

## What is CSS?

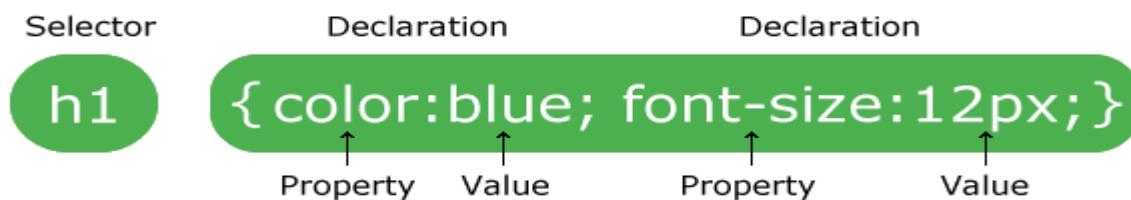
- CSS stands for Cascading Style Sheets
- Using CSS We can apply various styles to Html pages using Selectors.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once.

## Why Use CSS?

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

## CSS Syntax

A CSS rule-set consists of a selector and a declaration block:



- Selector is used to select an Html element which you want to apply the styles.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

## CSS Selectors

- If you want to apply the styles to the Html pages we use Selectors. Selectors can be classified into various types. Those are as follows:

- Element Selector
- id Selector
- class Selector
- Attribute Selector
- Pseudo Selector

### i. Element Selector

- Element Selector means we can select the specific Html Element apply the style on it.
- For Example apply the colour red for all the <p> Element then we will select the p element selector and apply the styles.

```
<style>
  p{
    color:red;
  }
</style>
```

## ii. ***Id Selector***

- The id selector uses apply the various styles to Specific Html element.
- This id should be unique it will not repeat anywhere in the document. Using id we can apply the styles. For applying the styles to Specific Html Element we can create id first for specific Html Element.
- To select an element with a specific id, write a hash (#) and, followed by the id of the element.
- For example we can apply the background image for the specific Html Element for that we need to create an id *bgImage* and apply the styles for that id.

In the *<style>* Element we will write the following code as:

```
#bgImage{  
    background-image: url("img/image5.jpeg");  
    width: 500px;  
    padding: 15px;  
}
```

In the *<body>* Element we will write the following code as:

```
<h2>BACKGROUND IMAGE</h2>  
<div id="bgImage">  
    <p>Lorem ipsum dolor sit amet,</p>  
</div>
```

## iii. ***Class Selector***

- The class selector uses apply the various styles to group of Html elements may or mayn't be the same element.
- If you apply various styles for group of Html Elements then we will use class selector.
- To select an element with a specific id, write a hash (#) and, followed by the id of the element.
- The below Example demonstrates, apply the background color, border-radius, and text-align to specific class & it will be applied the styles for group of Html Elements.

The screenshot shows a web page with the following structure:

```
<head>  
    <style>  
        .center{  
            text-align: center;  
            background-color: lightgray;  
            border-radius: 5px;  
            padding: 5px;  
        }  
    </style>  
</head>  
<body>  
    <h1 class="center">This is Heading</h1>  
    <p class="center">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aliquid, quos?  
    <p class="center">Lorem ipsum dolor sit amet.  
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.  
</body>
```

The page displays three paragraphs. The first paragraph has a light gray background, rounded corners, and centered text. The second and third paragraphs have standard black text and no styling applied.

This is Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aliquid, quos?

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

#### iv. Attribute Selector

- In this selector we will apply the styles by selecting Html attribute of a specific Html Element.
- To apply the background color green for a specific text box as follows:

```
<head>
  <style>
    input[type="text"]{
      background-color: red;
    }
  </style>
</head>
<body>
  <h1>Simple Form</h1>
  <form>
    <input type="text" placeholder="User Name">
    <input type="password" placeholder="Password">
  </form>
</body>
```

## Simple Form

User Name

Password

#### v. Pseudo Selector

- Using Pseudo Selector we will apply the styles for some special Events.

*Example 1: To change background color to red while we hover on the button*

```
<head>
  <style>
    button{
      background-color: #155084;
      color: white;
      font-size: 18px;
      padding: 5px 20px;
      border-radius: 8px;
      border: none;
    }
    button:hover{
      background-color: #dd3333;
    }
  </style>
</head>
<body>
  <h1>After Hover</h1>
  <button type="button">Hover</button>
</body>
</html>
```

## Simple Button

Hover

WS localhost:63342/CSS\_P

localhost:6

HTML Tutorial Movie

## After Hover

Hover

*Example 2:* To apply the background color red while we focus on the text box is as follows



```
<head>
  <style>
    input[type="text"]:focus {
      background-color: red;
    }
  </style>
</head>
<body>
  <h1>Simple Form</h1>
  <form>
    <input type="text" placeholder="User Name">
    <input type="password" placeholder="Password">
  </form>
</body>
```

## How Many Ways to Insert CSS

There are three ways of inserting a style sheet:

- Inline style
- Internal style sheet
- External style sheet

### Inline Style

- In this approach we will apply the styles to the Html Element by using style attribute.
- In this way we will apply the styles for specific Html Element.
- In this Inline approach each and every property separated with semicolon and each property name and value separated with colon.
- To apply the text color red for a paragraph using Inline approach as follows:

```
<body>
  <p style="color: red;">This is Inline Approach</p>
</body>
```

This is Inline Approach

### Drawbacks:

- In this approach we will apply the styles for specific element inorder to apply the same styles to multiple Html Elements we need to write style attribute for all the Html Elements. So, it is very difficult to apply styles to Html Element and it will increase the code duplication, redundancy and Increase maintainability of the application.
- Inorder to resolve this issue we have to use Internal Approach.

### Internal Style

- In this approach we will apply the style for single Html Page.
- Internal styles are defined within the <style> element, inside the <head> section of an Html page:

- The below example demonstrates apply the background color to div element.

```

<head>
  <style>
    div{
      background-color: lightgray;
      border-radius: 10px;
      padding: 10px;
      margin: 20px;
    }
  </style>
</head>
<body>
  <div>Internal Styles</div>
  <div>Internal Styles</div>
</body>

```

Internal Styles

Internal Styles

### **Drawbacks:**

- In this approach we will apply the styles for single Html page, inorder to apply the same styles for all Html Pages we need to write style element for all the Html documents. So, this is also increasing the code duplication, redundancy and Increase maintainability of the application.
- Inorder to resolve this issue we have to use External Approach.

### **External Styles**

- In this approach we will apply the styles for entire web page.
- With an external style sheet, you can change the look of an entire website by changing just one file!
- An External style sheet can be written in any text editor. The file should not contain any Html tags. The style sheet file must be saved with a .css extension.
- Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section:

```

<head>
  <link rel="stylesheet"
        type="text/css"
        href="externalStyles.css">
</head>
<body>
  <h1>External Style Sheet</h1>
  <p>Lorem ipsum dolor sit amet.</p>
</body>

```

## **External Style Sheet**

  Lorem ipsum dolor sit amet.

*In External Style Sheet with the filename as externalStyles.css*

```

p{
  color: red;
}

```

### **Cascading Order of css**

- An inline style (inside a specific Html element) has the highest priority, which means that it will override styles defined inside the <head> tag, or in an external style sheet, or a browser default value.

# CSS Colors

---

- We can specify the text color or background color for any Html Element by using various CSS Color values.
- The CSS Color values are classified as follows:
  - Textual Color
  - RGB (Red, Green, Blue)
  - HEX (Hexa Decimal)
  - HSL(Hue, Saturation, Lightness)
  - RGBA (Red, Green, Blue, Alpha)
  - HSLA (Hue, Sturation, Lightness, Alpha)

## ***Textual Color:***

- In this we will specify the color values in textual format.

*Ex:*

```
color: red;
```

## ***RGB (Red, Green, Blue)***

- In this we will specify color values in RGB format. RGB Values starts from 0 to 255.
- RGB Stands for RED, GREEN & BLUE.

*Ex:*

```
color: rgb(0,0,0);      //Black Color  
color: rgb(255,255,255); //Red Color
```

## ***HEX (Hexa Decimal)***

- In this we will specify color values in Hexa decimal format.
- The Hexa decimal values starts from 00 to FF.
- Hexa decimal must be starts with hash (#) and followed by hexa decimal values.

*Ex:*

```
color: #000000; //Black Color  
color: #ffffff; //Red Color
```

## ***HSL(Hue, Saturation, Lightness)***

- In this we will specify color values in HSL.
- HSL Stands for Hue, Sturation and Lightness

*Ex:*

```
color: hsl(360,0%,50%); // Dark Color
```

<b>Hue</b> is the Degree on the color wheel	0:Red
	120:Green
	240:Blue
	360:Black
<b>Saturation</b> will take % values from 0 to 100	0% - Shade of Gray
	100% - Full Color
<b>Lightness</b> will also take % values from 0 to 100	0% - Black Color
	50% - Neither light nor Dark Color
	100% - Full White Color

### ***RGBA (Red, Green, Blue, Alpha)***

- This is exactly same as RGB with an Extra parameter as alpha.
- Here alpha indicates the transparent values and the alpha values starts from 0 to 1.
- Here 0 indicates Full transparent and 1 indicates full Color.

*Ex:*

```
color: rgba(0,0,0,0.5);
```

### ***HSLA (Hue, Saturation, Lightness, Alpha)***

- This is exactly same as hsl with an Extra parameter as alpha.
- Here alpha indicates the transparent values and the alpha values starts from 0 to 1.
- Here 0 indicates Full transparent and 1 indicates full Color.

*Ex:*

```
color: hsla(0,0%,0%,0.5);
```

# CSS Backgrounds

---

- The CSS Backgrounds are used to apply various background styles for any specific Html Element.
- The CSS Backgrounds containing the following property values those are

***background-color:*** This property is used to apply the background color for any Html Element using CSS Color values.

Ex:

```
background-color: red;
```

***background-image:*** This property is used to apply the background Image for any Html Element.

Ex:

```
background-image: url("img/background.jpg");
```

***background-repeat:*** This property is used whether to repeat the Image x-axis or y-axis or no-repeat.

- This property containing the following property values
  - repeat-x : repeat the Image horizontally
  - repeat-y : repeat the Image vertically
  - no-repeat : Image will not repeat, only single Image will be displayed.
- The below example explains how the image is repeat in horizontally.

Ex:

```
background-image: url("img/background.jpg");
background-repeat:repeat-x;
```

***background-attachment:*** This property is used whether the Image is fixed to the background of the webpage or scrolls the image along with the scrollbar. This property contains two property values those are fixed and scroll.

- If the background-attachment is fixed it is fixed in the background it will not move.
- If the background-attachment is scroll, the Image is scrolled along with the scrollbar.

Ex:

```
background-image: url("img/background.jpg");
background-attachment: fixed;
```

***background-position:***

- This property is used to place the Image in specified position of the background of any Html Element.

```
background-image: url("img/background.jpg");
background-position: left top;
background-size: cover;
```

***Shortcut Method:***

- We can apply all the background styles by using a single CSS Property called background.  
Syntax:

```
background: color url repeat attachment position;
```

Ex:

```
background: white url("img/background.jpg")no-repeat fixed bottom right;
background-size: cover;
```

# CSS Borders

- CSS Borders are used to apply the various borders for any Html Elements.
- The CSS Borders have the following properties those are:

## ***border-width***

- Border width is used to apply the specified width of the border to the Html Element.
- The below example apply the 12px border width for the <p> Element.

```
border-width: 12px;
```

## ***border-style***

- Border style is used to apply various border styles for the Html Elements.
- We have an various border styles those are as follows: ***solid, double, dashed, dotted, ridge, groove, inset, outset***

```
#solid{  
    border-width: 2px;  
    border-style: solid;  
}  
#double{  
    border-width: 2px;  
    border-style: double;  
}  
#dash{  
    border-width: 2px;  
    border-style: dashed;  
}  
#dot{  
    border-width: 2px;  
    border-style: dotted;  
}  
#ridge{  
    border-width: 8px;  
    border-style: ridge;  
}  
#groove{  
    border-width: 8px;  
    border-style: groove;  
}  
#inset{  
    border-width: 8px;  
    border-style: inset;  
}  
#outset{  
    border-width: 8px;  
    border-style: outset;  
}
```

## Borders

Display Solid Border

Display Double Border

Display Dashed Border

Display Dotted Border

Display Ridge Border

Display Groove Border

Display Inset Border

Display Outset Border

## ***border-color***

- Border Color is used to apply the various colors to the borders for Html Elements using any css color properties.

```
#solid{  
    border-width: 2px;  
    border-style: solid;  
    border-color: red;  
}
```

## Borders

Display Solid Border

### ***Shortcut Method:***

- We can apply all the border styles by using a single CSS Property called border.  
Syntax:

```
border:width style color;
```

Ex:

```
border:2px solid red;
```

- We can also apply the borders for individual sides of an Html Elements using the properties as border-top, border-right, border-bottom, border-left

Ex:

```
#solid{  
    border-top: 4px solid red;  
    border-right: 4px solid green;  
    border-bottom: 4px solid darkred;  
    border-left: 4px solid blue;  
    padding: 10px;  
}
```

## Borders

Display Solid Border

## border-radius:

- We will specify rounded corners for any HTML Elements border we will use a special property called border-radius.

Ex: border-radius : 10px;

- This property applies the border radius for all the corners of the border.
- From CSS3 onwards we can apply the border radius for individual sides as follows:



**BTLR** : border-top-left-radius

**BTRR** : border-top-right-radius

**BBRR** : border-bottom-right-radius

**BBLR** : border-bottom-left-radius

## HTML File

```
<body>
  <section>
    <div id="br-1">
      <p>Border Radius Applied for all Sides</p>
    </div>
    <div id="br-2">
      <p>Border Radius Applied for 2 Sides</p>
    </div>
    <div id="br-3">
      <p>Border Radius Applied for Individual Sides</p>
    </div>
    <div id="br-4">
      <p>Complete Circle</p>
    </div>
  </section>
</body>
```

## CSS File

```
<style>
    section{
        width: 400px;
        margin: auto;
    }
    #br-1, #br-2, #br-3, #br-4{
        border: 2px solid green;
        background-color: green;
        border-radius: 20px;
        text-align: center;
        color: white;
        padding: 20px;
        margin-bottom: 10px;
    }
    #br-2{
        border-radius: 0;
        border-top-left-radius: 60px;
        border-bottom-right-radius: 60px;
    }
    #br-3{
        border-radius: 30px 50px 70px 100px;
    }
    #br-4{
        border-radius: 50%;
    }
</style>
```

## Result:



# CSS Margins

---

- Margin is used to provide some white space around the border (outside the border) is called as margin.
- Margin having the various properties those are as follows:
  - **margin-left:** To provide the left margin for a Html Element
  - **margin-right:** To provide the right margin for a Html Element
  - **margin-top:** To provide the top margin for a Html Element
  - **margin-bottom:** To provide the bottom margin for a Html Element
- These above properties are used to apply the margin for individual side of an Html Element.

Ex:

```
margin-left:10px;      -- apply 10px margin left to the Html Element  
margin-right:10px;     -- apply 10px margin right to the Html Element  
margin-top:10px;       -- apply 10px margin top to the Html Element  
margin-bottom:10px;    -- apply 10px margin bottom to the Html Element
```

## ***Shortcut Method:***

- In order to apply the margin for all the sides of any Html Element we will use the special property as margin.
- For example all the sides having 10px margin then we will apply margin property for the Html Element.

```
#solid{  
    margin: 10px;  
}
```

- The below example demonstrates first property value indicates *top, bottom* and second property value indicates *left and right*.

```
margin: 10px 20px;
```

# CSS Padding

---

- Padding Indicates some white spaces around the content and inside the border is called padding.
- We can specify padding for any Html Element of individual sides using the following CSS Padding properties.
  - **padding-top:** Apply top padding for any Html Element
  - **padding-right:** Apply right padding for any Html Element
  - **padding-bottom:** Apply bottom padding for any Html Element
  - **padding-left:** Apply left padding for any Html Element

Ex:

<code>padding-top:10px;</code>	-- Provide 10px top padding for Html Element
<code>padding-right:10px;</code>	-- Provide 10px right padding for Html Element
<code>padding-bottom:10px;</code>	-- Provide 10px bottom padding for Html Element
<code>padding-left:10px;</code>	-- Provide 10px left padding for Html Element

## ***Shortcut Method:***

- In order to apply the padding for all the sides of any Html Element we will use the special property as padding.
- For example all the sides having 10px padding then we will apply padding property for the Html Element.

Ex: `padding:10px;`

- The below example demonstrates first property value indicates *top, bottom* and second property value indicates *left and right*.

Ex:

`padding: 10px 20px;`

# CSS Outline

- Outline is also same as border but it will display outside the border.
- The CSS Borders have the following properties those are:

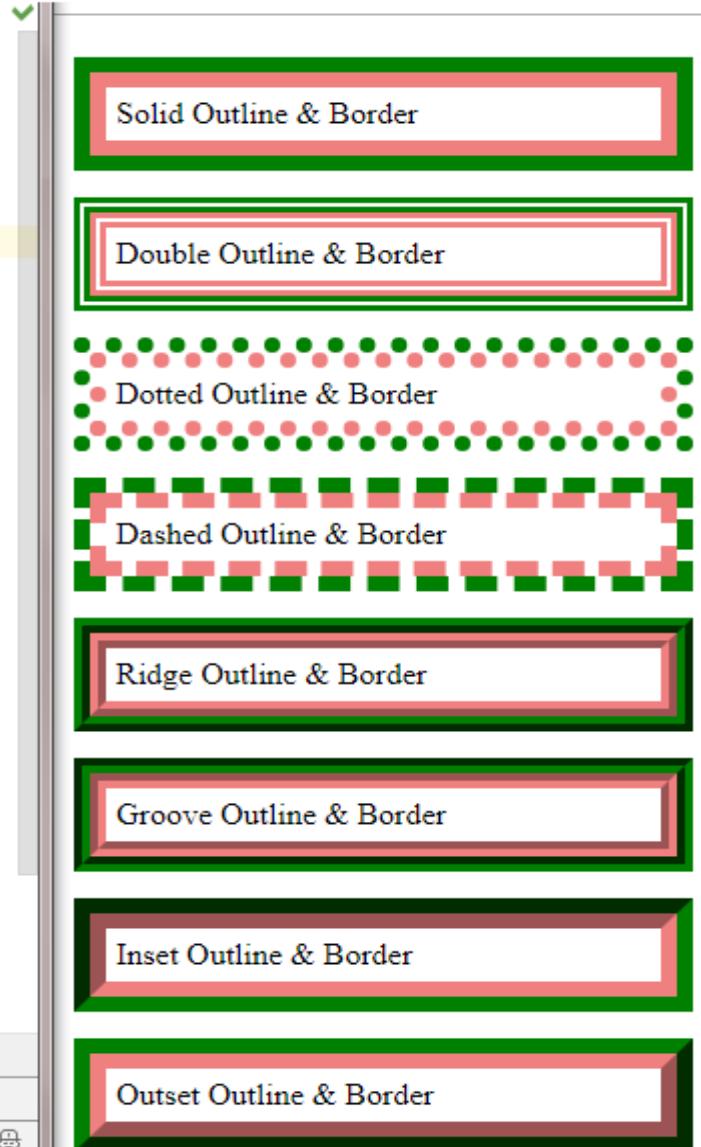
## *outline-width*

- It is used to apply the specified width of the outline to the Html Element.
- The below example apply the 12px outline width for the <p> Element.  
`outline: 12px;`

## *outline-style*

- Outline style is used to apply various outline styles for the Html Elements.
- We have an various outline styles those are as follows: **solid, double, dashed, dotted, ridge, groove, inset, outset**

```
#solid{
    outline:8px solid green;
    border: 8px solid lightcoral;
}
#double{
    outline:8px double green;
    border: 8px double lightcoral;
}
#dotted{
    outline:8px dotted green;
    border: 8px dotted lightcoral;
}
#dashed{
    outline:8px dashed green;
    border: 8px dashed lightcoral;
}
#ridge{
    outline:8px ridge green;
    border: 8px ridge lightcoral;
}
#groove{
    outline:8px groove green;
    border: 8px groove lightcoral;
}
#inset{
    outline:8px inset green;
    border: 8px inset lightcoral
}
#outset{
    outline:8px outset green;
    border: 8px outset lightcoral;
}
```



## **outline-color**

- It is used to apply the various colors to the outline for Html Elements using any css color properties.

```
#solid{  
    outline-width: 12px;  
    outline-style: solid;  
    outline-color: #00AEEF;  
}
```

### **Shortcut Method:**

- We can apply all the outline styles by using a single CSS Property called outline.  
Ex:

```
outline:12px solid darkred;
```

**Note:** There is no way to apply the Outline Individual sides

# Text Formatting

---

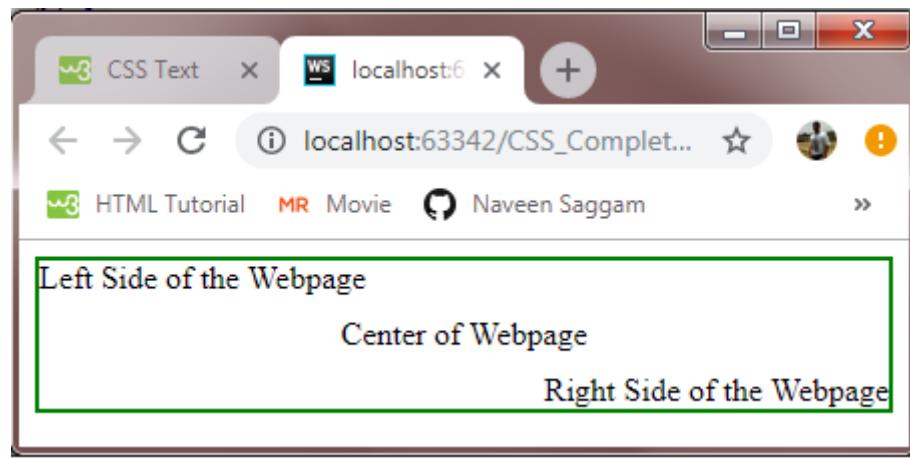
CSS Text formatting is used to apply various styles to the Html Element. CSS Text formatting consists of following properties those are:

**color :** This property is used to apply the various colors to the Html Content using various CSS Color values.

```
color:blue;
```

**text-align:** This property is used to align Html Content either left, right or centre of the webpage.

```
<head>  
    <style>  
        section{  
            border:2px solid green;  
        }  
        #left{  
            text-align: left;  
        }  
        #center{  
            text-align: center;  
        }  
        #right{  
            text-align: right;  
        }  
    </style>  
</head>  
<body>  
    <section>  
        <div id="left">Left Side of the Webpage</div>  
        <div id="center">Center of Webpage</div>  
        <div id="right">Right Side of the Webpage</div>  
    </section>  
</body>
```



**text-decoration:** Text Decoration is used to remove the underline from the anchor tag. If you define anchor tag by default underline will be appeared that is nothing but hyperlink, if you want to remove the underline then we will use text-decoration property.

Ex:

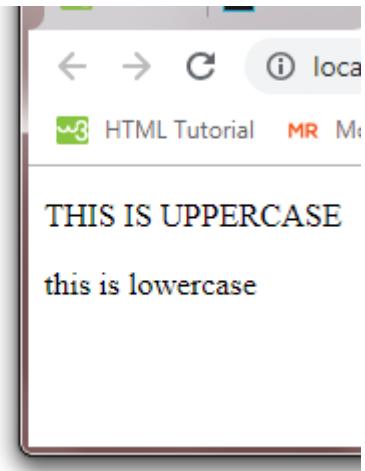
```
<head>
  <style>
    a{
      text-decoration: none;
    }
  </style>
</head>
<body>
  <h2> After adding Property Underline Removed<br>
    <a href="https://google.com">Google</a>
  </h2>
</body>
```

A screenshot of a web browser window titled "localhost:63342/CSS\_Complet...". The page content consists of two main sections: "By Default line Will be displayed" and "Google". The word "Google" is a blue hyperlink. Below this, there is another section with the text "After adding Property Underline Removed" followed by the word "Google" in blue, which is also a hyperlink. Both hyperlinks are displayed without an underline.

**text-transform:** Text transform is used to convert the Html Content into into uppercase, lowercase or capitalized.

Ex:

```
<head>
  <style>
    #upper{
      text-transform: uppercase;
    }
    #lower{
      text-transform: lowercase;
    }
  </style>
</head>
<body>
  <p id="upper">this is uppercase</p>
  <p id="lower">THIS IS LOWERCASE</p>
</body>
</html>
```



**text-indent:** text indent is used to some white space for the first line of the Html Content.

```
<body>
  <p>Normal paragraph without indent</p>
  <p style="text-indent: 50px">Lorem ipsum dolor sit amet,
    consectetur adipisicing elit. Asperiores delectus
    eum explicabo inventore minima nam odit qui
    repellat temporibus voluptates.</p>
</body>
```

Normal paragraph without indent

    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Asperiores delectus eum explicabo inventore minima nam odit qui repellat temporibus voluptates.

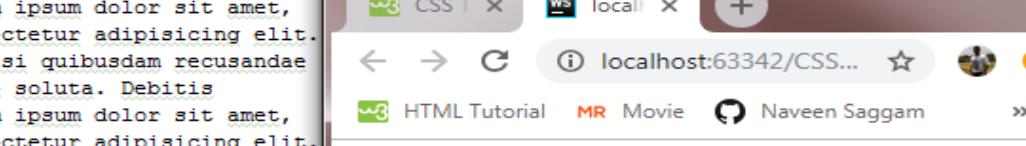
**letter-spacing:** letter spacing is used to provide some white space between the letters of a paragraph.

```
<p style="letter-spacing: 3px">  
    Letter Spacing paragraph</p>  Letter Spacing paragraph
```

**word-spacing:** Word spacing is used to provide some white space between the words of a paragraph.

```
<p style="word-spacing: 10px">  
    Word Spacing paragraph</p>
```

**line-height:** line height is used to increase or decrease the height of the each line of paragraph.

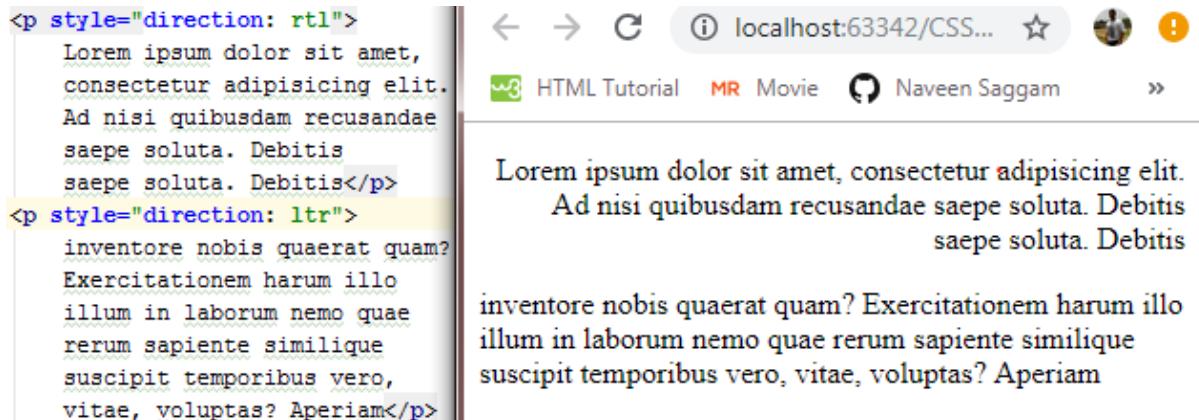


The screenshot shows a browser window with two tabs: 'CSS T' and 'local'. The local tab displays the URL 'localhost:63342/CSS...'. Below the tabs, there's a navigation bar with back, forward, and refresh buttons, along with a search bar containing 'localhost:63342/CSS...' and a star icon. The main content area shows the rendered HTML from the code editor. The first paragraph has a line height of 12px, and the second paragraph has a line height of 20px. The text is styled with various colors and fonts as defined in the CSS file.

```
<p style="line-height: 12px">
    Lorem ipsum dolor sit amet,
    consectetur adipisicing elit.
    Ad nisi quibusdam recusandae
    saepe soluta. Debitis
    Lorem ipsum dolor sit amet,
    consectetur adipisicing elit.
    Ad nisi quibusdam recusandae
    saepe soluta. Debitis</p>
<p style="line-height: 20px;">
    inventore nobis quaerat quam?
    Exercitationem harum illo
    illum in laborum nemo quae
    rerum sapiente similiqe
    suscipit temporibus vero,
    vitae, voluptas? Aperiam</p>
```

**direction:** direction is used to convert the direction either left to right or right to left. By default the direction is left to right.

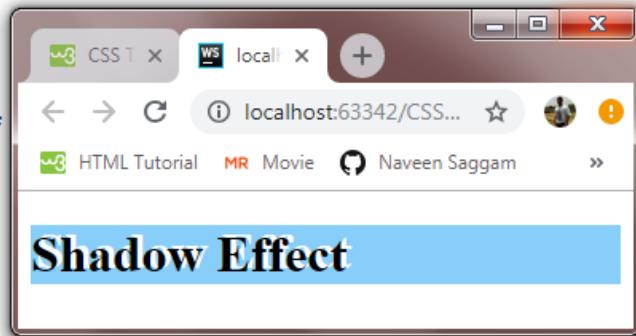
Ex:



The screenshot shows a browser window with two paragraphs of text. The first paragraph has a style of "direction: rtl", causing the text to be displayed from right to left. The second paragraph has a style of "direction: ltr", causing the text to be displayed from left to right. The browser interface includes a header with navigation icons and a sidebar with links like "HTML Tutorial", "Movie", and "Naveen Saggam".

```
<p style="direction: rtl">
    Lorem ipsum dolor sit amet,
    consectetur adipisicing elit.
    Ad nisi quibusdam recusandae
    saepe soluta. Debitis
    saepe soluta. Debitis</p>
<p style="direction: ltr">
    inventore nobis quaerat quam?
    Exercitationem harum illo
    illum in laborum nemo quae
    rerum sapiente similique
    suscipit temporibus vero,
    vitae, voluptas? Aperiam</p>
```

**text-shadow:** This property is used to apply shadow effect for the Html Content.



The screenshot shows a browser window displaying an 

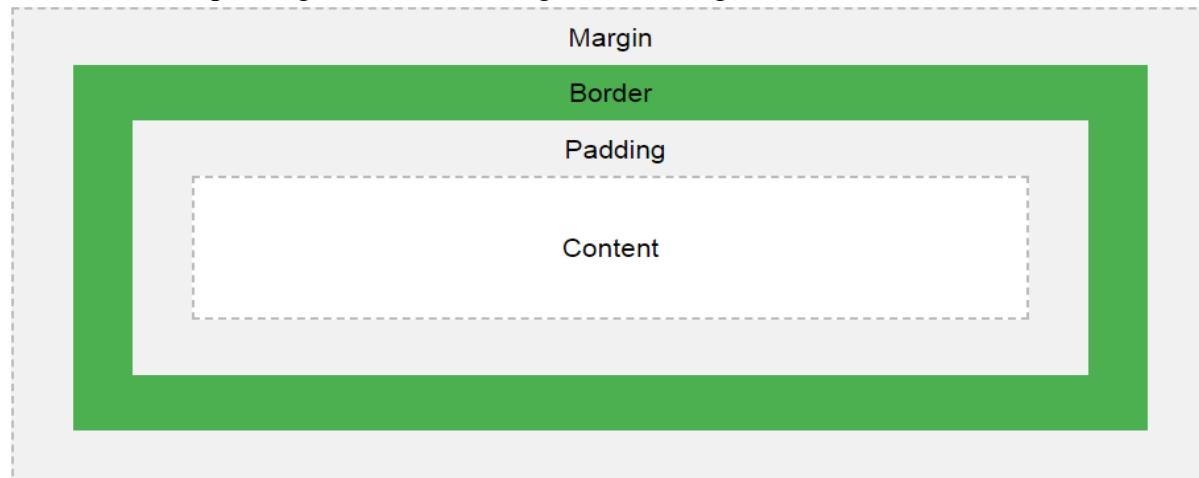
# Shadow Effect

. The heading has a text shadow of "3px -2px white", which creates a soft shadow effect below and to the left of the text. The browser interface includes a header with navigation icons and a sidebar with links like "CSS T", "localhost", "HTML Tutorial", "Movie", and "Naveen Saggam".

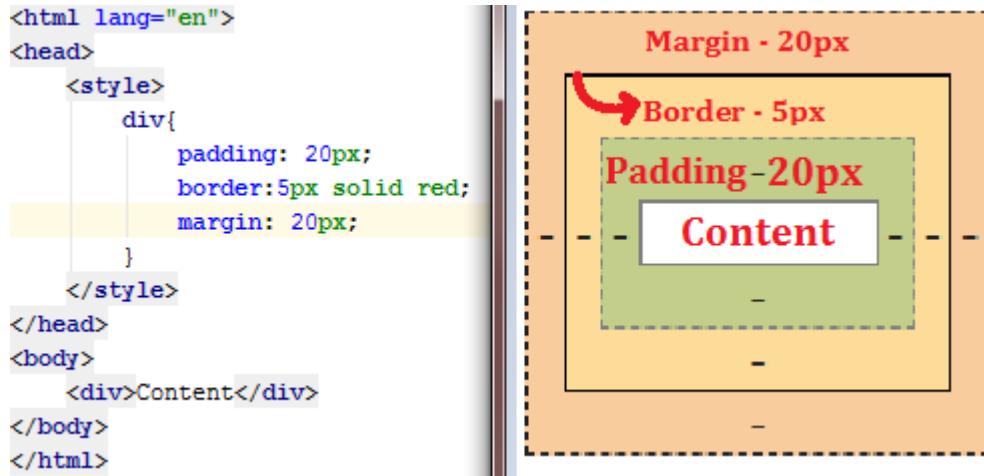
```
<html lang="en">
<head>
    <style>
        h1{
            text-shadow: 3px -2px white;
            color: black;
            background-color: lightskyblue;
            font-weight:bolder;
        }
    </style>
</head>
<body>
    <h1>Shadow Effect</h1>
</body>
</html>
```

## Box Model

- The CSS box model is essentially a box that wraps around every Html element. It consists of: Content, padding, borders and margins. The image below illustrates the box model.



- The Content Box is can be defined as where the Content and Image appear is called content box.
- Some space around the content box is called padding
- A border that goes around the content and padding
- Margin is nothing but outside the border is called margin.



- The above Example shows that the padding 20px white space created around the content and border 5px width displayed around the padding and Content and margin 20px white space is created outside the border.

## CSS Fonts

---

The CSS Fonts is used to apply the various fonts to the Html Content.  
The CSS Fonts contains the following properties:

### **font-family:**

This property is used to apply specific font to the Html Content.

In CSS fonts can be divided into 4-catagorized those are

- 1) serif (Times New Roman)
- 2) sans-Serif (Arial, Verdana, Georgia)
- 3) monospace (Courier new, Lucida console)
- 4) cursive (Google fonts)

### **font-style:**

This property is used to convert the Html Content to italic text.

### **font-size:**

This property is used to increase or decrease the size of the font. We will specify the size in px,inches,cm,mm etc..

### **font-weight:**

This property is used to convert normal text into bold or bolder text.

```
<html lang="en">
<body>
    <p style="font-family: 'Comic Sans MS', sans-serif">
        This is converted into normal text to another font
        using font family
    </p>
    <p style="font-style: italic">
        This is converted into Italic using font-style</p>
    <p style="font-size: 22px">
        This property convert text to bigger using font-size</p>
    <p style="font-weight: bolder">
        This is converted into normal text to Bolder using
        font-weight
    </p>
</body>
</html>
```

This is converted into normal text to another font using font family

*This is converted into Italic using font-style*

This property convert text to bigger using font-size

This is converted into normal text to Bolder using font-weight

## CSS Links

---

The CSS Links properties are used to apply the custom styles for the anchor tags for some special events.

For this we use the following pseudo selectors for hyperlinks.

**a:link** : This is a normal unvisited or untouched link.

**a:hover** : This pseudo selector is used to when mouse over the hyperlink is called hover.

**a:active** : This pseudo selector is used to moment is happened which is clicked that is active

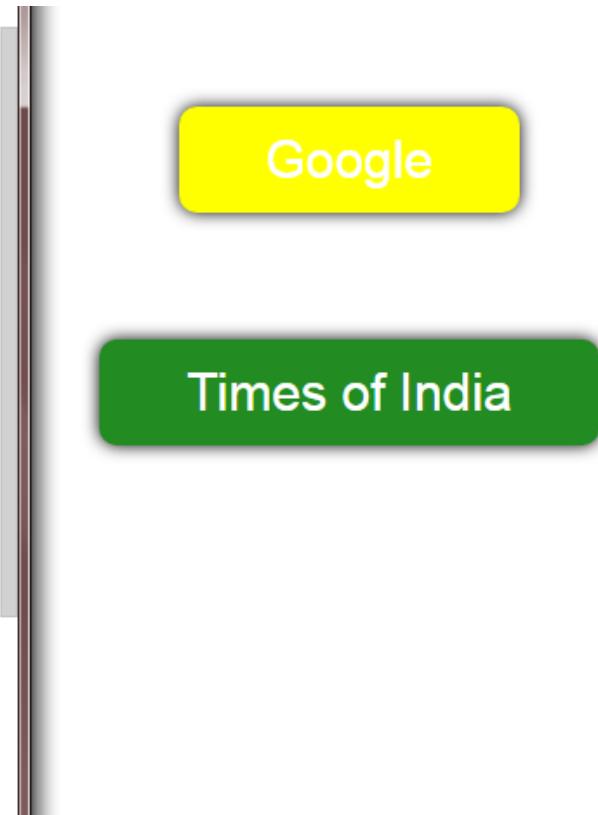
**a:visited** : The hyperlink has been visited is called visited link.

### Html file

```
<body>
    <!-- BUTTON LIKE LINK -->
    <div>
        <a href="https://www.google.com/">Google</a>
    </div>
    <div>
        <a href="https://timesofindia.indiatimes.com">
            Times of India</a>
    </div>
</body>
```

### CSS file and Output

```
<style>
    div{
        margin: 100px 0 5px 0;
        width: 500px;
        text-align: center;
    }
    a{
        font-family: Arial,sans-serif;
        text-decoration: none;
        color: white;
        font-size: 30px;
        padding: 14px 50px;
        box-shadow: 0 0 15px black;
        border-radius: 10px;
    }
    a:link{
        background-color: forestgreen;
    }
    a:hover{
        background-color: red;
    }
    a:active{
        background-color: blue;
    }
    a:visited{
        background-color: yellow;
    }
</style>
```



- Let us demonstrates the above example. The second hyperlink background color is *forest green* this is happened when user doesn't clicked or untouched this link.
- The first hyperlink background color is *yellow* this is happened when user has been visited the hyperlink.
- When the mouse over these hyperlink the background color turn into *red* color.

- When the user clicked on hyperlink but not released then the background color turn into *blue* color is called active.

## Difference between a.active and a:active

**a.active:** This represents a CSS selector. In this we select the specific anchor tag with a class name active.

Ex: <a class="active" href="https://www.google.co.in">Google</a>

**a:active:** This represents a pseudo selector for all the anchor tags, it means moment is happened which is clicked and which is applicable for all the anchor tags.

**a.active:active :** This represents the active moment for the specific anchor tag with the classname active.

# CSS POSITIONS

---

The CSS Positions are used to place the Html Element in the specific positions on the web page.

For CSS Positions we use a property called position. This Property contains the following values:

- position: static
- position: relative
- position: absolute
- position: fixed
- position: sticky

### position:static

By default all the Html Elements are with the position as static. Any Html Elements is defined with the position static then the moment which is not effected by top, left, right and bottom.

### position: relative

This is opposite to static position and its moment is effected by the properties as top, right, bottom and left but its moment is independent for any other Html Element. We normally use this position with CSS animations.

### position: absolute

This is almost same as the relative positive but its moment is purely depends on the parent Element or ancestor.

If the parent is defined with the static position then its element also be static if the parent is relative position then this also become relative. We normally use this position at dropdown lists.

## **position : fixed**

Any Html Element is defined with this position then it will move to the respective position and fix in the same position. We normally use this position to fix the Main Navigation bar at the top of the web page or fix the footer at bottom of the web page.

## **position : sticky**

Any Html Element is defined with this position then its moment is purely depends on scroll bar. We normally use this position to stick the advertisement at top of the web page.

### *CSS file*

```
<style>
    div{ font-family: "Comic Sans MS", sans-serif;
        background: white url("img/image6.jpeg") no-repeat scroll bottom;
        background-size: cover;
        font-size: 25px;
        border: 2px solid black;
        width: 300px;
        color: white;
        text-align: center;
        margin: 20px 0 20px 0;
        box-shadow: 0 0 15px black;
        border-radius: 15px; }
    .static{ position: static;
        top: 100px;
        left: 100px; }
    .relative{ position: relative;
        left: 500px; }
    .ancestor{ width: 500px;
        height: 300px;
        position: relative;
        left: 500px; }
    .absolute{ width: 200px;
        position: absolute;
    }
    .fixed{ position: fixed;
        top: 90px;
        right: 0; }
    .sticky{ width: 100%;
        top: 0;
        position: sticky; }
</style>
```

### *Html file*

```

<body>
  <div class="static">
    <p>Static</p>
  </div>
  <div class="relative">
    <p>Relative</p>
  </div>
  <div class="ancestor">
    <p>Ancestor</p>
    <div class="absolute">
      <p>Absolute</p>
    </div>
  </div>
  <div class="fixed">
    <p>Fixed</p>
  </div>
  <p>Lorem ipsum dolor sit amet, consectetur
  <div class="sticky">
    <p>Sticky</p>
  </div>
  <p>Lorem ipsum dolor sit amet, consectetur
</body>

```

*Output Screen 1:*



*Output Screen 2:*

*Lorem ipsum dolor sit amet, consectetur adipisciing elit. A delecti dicta distactio eligendi esse fugiat harum, incidunt ipsa maxime molestias necessitatibus omnis quse quidem repellat sapiente totam voluptatem voluptates voluptatum?*

*Lorem ipsum dolor sit amet, consectetur adipisciing elit. A delecti dicta distactio eligendi esse fugiat harum, incidunt ipsa maxime molestias necessitatibus omnis quse quidem repellat sapiente totam voluptatem voluptates voluptatum?*

*Lorem ipsum dolor sit amet, consectetur adipisciing elit. A delecti dicta distactio eligendi esse fugiat harum, incidunt ipsa maxime molestias necessitatibus omnis quse quidem repellat sapiente totam voluptatem voluptates voluptatum?*

*Lorem ipsum dolor sit amet, consectetur adipisciing elit. A delecti dicta distactio eligendi esse fugiat harum, incidunt ipsa maxime molestias necessitatibus omnis quse quidem repellat sapiente totam voluptatem voluptates voluptatum?*

*Lorem ipsum dolor sit amet, consectetur adipisciing elit. A delecti dicta distactio eligendi esse fugiat harum, incidunt ipsa maxime molestias necessitatibus omnis quse quidem repellat sapiente totam voluptatem voluptates voluptatum?*

*Lorem ipsum dolor sit amet, consectetur adipisciing elit. A delecti dicta distactio eligendi esse fugiat harum, incidunt ipsa maxime molestias necessitatibus omnis quse quidem repellat sapiente totam voluptatem voluptates voluptatum?*

*Lorem ipsum dolor sit amet, consectetur adipisciing elit. A delecti dicta distactio eligendi esse fugiat harum, incidunt ipsa maxime molestias necessitatibus omnis quse quidem repellat sapiente totam voluptatem voluptates voluptatum?*

*Lorem ipsum dolor sit amet, consectetur adipisciing elit. A delecti dicta distactio eligendi esse fugiat harum, incidunt ipsa maxime molestias necessitatibus omnis quse quidem repellat sapiente totam voluptatem voluptates voluptatum?*

### *Output Screen 3:*



## CSS Display

The Html Content is purely depends on the Html tags default display nature. In Html the Html tags are divided into two categories on their display nature as follows:

- ✓ Inline Elements
- ✓ Block Level Elements

### **Inline Elements:**

In Html `<a>` tag, `<span>` tag, `<input>` tag, `<button>` tag, `<img>` tag, `<b>` tag, `<i>` tag , `<abbr>` tags ,`<sub>` tag ,`<sup>` tags are comes under inline Element.

For these tags even though we write them line by line or single line in the Html Document but they will display with the same line on the web page these are called as Inline Element.

### **Block Level Elements:**

In Html `<p>` tag, `<div>` tag, `<h>` group tags, `<pre>` tag, `<ol>` & `<ul>` tags are comes under Block Level Element.

For these tags even though we write them line by line or single line in the Html Document but they will display line by line on the web page these are called as Block Level Element.

By Using CSS display property we can convert the block level elements to inline elements and vice versa.

*Display property contains the following values*

- `display : inline`
- `display : block`
- `display : inline-block`
- `display : none`

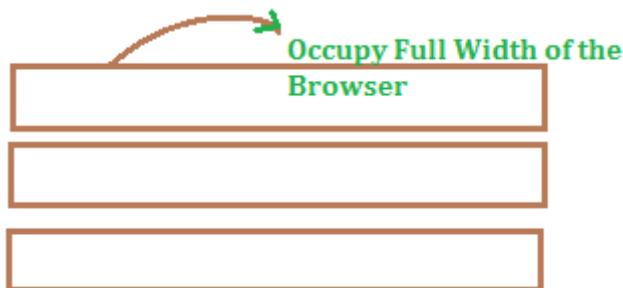
**display : inline**

- This property value is used to convert block level elements to inline.

**display : block**

- This property value is used to convert inline element to block level.

It Occupies the full width of the browser.



**display : inline-block** - This property value will display the Html element in the same line as block.



**display : none**

- This property value is used to hide any Html Element on the web page.

### *Html File*

```
<!-- DISPLAY INLINE -->
<div>
  <ul>
    <li><a href="#">HOME</a></li>
    <li><a href="#">ABOUT</a></li>
    <li><a href="#">CONTACT US</a></li>
    <li><a href="#">CAREERS</a></li>
  </ul>
</div>
<!-- Display Block -->
<div>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
    <span>1. At aut beatae cum cumque dolore</span>
    <span>2. iure libero maiores minus nisi!</span>
    <span>3. Lorem ipsum dolor sit amet,</span>
    At aut beatae cum cumque dolore exercitationem,
    iure libero maiores minus nisi odio sed temporibus!
  </p>
</div>
<!-- DISPLAY BLOCK -->
<div class="div_block">
  <a href="#">Home</a>
  <a href="#">About</a>
  <a href="#">Contact Us</a>
  <a href="#">Careers</a>
</div>
```

```

<!-- DISPLAY BLOCK -->


<form>
        <input type="text" placeholder="User Name">
        <input type="password" placeholder="Password">
        <input type="submit">
    </form>


<!-- DISPLAY NONE / HIDDEN -->


<h1>I am Hidden</h1>


```

### CSS File

```

<style>
    body{
        font-family: "Comic Sans MS", sans-serif;
    }
    div{
        padding: 10px;
        margin: 10px;
    }
    ul{
        background-color: lightgreen;
    }

    span{
        background-color: orange;
        display: block;
        margin-left: 50px;
    }
    li{
        display: inline;
    }
    .div block a{
        display: block;
    }
    form input{
        display: block;
        margin-bottom: 5px;
    }
</style>

```

### Output:

The screenshot shows a web page with the following structure:

- Header:** A green navigation bar with links: HOME, ABOUT, CONTACT US, CAREERS.
- Content Area:** A yellow rectangular box containing:
  - Text: "Lorem ipsum dolor sit amet, consectetur adipisicing elit."
  - List: "1. At aut beatae cum cumque dolore", "2. iure libero maiores minus nisi!", "3. Lorem ipsum dolor sit amet,"
  - Text: "At aut beatae cum cumque dolore exercitationem, iure libero maiores minus nisi odio sed temporibus!"
- Footer:** A white area with blue underlined links: Home, About, Contact Us, Careers.
- Form:** A user input section with three fields: "User Name", "Password", and a "Submit" button.

I am Hidden

# CSS Floats

---

The CSS float property is used to float the Html Elements either left side or right side of the web page.

Normally all the block level Html Elements will occupy full width of the browser and they will not sit the next Element to display aside to them.

For this we will apply the float effect for the first Html Elements to keep it either left side or right side of the web page.

In CSS float we use a property called float and that contains two values:

- ✓ float : left
- ✓ float : right

This float property may disturb the next element to display them.

In order to clear the float effect for the next Html Elements we use a property called “clear”

**Ex: clear : both**

**Note:** We normally use this property to display aside bar on either side of the web page. We also use this to display the newspaper article to surround the content with Images.

*Html File*

```
<body>
    <div class="left">
        <h1>Float Left</h1>
        <p>Lorem ipsum dolor sit amet, consectetur
            adipisicing elit. Beatae, sint!</p>
    </div>
    <div class="right">
        <h1>Float Right</h1>
        <p>Lorem ipsum dolor sit amet, consectetur
            adipisicing elit. Beatae, sint!</p>
    </div>
    <div>
        <h1 class="clear">clear me</h1>
    </div>
    <!-- NEWS SECTION -->
    <div class="news">
        
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing
    </div>
</body>
```

CSS File

```
<style>
body{
    font-family: "Comic Sans MS", sans-serif;
}
.left{
    border: 2px solid black;
    background-image: url("img/image4.jpeg");
    padding: 10px;
    width: 200px;
    float: left;
}
.right{
    border: 2px solid black;
    background-image: url("img/image4.jpeg");
    padding: 10px;
    width: 200px;
    float: right;
}
.clear{
    background-color: black;
    color: white;
    text-align: center;
    padding: 5px;
    clear: both;
}
.news{
    border: 2px solid black;
    padding: 10px;
}
.news img{
    float: right;
}
</style>
```

## *Output*



# CSS Transformations

- The transformation is an effect that changes an Html Shape, Size and Position. For CSS3 transformation we use a property called transform.
- The transform property allows us to visually manipulate an Element by skewing, rotating, translating and scaling.
- The CSS3 Supports 2D & 3D transform.

In CSS 2D transformation it contains the following methods for ‘transform’ property.

Method	Usage	Example
<b>translateX()</b>	This Method moves an element along X-axis by the given parameters.	transform: translateX(10px);
<b>translateY()</b>	This Method moves an element along Y-axis by the given parameters.	transform: translateY(10px);
<b>translate()</b>	This Method moves an element along X-axis and Y-axis by the given parameters.	transform: translate(10px,20px);
<b>scaleX()</b>	This Method is used to increase or decrease the size of an element along x-axis using the provided parameters.	transform: scaleX(2); - (Increases width 2 times)
<b>scaleY()</b>	This Method is used to increase or decrease the size of an element along y-axis using the provided parameters.	transform: scaleY(2); - (Increases height 2 times)
<b>scale()</b>	This Method is used to increase or decrease the size of an element along x-axis and y-axis using the provided parameters.	transform: scale(2,1); - (Increase width 2 times and height 1 times)
<b>skewX()</b>	The skewX() method is used to skew an element along the x-axis by the given parameters.	transform: skewX(45deg);
<b>skewY()</b>	The skewY() method is used to skew an element along the Y-axis by the given parameters.	transform: skewY(45deg);
<b>skew()</b>	The skew() method is used to skew an element along the x-axis and y-axis by the given parameters.	transform: skew(45deg,45deg);
<b>Matrix()</b>	This Method is a shortcut for all the above 2D transform Method. In this Method we will combine all the 2D transform methods using a mathematical function	transform : matrix(scaleX(),skewY(),skewX(), scaleY(),translateX(),translateY())
<b>rotate()</b>	This method is used to rotate an element in clock wise or counter clock wise based on the given parameters  If we provide the +ve degrees it rotates in clockwise, If we provide the -ve degrees it rotates in anti-clockwise.	transform : rotate (45deg)

### Html File

```
<body>
    <h3>Original Position</h3>
    <button>Without Transform Effect</button>
    <hr>

    <h3>translate(10px, 20px);</h3>
    <button class="translate">
        translate()
    </button>
    <br><br><hr>

    <h3>rotate(45deg);</h3>
    <button class="rotate_clock">
        rotate_clock
    </button>
    <hr>

    <h3>rotate(-45deg);</h3>
    <button class="rotate_counter">
        rotate_counter_clock
    </button>
    <hr>

    <h3>scale(2, 3);</h3>
    <button class="scale_up">
        scale up
    </button>
    <hr>

    <h3>scale(0.8, 0.8);</h3>
    <button class="scale_down">
        scale down
    </button>
    <br><br><hr>

    <h3>skewX(20deg);</h3>
    <button class="skewX_pos">
        skewX Pos
    </button>
    <br><br><hr>

    <h3>skewX(-20deg);</h3>
    <button class="skewX_neg">
        skewX Neg
    </button>
    <br><br><hr>

    <h3>skewY(20deg);</h3>
    <button class="skewY_pos">
        skewY Pos
    </button>
    <hr>
```

```

<h3>skewY (-20deg) ;</h3>
<button class="skewY_neg">
    skewY Neg
</button>
<hr>

<h3>skew(20deg, 20deg) ;</h3>
<button class="skew_pos">
    skew Pos
</button>
<br><br><hr>

<h3>skew(-20deg, -20deg) ;</h3>
<button class="skew_neg">
    skew Neg
</button>
<br><br><hr>

<h3>matrix(1, 2, -1, 1, 80, 80)</h3>
<button class="matrix">
    matrix
</button>
<hr>
</body>

```

*CSS File*

```

<style>
body{
    font-family: "Comic Sans MS", sans-serif;
}
button{
    background-image: url("img/image3.jpeg");
    color: white;
    border: none;
    font-size: 22px;
    padding: 12px 35px;
    box-shadow: 0 0 5px black;
}
.translate{
    transform: translate(10px,20px);
}
.rotate_clock{
    transform: rotate(45deg);
}
.rotate_counter{
    transform: rotate(-45deg);
}
.scale_up{
    transform: scale(2,3);
}
.scale_down{
    transform: scale(0.8,0.8);
}
.skewX_pos{
    transform: skewX(20deg);
}
.skewX_neg{
    transform: skewX(-20deg);
}

```

```

.skewY_pos{
    transform: skewY(20deg);
}
.skewY_neg{
    transform: skewY(-20deg);
}
.skew_pos{
    transform: skew(20deg,20deg);
}
.skew_neg{
    transform: skew(-20deg,-20deg);
}
.matrix{
    transform: matrix(1,2,-1,1,80,80);
}

```

### Output File



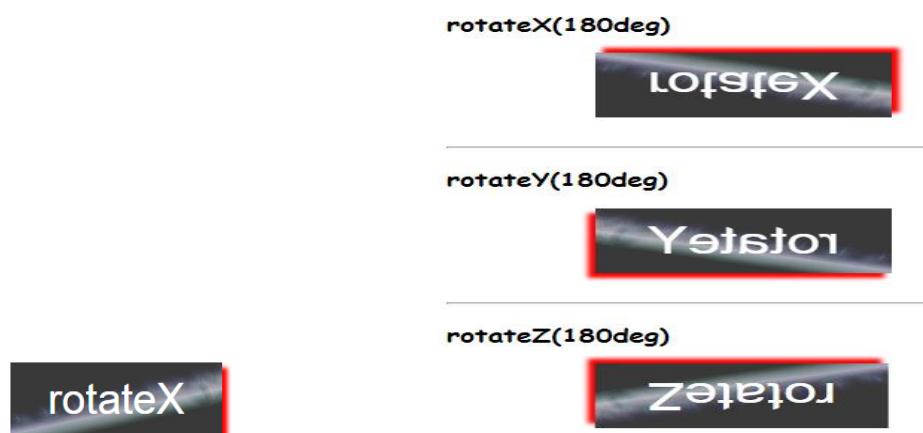
In CSS 3D transformation it contains the following methods for ‘transform’ property.

Method	Description	Example
<b>rotateX()</b>	This method rotates an element around its X-axis at a given degree.	transform : rotateX (45deg)
<b>rotateY()</b>	This method rotates an element around its Y-axis at a given degree.	transform : rotateY (45deg)
<b>rotateZ()</b>	This method rotates an element around its Z-axis at a given degree.	transform : rotateZ (180deg)

### CSS File

```
<style>
body{
    font-family: "Comic Sans MS", sans-serif;
}
button{
    background-image: url("img/image3.jpeg");
    color: white;
    font-size: 40px;
    border: none;
    padding: 12px 35px;
    margin-left: 100px;
    box-shadow: 5px 5px 5px red;
}
.rotateX:hover{
    transform: rotateX(180deg);
}
.rotateY:hover{
    transform: rotateY(180deg);
}
.rotateZ:hover{
    transform: rotateZ(180deg);
    transition:1s ease;
}
</style>
```

### Output File



### Html File

```
<body>
<h3>rotateX(180deg)</h3>
<button class="rotateX">
    rotateX
</button>
<br><br><hr>

<h3>rotateY(180deg)</h3>
<button class="rotateY">
    rotateY
</button>
<br><br><hr>

<h3>rotateZ(180deg)</h3>
<button class="rotateZ">
    rotateZ
</button>
</body>
```

# CSS3 Transitions

---

CSS transitions allows you to change property values smoothly (from one value to another), over a given duration.

To create a transition effect, you must specify two things:

- the CSS property you want to add an effect to
- the duration of the effect

**Note:** If the duration part is not specified, the transition will have no effect, because the default value is 0.

The CSS Transition consists of various properties those are as follows:

Property Name	Usage
<b>transition-name</b>	Indicates name of the transition
<b>transition-duration</b>	Indicates duration of the transition in seconds or milliseconds
<b>transition-delay</b>	To delay transition in the specified no.of seconds or milliseconds.
<b>transition-timing-function</b>	Indicates the timing function of the transition effect.

*The transition-timing-function property specifies the speed curve of the transition effect. This property can have the following values:*

- ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default).
- ease-in - specifies a transition effect with a slow start
- ease-out - specifies a transition effect with a slow end
- ease-in-out - specifies a transition effect with a slow start and end
- linear - specifies a transition effect with the same speed from start to end
- cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-Bezier function

## CSS File

```
<style>
    body{
        font-family: "Comic Sans MS", sans-serif;
    }
    div{
        height: 100px;
        width: 100px;
        background-image: url("img/image3.jpeg");
        border-radius: 10px;
        box-shadow: 0 0 15px black;
    }

```

### *Html File*

```
.t_width{
    transition:width 1s;
}
.t_width:hover{
    width:300px;
}
.t_height{
    transition: height 1s;
}
.t_height:hover{
    height: 300px;
}
.t_both{
    transition: all 1s;
}
.t_both:hover{
    width: 300px;
    height: 300px;
}
.t_rotate{
    transition: all 2s;
}
.t_rotate:hover{
    width: 300px;
    transform:rotate(180deg);
}
.t_timing{
    transition: width 2s linear;
}
.t_timing:hover{
    width: 500px;
}
.zoom{
    width: 600px;
    height: 350px;
    margin: auto;
    border-radius: 15px;
    box-shadow: 0 0 15px black;
    overflow: hidden;
}
.zoom img{
    width: inherit;
    height: inherit;
    transition: all 1s;
}
.zoom:hover img{
    transform: scale(1.5,1.5);
    cursor: pointer;
}

```

```
<body>
    <!-- Transition Width -->
    <h2>Transition Width</h2>
    <div class="t_width"></div>

    <!-- Transition Height -->
    <h2>Transition Height</h2>
    <div class="t_height"></div>

    <!-- Transition Both -->
    <h2>Transition Both</h2>
    <div class="t_both"></div>

    <!-- Transition Rotate -->
    <h2>Transition Rotate</h2>
    <div class="t_rotate"></div>

    <!-- Transition Timing Function -->
    <h2>Transition Timing Function</h2>
    <div class="t_timing"></div>

    <!--Image zoom effect-->
    <h2>Image with zoom effect</h2>
    <section class="zoom">
        
    </section>
    <br><br><br>
</body>
```

# CSS3 Animations

---

**CSS3 animations** make it possible to animate transitions from one CSS styles to another. You can change as many CSS properties you want, as many times you want.

To use CSS animation, you must first specify some *keyframes* for the animation. *Keyframes* hold what styles the element will have at certain times.

## The @keyframes Rule

When you specify CSS styles inside the `@keyframes` rule, the animation will gradually change from the current style to the new style at certain times.

We have specified when the style will change by using the keywords "from" and "to" (which represents 0% (start) and 100% (complete)).

The CSS3 Animation contains the following properties

Properties	Description
animation-name	Indicates the name of the animation
animation-duration	Indicates the duration of an animation in seconds or milliseconds
animation-direction	Indicates direction of an animation
animation-iteration-count	Indicates the no. of times an animation to be repeated.
animation-delay	Indicates the delay time frame of an animation in seconds or milliseconds.
animation-timing-function	Indicates the timing function of an animation. The values of this property are exactly same as transition-timing-function.

**Note:** The animation-duration property defines how long time an animation should take to complete. If the animation-duration property is not specified, no animation will occur, because the default value is 0s (0 seconds).

### Html File

```
<body>
    <h3>Simple Animation</h3>
    <div class="simple"></div>
    <hr>

    <h3>Spin Animation</h3>
    <div class="spin"></div>
    <hr>

    <h3>News Animation</h3>
    <div class="news">
        <p>Latest News: Lorem ipsum d
    </div>
    <hr>

    <!-- Box Animation -->
    <h3>Box Animation</h3>
    <div class="animate"></div>
    <hr>

    <!-- Foot Ball Animation -->
    <h3>Foot Ball Animation</h3>
    <div class="football"></div>
    <hr>
</body>
```

### CSS File

```
<style>
    body{
        font-family: "Comic Sans MS", sans-serif;
    }
    .simple{
        height: 100px;
        width: 100px;
        background-color: orangered;
        border-radius: 50%;
        border: 20px solid orange;
        box-shadow: 0 0 15px black;
        animation: simple 2s infinite linear;
    }
    @keyframes simple {
        0%, 50%, 100%{background-color: tomato; border-color: darkred}
        25%, 75%{background-color: lightsalmon; border-color: green}
    }
</style>
```

```

.spin{
    height: 100px;
    width: 100px;
    border-radius: 50%;
    border: 20px solid lightsalmon;
    border-top: 20px solid orangered;
    box-shadow: 0 0 15px black;
    animation:spin 2s infinite linear;
}
@keyframes spin {
    0%{transform: rotate(0deg)}
    100%{transform: rotate(360deg)}
}
.news{
    background-color: lightsalmon;
    text-align: center;
    font-size: 20px;
    width: 75%;
    margin: auto;
    padding: 10px;
    border-radius: 50px;
    font-weight: bolder;
    border: 15px solid orangered;
    overflow: hidden;
    box-shadow: 0 0 15px black;
}
.news p{
    margin: 0;
    animation: para 10s infinite linear;
}
@keyframes para {
    0%{transform: translateX(100%)}
    100%{transform: translateX(-100%)}
}
.animate{
    margin-bottom: 200px;
    height: 100px;
    width: 100px;
    background:white url("img/image5.jpeg")no-repeat fixed bottom;
    animation: animate 4s infinite linear;
    position: relative;
}
@keyframes animate {
    0%{top: 0;left: 0;transform: rotate(90deg);border-radius:0}
    25%{top: 0;left: 300px;transform: rotate(180deg);border-radius:50%}
    50%{top: 200px;left: 300px;transform: rotate(270deg);border-radius:0}
    75%{top: 200px;left: 0;transform: rotate(360deg);border-radius:50%}
    100%{top: 0;left: 0;transform: rotate(470deg);border-radius:0}
}

```

```

.football{
    width: 85px;
    height: 85px;
    margin-bottom: 250px;
    background-image: url("img/ball-clip-art.jpg");
    position: relative;
    animation: football 7s infinite linear;
}

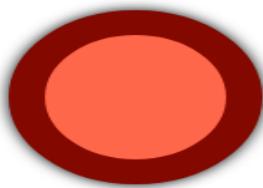
@keyframes football {
    0%{top: 0;left: 0}
    10%{top: 250px;left: 150px}
    20%{top:0; left: 300px;}
    30%{top: 250px;left: 450px}
    50%{top: 0;left: 600px}
    70%{top:250px; left: 750px}
    80%{ top: 0;left: 1000px}
    90%{ top: 250px;left: 1250px}
    100%{top: 250px;left: -10px}
}

}
</style>

```

### *Output File*

#### **Simple Animation**



#### **Spin Animation**



News Animation

Latest News: Lorem ipsum dolor sit amet, consectetur adipisicing elit. Iure, voluptatibus?

Box Animation



Foot Ball Animation



# CSS Buttons

---

*Html File : Sample Buttons*

```
<!-- Simple Buttons -->
<h2>Simple Buttons</h2>
<div>
    <a href="#" class="btn">Link</a>
    <button class="btn">Button</button>
    <input class="btn" type="button" value="Input">
</div>
```

*CSS File*

```
.btn{
    background-color: #321e10;
    color: white;
    text-decoration: none;
    border: none;
    font-family: Arial, sans-serif;
    font-size: 18px;
    padding: 14px 50px;
    outline: none;
}
```

*Output File*

**Simple Buttons**



*Html File: Sample Outline Buttons*

```
<!-- Simple Outline Buttons -->
<h2>Simple Outline Buttons</h2>
<div>
    <a href="#" class="btn-outline">Link</a>
    <button class="btn-outline">Button</button>
    <input class="btn-outline" type="button" value="Input">
</div>
```

*CSS File*

```
.btn-outline{
    border: 2px solid #321e10;
    background-color: white;
    color: #321e10;
    text-decoration: none;
    font-family: Arial, sans-serif;
    font-size: 18px;
    padding: 14px 50px;
    outline: none;
}
```

*Output File*

**Simple Outline Buttons**



### *Html File: Bootstrap Buttons*

```
<!-- Bootstrap Buttons -->
<h2>Bootstrap Buttons</h2>
<div>
  <button class="bs-btn primary">Primary</button>
  <button class="bs-btn secondary">Secondary</button>
  <button class="bs-btn success">Success</button>
  <button class="bs-btn warning">Warning</button>
  <button class="bs-btn danger">Danger</button>
</div>
```

### *CSS File*

```
.bs-btn{
  color: white;
  border: none;
  font-size: 18px;
  padding: 14px 50px;
  outline: none;
  border-radius: 50px;
}

.primary{background-color: blue}
.secondary{background-color: gray}
.success{background-color: forestgreen}
.warning{background-color: orangered}
.danger{background-color: darkred}
```

### *Output File*



### *Html File: Bootstrap Outline Buttons*

```
<!-- Bootstrap Outline Buttons -->
<h2>Bootstrap Outline Buttons</h2>
<div>
  <button class="bs-btn-outline outline-primary">Primary</button>
  <button class="bs-btn-outline outline-secondary">Secondary</button>
  <button class="bs-btn-outline outline-success">Success</button>
  <button class="bs-btn-outline outline-warning">Warning</button>
  <button class="bs-btn-outline outline-danger">Danger</button>
</div>
```

### CSS File

```
.bs-btn-outline{  
    border: 2px solid gray;  
    background-color: white;  
    color: white;  
    font-size: 18px;  
    padding: 14px 50px;  
    outline: none;  
    border-radius: 50px;  
}  
.outline-primary{color: blue; border-color: blue}  
.outline-secondary{color: gray; border-color: gray}  
.outline-success{color: forestgreen; border-color: forestgreen}  
.outline-warning{color: orangered; border-color: orangered}  
.outline-danger{color: darkred; border-color: darkred}
```

### Output File

#### Bootstrap Outline Buttons



## Vertical Navigation Bar

---

### HTML File

```
<body>  
    <aside>  
        <ul>  
            <li><a href="#" class="active">HTML</a></li>  
            <li><a href="#">CSS</a></li>  
            <li><a href="#">JavaScript</a></li>  
            <li><a href="#">JQuery</a></li>  
            <li><a href="#">AJAX</a></li>  
            <li><a href="#">Bootstrap</a></li>  
            <li><a href="#">Angular JS</a></li>  
        </ul>  
    </aside>  
</body>
```

### CSS File

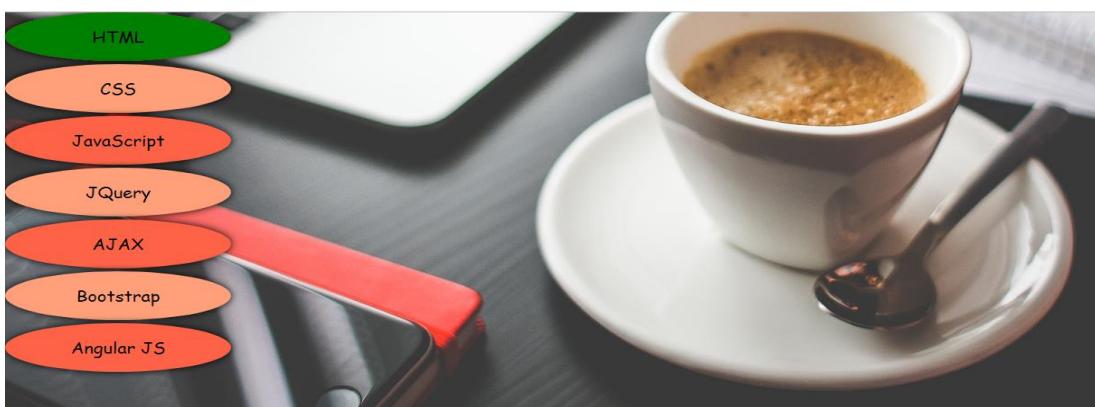
```
<style>  
    body{  
        background: white url("img/image3.jpeg") no-repeat fixed bottom;  
        margin: 0;  
    }
```

```

        aside{
            float:left;
            width: 250px;
            text-align: center;
            font-family: "Comic Sans MS", sans-serif;
            font-size: 20px;
        }
        aside ul{
            list-style-type: none;
            margin: 0;
            padding: 0;
        }
        aside ul li a{
            text-decoration: none;
            color: black;
            padding: 18px 25px;
            display: block;
        }
        aside ul li:nth-child(even){
            background-color: lightsalmon;
            border-radius: 50%;
            box-shadow: 0 0 15px black;
            margin-bottom:4px;
        }
        aside ul li:nth-child(odd){
            background-color: tomato;
            border-radius: 50%;
            box-shadow: 0 0 15px black;
            margin-bottom:4px;
        }
        aside ul li a:hover:not(.active){
            background-color: orange;
            border-radius: 50%;
        }
        aside ul li a.active{
            background-color: green;
            border-radius: 50%;
        }
    
```

</style>

### *Output File*



# Horizontal Navigation Bar

---

## Html File

```
<body>
    <div class="header">
        <h1>My Website</h1>
    </div>
    <nav>
        <ul>
            <li><a href="#" class="active">Home</a></li>
            <li><a href="#">About Us</a></li>
            <li><a href="#">Contact Us</a></li>
            <li><a href="#">Careers</a></li>
            <li><a href="#">Courses</a></li>
        </ul>
    </nav>

    <div class="mainContent">
        <h1>Welcome to our page</h1>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
    </div>
</body>
```

## CSS File

```
<style>
    body{
        font-family: "Baskerville Old Face", sans-serif;
        margin: 0;
    }
    nav{
        position: sticky;
        width: 100%;
        top: 0;
        background:white url("img/image6.jpeg") no-repeat scroll bottom;
        overflow: hidden;
    }
    nav ul{
        list-style-type: none;
        padding: 0;
        margin: 0;
    }
    nav ul li{
        float: left;
    }
```

```

nav ul li a{
    display: inline-block;
    padding: 18px 20px;
    text-decoration: none;
    color: white;
    font-size: 20px;
}
nav ul li a:hover{
    background-color: lightseagreen;
}
.mainContent{
    margin-top: 0px;
    padding: 20px;
}
.header{
    background-color: #eeeeee;
    text-align: center;
    padding: 10px;
}

```

</style>

### Output File



## Dropdown Menu

---

### Html File

```

<body>

<div class="dropdown">
    <button class="dropdown_button">Courses <i class="fa fa-caret-down"></i></button>
    <div class="dropdown_list">
        <a href="#">HTML</a>
        <a href="#">CSS</a>
        <a href="#">JavaScript</a>
        <a href="#">JQuery</a>
        <a href="#">AJAX</a>
        <a href="#">Bootstrap</a>
        <a href="#">Angular JS</a>
    </div>
</div>
</body>

```

## CSS File

```
<style>
body{
    font-family: Arial, sans-serif;
    background:white url("img/image6.jpeg")no-repeat fixed bottom;
}
.dropdown_button{
    background: white url("img/image3.jpeg")no-repeat fixed bottom;
    color: white;
    font-size: 25px;
    padding: 14px 60px;
    border: none;
    box-shadow: 0 0 15px white;
}
.dropdown_list{
    background:white url("img/image3.jpeg") no-repeat fixed bottom;
    width: 239px;
    text-align: center;
    position: absolute;
    display: none;
}

.dropdown_list a{
    display: block;
    text-decoration: none;
    color: white;
    padding: 12px 25px;
    font-size: 20px;
}
.dropdown_list a:hover{
    background-color: grey;
}
.dropdown:hover .dropdown_list{
    display: block;
}

/* To fix the bug hover effect*/
.dropdown{
    float: left;
}
</style>
```



# Dropdown Navbar

---

## Html File

```
<body>
<!-- Dropdown Navbar -->
<nav>
    <ul>
        <li><a href="#"><i class="fa fa-home"></i> Home</a></li>
        <li class="dropdown">
            <a href="#"><i class="fa fa-newspaper-o"></i> News</a>
            <div class="dropdown_list">
                <a href="#">Sports</a>
                <a href="#">Technology</a>
                <a href="#">Arts</a>
                <a href="#">Others</a>
            </div>
        </li>
        <li><a href="#"><i class="fa fa-user"></i> About Us</a></li>
        <li><a href="#"><i class="fa fa-address-book"></i> Contact Us</a></li>
        <li><a href="#"><i class="fa fa-book"></i> Careers</a></li>
        <li><a href="#"><i class="fa fa-book"></i> Courses</a></li>
    </ul>
</nav>
</body>
```

## CSS File

```
<style>
    body{
        font-family: Arial, sans-serif;
        margin: 0;
    }
    nav{
        background-color: orange;
        overflow: hidden;
    }
    nav ul{
        list-style-type: none;
        padding: 0;
        margin: 0;
    }
    nav ul li{
        float: left;
    }
    nav ul li a{
        color: black;
        font-size: 18px;
        display: inline-block;
        padding: 14px 25px;
        text-decoration: none;
    }
```

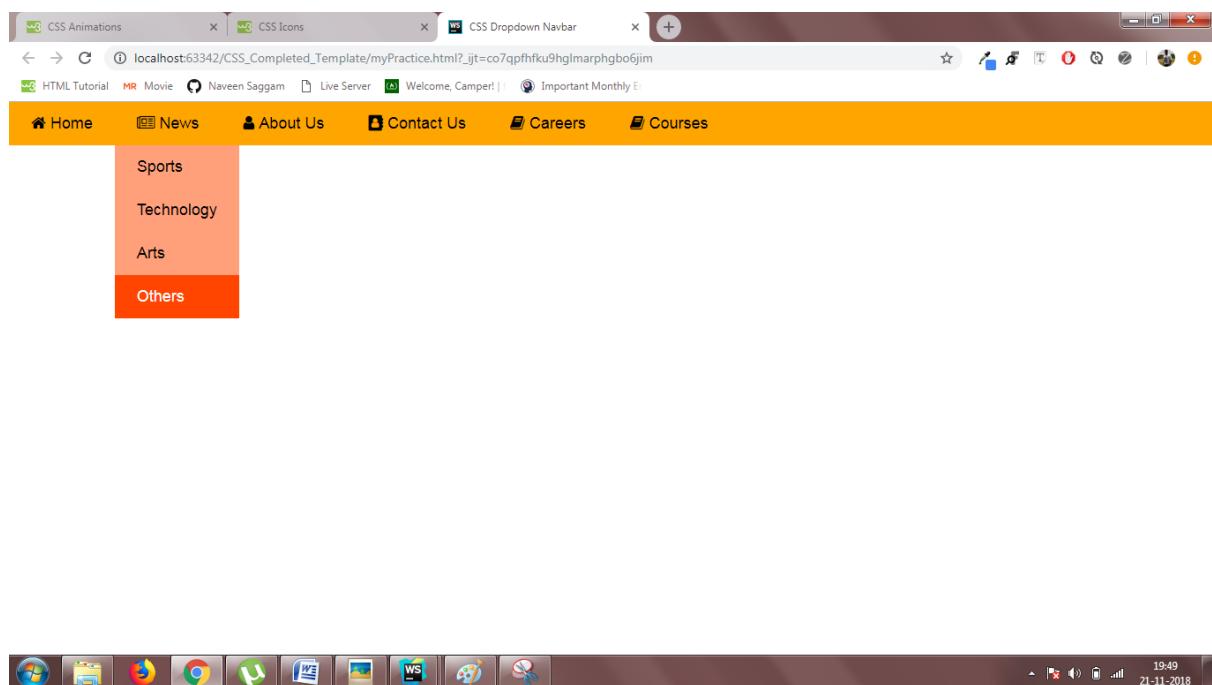
```

nav ul li a:hover{
    background-color: orangered;
    color: white;
}
.dropdown{
    float: left;
}
.dropdown_list{
    background-color: lightsalmon;
    position: absolute; /* To Stick with parent*/
    display: none;
}
.dropdown_list a{
    display: block;
}
.dropdown:hover .dropdown_list{
    display: block;
}

```

</style>

## Output



# CSS Forms

---

## *Html File*

```
<body>
    <!-- Login Form -->
    <div class="form-div">
        
        <h2>Login Here</h2>
        <form>
            <input type="text" placeholder="User Name"><br>
            <input type="password" placeholder="Password"><br>
            <input type="submit" value="Login">
        </form>
    </div>
</body>
```

## *CSS File*

```
<style>
    body{
        font-family: Arial, sans-serif;
        background: white url("img/image5.jpeg") no-repeat fixed top left;
    }
    .form-div{
        width: 250px;
        text-align: center;
        margin: 200px auto;
        background-color: rgba(0,0,0,0.5);
        box-shadow: 0 0 15px #00AEEF;
        border-radius: 10px;
    }
    .form-div img{
        width: 100px;
        height: 100px;
        margin-top: -50px;
    }
    .form-div h2{
        color: white;
        margin-top: 10px;
    }
</style>
```

```

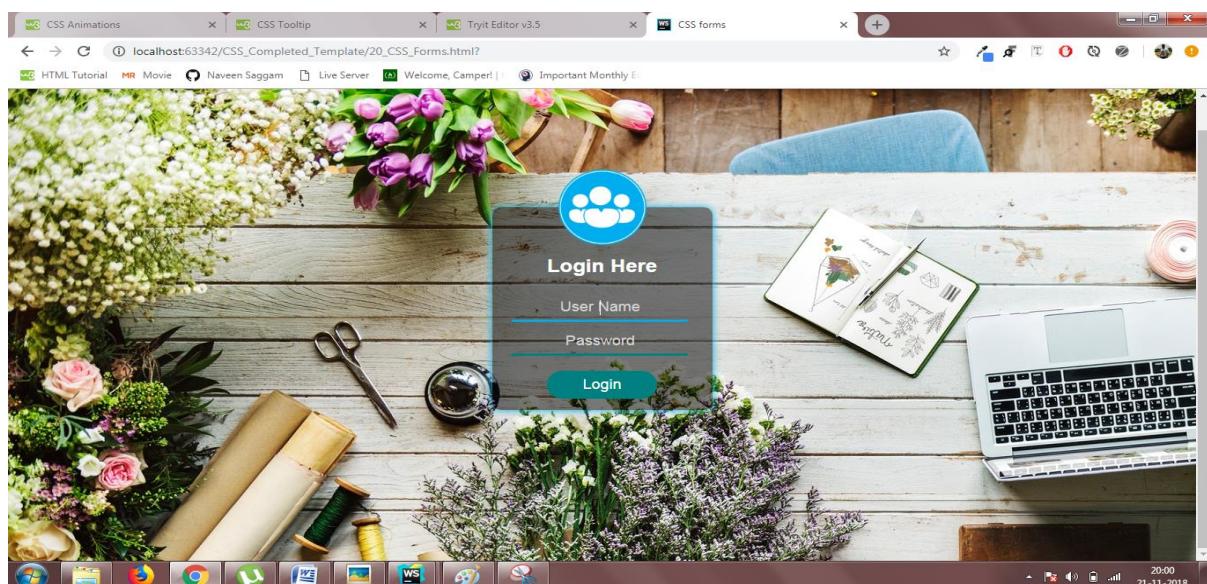
input[type='text'], input[type='password']{
    font-size: 18px;
    border: none;
    outline: none;
    height: 30px;
    margin: 5px;
    text-align: center;
    border-bottom: 3px solid teal;
    background: transparent;
    width: 200px;
    color: white;
}

::placeholder{
    color: lightgrey;
}
input[type='text']:focus,
input[type='password']:focus{
    border-bottom: 3px solid #00AEEF;
}
input[type='submit']){
    background-color: teal;
    color: white;
    border: none;
    font-size: 18px;
    margin: 15px;
    padding: 8px 40px;
    border-radius: 50px;
    outline: none;
}
input[type='submit']:hover{
    cursor: pointer;
    box-shadow: 0 0 15px white;
}

```

</style>

## Output File



# Image Gallery

---

## Html File

```
<body>
    <h1>Image Gallery</h1>

    <div class="image">
        
        <div class="desc">Image 1</div>
    </div>

    <div class="image">
        
        <div class="desc">Image 2</div>
    </div>

    <div class="image">
        
        <div class="desc">Image 3</div>
    </div>

    <div class="image">
        
        <div class="desc">Image 4</div>
    </div>

    <div class="image">
        
        <div class="desc">Image 5</div>
    </div>

    <div class="image">
        
        <div class="desc">Image 6</div>
    </div>

    <div class="image">
        
        <div class="desc">Image 7</div>
    </div>

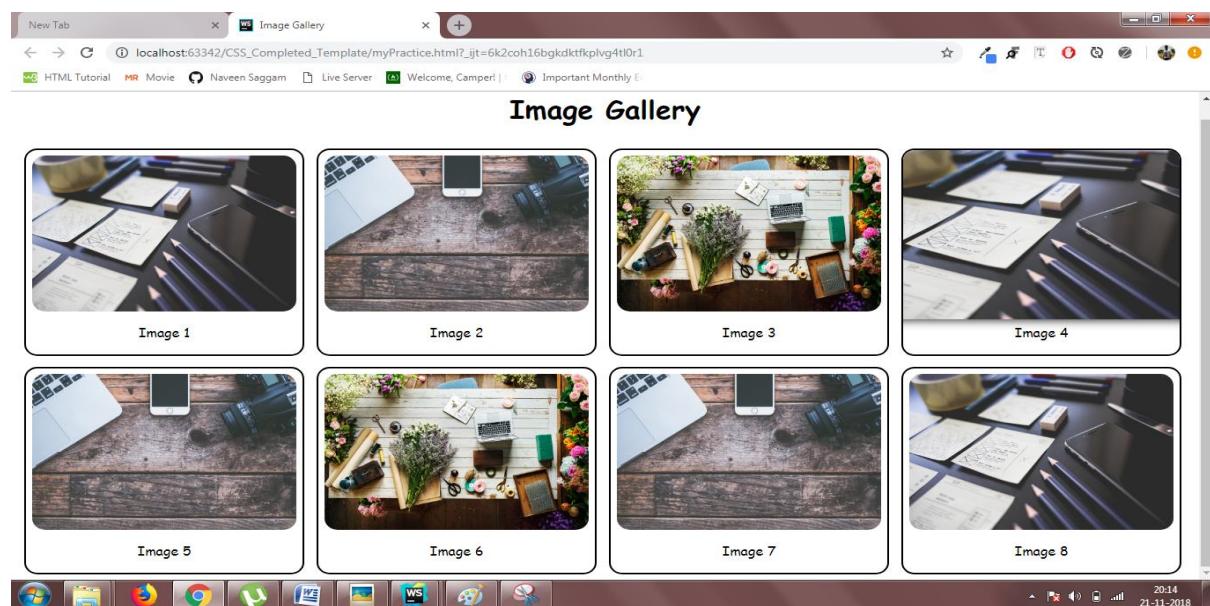
    <div class="image">
        
        <div class="desc">Image 8</div>
    </div>
</body>
```

## CSS File

```
<style>
body{
    font-family: "Comic Sans MS", sans-serif;
}
.image{
    border: 2px solid black;
    width: 300px;
    float: left;
    margin: 7px;
    padding: 7px;
    text-align: center;
    border-radius: 15px;
    overflow: hidden;
}
.image img{
    width: 100%;
    height: 200px;
    border-radius: 15px;
    transition: all 1s;
}
.desc{
    padding: 10px;
}

.image:hover img{
    transform: scale(1.5,1.1);
    cursor: pointer;
    box-shadow: 0 0 15px black;
}
h1{
    text-align: center;
}
</style>
```

## Output File



# CSS ICONS

---

- The simplest way to add an icon to your Html page, is with an icon library, such as Font Awesome. Font Awesome is a library which holds the various classes icons.
- Add the name of the specified icon using class to any inline Html element (like *<i>* or *<span>*).
- All the icons in the icon libraries below are scalable vectors that can be customized with CSS (size, color, shadow, etc.)

## Font Awesome Icons

To use the Font Awesome icons, add the following line inside the *<head>* section of your Html page:

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

**Note:** No downloading or installation is required!

- If you want to link from local; first we need to download that file and keep it into project folder from that we have to add that file to our Html Document.

**Ex:**

```
<!DOCTYPE html>
<html>
<head>
    <title>Font Awesome Icons</title>
    <meta name="viewport" content="width=device-width,
    initial-scale=1">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/
        ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
</head>
<body>
    <p>Some Font Awesome icons:</p>
    <i class="fa fa-cloud"></i>
    <i class="fa fa-heart"></i>
    <i class="fa fa-car"></i>
    <i class="fa fa-file"></i>
    <i class="fa fa-bars"></i>

    <p>Styled Font Awesome icons (size and color):</p>
    <i class="fa fa-cloud" style="font-size:24px;"></i>
    <i class="fa fa-cloud" style="font-size:36px;"></i>
    <i class="fa fa-cloud" style="font-size:48px;color:red;"></i>
    <i class="fa fa-cloud" style="font-size:60px;color:lightblue;"></i>
</body>
</html>
```

**Output**

Some Font Awesome icons:



Styled Font Awesome icons (size and color):



# Bootstrap Icons

To use the Bootstrap glyphicons, add the following line inside the <head> section of your Html page:

```
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

**Note:** No downloading or installation is required!

```
<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap Icons</title>
    <meta name="viewport" content="width=device-width,
        initial-scale=1">
    <link rel="stylesheet" href="https://
        maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>
<body class="container">
    <p>Some Bootstrap icons:</p>
    <i class="glyphicon glyphicon-cloud"></i>
    <i class="glyphicon glyphicon-remove"></i>
    <i class="glyphicon glyphicon-user"></i>
    <i class="glyphicon glyphicon-envelope"></i>
    <i class="glyphicon glyphicon-thumbs-up"></i>
    <br><br>

    <p>Styled Bootstrap icons (size and color):</p>
    <i class="glyphicon glyphicon-cloud"
        style="font-size:24px;"></i>
    <i class="glyphicon glyphicon-cloud"
        style="font-size:36px;"></i>
    <i class="glyphicon glyphicon-cloud"
        style="font-size:48px;color:red;"></i>
    <i class="glyphicon glyphicon-cloud"
        style="font-size:60px;
        color:lightblue;"></i>
</body>
</html>
```

## Output:

Some Bootstrap icons:



Styled Bootstrap icons (size and color):



# Google Icons

To use the Google icons, add the following line inside the `<head>` section of your Html page:  
`<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">`

**Note:** No downloading or installation is required!

**Ex:**

```
<!DOCTYPE html>
<html>
<head>
    <title>Google Icons</title>
    <meta name="viewport" content=
        "width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://
        fonts.googleapis.com/icon?family=Material+Icons">
</head>
<body>
    <p>Some Google icons:</p>
    <i class="material-icons">cloud</i>
    <i class="material-icons">favorite</i>
    <i class="material-icons">attachment</i>
    <i class="material-icons">computer</i>
    <i class="material-icons">traffic</i>
    <br><br>

    <p>Styled Google icons (size and color):</p>
    <i class="material-icons" style="font-size:24px;">cloud</i>
    <i class="material-icons" style="font-size:36px;">cloud</i>
    <i class="material-icons"
        style="font-size:48px;color:red;">cloud</i>
    <i class="material-icons"
        style="font-size:60px;color:lightblue;">cloud</i>
</body>
</html>
```

**Output:**

Some Google icons:



Styled Google icons (size and color):



## max-width

- A block-level element will occupy full width of the browser window (stretches out to the left and right as far as it can). The element will take up the specified width, and the remaining space will be white space. If resize the window smaller than you specified the width, then it adds horizontal scroll bar to the window.
- To overcome this problem using max-width property, will improve the browser's handling of small windows. This is important when making a site usable on small devices:

**Tip:** Resize the browser window to less than 500px wide, to see the difference between the two divs!

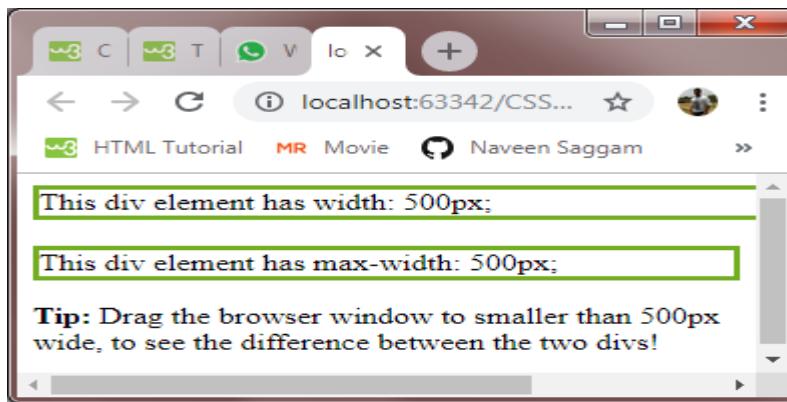
```
<!DOCTYPE html>
<html>
<head>
<style>
    div.ex1 {
        width:500px;
        border: 3px solid #73AD21;
    }
    div.ex2 {
        max-width:500px;
        border: 3px solid #73AD21;
    }
</style>
</head>
<body>

<div class="ex1">This div element has width: 500px;</div>
<br>

<div class="ex2">This div element has max-width: 500px;</div>

<p><strong>Tip:</strong> Drag the browser
window to smaller than 500px wide, to see the difference between
the two divs!</p>
</body>
</html>
```

**Output:**



**What is max-width and min-width?**

"max-width" is a **maximum width meaning** that the element cannot be bigger than the value you specified but can be smaller. "min-width" is a minimum **width meaning** that the element cannot be smaller than the value you specified but can be larger. "width" is a standard **width measurement** for defining the exact **width** of an element.

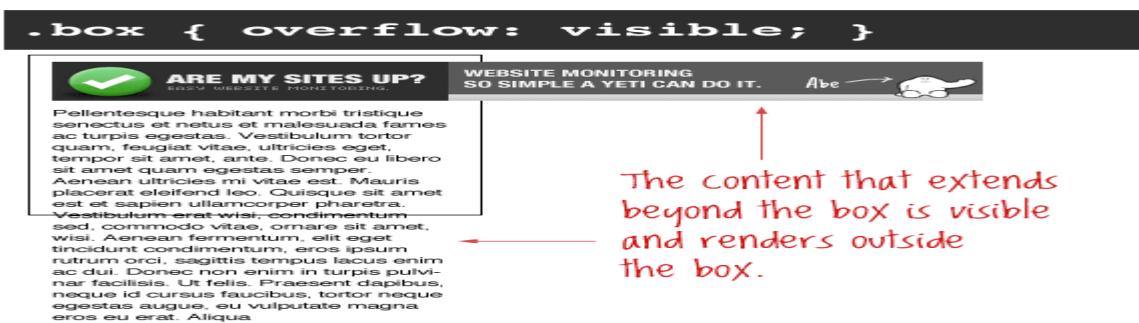
## CSS Overflow

---

- The CSS overflow property controls what happens to content that is too big to fit into an area.
- The overflow property specifies whether to clip content or to add scrollbars when the content of an element is too big to fit in a specified area.
- Imagine a div in which you've explicitly set to be 200px wide, but contains an image that is 300px wide. That image will stick out of the div and be visible by default. Whereas if you set the overflow value to hidden, the image will cut off at 200px.

*The overflow property has the following values:*

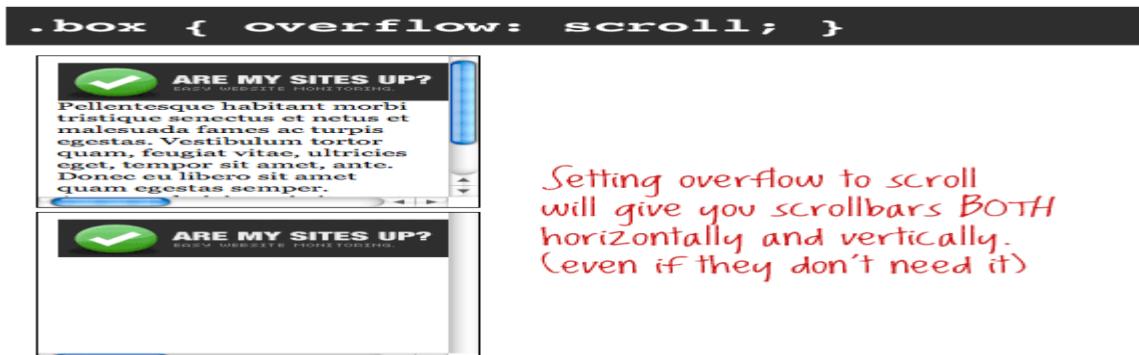
**overflow : visible** - content is not clipped when it proceeds outside its box. This is the default value of the property.



**overflow : hidden** - The overflow is clipped, and the rest of the content will be invisible



**overflow : scroll** - The overflow is clipped, but a scrollbar is added to see the rest of the content.



**overflow : auto** - If overflow is clipped, a scrollbar should be added to see the rest of the content.

```
.box { overflow: auto; }
```



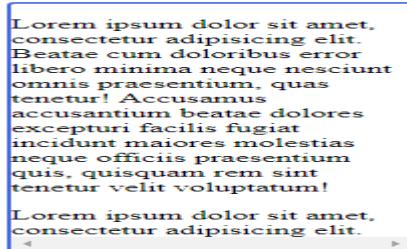
Auto overflow only gives the box the scroll bars it needs.  
(even none at all)

### overflow-x and overflow-y

- It's also possible to manipulate the overflow of content horizontally or vertically with the overflow-x and overflow-y properties.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>overflow</title>
    <style>
        #box{
            overflow-y: hidden;
            overflow-x: scroll;
            max-width: 200px;
            border: 2px solid royalblue;
            border-radius: 4px;
            height: 300px;
        }
    </style>
</head>
<body>
    <div id="box">
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
    </div>
</body>
</html>
```

Output:



initial: uses the default value which is visible

inherit: sets the overflow to the value of its parent element.

**Note:** The overflow property only works for block elements with a specified height.

**Note:** In OS X Lion (on Mac), scrollbars are hidden by default and only shown when being used (even though "overflow:scroll" is set).

## Clearfix

clearfix is used to clear the float effect of left side or right side of the webpage. If you not clear the float effect it will disturb the content because of this reason we need to use clearfix property.

#### Html File

```
<body>
    <div>
        <h1>Without Clear Fix</h1>
        
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ad, architecto
    </div>

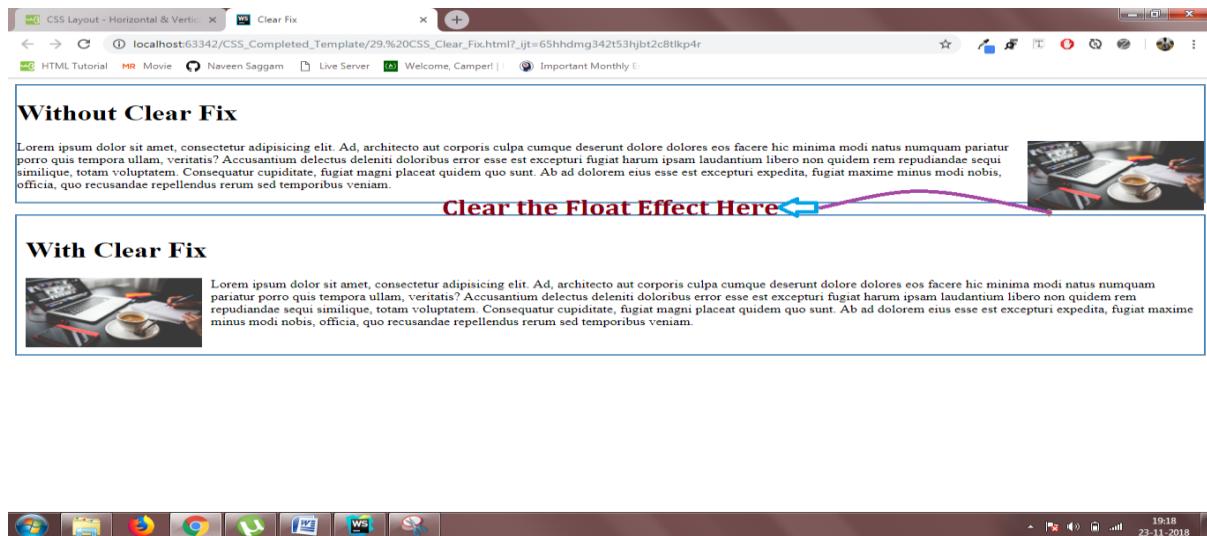
    <p id="clearboth"></p>

    <div id="clear">
        <h1>With Clear Fix</h1>
        
        <p id="space">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ad,
    </div>
</body>
```

#### CSS File

```
<style>
    div{
        border: 2px solid steelblue;
        margin-bottom: 10px;
    }
    .img1{
        float: right;
    }
    .img2{
        float: left;
    }
    #clear{
        overflow: auto;
        padding: 10px;
    }
    #clearboth{
        clear: both;
    }
    #space{
        margin-left: 210px;
    }
</style>
```

#### Output File



## Align Exactly Center

Using different property we have to handle the content exactly center of the element.

1. *Using padding and text-align we can set the content exactly to the center of the element.*

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Align Exactly Center</title>
    <style>
        div{
            border: 2px solid purple;
            border-radius: 4px;
        }
        #center{
            padding: 40px;
            text-align: center;
        }
    </style>
</head>
<body>
    <h1>Padding & text-align</h1>
    <div>
        <p id="center">Lorem ipsum dolor sit
    </div>
</body>
</html>
```

### Padding & text-align



2. *Using line-height property also we can set the content exactly to the element.*

### *Html File*

```
<body>
    <div id="line-height">
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
    </div>
</body>
```

### *CSS File*

```
<style>
    #line-height{
        border: 2px solid #04cac6;
        line-height: 200px;
        height: 200px;
        text-align: center;
    }
    #line-height p{
        line-height: 1.5;
        vertical-align: middle;
        display: inline-block;
    }
</style>
```

### *Output*

```
 Lorem ipsum dolor sit amet, consectetur adipisicing elit. Corporis, culpa dolor dolorum est modi nesciunt pariatur placeat reiciendis rerum suscipit?
```

3. Using position property and transform property also we can set the content to the center of the element.

```
<style>
    #postions{
        position: relative;
        border: 2px solid #5ED35E;
        height: 200px;
    }
    #postions p{
        margin: 0;
        position: absolute;
        top: 50%;
        left: 50%;
        transform: translate(-50%,-50%);
    }
</style>

<body>
    <div id="postions">
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
    </div>
</body>
```

### *Output*

```
 Lorem ipsum dolor sit amet, consectetur adipisicing elit. Corporis, culpa dolor dolorum est modi nesciunt pariatur placeat reiciendis rerum suscipit?
```

# CSS opacity

Opacity is used to convert the full color to transparent color. Opacity property consists of values from 0 to 1. Here 0 indicates the full transparent and 1 indicates the full color is called opacity.

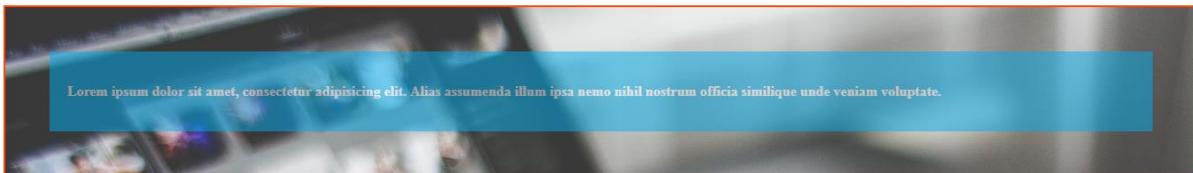
## CSS File

```
<style>
#container{
    background: url("img/image3.jpeg");
    border: 2px solid orangered;
    padding: 50px;
}
#sub-container{
    background-color: #00AEEF;
    opacity: 0.5;
    padding: 20px;
}
#sub-container p{
    font-weight: bold;
    color: white;
}
#sub-container:hover{
    opacity: 1;
}
</style>
```

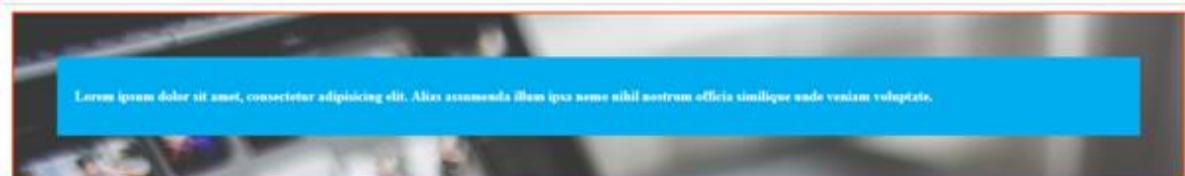
## Html File

```
<body>
    <div id="container">
        <div id="sub-container">
            <p>Lorem ipsum dolor sit amet,<br/>
            </p>
        </div>
    </div>
</body>
```

**Output:** Before hover the background color is transparent i.e., opacity value is 0.5



**Output:** hover the transparent color then it turn's into full color i.e., opacity value is 1.



# CSS Forms

---

## Html File:

```
<body>
    <div class="container">
        <!--Name of the person-->
        <div class="row">
            <div class="col-50">
                <p class="text-muted">First Name</p>
            </div>
            <div class="col-50">
                <p class="text-muted">Second Name</p>
            </div>
        </div>

        <div class="row">
            <div class="col-50">
                <form>
                    <input type="text" placeholder="First Name">
                </form>
            </div>
            <div class="col-50">
                <form>
                    <input type="text" placeholder="Second Name">
                </form>
            </div>
        </div>
        <div style="padding-bottom: 70px"></div>

        <!--Username of the person-->
        <div class="row" id="bottom">
            <div class="col-100">
                <p class="text-muted">Username</p>
            </div>
        </div>
        <div class="row">
            <div class="col-100">
                <form>
                    <input type="email">
                </form>
            </div>
        </div>
        <div class="row">
            <p class="sub-text">
                (Please enter email as your username,
                e.g. abc@gmail.com, xyz@yahoo.com)</p>
        </div>
        <div style="margin-bottom: -15px"></div>
```

```

<!--Password and Conform Password-->


<div class="col-50">
        <p class="text-muted">Password</p>
    </div>
    <div class="col-50">
        <p class="text-muted">Conform Password</p>
    </div>



<div class="col-50">
        <form>
            <input type="password">
        </form>
    </div>
    <div class="col-50">
        <form>
            <input type="password">
        </form>
    </div>


<div style="margin-bottom: 100px"></div>
<!--Language Preference-->


<div class="col-100">
        <label>Select Profile</label>
        <select style="margin-top: 10px">
            <option>Java</option>
            <option>.Net</option>
            <option>Cloud</option>
            <option>Phyton</option>
            <option>Ui Tech</option>
            <option>Oracle</option>
            <option>My Sql</option>
            <option>Angular 7</option>
        </select>
    </div>


<div style="margin-bottom: 15px"></div>
<!--Select Gender-->
<form>
    <label>Select Gender</label>
    <input type="radio" name="gender"> Male
    <input type="radio" name="gender"> Female
    <input type="radio" name="gender"> Transgender


```

```

<div style="margin-bottom: 15px"></div>
<!--Select DateofBirth-->
<form>
    <label style="margin-right: 40px">Select Date of Birth</label>
    <input type="date">
</form>

<div style="margin-bottom: 15px"></div>
<!--File Upload-->
<form>
    <label style="margin-right: 65px">Upload Resume</label>
    <input type="file">
</form>
<!--Register Button-->
<div class="row">
    <button type="button" class="button"
           style="margin-top: 20px">Register</button>
</div>
<p style="color: red; letter-spacing: 1px;font-size: 18px">
    If you are registered user you can login here.</p>
    <button type="button" class="button">Login</button>
</div>
</body>

```

### CSS File

```

<head>
    <style>
        .container{
            margin-left: 200px;
            width: 600px;
        }
        .col-50{
            width: 50%;
            float: left;
        }
        .col-100{
            width: 100%;
        }
        .text-muted, label{
            font-weight: bolder;
            color: rosybrown;
            margin-bottom: 3px;
        }
    </style>

```

```

input[type='text'], input[type='email'],
input[type='password'], select, input[type='date']{
    height: 22.5px;
    border-radius: 5px;
    border: 2px solid lightgray;
    padding: 5px;
    width: 80%;
}
input[type='radio']{
    margin-left: 72px;
}
input[type='date'] {
    width: 60%;
    margin-left: -5px;
}
input[type='file'] {

}
select{
    height: 35px;
    width: 92%;
}

input[type='text']:focus, input[type='email']:focus,
input[type='password']:focus, select:focus,
input[type='date']:focus{
    border: 1px solid #eaffff;
    box-shadow: 0 0 15px #eaffff;
}
input[type='email']){
    width: 90%;
}
.button{
    background-color: red;
    color: white;
    padding: 5px 20px;
    border: 1px solid red;
    border-radius: 4px;
}
.button:hover{
    box-shadow: 0 0 15px orangered;
}
.sub-text{
    margin-top: 2px;
    color: darkgray;
}

```

</style>

**Result:**

<b>First Name</b>	<b>Second Name</b>
<input type="text" value="First Name"/>	<input type="text" value="Second Name"/>
<b>Username</b>	
<input type="text"/>	(Please enter email as your username, e.g. abc@gmail.com, xyz@yahoo.com)
<b>Password</b>	<b>Conform Password</b>
<input type="text"/>	<input type="text"/>
<b>Select Profile</b>	
<input type="text" value="Java"/>	▼
<b>Select Gender</b>	<input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Transgender
<b>Select Date of Birth</b>	<input type="text" value="dd - mm - yyyy"/>
<b>Upload Resume</b>	<input type="button" value="Choose file"/> No file chosen
<input type="button" value="Register"/>	
If you are registered user you can login here.	
<input type="button" value="Login"/>	

# CSS Gradients

---

CSS gradients will display smooth transitions between two or more specified colors.

CSS defines two types of gradients:

- **Linear Gradients (goes down/up/left/right/diagonally)**
- **Radial Gradients (defined by their center)**

## CSS Linear Gradients

To create a linear gradient you must define at least two color stops. You can also set a starting point and a direction (or an angle) along with the gradient effect.

**Syntax :** background-image: linear-gradient(direction, color1, color2, ...);

### Html File

```
<body>
    <section>
        <div id="linear">Linear Gradient toRight</div>
        <div id="linear-tobottom">Linear Gradient toBottom</div>
        <div id="linear-diagonal">Linear Gradient toDiagonal</div>
        <div id="linear-angle">Linear Gradient toAngle</div>
    </section>
</body>
```

### CSS File

```
<style>
    section{
        margin-bottom: 10px;
        text-align: center;
        color: white;
    }
    #linear{
        background-image: linear-gradient(to right, black, red, black);
        height: 150px;
        line-height: 150px;
        font-size: 20px;
    }
    #linear-tobottom{
        background-image: linear-gradient(red, white, green);
        margin-top: 10px;
        color: black;
        height: 150px;
        line-height: 150px;
        font-size: 20px;
    }
</style>
```

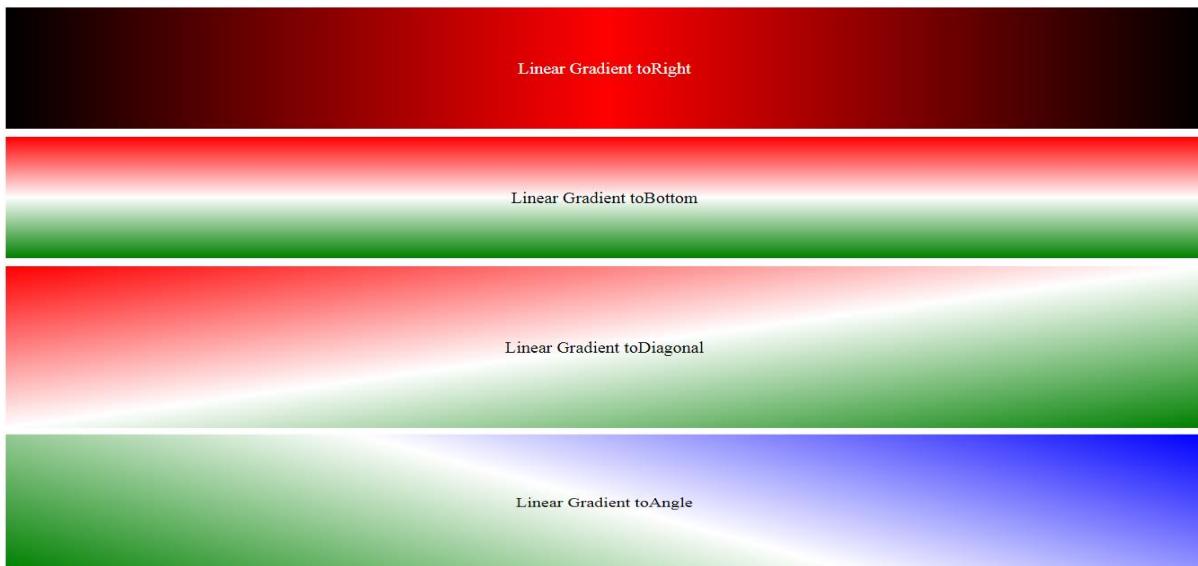
```

#linear-diagonal{
    background-image: linear-gradient(to bottom right, red, white, green);
    margin-top: 10px;
    color: black;
    height: 200px;
    line-height: 200px;
    font-size: 20px;
}

#linear-angle{
    background-image: linear-gradient(200deg, blue, white, green);
    margin-top: 10px;
    color: black;
    height: 200px;
    line-height: 200px;
    font-size: 20px;
}

```

### Result



## CSS Radial Gradients

A radial gradient is defined by its center. To create a radial gradient you must also define at least two color stops.

**Syntax :** `background-image: radial-gradient(shape size at position, color, ..., color);`

By default, shape is ellipse, size is farthest-corner, and position is center.

### Html File

```

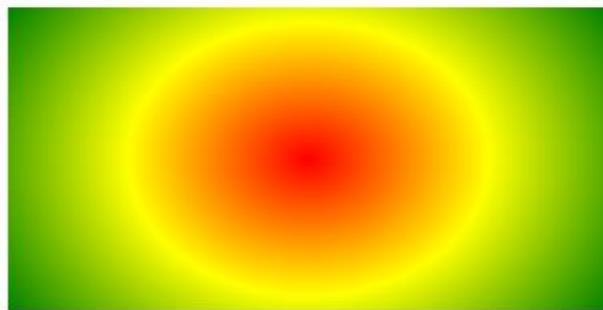
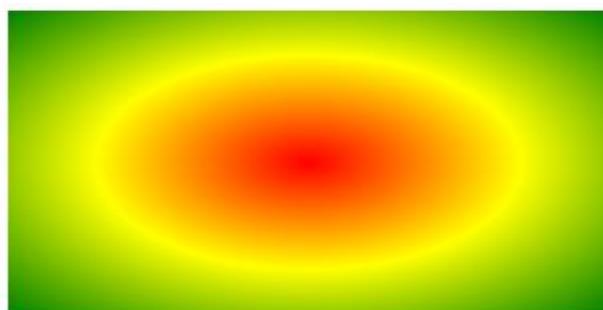
<body>
    <section>
        <div id="grad1"></div>
        <div id="grad2"></div>
    </section>
</body>

```

### CSS File

```
<style>
    section{
        margin-bottom: 10px;
        text-align: center;
        color: white;
    }
    #grad1 {
        margin-top: 10px;
        height: 200px;
        width: 300px;
        background: red radial-gradient(red, yellow, green);
    }
    #grad2 {
        margin-top: 10px;
        height: 200px;
        width: 300px;
        background: red radial-gradient(circle, red, yellow, green);
    }
</style>
```

### Result



# Pagination

---

## Html File

```
<body>
    <div id="pagination">
        <a href="#">&laquo;</a>
        <a href="#" id="active">1</a>
        <a href="#">2</a>
        <a href="#">3</a>
        <a href="#">4</a>
        <a href="#">5</a>
        <a href="#">6</a>
        <a href="#">7</a>
        <a href="#">8</a>
        <a href="#">9</a>
        <a href="#">&raquo;</a>
    </div>
</body>
```

## CSS File

```
<style>
    #pagination{
        display: inline-block;
        margin: 20px 400px;
    }
    #pagination a{
        text-decoration: none;
        padding: 10px;
        color: black;
        float: left;
        border: 1px solid green;
        border-right: none;
    }
    #pagination a:last-child{
        border-right: 1px solid green;
    }
    #pagination a:hover:not(#active) {
        background-color: lightgreen;
        color: white;
        padding: 10px;
    }
    #active{
        background-color:green;
        padding: 10px;
    }
    #pagination #active{
        color: white;
    }
</style>
```

## Result

«	1	2	3	4	5	6	7	8	9	»
---	---	---	---	---	---	---	---	---	---	---

# CSS Multi-column Layout

The CSS multi-column layout allows easy definition of multiple columns of text - just like in newspapers.

Properties	Usage	Example
column-count	This property specifies the number of columns an element	column-count: 3;
column-gap	This property specifies the gap between the columns.	column-gap: 40px;
column-rule-style	This property specifies the style of the rule between columns	column-rule-style: solid;
column-rule-width	This property specifies the width of the rule between columns	column-rule-width: 1px;
column-rule-color	This property specifies the color of the rule between columns	column-rule-color: lightblue;
column-rule	This property is a shorthand property for setting all the column-rule.	column-rule: 1px solid lightblue;
column-span	This property specifies how many columns an element should span across	column-span: all;
column-width	This property specifies a suggested, optimal width for the columns	column-width: 100px;

## Html File

```
<body>
  <p>
    <b>Note:</b> Firefox and Internet Explorer 9 (and earlier versions)
    do not support the column-span property.</p>

  <div class="newspaper">
    <h2>Lorem Ipsum Dolor Sit Amet</h2>
    Lorem ipsum dolor sit amet, consectetuer adipiscing elit,
    sed diam nonummy nibh euismod tincidunt ut laoreet dolore
    magna aliquam erat volutpat. Ut wisi enim ad minim veniam,
    quis nostrud exerci tation ullamcorper suscipit lobortis
    nisl ut aliquip ex ea commodo consequat. Duis autem vel
  </div>
</body>
```

## CSS File

```
<style>
  .newspaper {
    column-count: 3;
    column-gap: 40px;
    column-rule: 1px solid lightblue;
  }
  h2 {
    column-span: all;
  }
</style>
```

## Result

Note: Firefox and Internet Explorer 9 (and earlier versions) do not support the column-span property.

### **Lorem Ipsum Dolor Sit Amet**

  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliqua. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel

# CSS User Interface

- This User Interface is used to resize the text field as per user convenient. We can resize the text field by dragging horizontally or vertically. User Interface has two properties `resize` and `outline-offset`.

## CSS Resizing

- The `resize` property specifies if (and how) an element should be resizable by the user.

### *Html File*

```
<body>
  <section>
    <div id="resize-horizontal">
      <h3>Horizontal Resize</h3>
      <p>Lorem ipsum dolor sit amet,<br/>
        consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliqua. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel
    </div>

    <div id="resize-vertical">
      <h3>Vertical Resize</h3>
      <p>Lorem ipsum dolor sit amet,<br/>
        consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliqua. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel
    </div>

    <div id="resize-both">
      <h3>Both Resize</h3>
      <p>Lorem ipsum dolor sit amet,<br/>
        consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliqua. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel
    </div>
  </section>
</body>
```

### *CSS File*

```
<style>
  section{
    width: 500px;
    margin: auto;
  }
  div{
    border: 1px solid red;
    margin-bottom: 20px;
    padding: 10px;
  }
  h3{
    text-align: center;
  }
```

```

#resize-horizontal{
    resize: horizontal;
    overflow: auto;
}
#resize-vertical{
    resize: vertical;
    overflow: auto;
}
#resize-both{
    resize: both;
    overflow: auto;
}

```

</style>

### *Result*

The image shows three separate boxes demonstrating different resize behaviors:

- Horizontal Resize:** A box with a red border containing the text "Horizontal Resize". Inside the box, the text "Lorem ipsum dolor sit amet, consectetur adipisicing elit. Delectus harum impedit nostrum quasi quibusdam unde! Eligendi quae quia quis repellat." is displayed.
- Vertical Resize:** A box with a red border containing the text "Vertical Resize". Inside the box, the text "Lorem ipsum dolor sit amet, consectetur adipisicing elit. Delectus harum impedit nostrum quasi quibusdam unde! Eligendi quae quia quis" is displayed, with a vertical scroll bar on the right side.
- Both Resize:** A box with a red border containing the text "Both Resize". Inside the box, the text "Lorem ipsum dolor sit amet, consectetur adipisicing elit. Delectus harum impedit nostrum quasi quibusdam unde! Eligendi quae quia quis repellat." is displayed.

## CSS Outline Offset

- The outline-offset property adds space between an outline and the edge or border of an element.

Outlines differ from borders in three ways:

- An outline is a line drawn around elements, outside the border edge
- An outline does not take up space

This div has an outline 15px outside the border edge.

The following example uses the outline-offset property to add space between the border and the outline:

## *Html File*

```
<body>
    <h1>The outline-offset Property</h1>
    <div class="ex1">This div has a 4 pixels solid</div>
    <br>
    <div class="ex2">This div has a 5 pixels dashed</div>
</body>
```

## *CSS File*

```
<style>
    div.ex1 {
        margin: 20px;
        border: 1px solid black;
        outline: 4px solid red;
        outline-offset: 15px;
    }
    div.ex2 {
        margin: 10px;
        border: 1px solid black;
        outline: 5px dashed blue;
        outline-offset: 5px;
    }
</style>
```

## *Result*

### **The outline-offset Property**

This div has a 4 pixels solid red outline 15 pixels outside the border edge.

This div has a 5 pixels dashed blue outline 5 pixels outside the border edge.

# CSS Box Sizing

---

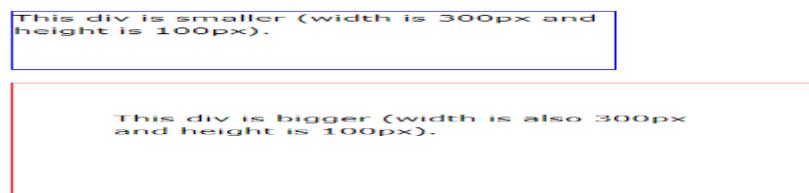
The CSS box-sizing property allows us to include the padding and border in an element's total width and height.

## Without the CSS box-sizing Property

By default, the width and height of an element is calculated like this:

- width + padding + border = actual width of an element
- height + padding + border = actual height of an element

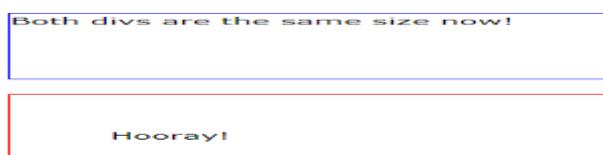
The following illustration shows two `<div>` elements with the same specified width and height:



## With the CSS box-sizing Property

The box-sizing property allows us to include the padding and border in an element's total width and height.

If you set `box-sizing: border-box;` on an element padding and border are included in the width and height:



### Html File

```
<body>
    <div class="div1">Both divs are the same size now!</div>
    <br>
    <div class="div2">Hooray!</div>
</body>
```

### CSS File

```
<style>
    .div1 {
        width: 300px;
        height: 100px;
        border: 1px solid blue;
        box-sizing: border-box;
    }
    .div2 {
        width: 300px;
        height: 100px;
        padding: 50px;
        border: 1px solid red;
        box-sizing: border-box;
    }
</style>
```

# CSS Flexbox Layout Module

Before the Flexbox Layout module, there were four layout modes:

- For sections in a webpage we are using Block level.
- For text we are using Inline
- For two-dimensional table data we are using Table
- For explicit position of an element we are using positioned.

The Flexible Box Layout Module makes it easier to design flexible responsive layout structure without using float or positioning. Flexible consists of flex containers and flex items.

Property	Usage	Example
<b>flex-direction</b>	This property defines in which direction the container wants to stack the flex items.	<code>flex-direction: column;</code> <code>flex-direction: column-reverse;</code> <code>flex-direction: row;</code> <code>flex-direction: row-reverse;</code>
<b>flex-wrap</b>	This property specifies whether the flex items should wrap or not.	<code>flex-wrap: wrap;</code> <code>flex-wrap: nowrap;</code> <code>flex-wrap: wrap-reverse;</code>
<b>flex-flow</b>	This property is a shorthand property for setting both the <code>flex-direction</code> and <code>flex-wrap</code> properties.	<code>flex-flow: row wrap;</code> <code>flex-flow: row nowrap;</code> <code>flex-flow: row wrap-reverse;</code> <code>flex-flow: column wrap;</code> <code>flex-flow: column nowrap;</code> <code>flex-flow: column wrap-reverse;</code>
<b>justify-content</b>	This property is used to align the flex Items center, flex-start, flex-end, space around the flex & space between the flex.	<code>justify-content: center;</code> <code>justify-content: flex-start;</code> <code>justify-content: flex-end;</code> <code>justify-content: space-around;</code> <code>justify-content: space-between;</code>
<b>align-items</b>	This property is used to align the flex Items middle of the flex-container, Starts, End of the flex-container.	<code>align-items: center;</code> <code>align-items: flex-start;</code> <code>align-items: flex-end;</code> <code>align-items: stretch;</code> <code>align-items: baseline;</code>
<b>align-content</b>	This property is used to align the flex lines.	<code>align-content: center;</code> <code>align-content: flex-start;</code> <code>align-content: flex-end;</code> <code>align-content: space-between;</code> <code>align-content: space-around;</code> <code>align-content: stretch;</code>
<b>align-self</b>	This Property is used to align the Self flex item to center, flex-start or flex-end	<code>align-self: center;</code> <code>align-self: flex-start;</code> <code>align-self: flex-end;</code>

## For Defining Flex the property is flex

CSS File

```
.flex-container{  
    display: flex;  
    background-color: #00AEEF;  
}  
  
.flex-container div{  
    background-color: ghostwhite;  
    font-size: 30px;  
    padding: 25px 20px;  
    margin: 5px;  
    text-align: center;  
}
```

Html File

```
<!--Main root Element -->  
<h1>Main root Flex Box Element</h1>  
<div class="flex-container">  
    <div>1</div>  
    <div>2</div>  
    <div>3</div>  
</div>
```

Result

### Main root Flex Box Element



## For Defining alignment of the flex direction the property is flex-direction

CSS File

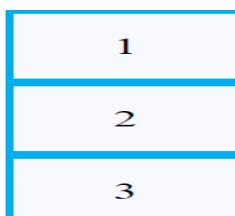
```
.flex-direction-column{  
    display: flex;  
    flex-direction: column;  
    background-color: #00AEEF;  
}  
  
.flex-direction-column > div{  
    background-color: ghostwhite;  
    font-size: 30px;  
    padding: 25px 20px;  
    margin: 5px;  
    width: 100px;  
    text-align: center;  
}
```

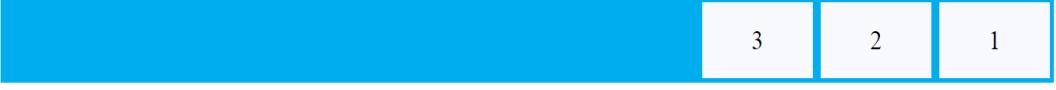
Html File

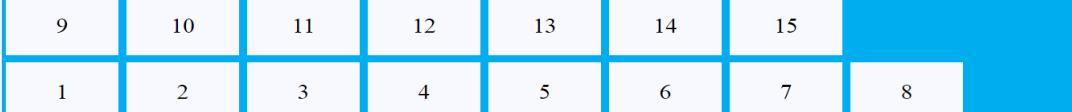
```
<!--Flex Direction from top to bottom-->  
<h1>Flex Direction from top to bottom</h1>  
<div class="flex-direction-column">  
    <div>1</div>  
    <div>2</div>  
    <div>3</div>  
</div>
```

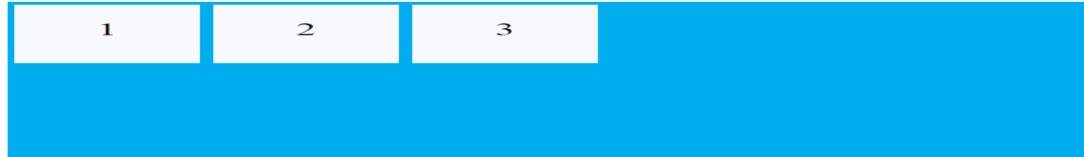
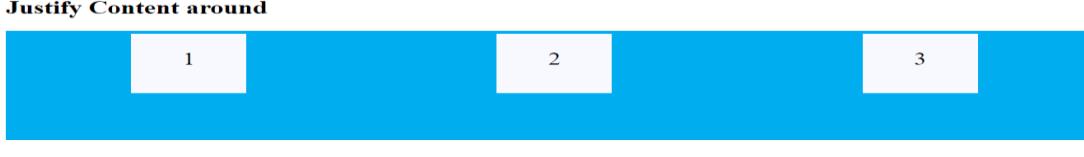
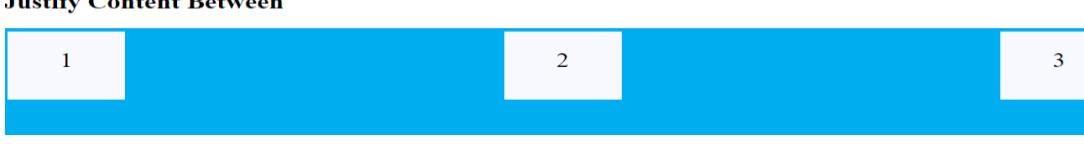
Result

### Flex Direction from top to bottom

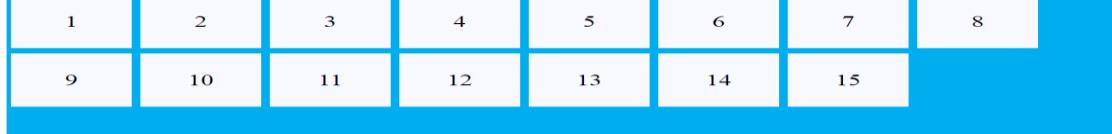
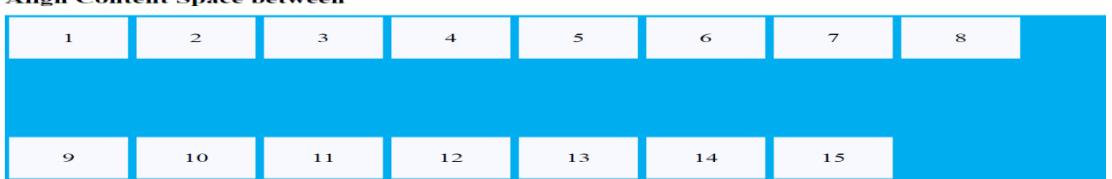
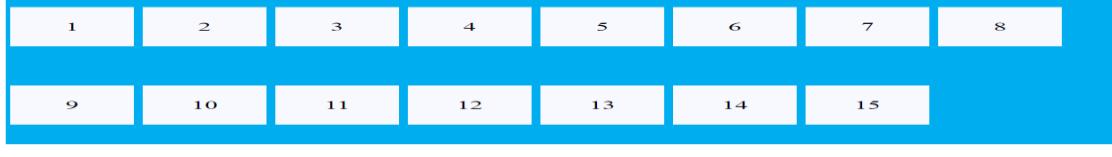


Property	Result
<code>flex-direction: column;</code>	<b>Flex Direction from top to bottom</b> 
<code>flex-direction: column-reverse;</code>	<b>Flex Direction from Bottom to Top</b> 
<code>flex-direction: row;</code>	<b>Flex Direction from left to right</b> 
<code>flex-direction: row-reverse;</code>	Flex Direction from right to left 

Property	Result
	<b>Flex Wrap</b>
<code>flex-wrap: wrap;</code>	
	<b>Flex No-Wrap</b>
<code>flex-wrap: nowrap;</code>	
	<b>Flex Wrap-reverse</b>
<code>flex-wrap: wrap-reverse;</code>	

Property	Result
<code>justify-content: center;</code>	<b>Justify Center</b> 
<code>justify-content: flex-start;</code>	<b>Justify Content Starts</b> 
<code>justify-content: flex-end;</code>	<b>Justify Content End</b> 
<code>justify-content: space-around;</code>	<b>Justify Content around</b> 
<code>justify-content: space-between;</code>	<b>Justify Content Between</b> 

Property	Result
<code>align-items: center;</code>	<b>Align items center</b> 
<code>align-items: flex-start;</code>	<b>Align items Starts</b> 
<code>align-items: flex-end;</code>	<b>Align items End</b> 
<code>align-items: stretch;</code>	<b>Align items stretch</b> 
<code>align-items: baseline;</code>	<b>Align items baseline</b> 

Property	Result
<code>align-content: center;</code>	<b>Align Content Center</b> 
<code>align-content: flex-start;</code>	<b>Align Content Starts</b> 
<code>align-content: flex-end;</code>	<b>Align Content Ends</b> 
<code>align-content: space-between;</code>	<b>Align Content Space between</b> 
<code>align-content: space-around;</code>	<b>Align Content Space-around</b> 
<code>align-content: stretch;</code>	<b>Align Content Stretch</b> 

Property	Result
<b>Align Items Self</b>	

## Exactly Center

Property	Result
justify-content: center; align-items: center;	<b>Align Items Exactly Center</b>

# Media Queries

- Media queries allow you to customize the presentation of your web pages for a specific range of devices like mobile phones, tablets, desktops, etc. without any change in markups.
- A media query consists of a media type and zero or more expressions that match the type and conditions of a particular media features such as device width or screen resolution.
- Since media query is a logical expression it can be resolved to either true or false. If it is true execute media query.
- When a media query is true, the related style sheet or style rules are applied to the target device.

### Html File

```

<body>
    <h1>CSS Media Queries</h1>
    <p>The background of the output area
        is different in different media or devices.</p>
    <p><strong>Alternative:</strong>
        You can also see the effect of this media query
        by opening the output in a new window and resize
        it to different sizes.</p>
</body>

```

### CSS File

```
<style type="text/css">
    /* Smartphones (portrait and landscape) ----- */
    @media screen and (min-width: 320px) and (max-width: 480px){
        body{
            background: #7ce7e1;
        }
    }
    /* Smartphones (portrait) ----- */
    @media screen and (max-width: 320px){
        body{
            background: #ffd280;
        }
    }
    /* Smartphones (landscape) ----- */
    @media screen and (min-width: 321px){
        body{
            background: #9ddfb9;
        }
    }
    /* tablets, iPads (portrait and landscape) ----- */
    @media screen and (min-width: 768px) and (max-width: 1024px){
        body{
            background: #ffb497;
        }
    }
    /* tablets, iPads (portrait) ----- */
    @media screen and (min-width: 768px){
        body{
            background: #f0e68c;
        }
    }
    /* tablets, iPads (landscape) ----- */
    @media screen and (min-width: 1024px){
        body{
            background: #d6b3f4;
        }
    }
    /* Desktops and laptops ----- */
    @media screen and (min-width: 1224px){
        body{
            background: #d8ff9d;
        }
    }
    /* Large screens ----- */
    @media screen and (min-width: 1824px){
        body{
            background: #ffc0cb;
        }
    }
</style>
```

# Responsive Image gallery

---

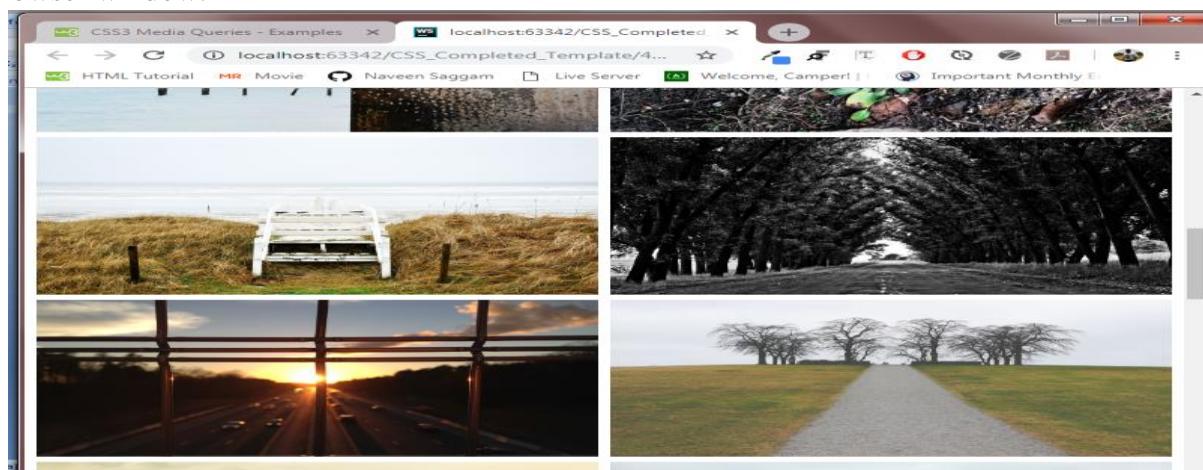
## Html File

```
<body>
    <!-- Header -->
    <div class="header">
        <h1>Responsive Image Grid</h1>
        <p>Resize the browser window to see the responsive effect.</p>
    </div>
    <!-- Photo Grid -->
    <div class="row">
        <div class="column">
            
            
            
            
            
            
        </div>
        <div class="column">
            
            
            
            
            
            
        </div>
        <div class="column">
            
            
            
            
            
            
        </div>
        <div class="column">
            
            
            
            
            
            
        </div>
    </div>
</body>
```

### CSS File

```
<style>
  * {
    box-sizing: border-box;
  }
  body {
    margin: 0;
    font-family: Arial;
  }
  .header {
    text-align: center;
    padding: 32px;
  }
  .row {
    display: flex;
    flex-wrap: wrap;
    padding: 0 4px;
  }
  /* Create four equal columns that sits next to each other */
  .column {
    flex: 25%;
    max-width: 25%;
    padding: 0 4px;
  }
  .column img {
    margin-top: 8px;
    vertical-align: middle;
  }
  /* Responsive layout - makes a two column-layout instead of four columns */
  @media screen and (max-width: 800px) {
    .column {
      flex: 50%;
      max-width: 50%;
    }
  }
  /* Responsive layout - makes the two columns stack on top of
   each other instead of next to each other */
  @media screen and (max-width: 600px) {
    .column {
      flex: 100%;
      max-width: 100%;
    }
  }
</style>
```

**Result:** Output is different from one width of the browser window to another width of the browser window.



## What is The Viewport?

- The viewport is the user's visible area of a web page.
- The viewport varies from one device width to another, and will be smaller on a mobile phone than on a computer screen.
- Before tablets and mobile phones, web pages were designed only for computer screens, and it was common for web pages to have a static design and a fixed size.
- When we started surfing the internet using tablets and mobile phones, fixed size web pages were too large to fit the viewport. To fix this, browsers on those devices scaled down the entire web page to fit the screen.

### Setting the Viewport

```
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

## What is a Grid-View?

- Many web pages are based on a grid-view, which means that the page is divided into columns.
- Using a grid-view is very helpful when designing web pages. It makes it easier to place elements on the page.
- A responsive grid-view often has 12 columns, and has a total width of 100%, and will shrink and expand as you resize the browser window.



## List out CSS3 modules?

Below is list of most important CSS3 modules are

- o Selectors
- o Box Model
- o Backgrounds and Borders
- o Text Effects
- o 2D/3D Transformations
- o Animations
- o Multiple Column Layout
- o User Interface

## Explain how flexibility is achieved more in CSS3?

Flexibility achieved is in the greater ratio in CSS3 because of the feature of handling multiple style sheets in CSS3 and because of the modularized approach of CSS3.

## Explain what are the values that can be taken by property white-space of CSS3?

The five values that can be taken by property white-space of CSS3 are

- Normal
- pre
- nowrap
- pre-wrap
- pre-line

## What is the word wrap/word wrapping in CSS3?

**Word wrap/word wrapping** is used to allow long words to be able to break and wrap onto the next line in css3 we used word-wrap property like below class.

```
.wrapWord{ word-wrap : break-word; }
```

## List some advantages to CSS3 animations over script-based animation?

Advantages of using CSS3 animations over script-based animation techniques are as follows:

1. Easy to use and anybody can create them without the knowledge of JavaScript.
2. Executes well even under reasonable system load. As simple animations perform poorly in JavaScript, the rendering engine uses the frame-skipping techniques to allow smooth flow of animation.
3. Allows the browser to control the animation sequence, optimize performance and efficiency by reducing the update frequency of animations executing in tabs that aren't currently visible.

## What is the purpose of the z-index and how is it used?

The z-index helps specify the stack order of positioned elements that may overlap one another. The z-index default value is zero, and can take on either a positive or negative number.

An element with a higher z-index is always stacked above than a lower index.  
Z-Index can take the following values:

- **Auto:** Sets the stack order equal to its parents.
- **Number:** Orders the stack order.
- **Initial:** Sets this property to its default value (0).
- **Inherit:** Inherits this property from its parent element.

### **Enlist the media types CSS allows?**

Media is one of the most important features of CSS. Media renders the design and customization of documents. By applying media control over the external style sheets, they can be retrieved and used by loading it from the network. Some of the media types are:

- All – for all devices
- Aural – speech synthesizers
- Braille – feedback devices
- Embossed – paged Braille printers
- Handheld – typically small screen, limitation of bandwidth
- Projection – basically for projectors
- TV – television type devices
- Screen – color computer screens

### **Differentiate logical tags from physical tags?**

There are several pointers that make local tags different from physical tags:

Physical Tags	Local Tags
Physical tags are used to indicate how a particular character is to be formatted	logical tags are used to indicate by the visually impaired and put emphasis on the text.
Physical tags are also referred to as presentational markup	Logical tags are useless for appearances
Physical tags are newer versions	Logical tags are old and concentrate on content

### **What are the different versions of CSS?**

Different versions of CSS are:

- CSS 1
- CSS 2
- CSS 2.1
- CSS 3
- CSS 4

### **What are the limitations of CSS?**

There are several limitations of CSS such as:

- CSS can't fulfil turning completeness hence, it can never perform logical like 'if/else', for/while, etc, or arithmetical tasks.
- It cannot provide total control over document display and allows the contents of the page to come through whatever the browser is used.
- Ascending by selectors is not possible.
- Limitations of vertical control.
- No expression as it is a text-based coding language.

- Rules, styles, targeting specific text not possible.

### Describe 'rule set'?

- It is an instruction that tells browser on how to render a specific element on the Html page.
- It consists of a selector with a declaration block that follows. Rule set: Selectors can be attached to other selectors to be identified by rule set.

It has two parts:

- Selector, e.g. R and
- Declaration block {text-indent: 11pt}

### What are the advantages of CSS?

Advantages are:

- Bandwidth
- Site-wide consistency
- Page reformatting
- Accessibility
- Content separated from presentation
- CSS Selectors
- In CSS, selectors are patterns used to select the element(s) you want to style. Each declaration includes a **CSS** property name and a value, separated by a colon.

### Differentiate Style Sheet concept from Html?

While Html provides easy structure method, it lacks styling, unlike Style sheets. Moreover, style sheets have better browser capabilities and formatting options.

### Can default property value be restored through CSS? If yes, how?

In CSS, you cannot revert back to old values due to lack of default values. The property can be re-declared to get the default property.

### What is contextual selector?

Selector used to select special occurrences of an element is called contextual selector. A space separates the individual selectors. Only the last element of the pattern is addressed in this kind of selector. For e.g.: TD P TEXT {color: blue}

### Compare Grouping and Nesting in CSS?

**Grouping:** Selectors can be grouped having the same values of property and the code be reduced. Rewriting can be avoided by writing each selector separated by a comma.

```

<style>
  h1 {
    color: blue;
  }
  h2 {
    color: blue;
  }
  p {
    color: blue;
  }
</style>

```

```

<style>
  h1, h2, p {
    color: blue;
  }
</style>

```

**Nesting:** If the CSS is structured well, there shouldn't be a need to use many class or ID selectors. This is because you can specify properties to selectors **within** other selectors are called Nesting.



## How can the dimension be defined of an element?

Dimension properties can be defined by:

- Height
- Max-height
- Max-width
- Min-height
- Min-width
- Width

## What is graceful degradation?

- In case the component fails, it will continue to work properly in the presence of a graceful degradation.
- The latest browser application is used when a webpage is designed, As it is not available to everyone, there is a basic functionality, which enables its use to a wider audience. In case the image is unavailable for viewing, text is shown with the alt tag.

## What is progressive enhancement?

- It's an alternative to graceful degradation, which concentrates on the matter of the web. The functionality is same, but it provides an extra edge to users having the latest bandwidth. It has been into prominent use recently with mobile internet connections expanding their base.

## How can backward compatibility be designed in CSS?

- Html sheet methods is collaborated with CSS and used accordingly.