So, it is possible only to float elements left or right, not up or down.



Example:

```
<html>
<head>
  <title>Float Left</title>
  <style>
     .i1 {float: right;height: 100px;}
     .i2 {float: left;height: 100px;}
     div{height: 100px;}
  </style>
</head>
```

```
<body>
    <div>
      <img src="smiley.png" class="i1">
      <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit.Voluptatibus quae voluptates fugiat?</p>
    </div>
    <div>
      <img src="smiley.png" class="i2">
      <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Voluptatibus quae voluptates fugiat?</p>
    </div>
</body>
</html>
```



Example

Write HTML and CSS script to display two sections of 40% width using div tag. Both these sections are having title and description. Both these div's should be horizontally adjacent to each other.

```
<html>
<head>
<style>
 .d1{
   width:40%; border:2px solid green; text-align:justify;float:left; margin:20px 5px;padding: 10px;
 }
 .d2{
   width:40%; border:2px dashed red; float:left; text-align:justify; margin:20px 5px;padding: 10px;
 }
 h1{
   text-align: center;
```

```
 }
</style>
</head>
<div class="d1">
 <h1>INDIA</h1>
 <p>India, officially the Republic of India (Hindi: Bhārat Gaṇarājya),[26] is a country in South Asia. It is the seventh-largest country by area, the second-most populous country, and the most populous democracy in the world. Bounded by the Indian Ocean on the south, the Arabian Sea on the southwest, and the Bay of Bengal on the southeast, it shares land borders with Pakistan to the west;[f] China, Nepal, and Bhutan to the north; and Bangladesh and Myanmar to the east. In the Indian Ocean, India is in the vicinity of Sri Lanka and the Maldives; its Andaman and Nicobar Islands share a maritime border with Thailand, Myanmar, and Indonesia. The nation's capital city is New Delhi.</p>
</div>
<div class="d2">
 <h1>TAJ MAHAL</h1>
 <p>The Taj Mahal, is an Islamic ivory-white marble mausoleum on the right bank of the river Yamuna in the Indian city of Agra. It was commissioned in 1632 by the Mughal emperor Shah Jahan to house the tomb of his favourite wife, Mumtaz Mahal; it also houses the tomb of Shah Jahan himself. The tomb is the centrepiece of a 17-hectare (42-acre) complex, which includes a mosque and a guest house, and is set in formal gardens bounded on three sides by a crenellated wall.The Taj Mahal was commissioned by Shah Jahan in 1631, to be built in the memory of his wife Mumtaz Mahal, who died on 17 June that year, while giving birth to their 14th child, Gauhara Begum.</p>
</div>
</html>
```