Full Stack Development Using Javascript-1

Unit-4 Introduction to CSS

Significance of CSS

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

Structure, syntax and types of CSS

- To apply style(formatting) to specific tags, we can use CSS.
- The syntax of CSS is slightly different from that of an HTML. Instead of using (< & >), =, &, " ", CSS uses curly braces, colons and semicolon.
- **Syntax:**
  Selector
  {
  Property1:value;
  Property2:value;

  }

- In above syntax "selector" is the element(tag) that the rule defines and "property1", "property2" are the values assigned to these properties.

Example:

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

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

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 HTML element.
- The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:

Example:

```
<html>
    <body>

        <h1 style="color:green;">A Green Heading</h1>

        <p style="color:red;">A red paragraph.</p>

    </body>
</html>
```

Output:

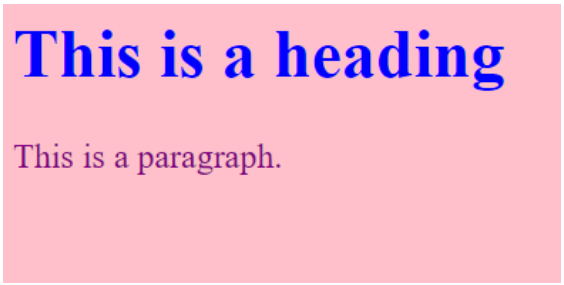
2) Internal Style

- 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.
- The following example sets the text color of ALL the <h1> elements (on that page) to blue, and the text color of ALL the <p> elements to purple. In addition, the page will be displayed with a "pink" background color:

Example:

```
<html>
    <head>
    <style>
        body {background-color: pink;}
        h1{color: blue;}
        p{color: purple;}
    </style>
    </head>
<body>

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

</body>
</html>
```

Output:



3) External CSS

- External CSS contains separate CSS file which contains only style property with the help of tag attributes (For example class, id, heading, … etc). CSS property written in 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.

Below is the HTML file that is making use of the created external style sheet

- **link** tag is used to link the external style sheet with the html webpage.
- **href** attribute is used to specify the location of the external style sheet file.

Example

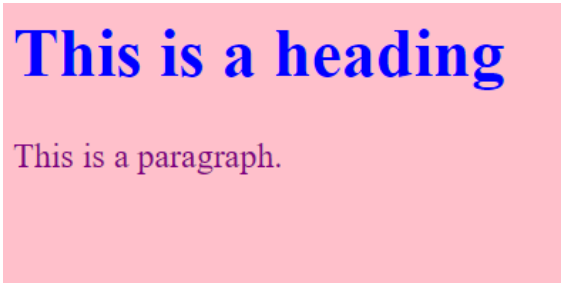
demo.html

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

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

</body>
</html>
```

style.css

```
body {
  background-color: pink;
     }
h1 {
  color: blue;
   }
p {
  color: purple;
  }
```

Output:



Various CSS Selectors:

- A CSS selector selects the HTML element(s) you want to style.
- 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

{

CSS property: value; CSS 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: blue;
}
</style>
</head>
<body>
<p>This is element selector and we have selected paragraph to provide style<p>
<p>Both paragraph will be in blue color</p>
</body>
</html>
```

Output:

This is element selector and we have selected paragraph to provide style

Both paragraph will be in blue color

2) 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: *An id name cannot start with a number!*

Synatx:

#element-id{

CSS property1: value-1;

CSS property2: value-2;

}

Example:

```
<html>
<head>
<style>
#para1 {
  text-align: center;
  color: green;
}
</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) 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

{

css property: value;

}

Example 1:

```
<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>
```

Output:

P tag using class selector

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 red and center-aligned:

```
<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 class="center"> Red and Center aligned</p>

</body>
</html>
```

Output:

No effect of center class

Red and Center aligned

Red and Center aligned

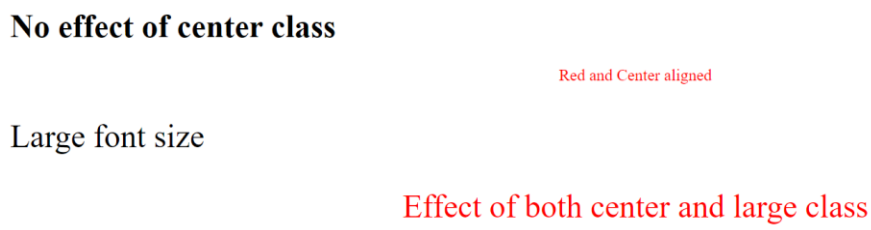
Example 3

- 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;
}

p.large {
  font-size: 200%;
}
```

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

<h1 class="center">No effect of center class</h1>
<p class="center">Red and Center aligned</p>
<p class="large">Large font size</p>
<p class="center large">Effect of both center and large class</p>

</body>
</html>
```

Output:



Note: *A class name cannot start with a number!*

4) Universal Selector

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

Syntax:

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

Example:

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

```
<html>
<head>
<style>
* {
  text-align: center;
  color: brown;
}
</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:

5) Descendant selector

- If some tag is nested in the other tag then nested tag is called as descendant of parent tag.

Syntax:

element1 element2

{

css property: value;

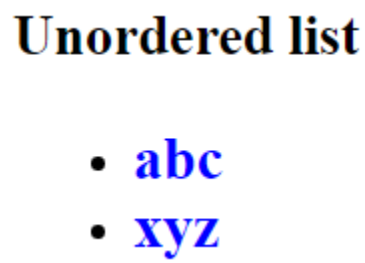
}

Example:

```
<html>
  <head>
    <style>
      ul b
      {
      font-size:20px;
      color:blue;
      }
    </style>
  </head>
```

```
  <body>
    <h3>Unordered list</h3>
    <ul>
    <li><b>abc</b></li>
    <li><b>xyz</b></li>
    </ul>
  </body>
</html>
```

Output:



6) 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.

Syntax:

element1>element2

{

css property:value;

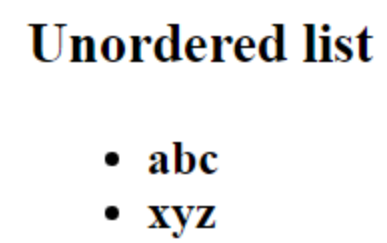
}

Example 1:

```
<html>
  <head>
    <style>
      ul>b
      {
      font-size:20px;
      color:blue;
      }
    </style>
```

```
  </head>

  <body>
    <h3>Unordered list</h3>
    <ul>
    <li><b>abc</b></li>
    <li><b>xyz</b></li>
    </ul>
  </body>
</html>
```

Output:

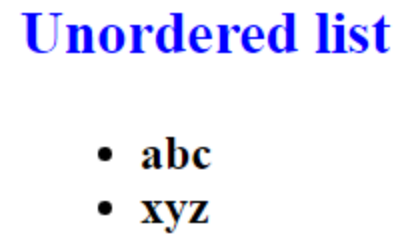


Note: CSS will not work for example 1 as nested tag <b> is not a direct child of <ul> tag.

Example 2:

```
<html>
  <head>
    <style>
      body>h3
      {
      font-size:20px;
      color:blue;
      }
    </style>
  </head>

  <body>
    <h3>Unordered list</h3>
    <ul>
    <li><b>abc</b></li>
    <li><b>xyz</b></li>
    </ul>
  </body>
</html>
```

Output:



7) 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 without class or ID:

Syntax:

element-name[type=”text”]

{

css property: value;

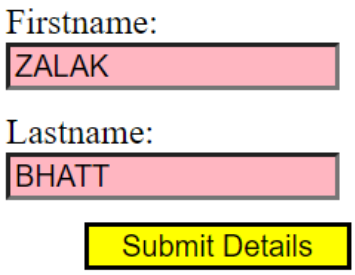
}

Example:

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

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

```
<form name="input" action="" method="get">
  Firstname:<input type="text" name="Name" value="ZALAK" size="20">
  Lastname:<input type="text" name="Name" value="BHATT" size="20">
  <input type="button" value="Submit Details">
</form>

</body>
</html>
```

Output:



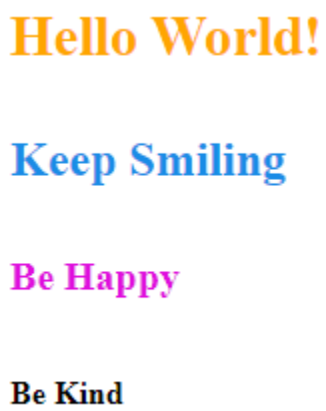
CSS colors

- Colors are specified using predefined color names, or RGB, HSL, RGBA, HSLA values.
- Colors are specified using HEX code #rr gg bb, Simple #rgb.
- 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:orange;">Hello World!</h3>

<h4 style="color:rgb(25,140,230);">Keep Smiling</h4>

<h5 style="color: #df0fdd;">Be Happy</h5>

<h6 style="color: #000;">Be Kind</h6>

</body>
</html>
```

Output:



Background Rules: background-color

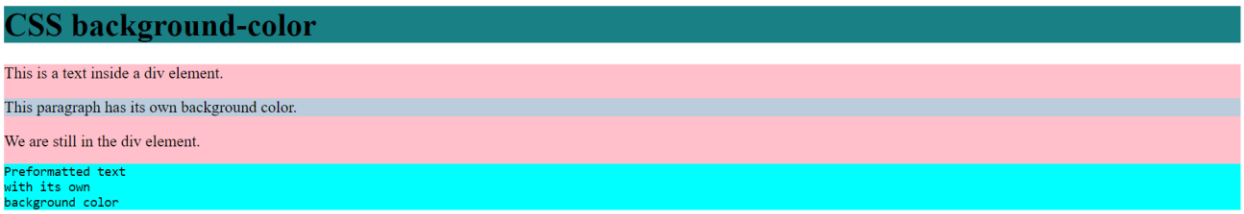
- The background-color property specifies the background color of an element.

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:



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("D:/Full Stack/flwr.jpg");
}
</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

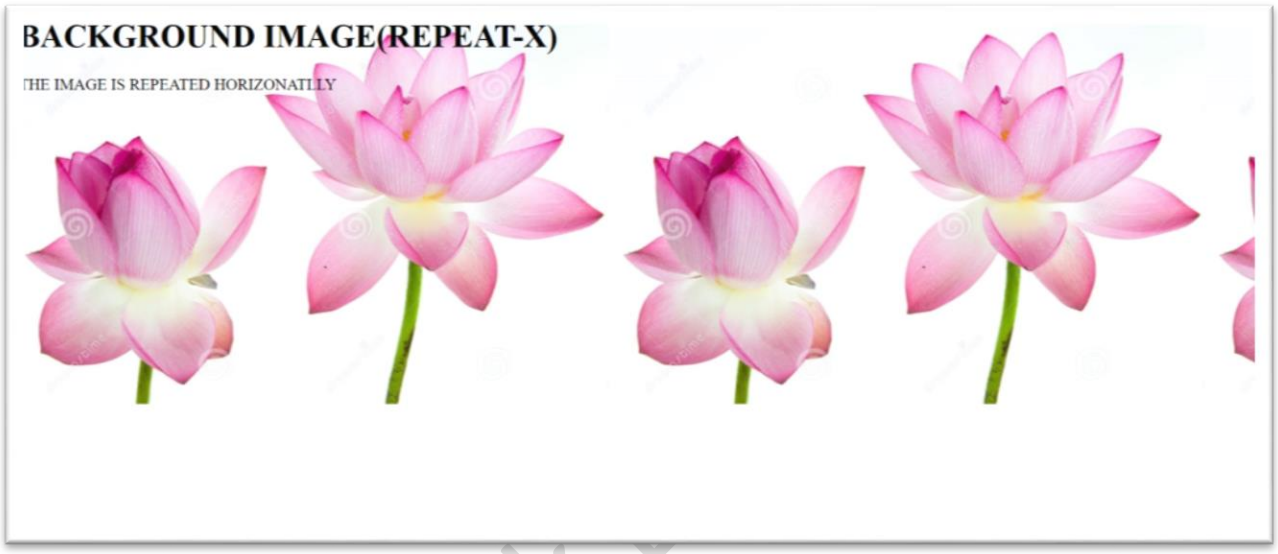
- By default, the background-image property repeats an image both horizontally and vertically.
- Some images should be repeated only horizontally or vertically, or they will look strange.
- To repeat an image vertically, set **background-repeat: repeat-y**;
- To repeat an image horizontally, set **background-repeat: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

Example (repeat-x):

```
<head>
<style>
body {
  background-image: url("D:/Full Stack/flwr.jpg");
  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("D:/Full Stack/flwr.jpg");
  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("D:/Full Stack/flwr.jpg");
  background-repeat: no-repeat;
}
</style>
</head>
<body>
```

```
<h1>BACKGROUND IMAGE(no-repeat)</h1>
<p>THE IMAGE IS REPEATED ONLY ONCE</p>
</body>
</html>
```

Output:

Example (Shorthand):

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

Output:

Text Manipulation: 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.

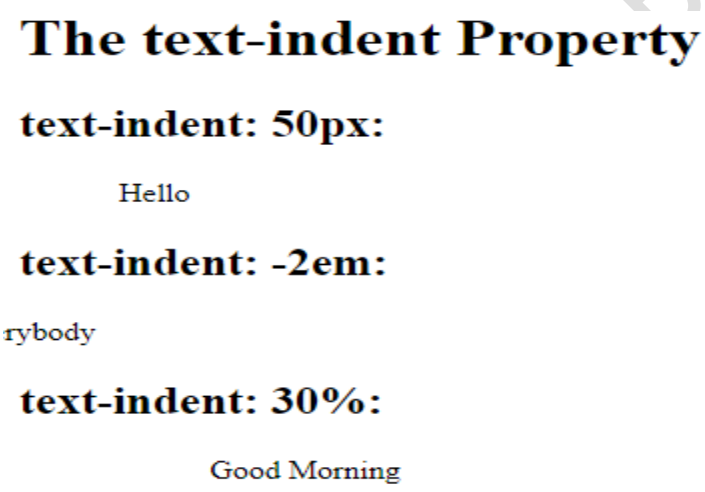
Note: Negative values are allowed. The first line will be indented to the left if the value is negative.

Example:

```
<html>
<head>
<style>
.a {
  text-indent: 50px;
}
.b {
  text-indent: -2em;
}
.c {
  text-indent: 5%;
}
</style>
</head>
<body>
```

```
<h1>The text-indent Property</h1>

<h2>text-indent: 50px:</h2>
<div class="a">
  <p>Hello</p>
</div>

<h2>text-indent: -2em:</h2>
<div class="b">
  <p>Everybody</p>
</div>

<h2>text-indent: 30%:</h2>
<div class="c">
  <p>Good Morning</p>
</div>

</body>
</html>
```

Output:



Text Manipulation: text-decoration

- This property is used to decorate a text.
- Possible values are: none, underline, overline, line-through.

Example:

```
<html>

<head>
```

```
<style>
h1 {
  text-decoration: overline;
}

h2 {
  text-decoration: line-through;
}

h3 {
  text-decoration: underline;
}

h4 {
  text-decoration: underline overline;
}

h5 {
  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</h4>
<h5>Heading with none</h5>
</body>
</html>
```

Output:

Heading with overline

~~Heading with line-through~~

Heading with underline

Heading with underline overline

Heading with none

Text Manipulation: text-transform

- The text-transform property controls the capitalization of text.
- Possible values are: uppercase, lowercase, capitalize.
- **none** No capitalization. The text renders as it is. This is default
- **capitalize:** Transforms the first character of each word to uppercase
- **lowercase:** Transforms all characters to lowercase
- **uppercase:** Transforms all characters to uppercase

Example:

```
<html>
<head>
<style>
.a {
  text-transform: uppercase;
}

.b {
  text-transform: lowercase;
}

.c {
  text-transform: capitalize;
}

.d {
  text-transform: none;
}
```

```
</style>
</head>
<body>
<h1>The text-transform Property</h1>

<h2>text-transform: Uppercase</h2>
<div class="a">Have a nice day</div>

<h2>text-transform: Lowercase</h2>
<div class="b">Have a nice day</div>

<h2>text-transform: Capitalize</h2>
<div class="c">Have a nice day</div>

<h2>text-transform: None</h2>
<div class="d">Have a nice day</div>

</body>
</html>
```

Output:

The text-transform Property

text-transform: uppercase:

LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT.

text-transform: lowercase:

lorem ipsum dolor sit amet, consectetur adipiscing elit.

text-transform: capitalize:

Lorem Ipsum Dolor Sit Amet, Consectetur Adipiscing Elit.

text-transform: none:

Lorem ipsum

Text Manipulation: text-align

- The text-align property specifies the horizontal alignment of text in an element.

Syntax:

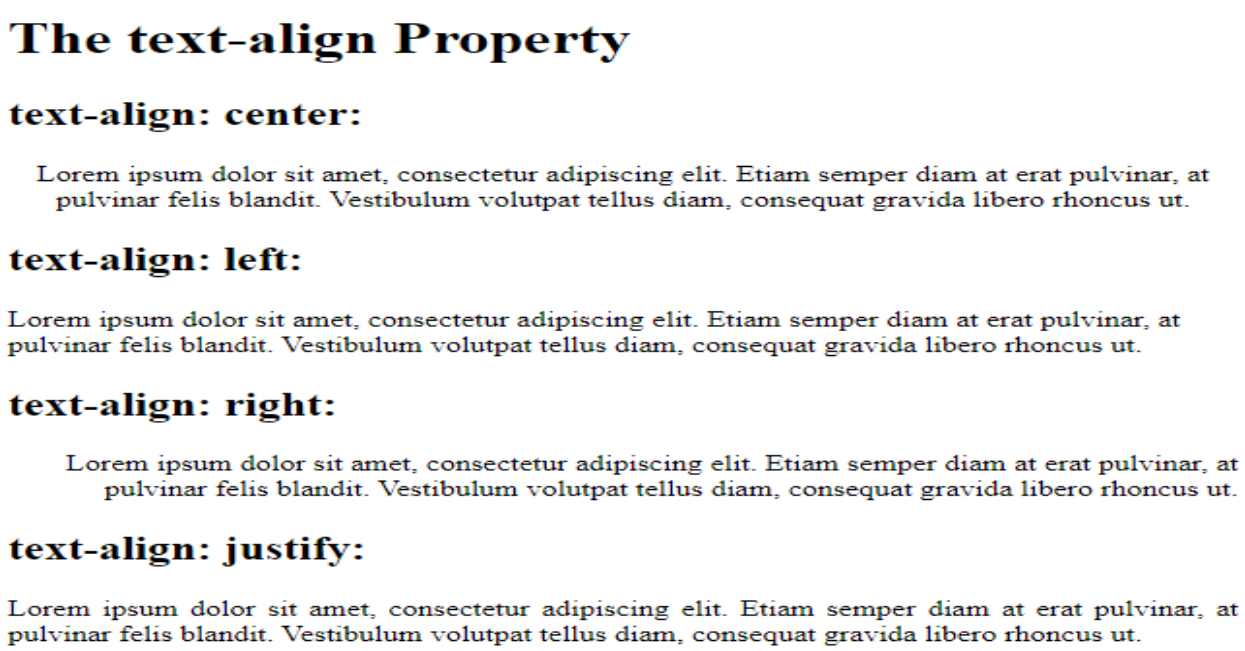
text-align: left|right|center|justify;

Example:

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

.b {
  text-align: left;
}

.c {
  text-align: right;
}
.d {
  text-align: justify;
}
</style>
</head>
<body>

<h1>The text-align Property</h1>

<h2>text-align: center:</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.</p>


<h2>text-align: left:</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.</p>

<h2>text-align: 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.</p>

<h2>text-align: 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.</p>

</body>
</html>
```

Output:



CSS Fonts

- 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.

Font families:

- In CSS, we use the **font-family** property to specify the font of a text.

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

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 Font Family	Examples of Font Names
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

Example:

```
<html>
<head>
<style>
.p1 {
```

```
  font-family: "Times New Roman", Times, serif;
}
.p2 {
  font-family: Arial, Helvetica, sans-serif;
}
.p3 {
  font-family: "Lucida Console", "Courier New", monospace;
}
</style>
</head>
<body>
<h1>CSS font-family</h1>
<p class="p1">This is a paragraph, shown in the Times New Roman font.</p>
<p class="p2">This is a paragraph, shown in the Arial font.</p>
<p class="p3">This is a paragraph, shown in the Lucida Console font.</p>
</body>
</html>
```

Output:

Font style:

- The `font-style` property is mostly used to specify italic text.
- Possible values for this properties are:
  1) **Normal**: The text is shown normally
  2) **Italic**: The text is shown in italics

Example:

```
<html>
<head>
<style>
p.normal {
```

```
  font-style: normal;
}
p.italic {
  font-style: italic;
}

</style>
</head>
<body>
<h1>The font-style property</h1>
<p class="normal">This is a paragraph in normal style.</p>
<p class="italic">This is a paragraph in italic style.</p>
</body>
</html>
```

Output:

The font-style property

This is a paragraph in normal style.

This is a paragraph in italic style.

Font weight

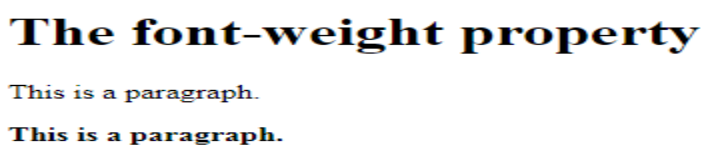
- The `font-weight` property specifies the weight of a font:

Example:

```
<html>
<head>
<style>
p.normal {
  font-weight: normal;
}

p.thick {
  font-weight: bold;
}

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

```
<h1>The font-weight property</h1>

<p class="normal">This is a paragraph.</p>
<p class="thick">This is a paragraph.</p>

</body>
</html>
```

Output:



Font variant:

- The font-variant property specifies whether or not a text should be displayed in a small-caps font.
- In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text.

Example:

```
<html>
<head>
<style>
p.normal {
  font-variant: normal;
}

p.small {
  font-variant: small-caps;
}
</style>
</head>
<body>

<h1>The font-variant property</h1>

<p class="normal">My name is Hege Refsnes.</p>
<p class="small">My name is Hege Refsnes.</p>
```

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

Output:

Font size:

- The `font-size` property sets the size of a font.

Syntax:

font-size: medium|xx-small|x-small|small|large|x-large|xx-large|px|%

Example:

```
<html>
<head>
<style>
.a {
  font-size: 15px;
}

.b {
  font-size: large;
}

.c {
  font-size: 150%;
}
.d {
  font-size: small;
}

.e {
```

```
  font-size: x-small;
}

.f {
  font-size: xx-small;
}
</style>
</head>
<body>
<h1>The font-size Property</h1>

<p class="a">This is some text.</p>

<p class="b">This is some text.</p>

<p class="c">This is some text.</p>

<p class="d">This is some text.</p>

<p class="e">This is some text.</p>

<p class="f">This is some text.</p>

</body>
</html>
```

Output:

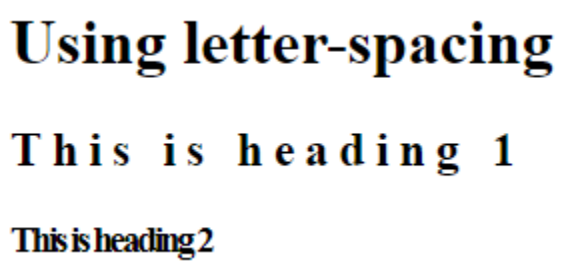
CSS text controls: Letter spacing

- The `letter-spacing` property is used to specify the space between the characters in a text.

Example:

```
<html>
<head>
<style>
h2 {
  letter-spacing: 5px;
}

h3 {
  letter-spacing: -2px;
}
</style>
</head>
<body>
<h1>Using letter-spacing</h1>
<h2>This is heading 1</h2>
<h3>This is heading 2</h3>
</body>
</html>
```

Output:



CSS text controls: Word spacing

- The `word-spacing` property is used to specify the space between the words in a text.

Example:

```
<html>
<head>
<style>
p.one {
  word-spacing: 10px;
}

p.two {
  word-spacing: -2px;
}
</style>
</head>
<body>

<h1>Using word-spacing</h1>

<p>This is a paragraph with normal word spacing.</p>

<p class="one">This is a paragraph with larger word spacing.</p>

<p class="two">This is a paragraph with smaller word spacing.</p>

</body>

</html>
```


Output:

Using word-spacing

This is a paragraph with normal word spacing.

This is a paragraph with larger word spacing.

This is a paragraph with smaller word spacing.

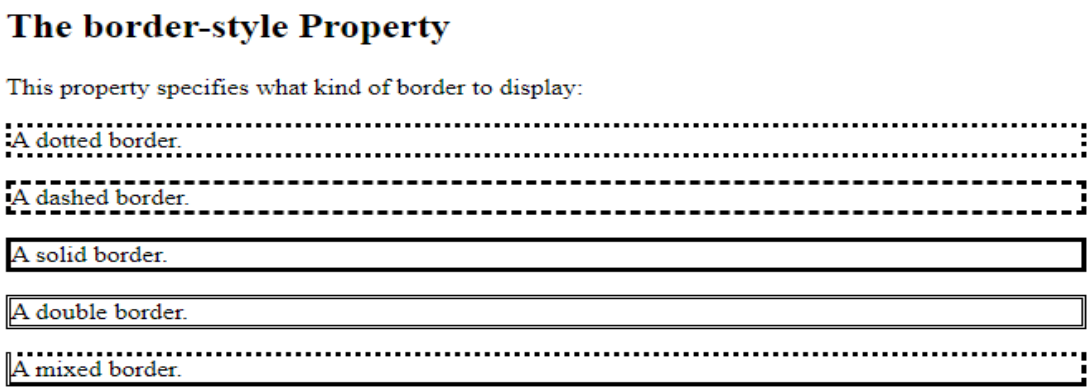
CSS Borders:

- The CSS border properties allow you to specify the style, width, and color of an element's border.
- The `border-style` property specifies what kind of border to display.
- Possible values are: solid, dotted, dashed, double.

Example:

```
<html>
<head>
<style>
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.mix {border-style: dotted dashed solid double;}
</style>
</head>
<body>

<h2>The border-style Property</h2>
<p>This property specifies what kind of border to display:</p>

<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="mix">A mixed border.</p>
</body>
</html>
```

Output:



Border color:

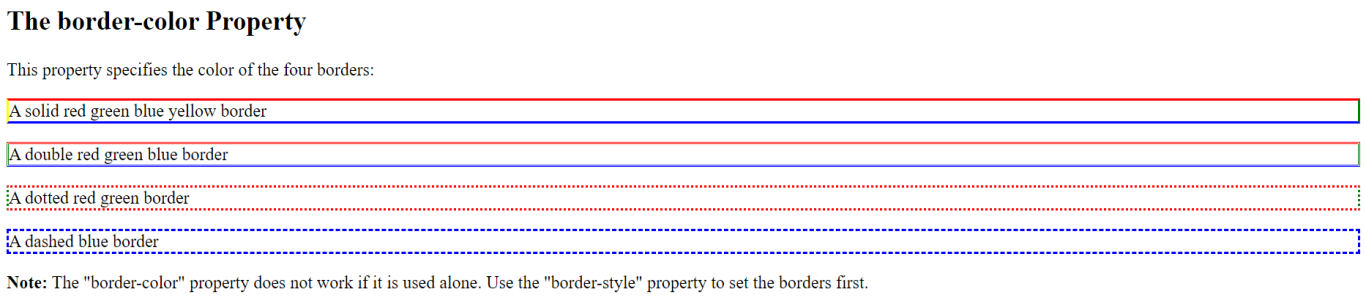
- The `border-color` property is used to set the color of the four borders.
- The `border-color` property can have from one to four values (for the top border, right border, bottom border, and the left border).

Example:

```
<html>
<head>
<style>
p.one {
  border-style: solid;
  border-color: red green blue yellow; /* red top, green right, blue bottom and
yellow left */
}

p.two {
  border-style: double;
  border-color: red green blue;
}

p.three {
  border-style: dotted;
  border-color: red green;
}
p.four {
  border-style: dashed;
  border-color: blue;
}
</style>
</head>
<body>

<h2>The border-color Property</h2>
<p>This property specifies the color of the four borders:</p>

<p class="one">A solid red green blue yellow border</p>
<p class="two">A double red green blue border</p>
<p class="three">A dotted red green border</p>
<p class="four">A dashed blue border</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:



Border width:

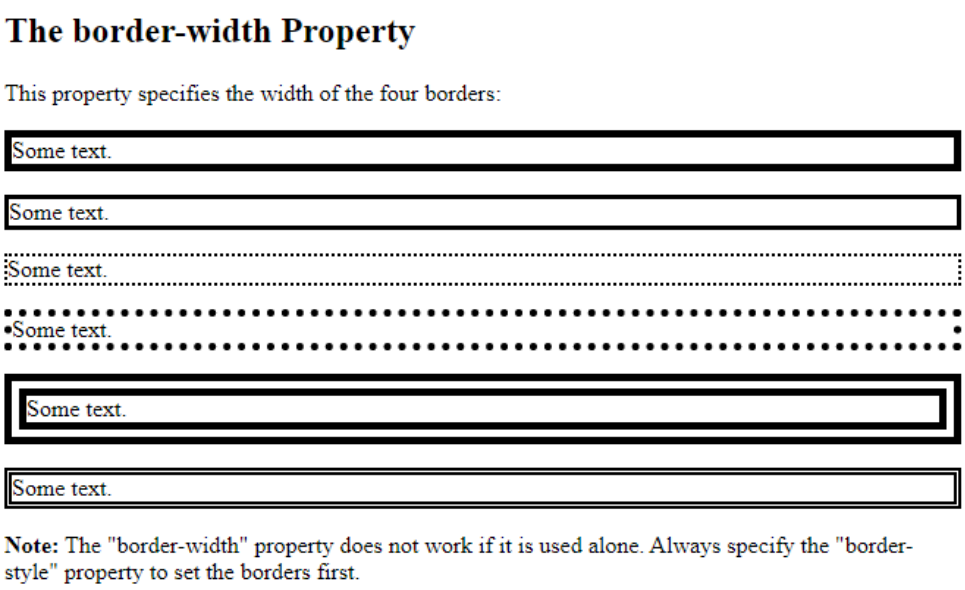
- The `border-width` property specifies the width of the four borders.
- The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick:

Example:

```
<html>
<head>
<style>
p.one {
  border-style: solid;
  border-width: 5px;
}

p.two {
  border-style: solid;
  border-width: medium;
}

p.three {
  border-style: dotted;
  border-width: 2px;
}

p.four {
  border-style: dotted;
  border-width: thick;
}
```

```
p.five {
  border-style: double;
  border-width: 15px;
}

p.six {
  border-style: double;
  border-width: thick;
}
</style>
</head>
<body>

<h2>The border-width Property</h2>
<p>This property specifies the width of the four borders:</p>

<p class="one">Some text.</p>
<p class="two">Some text.</p>
<p class="three">Some text.</p>
<p class="four">Some text.</p>
<p class="five">Some text.</p>
<p class="six">Some text.</p>
<p><b>Note:</b> The "border-width" property does not work if it is used alone.
Always specify the "border-style" property to set the borders first.</p>
</body>
</html>
```


Output:



Border Shorthand:

- 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.

Example:

```
<html>
<head>
<style>
.a {
  border: 5px solid red;
}
.b{
  border-left: 5px solid red;
}
</style>
</head>
<body>
<h1>The border short hand Property</h1>

<p class="a">This is some text.</p>
<p class="b">This is some text.</p>

</body>
</html>
```

Output:

Pseudo classes

- A pseudo-class is used to define a special state of an element.
- it can be used to:
  1) Style an element when a user mouses over it
  2) Style visited and unvisited links differently
  3) Style an element when it gets focus

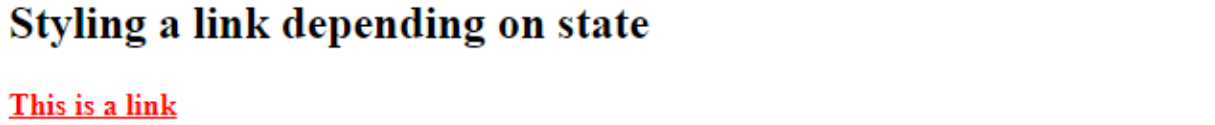
Syntax:

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

Example:

```
<html>
<head>
<style>
/* unvisited link */
a:link {
  color: red;
}

/* visited link */
a:visited {
  color: green;
}

/* mouse over link */
a:hover {
  color: hotpink;
}

/* selected link */
a:active {
  color: blue;
}
</style>
</head>
<body>

<h2>Styling a link depending on state</h2>

<p><b><a href="default.asp" target="_blank">This is a link</a></b></p>
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS
definition in order to be effective.</p>
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order
to be effective.</p>

</body>
</html>
```

Output:



Example:

```
<html>
	<head>
		<style>
			a:link
			{
				color:blue;
			}
			a:visited
			{
				color:blue;
			}
			a:hover
			{
				color:yellow;
			}
			a:active
			{
				color:cyan;
			}
			ol:hover
			{
				color:pink;
			}
			ol:active
			{
				color:purple;
			}
			a.highlight:hover
			{
				font-size:100px;
			}

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

```
            <a href="http://www.facebook.com" class="highlight">FACEBOOK</a>
            <ol>
                  <li>ABC</li>
                  <li>PQR</li>
            </ol>

      </body>
</html>
```

Output:

FACEBOOK

1. ABC
2. PQR

Pseudo Elements

- A CSS pseudo-element is used to style specified parts of an element.
- it can be used to:
  1) Style the first letter, or line, of an element
  2) Insert content before, or after, the content of an element

Syntax:

```
selector::pseudo-element

{
  property: value;
}
```

The ::first-line Pseudo-element

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

Example:

```
<html>
      <head>
            <style>
                  p::first-line
                  {
                        font-size:30px;
                        font-family:times new roman,sarif;
                        font-style:italic;
                        font-weight:bold;
                        font-variant:small-caps;
                  }
```

```
		</style>
	</head>
	<body>
		<p>Happiness, in the context of mental or emotional states, is
positive or pleasant emotions ranging from contentment to intense joy. Other
forms include life satisfaction, well-being, subjective well-being, flourishing
and eudaimonia.
Since the 1960s, happiness research has been conducted in a wide variety of
scientific disciplines, including gerontology, social psychology and positive
psychology, clinical and medical research and happiness economics.</p>
	</body>
</html>
```

Output:

HAPPINESS, IN THE CONTEXT OF MENTAL OR EMOTIONAL STATES, IS POSITIVE OR PLEASANT EMOTIONS
ranging from contentment to intense joy. Other forms include life satisfaction, well-being, subjective well-being, flourishing and eudaimonia. Since the 1960s, happiness research has been conducted in a wide variety of scientific disciplines, including gerontology, social psychology and positive psychology, clinical and medical research and happiness economics.

The ::first-letter Pseudo-element

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

Example:

```
<html>
	<head>
		<style>
			p::first-letter
			{
				font-size:100px;
			}
		</style>
	</head>
	<body>
		<p>This is demo of first-letter pseudo element..</p>
	</body>
</html>
```

Output:

This is demo of first-letter pseudo element..

CSS - The ::before Pseudo-element

- The ::before pseudo-element can be used to insert some content before the content of an element.

Example:

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

Output:

Spread smile wherever you go.....

CSS - The ::after Pseudo-element

- The `::after` pseudo-element can be used to insert some content after the content of an element.

Example:

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

Output:

CSS - The ::marker Pseudo-element

- The ::marker pseudo-element selects the markers of list items.

Example:

```
<html>
	<head>
		<style>
			::marker
			{
				color:red;
				font-size:20px;
			}
```

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

            <ul>
                  <li>ABC</li>
                  <li>PQR</li>
                  <li>XYZ</li>
            </ul>
            <ol>
                  <li>ABC</li>
                  <li>PQR</li>
                  <li>XYZ</li>
            <ol>
      </body>
</html>
```

Output:

- ABC
- PQR
- XYZ

1. ABC
2. PQR
3. XYZ

CSS - The ::selection Pseudo-element

- The ::selection pseudo-element matches the portion of an element that is selected by a user.
- The following CSS properties can be applied to ::selection: color, background, cursor, and outline.

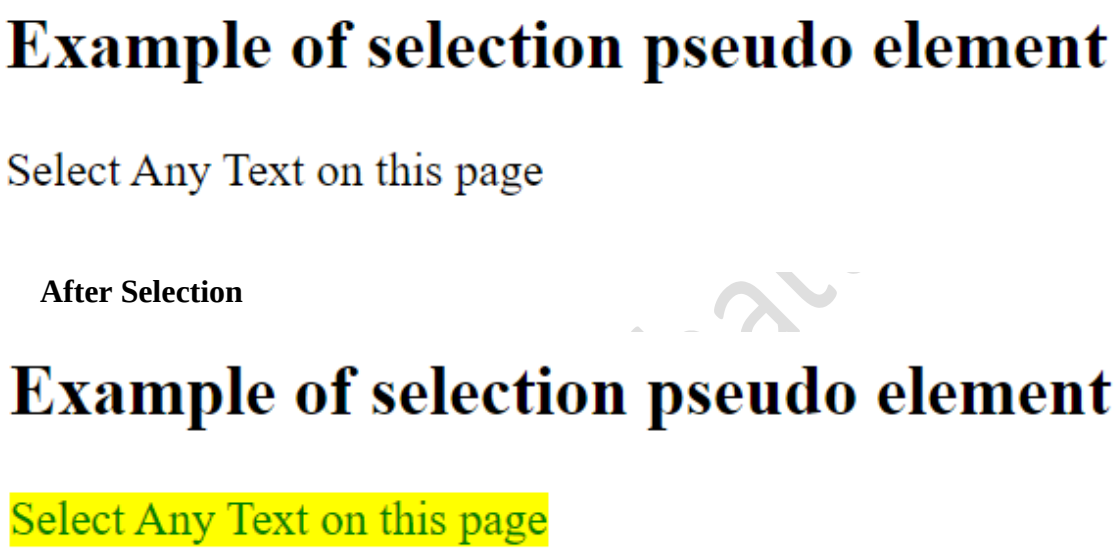
Example:

```
<html>
      <head>
            <style>
                  ::selection
                  {
                        color:green;
                        background-color:yellow;
                  }
            </style>
```

```
	</head>
	<body>
		<h2>Example of selection pseudo element</h2>
		<p>Select Any Text on this page</p>
	</body>
</html></html>
```

Output

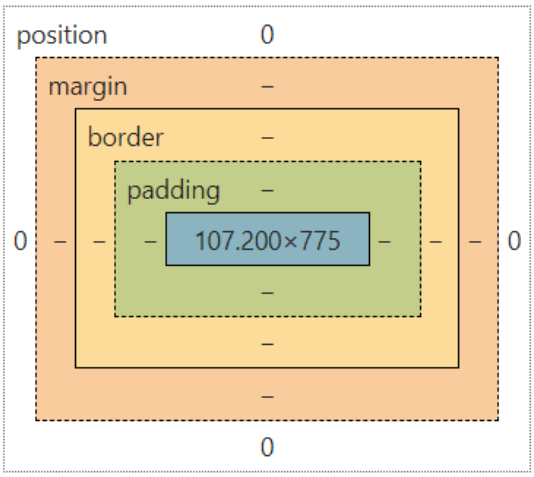
Before Selection:



After Selection

CSS Box Model:

- All HTML elements can be considered as boxes.
- In CSS, the term "box model" is used when talking about design and layout.
- The box model allows us to add a border around elements, and to define space between elements.
- The CSS box model is essentially a box that wraps around every HTML element. It consists of: **margins**, **borders**, **padding**, and the actual **content**.



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

Example:

```
<html>
	<head>
	</head>
	<body>
		<p style="margin:auto; padding:50px; border:4px solid blue">
		THIS IS P1
		</p>
		<p style="margin:10px; padding:50px 50px; border:5px dashed red">
		THIS IS P2
		</p>
		<p style="margin:100px 100px; border:15px dotted green">
		THIS IS P3
		</p>
		<p style="margin:15px 20px 25px 30px;border:4px double purple">
		THIS IS P4
		</p>
	</body>
</html>
```

Output:

CSS positioning:

- The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).
- Possible values for the position property are:
  1) Absolute
  2) Relative
  3) Fixed
  4) Static

position: static:

- HTML elements are positioned static by default.
- Static positioned elements are not affected by the top, bottom, left, and right properties.
- An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page

position :relative

- An element with position: relative; is positioned relative to its normal position.
- Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

position: fixed

- An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled.
- The top, right, bottom, and left properties are used to position the element.
- A fixed element does not leave a gap in the page where it would normally have been located.

position-absolute:

- An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).
- However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

Note: Absolute positioned elements are removed from the normal flow, and can overlap elements.

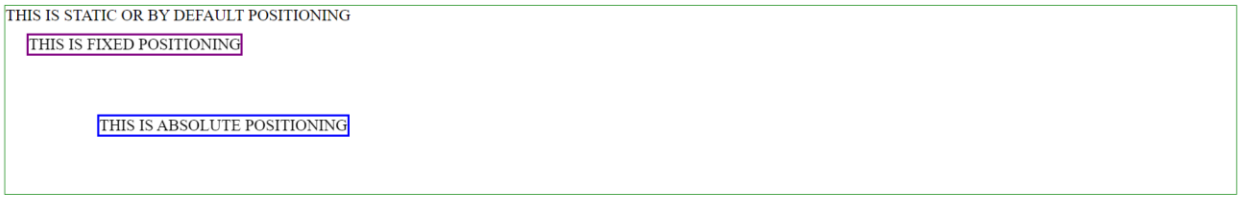
Example:

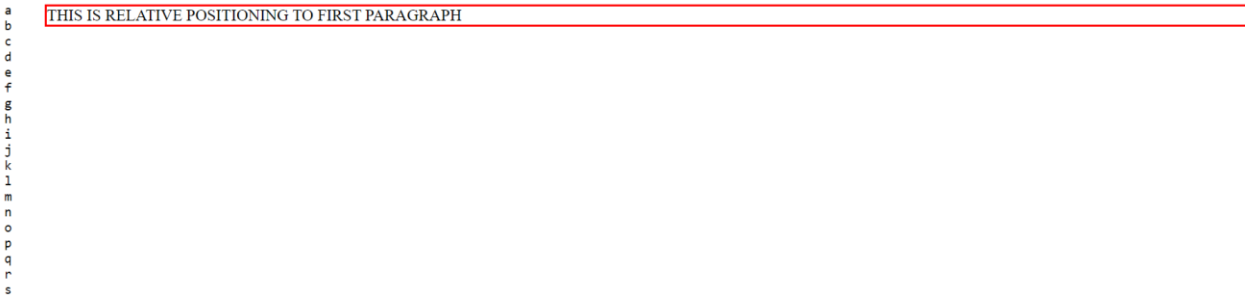
```
<html>
	<head>
		<style>
			#p1
			{
				position:static;
				border:1px solid green;
				height:30%;
			}
			#p2
			{
				border:2px solid red;
				position:relative;
				top:40px;
				left:40px;
			}
			#p3
			{
				border:2px solid purple;
				position:fixed;
				top:20px;
				left:30px;
			}
			#p4
			{
				border:2px solid blue;
				position:absolute;
				top:100px;
				left:100px;
			}
		</style>
	</head>
	<body>
		<p id="p1">
		THIS IS STATIC OR BY DEFAULT POSITIONING
```

```
            </p>
            <p id="p2">
            THIS IS RELATIVE POSITIONING TO FIRST PARAGRAPH
            </p>
            <p id="p3">
            THIS IS FIXED POSITIONING
            </p>
            <p id="p4">
            THIS IS ABSOLUTE POSITIONING
            </p>
            <pre>
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
*
*
*
*
*
*
*
*
```

```
*
*
*
*
*
*

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

Output:





CSS float property

- The float property specifies whether an element should float to the left, right, or not at all.
- Allowing elements to wrap around it.
- Elements are floated horizontally only.

Note: Absolutely positioned elements ignore the float property!

Example:

```
<html>
	<head>
	<title>css float property</title>
	<style>
		img
		{
			float:left;
			width:50px;
```

```
                height:50px;
            }
        </style>
        </head>
        <body>
            <img src="D:/Full Stack/flow.jfif"/>

            <strong><p>Lotus plants are adapted to grow in the flood plains of
slow-moving rivers and delta areas.
Stands of lotus drop hundreds of thousands of seeds every year to the bottom of
the pond. While some sprout immediately,
and most are eaten by wildlife, the remaining seeds can remain dormant for an
extensive period of time as the pond silts in and dries out.
During flood conditions, sediments containing these seeds are broken open, and
the dormant seeds rehydrate and begin a new lotus colony. </p></strong>

        </body>
</html>
```

Output:

Lotus plants are adapted to grow in the flood plains of slow-moving rivers and delta areas. Stands of lotus drop hundreds of thousands of seeds every year to the bottom of the pond. While some sprout immediately, and most are eaten by wildlife, the remaining seeds can remain dormant for an extensive period of time as the pond silts in and dries out. During flood conditions, sediments containing these seeds are broken open, and the dormant seeds rehydrate and begin a new lotus colony.

Note: In this example image will float to the left and text in the paragraph will wrap around the image.

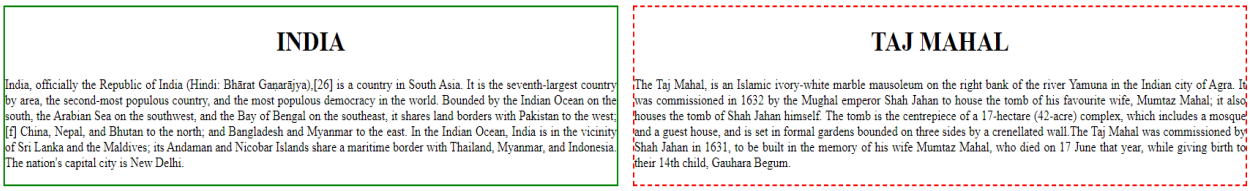
Examples:

1) 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>
            <title>css float property example</title>
        </head>

<div style="width:45%; border:2px solid green; text-align:justify;float:left">
            <h1 align="center">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 style="width:45%; border:2px dashed red; float:left; text-
align:justify; margin:0 0 0 20px">
            <h1 align="center">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>
```

Output:



2) Write following style in separate css file.

1. Heading should have normal font style and size should be 120%

2. Define a class arial for paragraph with arial face, bold text and 3 cm spacing for paragraph initialization.

3. Apply a background image and it should be repeated vertically only.

example.html

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

```
<link type="text/css" rel="stylesheet" href="demo.css"/>
</head>
<body class="body1">
<h1 class="class1">
      CSS EXAMPLE
</h1>
<p class="arial">
paragraph text in center with indentation
</p>
</body>
</html>
```

demo.css

```
.class1
{
    font-style : normal;
    font-size : 120%;
    text-align:center;
}
.arial
{
    font-family : arial;
    font-weight : bold;
    text-indent : 3cm;
    color:white;
}
.body1
{
    background : url("D:/ZPB_FullStack/flow.jfif") repeat-y;
}
```

Output:

3) Write an HTML and CSS script to create a table with 5 rows and 3 columns. Even no of rows should be displayed in red color and odd no of rows should be displayed in yellow color.

table.html

```
<html>
	<head>
		<title>CSS DEMO</title>
		<link type="text/css" rel="stylesheet" href="demo.css"/>
	</head>
	<body>
	<table border="2" width="30%">
			<tr align="center" class="tr2">
				<td>x</td>
				<td>x</td>
				<td>x</td>
			</tr>
			<tr align="center" class="tr1">
				<td>y</td>
				<td>y</td>
				<td>y</td>
			</tr><tr align="center" class="tr2">
```

```
                        <td>z</td>
                        <td>z</td>
                        <td>z</td>
                    </tr><tr align="center" class="tr1">
                        <td>w</td>
                        <td>w</td>
                        <td>w</td>
                    </tr><tr align="center" class="tr2">
                        <td>v</td>
                        <td>v</td>
                        <td>v</td>
                    </tr>
                </table>
        </body>
</html>
```

demo.css

```
.tr1
{
        background-color:red;
}
.tr2
{
        background-color:yellow;
}
```

Output:

x	x	x
y	y	y
z	z	z
w	w	w
v	v	v