CSS Introduction

1. CSS stands for Cascading Style Sheet.
2. CSS is used to design web pages created by html.
3. CSS is widely used language on the web.
4. HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.
5. CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
6. CSS saves a lot of work. It can control the layout of multiple web pages all at once.

CSS Syntax

1. A CSS rule set contains a selector and a declaration block.
2. Selector indicates the HTML element you want to style. It could be any tag like <h1>, <p> etc.
3. The declaration block can contain one or more declarations separated by a semicolon.
4. Each declaration contains a property name and value, separated by a colon.
5. A Property is a type of attribute of HTML element. It could be color, border etc.
6. Values are assigned to CSS properties.

```
<!DOCTYPE HTML>
<html>
    <head>
        <title>CSS Syntax</title>
    <style>
      h1{
        color:red;
      } <!-- Here h1 is selector, color is property and red is value-->
    </style>
    </head>

    <body>
        <h1>CSS Tutorials 2018</h1>
    </body>
</html>
```

CSS Selectors

1. CSS selectors are used to select the content you want to style.
2. There are several different types of selectors in CSS.

- **CSS Element Selector:**The element selector selects the HTML element by name like h1, p, table, ol, img etc...
- **CSS Id Selector:**The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element. It is written with the hash character (#), followed by the id of the element.
- **CSS Class Selector:**The class selector selects HTML elements with a specific class attribute. It is used with a period character . (full stop symbol)
- **CSS Universal Selector:**The universal selector is used as a wildcard character. It selects all the elements on the pages. It is used with a * symbol.
- **CSS Group Selector:**The grouping selector is used to select all the elements with the same style definitions. Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

```
<!DOCTYPE HTML>
<head>
        <title>CSS Selectors</title>
    <style>
        h1{
            color:red;
        }
        #heading2{
            color:blue;
        }
        .heading3{
            color:brown;
        }
        *{
            color:green;
        }
        h6, p{
            color:aqua;
        }
    </style>
</head>

<body>
        <h1> CSS Tutorials 2018 </h1>
        <h2 id="heading2"> CSS Tutorials 2018 </h2>
        <h3 class="heading3"> CSS Tutorials 2018 </h3>
        <h4> CSS Tutorials 2018 </h4>
        <h5> CSS Tutorials 2018 </h5>
        <h6> CSS Tutorials 2018 </h6>
        <p> CSS Tutorials 2018 </p>
</body>
</html>
```

How & Where to add CSS code

1. CSS is added to HTML pages to format the document according to information in the style sheet. There are three ways to insert CSS in HTML documents.

- **Inline CSS:**Inline CSS is used to apply CSS on a single line or element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<!DOCTYPE HTML>
<head>
        <title>Inline CSS</title>
</head>

<body>
        <h1 style="color:orange;">This is heading 1</h1>
</body>
</html>
```

- **Internal CSS:**Internal CSS is used to apply CSS on a single document or page. It can affect all the elements of the page. It is written inside the style tag within head section of html.

```
<!DOCTYPE HTML>
<head>
        <title>Internal CSS</title>
    <style>
        body {
            background-color: aqua;
        }
        h1 {
            color: red;
        }
    </style>
</head>

<body>
        <h1>This is heading 1</h1>
        <h2 style="color:blue">This is heading 2</h2>
</body>
</html>
```

- **External CSS:**External CSS is used to apply CSS on multiple pages or all pages. Here, we write all the CSS code in a css file. Its extension must be .css for example style1.css. With an external style sheet, you can change the look of an entire website by changing just one file! Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section

```
<!DOCTYPE HTML>
<head>
        <title>External CSS</title>
    <link rel="stylesheet" type="text/css" href="mystyle1.css">
    <style>
        h1 {
            color: red;
        }
    </style>
</head>

<body>
        <h1>This is heading 1</h1>
        <h2 style="color:blue">This is heading 2</h2>
        <h3>This is heading 3</h3>
</body>
</html>
```

mystyle1.css (seperate CSS file)

```
        body{
            background-color:orange;
        }
        h3{
            color:darkgreen;
        }
```

Priority:inline(highest) > internal(middle) > external(low)
When 3 types of CSS added to a single file then first it will check for inline CSS, if Inline CSS is not there it will check for Internal CSS and atlast it will check for External CSS.

```
<!DOCTYPE HTML>
<head>
        <title>CSS Priority</title>
    <link rel="stylesheet" type="text/css" href="mystyle2.css">
    <style>
        h1{
            color: red;
        }
    </style>
</head>

<body>
        <h1>This is heading 1</h1>
        <h2 style="color:blue">This is heading 2</h2>
        <h3>This is heading 3</h3>
    <h4>This is heading 4</h4>
</body>
</html>
```

mystyle2.css (seperate CSS file)

```
*{
   color:darkgreen;
}
```

CSS Background

1. The CSS background properties are used to define the background effects for elements. There are 5 CSS background properties are available.

- **CSS background-color:** The background-color property is used to specify the background color of the element. Directly you can give color name (or) you can use HEX value (or) you can give RGB value. And you can give different colors for different elements like h1, p, body, div etc...

```
<!DOCTYPE HTML>
<head>
      <title>CSS Background Color</title>
   <style>
      body {
         background-color: lightblue;
      }
      h1 {
         background-color: green;
      }
      p {
         background-color: yellow;
      }
</style>
</head>

<body>
      <h1>Programming Tutorials In Telugu</h1>
      <p>This is paragraph.</p>
   This is normal text.
</body>
</html>
```

- **CSS background-image:** The background-image property is used to set an image as a background of an element. By default, the image is repeated so it covers the entire element. And you can also insert different images for different elements like h1, p, body, div etc...

```
<!DOCTYPE HTML>
<head>
        <title>CSS Background Image</title>
    <style>
        body {
            background-image: url("paper1.jpg");
        }
        h1 {
            background-image: url("paper2.png");
        }
        p {
            background-image: url("paper3.gif");
        }
</style>
</head>

<body>
        <h1>CSS Background</h1>
        <p>This is paragraph.</p>
    This is normal text.
</body>
</html>
```

- **CSS background-repeat:** By default, the background-image property repeats the background image horizontally or vertically or both to cover full place. But you can control that using this property. If you use value as **repeat-x** it will repeat only horizontally. If you give value as **repeat-y** it will repeat only vertically. If you give value as **no-repeat** it will display only single time.

```
<!DOCTYPE HTML>
<head>
        <title>CSS Background Repeat</title>
    <style>
        body {
            background-image: url("hello.png");
            background-repeat: repeat-x;
        }
</style>
</head>

<body>
        <h1>Programming Tutorials In Telugu</h1>
        <p>This is paragraph.</p>
    This is normal text.
</body>
</html>
```

- **CSS background-position:** By using this property you can give position to background image with the help of **background-repeat: no-repeat;**. You can give position as **right top, left top, bottom right, bottom left** etc....

```
<!DOCTYPE HTML>
<head>
    <title>CSS Background Position</title>
  <style>
    body {
      background-image: url("hello.png");
      background-repeat: no-repeat;
      background-position: right top;
    }
</style>
</head>

<body>
    <h1>Programming Tutorials In Telugu</h1>
    <p>This is paragraph.</p>
  This is normal text.
</body>
</html>
```

- **CSS background-attachment:** The background-attachment property is used to specify if the background image is fixed or scroll with the rest of the page in browser window. If you set fixed the background image then the image will not move during scrolling in the browser.

```
<!DOCTYPE HTML>
<head>
    <title>CSS Background Attachment</title>
  <style>
    body {
      background-image: url("hello.png");
      background-repeat: no-repeat;
      background-position: right top;
      background-attachment: fixed;
    }
</style>
</head>

<body>
    <h1>Programming Tutorials In Telugu</h1>
    <p>This is paragraph.</p>
  This is normal text.
</body>
</html>
```

- **CSS Background - Shorthand property:** To shorten the code, it is also possible to specify all the background properties in one single property. This is called a shorthand property. The shorthand property for background is **background**
- When using the shorthand property the order of the property values is: **background-color background-image background-repeat background-attachment background-position**. It does not matter if one of the property values is missing, as long as the other ones are in this order.

```
<!DOCTYPE HTML>
<head>
        <title>CSS Background</title>
    <style>
        body {
            background: red url("abhi.png") no-repeat right top;
        }
</style>
</head>

<body>
        <h1>Programming Tutorials In Telugu</h1>
        <p>This is paragraph.</p>
    This is normal text.
</body>
</html>
```

CSS Borders

1. 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 is used to specify the border type which you want to display on the web page. Various border styles are **dotted, dashed, solid, double, groove, ridge, inset, outset, none and hidden**.
- **CSS border-width:** The border-width property is used to set the border's width. It is set in pixels. You can also use the one of the three pre-defined values, thin, medium or thick to set the width of the border. The border-width property is not used alone. It is always used with other border properties like "border-style" property to set the border first otherwise it will not work.
- **CSS border-color:** The border-color property is used to set the border's color. The border-color property is not used alone. It is always used with other border properties like "border-style" property to set the border first otherwise it will not work.

```
<!DOCTYPE HTML>
<head>
        <title>CSS Borders (Style, width, color)</title>
    <style>
        p{
        border-style: dashed;
        border-width:2px;
        border-color:red;
        }
    <style>
</head>

<body>
        <p>Hello World</p>
</body>
</html>
```

- **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. In this border style is mandatory.

```
<!DOCTYPE HTML>
<head>
      <title>CSS Border</title>
   <style>
      p{
      border:2px solid green;
      }
   <style>
</head>

<body>
      <p>Hello World</p>
</body>
</html>
```

- **CSS Border - Different Sides:** You can also apply different borders for different sides like top, bottom, left and right with different styles and colors. And you can use anyone of them also.

```
<!DOCTYPE HTML>
<head>
      <title>CSS Borders for different sides</title>
   <style>
      p{
      border-top:2px solid green;
      border-bottom:3px dotted red;
      border-left:4px dashed black;
      border-right:2px dashed aqua;
      }
   <style>
</head>

<body>
      <p>Hello World</p>
</body>
</html>
```

- **CSS Border Radius:** The border-radius property is used to add rounded borders to an element.

```
<!DOCTYPE HTML>
<head>
        <title>CSS Border Radius</title>
    <style>
        p{
        border:2px solid green;
        border-radius: 15px;
        }
    <style>
</head>

<body>
        <p>Hello Welcome to CSS Round Borders</p>
</body>
</html>
```

CSS Margin

1. The CSS margin properties are used to create space around elements, outside of any defined borders.
2. Top, bottom, left and right margin can be changed independently using separate properties. You can also change all properties at once by using shorthand margin property.

```
<!DOCTYPE HTML>
<head>
        <title>CSS Margin</title>
    <style>
        h1{
            margin-top: 100px;
            margin-bottom: 2%;
            margin-right: auto;
            margin-left: auto;
        }
    <style>
</head>

<body>
        <h1>Hello Welcome to CSS by Programming Tutorials</h1>
</body>
</html>
```

- **CSS Margin Shorthand Property:** To shorten the code, it is possible to specify all the margin properties in one property.
- For example if **margin: 25px 50px 75px 100px;** as given then top margin is 25px, right margin is 50px, bottom margin is 75px, left margin is 100px.
- For example if **margin: 25px 50px 75px;** as given then top margin is 25px, right and left margins are 50px, bottom margin is 75px.
- For example if **margin: 25px 50px;** as given then top and bottom margins are 25px, right and left margins are 50px
- For example if **margin: 25px;** as given then all four margins are 25px.
- For example if **margin: auto;** as given then the element will then take up the specified width, and the remaining space will be split equally between the left and right margins. You can set the margin property to auto to horizontally center the element within its container.

```
<!DOCTYPE HTML>
<head>
        <title>CSS Margin Shorthand property</title>
    <style>
        h1{
            margin:10px;
        }
    <style>
</head>

<body>
        <h1>Hello Welcome to CSS by Programming Tutorials</h1>
</body>
</html>
```

CSS Padding

1. CSS Padding property is used to define the space between the element content and the element border.
2. It is different from CSS margin in the way that CSS margin defines the space around elements. CSS padding is affected by the background colors. It clears an area around the content.
3. Top, bottom, left and right padding can be changed independently using separate properties. You can also change all properties at once by using shorthand padding property.
4. **padding-bottom Property:** The padding-bottom property sets the bottom padding of an element.
5. **padding-top Property:** The padding-top property sets the top padding of an element.
6. **padding-left Property:** The padding-left property sets the left padding of an element.
7. **padding-right Property:** The padding-right property sets the right padding of an element.

```
<!DOCTYPE HTML>
<head>
    <title>CSS Padding</title>
  <style>
    h1{
      padding-top: 50px;
      padding-right: 100px;
      padding-bottom: 150px;
      padding-left: 200px;
    }
  <style>
</head>

<body>
    <h1>Hello Welcome to CSS by Programming Tutorials</h1>
</body>
</html>
```

- **Padding - Shorthand Property:** To shorten the code, it is possible to specify all the padding properties in one property.
- If the padding property has four values like **padding: 15px 25px 35px 45px;** then **top padding is 15px, right padding is 25px, bottom padding is 35px, left padding is 45px**

```
<!DOCTYPE HTML>
<head>
    <title>CSS Padding</title>
  <style>
    h1{
      border: 1px solid red;
      background-color: dodgerblue;
      padding:15px 25px 35px 45px;
    }
  <style>
</head>

<body>
    <h1>Hello Welcome to CSS by Programming Tutorials</h1>
</body>
</html>
```

- If the padding property has three values like **padding: 25px 50px 75px;** then top padding is 25px, right and left paddings are 50px, bottom padding is 75px
- If the padding property has two values like **padding: 25px 50px;** then top and bottom paddings are 25px, right and left paddings are 50px
- If the padding property has one value like **padding: 25px;** then all four paddings are 25px

```
<!DOCTYPE HTML>
<head>
    <title>CSS Padding</title>
  <style>
    h1{
      border: 1px solid black;
      background-color: aqua;
      padding:25px;
    }
  <style>
</head>

<body>
    <h1>Hello Welcome to CSS by Programming Tutorials</h1>
</body>
</html>
```

CSS Font

1. 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.
2. The CSS font properties define the font family, boldness, size, and the style of a text.
3. **CSS Font Family:** which demonstrates how to set the font family of an element. Possible value could be any font family name. The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font, and so on. If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman". More than one font family is specified in a comma-separated list.
4. **CSS Font Size:** CSS font size property is used to change the size of the font. Possible values could be xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, size in pixels or in %.
5. **CSS Font Style:** CSS Font style property defines what type of font you want to display. It may be italic, oblique, or normal.
6. **CSS Font Variant:** CSS font variant property specifies how to set font variant of an element. It may be normal and small-caps.
7. **CSS 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).
8. **CSS Font Shorthand Property:** You can use the font property to set all the font properties at once.

```
<!DOCTYPE HTML>
<head>
      <title>CSS Font</title>
   <style>
      h1{
      font:italic small-caps bold 15px georgia;
      }
   <style>
</head>

<body>
      <h1>Hello Welcome to CSS by Programming Tutorials</h1>
</body>
</html>
```

```
<!DOCTYPE HTML>
<head>
      <title>CSS Font</title>
   <style>
      h1{
         font-family:georgia,garamond,serif;
         font-style:italic;
         font-variant: small-caps;
         font-weight:bolder;
         font-size:large;
      }
   <style>
</head>

<body>
      <h1>Hello Welcome to CSS by Programming Tutorials</h1>
</body>
</html>
```

CSS Text

1. By using text property we can apply some special styles to text like:
2. **CSS Text Color:** The color property is used to set the color of the text. Possible value could be any color name in any valid format.
3. **CSS Text Direction:** It demonstrates how to set the direction of a text. Possible values are ltr or rtl.
4. **CSS Set the Space between Characters:** It demonstrates how to set the space between characters. Possible values are normal or a number specifying space.
5. **CSS Set the Space between Words:** It demonstrates how to set the space between words. Possible values are normal or a number specifying space.
6. **CSS Text Indent:** It demonstrates how to indent the first line of a paragraph. Possible values are % or a number specifying indent space.
7. **CSS Text Alignment:** It demonstrates how to align a text. Possible values are left, right, center, justify.

8. **CSS Text Decoration:** It demonstrates how to decorate a text. Possible values are none, underline, overline, line-through, blink.
9. **CSS Text Transform:** It demonstrates how to set the cases for a text. Possible values are none, capitalize, uppercase, lowercase. It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.
10. **CSS Text Shadow:** It demonstrates how to set the shadow around a text.
11. **CSS Line Height:** The line-height property is used to specify the space between lines.

```
<!DOCTYPE HTML>
<head>
        <title>CSS Text</title>
    <style>
        h1{
            color:red;
            direction:ltr;
            letter-spacing:5px;
            word-spacing:5px;
            text-indent:1cm;
            text-align:center;
            text-decoration:line-through;
            text-transform:capitalize;
            text-shadow:4px 4px 8px blue;
            line-height: 0.8;
        }
    <style>
</head>

<body>
        <h1>Hello Welcome to CSS by Programming Tutorials</h1>
</body>
</html>
```

CSS Links

1. A link or hyperlink is a connection from one web resource to another.
2. A link has four different states — link, visited, active and hover.
3. **a:link** - a normal, unvisited link.
4. **a:visited** - a link the user has visited.
5. **a:hover** - a link when the user mouses over it.

6. **a:active -** a link the moment it is clicked.

```
<!DOCTYPE HTML>
<head>
        <title>CSS Links</title>
    <style>
        a:link {
            color: red;
        }
        a:visited {
            color: green;
        }
        a:hover {
            color: hotpink;
        }
        a:active {
            color: blue;
        }
    <style>
</head>

<body>
        <h1>Hello Welcome to CSS by Programming Tutorials</h1>
    <a href="html.html">HTML Tutorials</a>
    <a href="css.html">CSS Tutorials</a>
    <a href="javascript.html">JavaScript Tutorials</a>
    <a href="php.html">PHP Tutorials</a>
</body>
</html>
```

- The **target attribute** specifies where to open the linked document.
  target="_blank": Opens the linked document in a new window or tab
  target="_self": Opens the linked document in the same frame as it was clicked (this is default)
- The **text-decoration** property is mostly used to remove underlines from links.

```
<!DOCTYPE HTML>
<head>
   <title>CSS Links</title>
   <style>
      a{
         text-decoration:none;
      }
   <style>
</head>

<body>
   <h1>Hello Welcome to CSS by Programming Tutorials</h1>
   <a href="html.html" target="_blank">HTML Tutorials</a>
   <a href="css.html" target="_blank">CSS Tutorials</a>
   <a href="js.html" target="_blank">JavaScript Tutorials</a>
   <a href="php.html" target="_blank">PHP Tutorials</a>
</body>
</html>
```

CSS Images

You can set the following image properties using CSS.

1. The **border** property is used to set the width of an image border.
2. The **height** property is used to set the height of an image.
3. The **width** property is used to set the width of an image.
4. The **-moz-opacity** property is used to set the opacity of an image.

```
<!DOCTYPE HTML>
<head>
   <title>CSS Images</title>
   <style>
      img{
      border:3px dashed red;
      height:100px;
      width:150px;
      -moz-opacity:0.4;
      }
   <style>
</head>

<body>
   <h1>Hello Welcome to CSS by Programming Tutorials</h1>
   <img src="css1.jpg" />
</body>
</html>
```

5. Use the **border-radius** property to create rounded images.
6. To center an image, set left and right margin to auto and make it into a block element.

```
<!DOCTYPE HTML>
<head>
   <title>CSS Images</title>
   <style>
      img{
         border-radius: 50%;
         display: block;
         margin-left: auto;
         margin-right: auto;
      }
   <style>
</head>

<body>
   <h1>Hello Welcome to CSS by Programming Tutorials</h1>
   <img src="css2.jpg" />
</body>
</html>
```

CSS Display

The display property is the most important CSS property for controlling layout. Every element has a default display value according to its nature. The default display value for most elements is block or inline.

- The **Block-level Elements** always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can). Examples of block-level elements: **<div>**, **<h1> - <h6>**, **<p>**, **<form>**, **<header>**, **<footer>**, **<section>**.
- The **Inline Elements** does not start on a new line and only takes up as much width as necessary. Examples of inline elements: **<span>**, **<a>**, **<img>**.
- As mentioned, every element has a default display value. However, we can override this. Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way.

1. **display: inline** It is used to display an element as an inline element.

```
<!DOCTYPE HTML>
<head>
    <title>CSS Display</title>
    <style>
        p {
            display: inline;
        }
    <style>
</head>

<body>
    <p>Hello, Welcome to kothaabhishek.co.in</p>
    <p>Click here for Java Tutorial.</p>
    <p>Click here for SQL Tutorial.</p>
    <p>Click here for HTML Tutorial.</p>
</body>
</html>
```

2. **display: block** It is used to display an element as a block element

```
<!DOCTYPE HTML>
<head>
    <title>CSS Display</title>
    <style>
        span {
                display: block;
        }
    <style>
</head>

<body>
    <span>A display property with a value of "block" results in</span>
    <span>a line break between the two elements.</span>
</body>
</html>
```

3. **display: inline-block** element is very similar to inline element but the difference is that you are able to set the width and height.

```
<!DOCTYPE HTML>
<head>
  <title>CSS Display</title>
  <style>
    li {
      display: inline-block;
      height: 100px;
      width:150px;
    }
  <style>
</head>

<body>
  <ol>
    <li><a href="HTML.html">HTML Tutorials</a></li>
    <li><a href="CSS.html">CSS Tutorials</a></li>
    <li><a href="SQL.html">SQL Tutorials</a></li>
    <li><a href="Java.html">Java Tutorials</a></li>
  </ol>
</body>
</html>
```

4. **display: none** value totally removes the element from the page. It will not take any space.

```
<!DOCTYPE HTML>
<head>
  <title>CSS Display</title>
  <style>
    h1.hidden {
          display: none;
    }
  <style>
</head>

<body>
  <h1>This is a normal heading and is visible.</h1>
  <h1 class="hidden">This is normal heading but it is not visible.</h1>
  <p>You can see that the hidden heading does not contain any space.</p>
</body>
</html>
```

CSS Float

1. The CSS float property is a positioning property. It is used to push an element to the left or right, allowing other element to wrap around it. It is generally used with images and layouts. e.g. let an

image float left to the text in a container.

- **float: right** The element floats on the right side of its containing block.

```
<!DOCTYPE HTML>
<head>
   <title>CSS Float</title>
   <style>
      img {
         float: right;
      }
   <style>
</head>

<body>
   <p><img src="abhishek.jpg" alt="image" style="width:150px;height:150px;">
   Earth is the third planet from the Sun and the only astronomical object known to harbor life.
   According to radiometric dating and other sources of evidence, Earth formed over 4.5 billion years ag
   Earth's gravity interacts with other objects in space, especially the Sun and the Moon, which is Earth
   only natural satellite.
   Earth orbits around the Sun in 365.256 days, a period known as an Earth sidereal year.
   </p>
</body>
</html>
```

- **float: left** The element floats on the left side of its containing block.
- **float: none** Removes the float property from an element.

2. The **clear** property specifies what elements can float beside the cleared element and on which side. When clearing floats, you should match the clear to the float: If an element is floated to the left, then you should clear to the left.

- **clear: left** No floating elements allowed on the left side.
- **clear: right** No floating elements allowed on the right side.
- **clear: both** No floating elements allowed on the left&right side.
- **clear: none** Allows floating elements on both sides. This is default.

```
<!DOCTYPE HTML>
<head>
   <title>CSS Float</title>
   <style>
      .div1 {
            float: left;
            width: 100px;
            height: 50px;
            margin: 10px;
            border: 3px solid #73AD21;
      }
      .div2 {
            border: 1px solid red;
      }
      .div3 {
            float: left;
            width: 100px;
            height: 50px;
            margin: 10px;
            border: 3px solid #73AD21;
      }
      .div4 {
            border: 1px solid red;
            clear: left;
      }
   <style>
</head>

<body>
<div class="div1">div1</div>
<div class="div2">div2 - Notice that div2 is after div1 in the HTML code. However, since div1 floats to th
the text in div2 flows around div1.</div>
<br><br>

<div class="div3">div3</div>
<div class="div4">div4 - Here, clear: left; moves div4 down below the floating div3. The value "left" clea
elements floated to the left. You can also clear "right" and "both".</div>
</body>
</html>
```

3. If an element is taller than the element containing it, and it is floated, it will "overflow" outside of its container. Then we can add **overflow: auto;** to the containing element to fix this problem.

```
<!DOCTYPE HTML>
<head>
    <title>CSS Float</title>
    <style>
        div {
                border: 3px solid #4CAF50;
                padding: 5px;
        }
        .img1 {
                float: right;
        }
        .clearfix {
                overflow: auto;
        }
        .img2 {
                float: right;
        }
    <style>
</head>

<body>
<div>
  <img class="img1" src="html.jpg" alt="img1" width="150" height="150">
  If an element is taller than the element containing it, and it is floated, it will "overflow" outside of its cor
  Then we can add overflow: auto; to the containing element to fix this problem.
</div>

<div class="clearfix">
  <img class="img2" src="html1.jpg" alt="img2" width="150" height="150">
  If an element is taller than the element containing it, and it is floated, it will "overflow" outside of its cor
  Then we can add overflow: auto; to the containing element to fix this problem.
</div>
</body>
</html>
```

4. We can use float with a list of hyperlinks to create a horizontal menu

```
<!DOCTYPE HTML>
<head>
   <title>CSS Float</title>
   <style>
      ul {
         list-style-type: none;
         margin: 5px;
         padding: 10px;
         overflow: auto;
         background-color: #333;
      }

      li {
         float: left;
         margin: 5px;
         padding: 10px;
      }
   <style>
</head>

<body>
<ul>
  <li><a href="#home" >Home</a></li>
  <li><a href="#Tutorials">Tutorials</a></li>
  <li><a href="#Programs">Programs</a></li>
  <li><a href="#Contact Us">Contact Us</a></li>
</ul>
</body>
</html>
```

CSS Line Height

1. It sets the differences between two lines of your content. It defines the amount of space above and below inline elements. It allows you to set the height of a line of independently from the font size.
2. You can give value in different formats like number, length & %. For example line-height:50%; (or) line-height:2; (or) line-height:20px; etc...

```
<!DOCTYPE HTML>
<head>
   <title>CSS Line Height</title>
   <style>
      p{
         line-height: 70%;
      }
   <style>
</head>

<body>
<p> It sets the differences between two lines of your content. It defines the amount of space above and
inline elements. It allows you to set the height of a line of independently from the font size.
</p>
</body>
</html>
```

CSS Opacity

1. The CSS opacity property is used to specify the transparency of an element. In simple word, you can say that it specifies the clarity of the image.
2. The opacity property can take a value from 0.0 - 1.0. The lower value, the more transparent.
3. The opacity property is often used together with the :hover selector to change the opacity on mouse-over.

```
<!DOCTYPE HTML>
<head>
   <title>CSS Opacity</title>
   <style>
      img{
         opacity: 0.5;
      }
      img:hover{
         opacity: 1.0;
      }
   <style>
</head>

<body>
<img src="abhi.jpg" alt="Image1" width="200" height="200">
<img src="abhishek.jpg" alt="Image2" width="200" height="200">
</body>
</html>
```

CSS Overflow

1. The CSS overflow property specifies how to handle the content when it overflows (that is too big to fit into an area) its block level container.
2. Let's take an example: If you don't set the height of the box, it will grow as large as the content. But if you set a specific height or width of the box and the content inside cannot fit then what will happen. The CSS overflow property is used to overcome this problem. It specifies whether to clip content, render scroll bars, or just display content.
3. This property can take one of the following values: visible (default), hidden, scroll, and auto. CSS3 also defines the overflow-x and overflow-y properties which allow for independent control of the vertical and horizontal clipping.

- **visible:** Default. The overflow is not clipped. The content renders outside the element's box.
- **hidden:** The overflow is clipped, and the rest of the content will be invisible.
- **scroll:** The overflow is clipped, and a scrollbar is added to see the rest of the content.
- **auto:** Similar to scroll, but it adds scrollbars only when necessary.

```
<!DOCTYPE HTML>
<head>
   <title>CSS Overflow</title>
   <style>
      .p1{
         width: 100px;
         height: 100px;
         overflow: scroll;
      }
      .p2{
         width: 100px;
         height: 100px;
         overflow-x: scroll;
      }
   <style>
</head>

<body>
<p class="p1">The property specifies what to do if the content of an element exceeds the size of the el
The property specifies what to do if the content of an element exceeds the size of the element's box.
The property specifies what to do if the content of an element exceeds the size of the element's box.
The property specifies what to do if the content of an element exceeds the size of the element's box.</p

<p class="p2">The property specifies what to do if the content of an element exceeds the size of the el
The property specifies what to do if the content of an element exceeds the size of the element's box.
The property specifies what to do if the content of an element exceeds the size of the element's box.
The property specifies what to do if the content of an element exceeds the size of the element's box.</p
</body>
</html>
```

CSS Position

1. The position property specifies the type of positioning method used for an element.
2. There are five different position values: **static**, **relative**, **fixed**, **absolute**, **sticky**
3. Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

- **static:** 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.
- **relative:** Relative positioning changes the position of the HTML element relative to where it normally appears. So "left:20" adds 20 pixels to the element's LEFT position. You can use two values **top** and **left** along with the position property to move an HTML element anywhere in the HTML document. You can use bottom or right values as well in the same way as top and left. A relatively positioned element can be moved and overlap other elements, but it keeps the space originally reserved for it in the normal flow.

```
<!DOCTYPE HTML>
<head>
   <title>CSS Position</title>
   <style>
      .p1{
         position: relative;
         top: 100px;
         color: #fff;
         background: #00c4cc;
      }
      .p2{
         position: relative;
         left: 100px;
         color: #fff;
         background: #00c4cc;
      }
   <style>
</head>

<body>
<p class="p1">Relative positioning changes the position of the HTML element relative to where it norm
appears. So "left:20" adds 20 pixels to the element's LEFT position. </p>

<p class="p2">Relative positioning changes the position of the HTML element relative to where it norm
appears. So "left:20" adds 20 pixels to the element's LEFT position. </p>
</body>
</html>
```

- **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.

```
<!DOCTYPE HTML>
<head>
    <title>CSS Position</title>
    <style>
        .p1{
            position: fixed;
            top: 100px;
            color: #fff;
            background: #00c4cc;
        }
        .p2{
            position: fixed;
            left: 100px;
            color: #fff;
            background: #00c4cc;
        }
    <style>
</head>

<body>
<p class="p1">Relative positioning changes the position of the HTML element relative to where it norm
appears. So "left:20" adds 20 pixels to the element's LEFT position. </p>

<p class="p2">Relative positioning changes the position of the HTML element relative to where it norm
appears. So "left:20" adds 20 pixels to the element's LEFT position. </p>
</body>
</html>
```

- **absolute:** 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.

```
<!DOCTYPE HTML>
<head>
    <title>CSS Position</title>
    <style>
        div.absolute {
            position: absolute;
            top: 80px;
            right: 30px;
            width: 200px;
            height: 100px;
            border: 3px solid red;
        }

        div.relative {
            position: relative;
            width: 400px;
            height: 200px;
            border: 3px solid green;
        }
    <style>

</head>

<body>
<div class="absolute">This div element has position: absolute;</div>

<div class="relative">This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>
</body>
</html>
```

- **sticky:** Sticky positioning is a hybrid of relative and fixed positioning. The element is treated as relative positioned until it crosses a specified threshold, at which point it is treated as fixed positioned. An element with position: sticky is positioned based on the user's scroll position.

```
<!DOCTYPE HTML>
<head>
   <title>CSS Position</title>
   <style>
      div.sticky {
         position: sticky;
         top: 0;
         padding: 5px;
         background-color: #cae8ca;
         border: 2px solid #4CAF50;
      }
   <style>
</head>

<body>
   <p>Sticky positioning is a hybrid of relative and fixed positioning. The element is treated as relative p

   <div class="sticky">I am sticky!</div>
</body>
</html>
```

- **Overlapping Elements:** When elements are positioned, they can overlap other elements. The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others). An element can have a positive or negative stack order. An element with greater stack order is always in front of an element with a lower stack order.
  Note: If two positioned elements overlap without a z-index specified, the element positioned last in the HTML code will be shown on top.

```
<!DOCTYPE HTML>
<head>
   <title>CSS Position</title>
   <style>
      img {
         position: absolute;
         left: 0px;
         top: 0px;
         z-index: -1;
      }
   <style>
</head>

<body>
   <h1>This is a heading</h1>
   <img src="abhi.jpeg" width="100" height="140">
   <p>Because the image has a z-index of -1, it will be placed behind the text.</p>
</body>
</html>
```

CSS Box Model

1. All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.
2. The CSS box model is essentially a box that wraps around every HTML element.
3. 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.
  The box model allows us to add a border around elements, and to define space between elements.

```
<!DOCTYPE HTML>
<head>
    <title>CSS Box Model</title>
    <style>
        .p1{
            position: relative;
            top: 100px;
            color: #fff;
            background: #00c4cc;
        }
        .p2{
            position: relative;
            left: 100px;
            color: #fff;
            background: #00c4cc;
        }
    <style>
</head>

<body>
<p class="p1">Relative positioning changes the position of the HTML element relative to where it norm
appears. So "left:20" adds 20 pixels to the element's LEFT position. </p>

<p class="p2">Relative positioning changes the position of the HTML element relative to where it norm
appears. So "left:20" adds 20 pixels to the element's LEFT position. </p>
</body>
</html>
```