“CSS BASICS”

SUBMITTED BY:

MANASWI M. PATIL

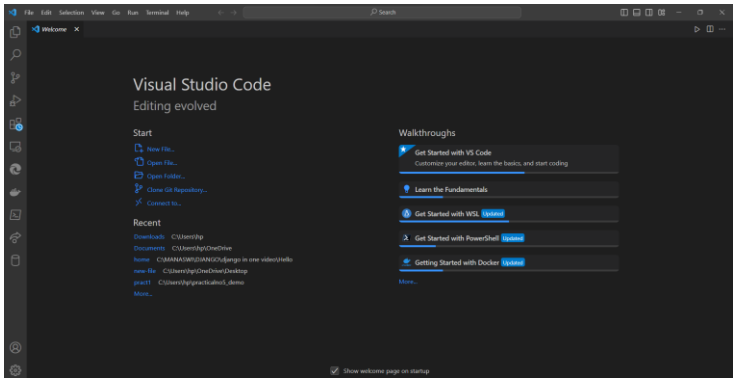
INDEX

1. INTRODUCTION

CSS (Cascading Style Sheets) is like a set of instructions for web browsers. It tells them how to make a website look nice and organized. Using CSS, you can make sure your website looks good on different devices, like phones or computers. It helps web developers keep things organized and makes it easier to change the look of a whole website at once. It means CSS is the design tool for websites, making them visually appealing and user-friendly.

I. Steps for Editor:

Go on following link→ https://code.visualstudio.com/download →click→download this editor→install.



II. Why CSS used?

CSS is used in programming for presentation and layout of web pages. It is a style sheet language that defines how HTML elements should be displayed on the screen, in print, or in other media.

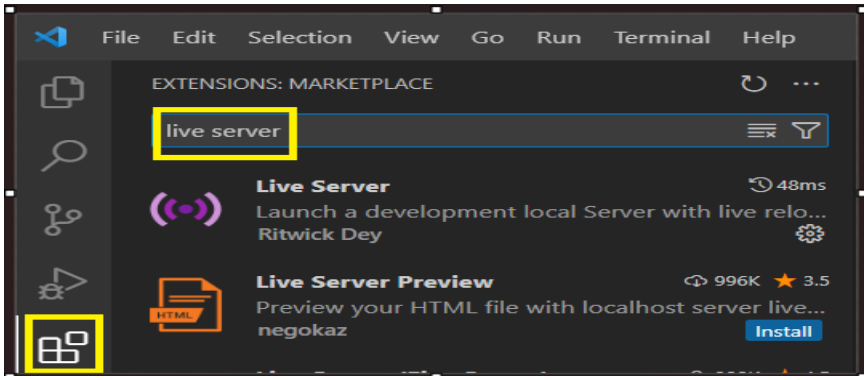
- Make Things Look Nice: CSS helps in styling web pages, making them visually appealing to users.
- Keep Things Organized: It allows programmers to keep the structure of a webpage (HTML) separate from its appearance (CSS), making it easier to manage.
- Consistent Look: You can use CSS to make sure all pages on a website have a similar style, creating a cohesive look and feel.
- Work on Different Devices: CSS helps in creating websites that look good on computers, tablets, and phones—adjusting to different screen sizes.

- Easy Updates: Changes to the design can be made quickly in CSS without altering the underlying structure of the webpage.
- Reuse Styles: CSS allows for the creation of styles that can be used across different parts of a website, avoiding repetition.
- Print-Friendly: It helps in creating versions of web pages that are easy to print, with optimized layouts for paper.
- Add Cool Effects: CSS supports animations and transitions, allowing for interactive and visually interesting effects on a webpage.
- Accessibility: CSS features can be used to improve the accessibility of web content, making it more user-friendly for people with disabilities.

III. Install Live-Server

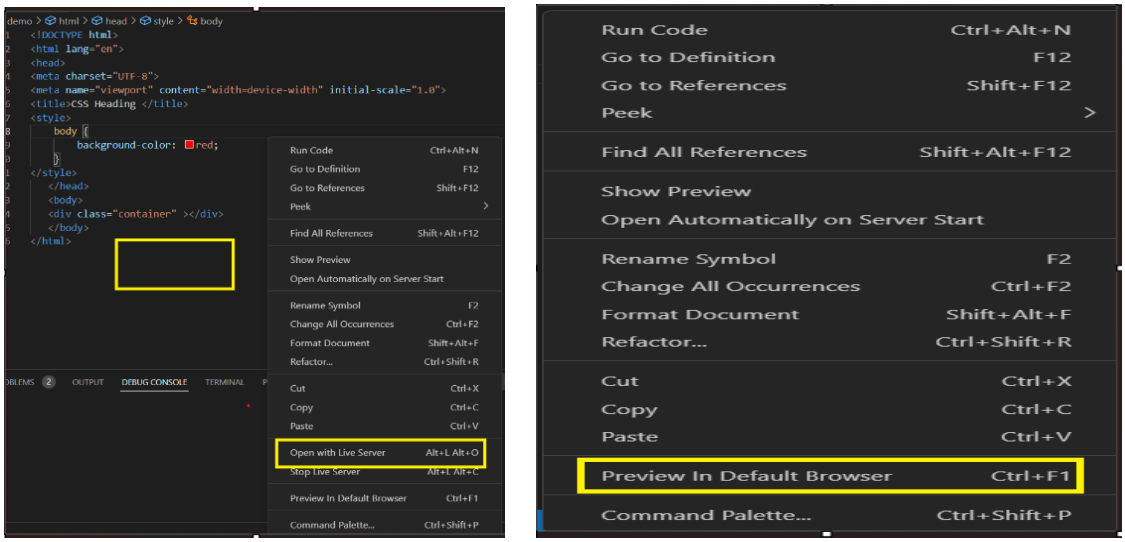
Using this extension in vs code it will reflect your code in browser, suppose you are doing something in the editor window then it will automatically reflect in browser and get desired output .

➢ Go on extension symbol→click→search for live server→install.

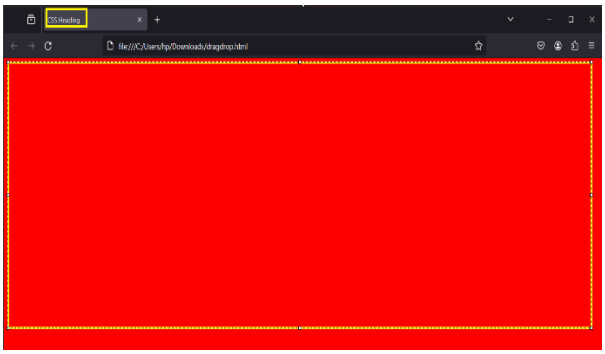


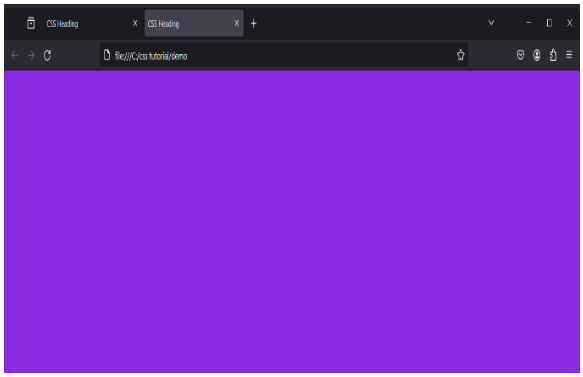
IV. How to use CSS in programming?

Type code in your vs code editor →right click on code window→select option 'open with live server→select browser→preview in default browser→select browser(must select 'firefox' browser for CSS)→STEPS TO INSTALL FIREFOX→click link → https://www.mozilla.org/en-US/firefox/new/ →download.



After adding ‘CSS’ in programming, page will look like this. You can add CSS by adding ‘style tag ‘in program .

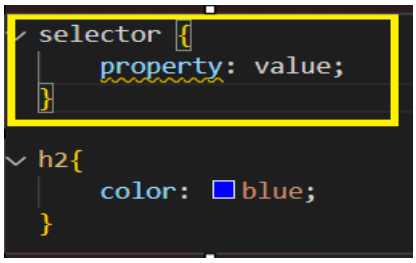




V. Syntax of CSS:

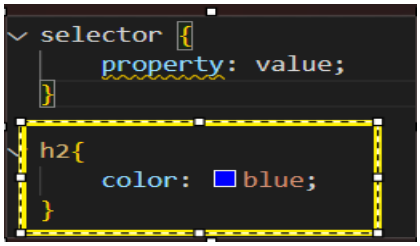
CSS follows a rule-based structure. Each rule consists of a selector and a declaration block. Selectors pick the HTML elements, while declaration blocks contain pairs of properties and values.

Selector: The selector is like a target. It picks the HTML element(s) that you want to style.For example, body is a selector that targets the entire body of the HTML document, and h1 targets all <h1> heading elements.



Declaration Block: The declaration block is a set of instructions enclosed in curly braces {}.Inside the declaration block, you have pairs of properties and values. These define how the selected elements should look.

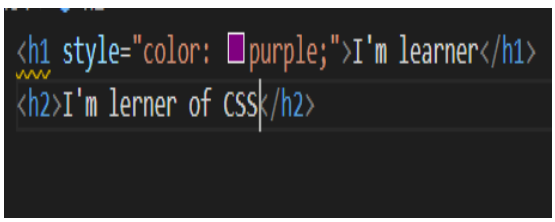
From this below example 'h2' is the **selector** , 'color' is the **property** and 'blue color' is the **value** and the curly braces '{}' is the **declaration block**.

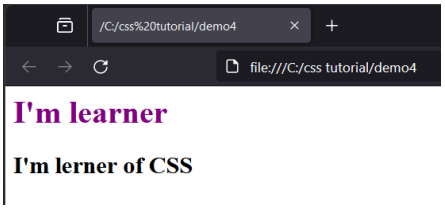


VI. Ways to add CSS in html:

There are three ways to add CSS in html

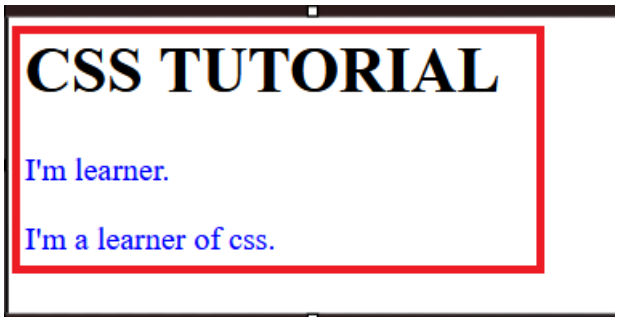
a) <Style> tag→ Adding style <style>…….</style > to html. The <style> tag is placed within the <head> section of an HTML document and contains CSS code that specifies how the HTML elements should be styled.
b) Inline CSS→Adding CSS using style attribute.





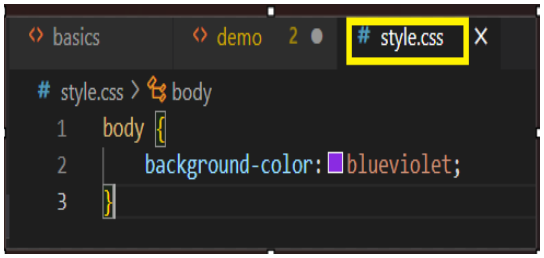
c) Internal CSS→It is used to apply custom style to the multiple elements on the same page. Internal CSS is defined in a style block, which will be inside the head section.

```
<!DOCTYPE html>
<head>
    <style>
        p {
            color: blue;
        }
    </style>
</head>

<body>
    <h1>CSS TUTORIAL</h1>
    <p>I'm learner.</p>
    <p>I'm a learner of css.</p>
</body>

</html>
```



d) External CSS→ External CSS is used in HTML to separate the styling from the structure (content) of a web page. Instead of adding the styles within the HTML file, we create a separate file with **'.css extension'**. This file will hold all the styling details. Then, we **link** this file to the HTML page. link tag has rel and href properties

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width" initial-scale="1.0">
<title>CSS Heading </title>
<link rel="stylesheet" href="style.css"
```



VII. Types of Selector in CSS:

Suppose you want to add a custom style to only a specific tag. There, We can make use of CSS selector. CSS selectors allow us to choose specific elements and apply styles to them.

i. **Universal Selector** → Universal selector represented by "*" targets all the HTML elements on the page.

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

```html
<html>

<head>
    <style>
        * {
            color: purple;
            text-align: center;
        }
    </style>
</head>

<body>
    <p>Welcome to </p>
    <h1>CSS TUTORIAL</h1>
</body>

</html>
```

Welcome to

CSS TUTORIAL

ii. **Element Selector→**The element selector selects the target element based on the specific type. If you want to underline all the <p> tags then in this case, the **element selector** will be the best choice.

```
p {
    property : value;
}
```

```html
<html>

<head>
    <title>CSS</title>
    <style>
        p{
            text-decoration: underline;
        }
    </style>
</head>

<body>
    <h1>CSS TUTORIAL</h1>
    <h2>CSS Topics:</h2>
    <p>css properties</p>
    <p>csss attributes</p>

</body>

</html>
```

CSS TUTORIAL

CSS Topics:

css properties

csss attributes

iii. **Id Selector→** It is written with the hash "#" character followed by the ID name in the style sheet.

```html
<html>

<head>
    <style>
        #title {
            text-align: left;
            color: red;
        }
    </style>
</head>

<body>
    <h1 id="title">CSS TUTORIALS</h1>
    <p>I'm a learner</p>
</body>

</html>
```

CSS TUTORIALS

I'm a learner

iv. **Class Selector→**If you want to give a custom style to a specific group of elements. In this case, the class selector is the best option to use. The class selector does the same job as the id selector, a class selector helps group various types of elements.

```
.class {
    property : value;
}
```

```
<html>

<head>
    <title>CSS</title>
    <style>
        .red {
            color: red;
        }
    </style>
</head>

<body>
    <p>This is simple p tag</p>
    <p class="red">This p tag has class red</p>
    <p>This is simple p tag</p>
    <p class="red">This p tag has class red</p>
</body>

</html>
```

This is simple p tag

This p tag has class red

This is simple p tag

This p tag has class red

v. **Group Selector→** Commas **","** are used to separate each selector in a grouping. This reduces the number of lines of code. The group selector is used to minimise the code.

```
div, p, a {
    property : value;
}
```

```
<html>

<head>
    <title>CSS</title>
    <style>
        h1 {
            color: green;
        }

        p,a {
            color: purple;
        }
    </style>
</head>

<body>
    <h1>CSS TUTORIAL</h1>
    <p>This is the p tag</p>
    <a href="#">This is the anchor (a) tag</a>
</body>

</html>
```

CSS TUTORIAL

This is the p tag

This is the anchor (a) tag

VIII. CSS comments:

CSS comments are used to write brief description of line. It will help to undertand the concept behind line. You can used single line comment or multiline comment in during task. **Single-line** comments are contained within one line. They are useful for short annotations. **Multi-line** comments span across multiple lines, making them ideal for detailed explanations or temporarily disabling blocks of code. **CSS Comments are enclosed between /* and */.**

2. CSS PROPERTIES

I. **CSS Colors:** Colors play a crucial role in web design, and CSS provides several ways to specify colors. In CSS use 'color, not colour.'

a. **Hexadecimal (hex) values** represent colors using a six-digit code composed of numbers and letters. Each pair of digits represents the intensity of the red, green, and blue components of the color.

```
<head>
    <style>
        h1 {
            color: #FF0000;
            /*Pure Red*/
        }
        h2 {
            color: #00FF00;
            /* Pure Green */
        }
        h3 {
            color: #0000FF;
            /* Pure Blue */
        }
        h4 {
            color: #b700ff;
            /* Custom Color */
        }
    </style>
</head>
<body>
    <h1>CSS TUTORIAL</h1>
    <h2>CSS learner</h2>
    <h3>css basics includes</h3>
    <h4>apply css to page</h4>
```

CSS TUTORIAL

CSS learner

css basics includes

apply css to page

b. **RGB** stands for Red, Green, Blue. This color model represents colors using three values ranging from 0 to 255 for each of the three color components.

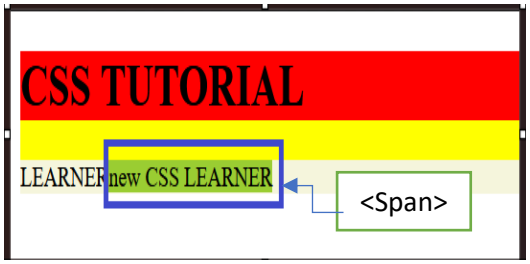
c. **RGBA** is an extension of RGB that includes an additional parameter for opacity (alpha), ranging from 0 to 1. A value of 0 means completely transparent, and 1 means fully opaque.

d. **HSL** stands for Hue, Saturation, Lightness. This color model is based on the way humans perceive colors. Hue represents the type of color, saturation controls the vividness, and lightness controls the brightness.

e. **HSLA** includes an additional parameter for alpha (opacity).

II. **CSS Backgrounds:** The CSS background property helps set the background style, property, and effects of the element. The background color property sets the background colour of HTML tags such as div, section, p, etc.** tag** in HTML is an inline container used to **group and apply styles to a specific portion** of text or inline content within a larger block of content.

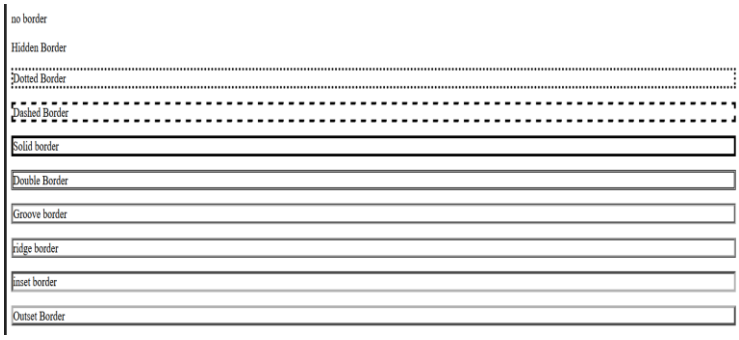
```html
<html>
<head>
    <style>
        div{
            background-color: yellow;
        }
        h1{
            background-color: #FF0000;
        }
        p{
            background-color: beige;
        }
        span{
            background-color: yellowgreen;
        }
    </style>
</head>
<body>
    <div>
        <h1>CSS TUTORIAL</h1>
        <p>LEARNER<span>new CSS LEARNER</span></p>
    </div>
</body>
</html>
```



The Background Image property sets an image as a background by providing the image URL within the url() function.

III. **CSS Borders** help define the visual boundaries of HTML elements. **Border styles** define the style of the border.

```html
<html lang="en">
<head>
    <style>
        .none {
            border-style: none;
        }
        .hidden {
            border-style: hidden;
        }
        .dotted {
            border-style: dotted;
        }
        .dashed {
            border-style: dashed;
        }
        .solid {
            border-style: solid;
        }
        .double {
            border-style: double;
        }
        .groove {
            border-style: groove;
        }
        .ridge {
            border-style: ridge;
        }
        .inset {
            border-style: inset;
        }
        .outset {
            border-style: outset;
        }
    </style>
</head>
<body>
    <p class="none">no border</p>
```



Border color property sets the colour of the border. We can use colourname, hex, rgb, or hsl to set the color.

```html
<html lang="en">
<head>
    <style>
        .dotted {
            border-style: dotted;
            color: red;
        }
        .dashed {
            border-style: dashed;
            border-color: #FF0000;
        }
        .solid {
            border-style: solid;
            border-color: rgb(100, 233, 12);
        }
        .double {
            border-style: double;
            border-color: hsl(10, 50, 30);
        }
    </style>
</head>
<body>
    <p class="dotted">Dotted Border</p>
    <p class="dashed">Dashed Border</p>
    <p class="solid">Solid border</p>
    <p class="double">Double Border</p>
</body>
</html>
```

Border-collapse helps to control how table borders interact with each other.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Collapsing Borders Example</title>
  <style>
    table {
      border-collapse: collapse;
      width: 100%;
    }

    th, td {
      border: 1px solid black;
      padding: 8px;
      text-align: left;
    }
  </style>
</head>
<body>
  <table>
    <tr>
      <th>HEADING1</th>
      <th>HEADING2</th>
    </tr>
    <tr>
      <td>Row 1</td>
      <td>Row 1</td>
    </tr>
    <tr>
      <td>Row 2</td>
      <td>Row 2 </td>
    </tr>
  </table>
</body>
</html>
```

HEADING1	HEADING2
Row 1	Row 1
Row 2	Row 2

HEADING1	HEADING2
Row 1	Row 1
Row 2	Row 2

IV. CSS Images having two attributes.

i. src: The path of the image.

ii. alt: Alternate text for the image, if the image is not displayed due to any issue.

Image opacity controls the transparency of the image. The property value ranges from 0 to 1.

V. CSS Video Embedding

```
<body>
    <video controls>
        <source src="./Sigma Web Development Course.mp4" type="video/mp4"> Your browser does not support the video tag.
    </video>
</body>
```

Add 'style.css' file in your window to run video.

```
# style.css > video
1   video {
2       width: 500px;
3       height: 200px;
4   }
```

VI. CSS Fonts decide how texts will look on the screen; depending on the website, different kinds of fonts are used.

Font color defines the colour of the text or typography.

Font size property sets the size of the fonts.

px: px is the absolute unit. This is useful for setting precise sizes.

em: em is the relative unit, based on the font size of the parent element. If the element's font size is 1.5 em, that means the element will be 1.5 times the size of the parent.

rem: rem is the relative unit, based on the font size of the root element, i.e., <html>.

```
<html lang="en">
<head>
    <style>
        #p1 {
            font-size: small;
        }
        #p1{
          font-style: italic;
        }
        #p2 {
            font-size: medium;
        }
        #p2 {
          font-style: normal;
        }
        #p3 {
            font-size: large;
        }
        #p3 {
          font-style: oblique;
        }
        #p4 {
            font-size: larger;
        }
        #p5 {
            font-size: 25px;
        }
        #p6 {
            font-size: 2em;
        }
        #p7 {
            font-size: 2rem;
        }
        #p8 {
            font-size: 50%;
        }
    </style>
```

font-size: small font-style: italic

font-size: medium font-style:normal

font-size: largr font-style:oblique

font-size: larger

font-size: 25px

font-size: 2em

font-size: 2rem

font-size: 50%

VII. CSS Text Styling

i. overline: adds a line above the text

ii. underline: adds a line below the text

iii. line-through: Adds a line to the text.

iv. none: To remove decoration.

There are four values for text alignment:

a. left: align the text to the left.

b. centre: align the text to the centre.

c. right: align the text to the right.

d. justify: spread the text evenly between the left and right margins.

There are four values for text-transform:

a. uppercase: Transform text to uppercase (all capital letters).

b. lowercase: transform text to lowercase (all small letters).

c. capitalise: capitalise the first character from each word.

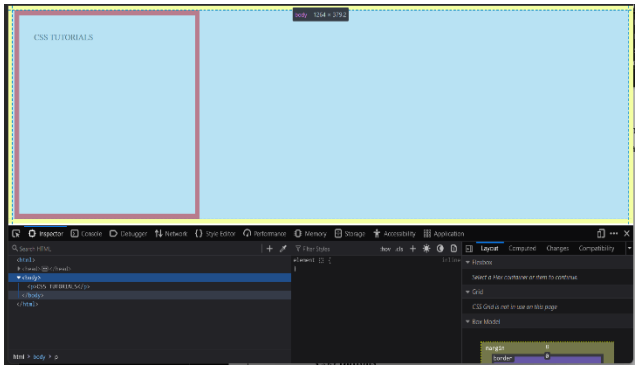
d. none: To remove text transformation.

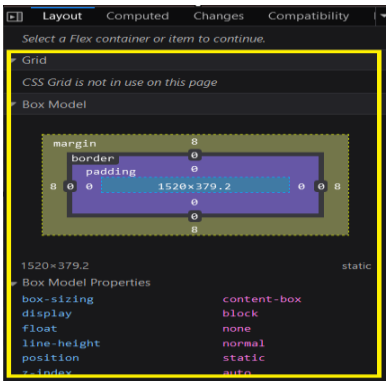
The line-height property controls the spacing between the two lines of text.

The **text-shadow** property adds a shadow to the text.

VIII. **CSS Box Model** is a fundamental concept in web design and layout. It describes the structure of HTML elements as rectangular boxes with content, padding, borders, and margins. Right click on output→inspect→box model.

```
<head>
    <style>
        p{
            width: 300px;
            height: 300px;
            padding: 30px;
            border: 10px solid red;
            margin: 5px;
        }
    </style>
</head>
<body>
    <p>CSS TUTORIALSy</p>
</body>
</html>
```





IX. **CSS Padding** refers to the **space between the content of an element** and its border. It is one of the components of the CSS Box Model.

X. **CSS Margin** is the **space outside an element's border, providing external spacing between the element** and other elements in the layout.

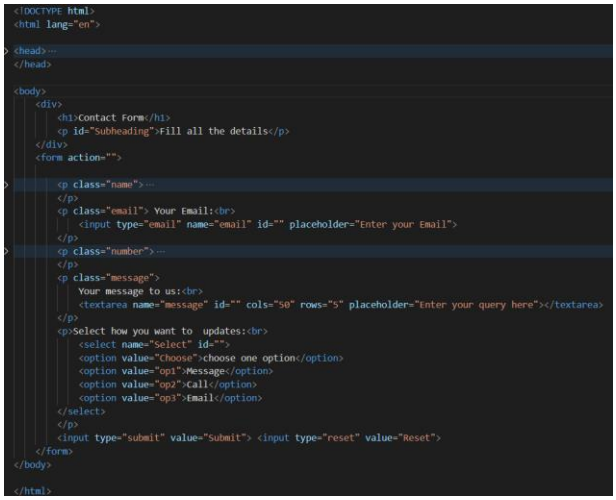
XI. **CSS Hover** ':hover' pseudo-class is used to select and style an element when the user hovers over it with their mouse

XII. **CSS Combinators** the relation between multiple or single selector.

XIII. **CSS Z-index** property is used to control the stacking order of positioned elements along the z-axis. The z-axis is perpendicular to both the x-axis and y-axis and determines the depth or layering of elements on the web page.

XIV. CSS Forms

Before adding CSS

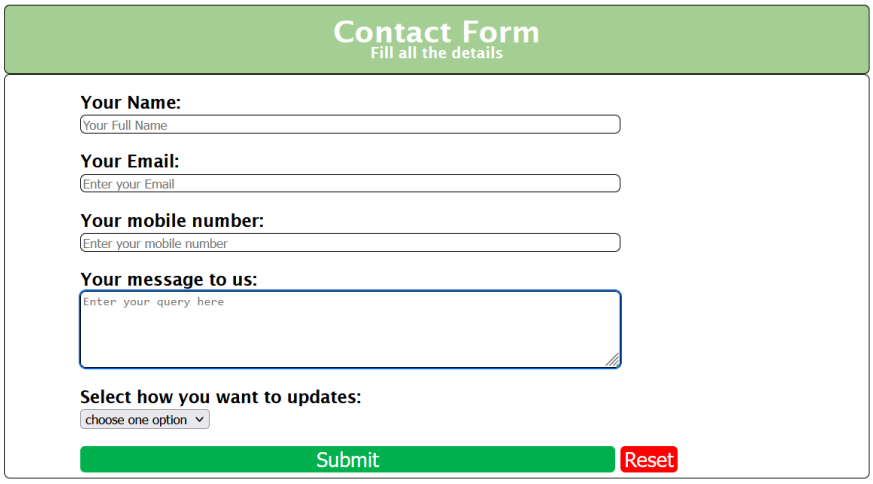


```
<!DOCTYPE html>
<html lang="en">

<head>...
</head>

<body>
    <div>
        <h1>Contact Form</h1>
        <p id="Subheading">Fill all the details</p>
    </div>
    <form action="">
        <p class="name">...
        </p>
        <p class="email"> Your Email:<br>
            <input type="email" name="email" id="" placeholder="Enter your Email">
        </p>
        <p class="number">...
        </p>
        <p class="message">
            Your message to us:<br>
            <textarea name="message" id="" cols="50" rows="5" placeholder="Enter your query here"></textarea>
        </p>
        <p>Select how you want to  updates:<br>
            <select name="Select" id="">
            <option value="Choose">choose one option</option>
            <option value="op1">Message</option>
            <option value="op2">Call</option>
            <option value="op3">Email</option>
        </select>
        </p>
        <input type="submit" value="Submit"> <input type="reset" value="Reset">
    </form>
</body>

</html>
```

After adding styles form will look like this.

```
<style>
    body {
      padding: 0px 300px;
      font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande',
'Lucida Sans Unicode', Geneva, Verdana, sans-serif;
      text-decoration: solid;
      font-weight: 900;
  }

  div {
      padding: 5px 5px;
      background-color: rgba(164, 206, 147);
      width: auto;
      color: white;
      line-height: 10%;
      text-align: center;
      border-radius: 5px;
      border: 1px solid black;
  }

  form {
      background-color:Yellow (238, 239, 235);
      border-radius: 5px;
      border: 1px solid black;
      padding: 0px 80px 5px;
      font-size: large;
  }

  input[type="text"],
  input[type="email"],
  textarea {
      width: 75%;
```

```
    border: 1px solid black;
    border-radius: 5px;
    background-color: white;
}

input[type="submit"] {
    background-color:yellow (164, 206, 147);
    font-size: larger;
    color: white;
    border-radius: 5px;
    border: none;
    cursor: pointer;
    width: 75%;
}

input[type="Reset"] {
    background-color: red;
    font-size: larger;
    color: white;
    border-radius: 5px;
    cursor: pointer;
    border: none;
}

input[type="submit"]:hover {
    background-color: rgb(0, 176, 76);
}

input[type="reset"]:hover {
    background-color: rgb(191, 0, 0);
}

</style>
```



XV. CSS Navigation Bar

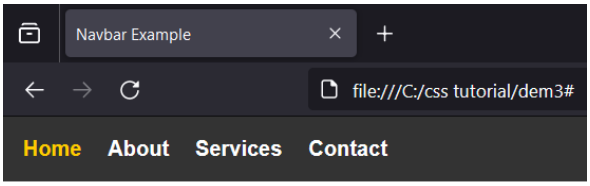
Before adding style tag in program.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="styles.css">
    <title>Navbar Example</title>
</head>
<body>
    <style>...
    </style>
    <nav class="navbar">
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">About</a></li>
            <li><a href="#">Services</a></li>
            <li><a href="#">Contact</a></li>
        </ul>
    </nav>
</body>
</html>
```

- Home
- About
- Services
- Contact

After adding style in programming.

```css
<style>
    body {
        margin: 0;
        font-family: 'Arial', sans-serif;
    }

    .navbar {
        background-color: #333;
        padding: 15px;
    }

    ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
        display: flex;
    }

    li {
        margin-right: 20px;
    }

    a {
        text-decoration: none;
        color: white;
        font-weight: bold;
        font-size: 16px;
        transition: color 0.3s;
    }

    a:hover {
        color: #ffcc00; /* Change color on hover */
    }
</style>
```



3. CSS DESIGNING

I. CSS Display

Display inline only takes the space required for content, leaving the rest space for other elements to come.

Syntax:

{display: inline;}

Display box takes the full width available across the website page leaving a new line before and after the element.

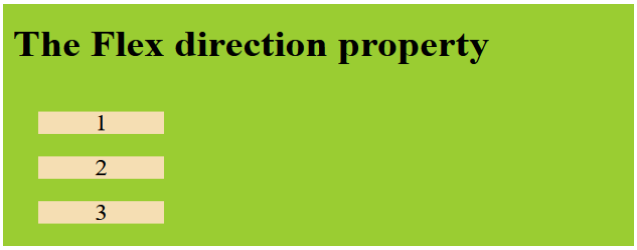
Syntax:

{display: block;}

II. CSS FlexBox

'Flexible Box Layout ' is a layout model in CSS that allows you to design complex layouts and distribute space among items within a container, even when their size is unknown or dynamic. Flexbox is particularly useful for creating responsive and flexible designs.

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            body{
                background-color: yellowgreen;
            }
        .flex-container {
        display: flex;
        flex-direction: column;
        background-color: yellowgreen;
    }
    div{
        background-color: wheat;
        width:100px;
        margin:20px;
        text-align:center;
        line-height: 20px;
        font-size: 20px;
    }
    </style>
    </head>
    <body>
        <h1>The Flex direction property</h1>
<span class ="flex-container"></span>
<div>1</div>
<div>2</div>
<div>3</div>
</body>
</html>
```



Float property is used to push an element to one side of its containing element, allowing other elements to wrap around it. Common values are left and right.

Clear property is used to control the behavior of an element concerning floated elements. It determines whether an element should be moved below (cleared) any floated elements that precede it.

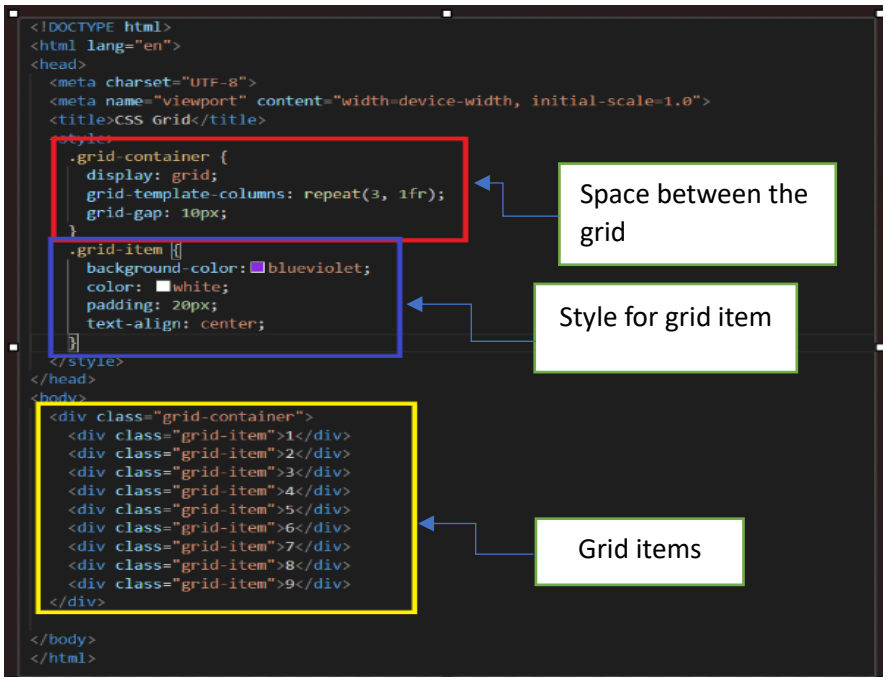
III. **CSS Grid** Layout is a powerful layout system that allows you to create complex two-dimensional layouts with rows and columns. It provides a way to structure content in a more efficient and flexible manner compared to traditional layout methods.

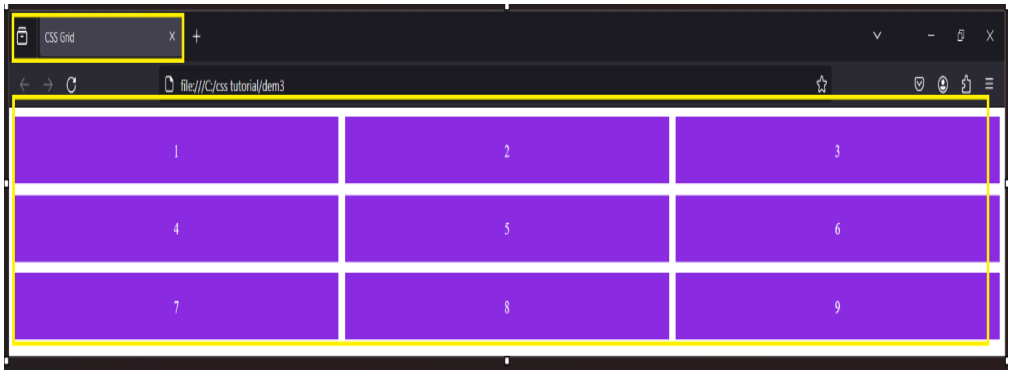
Syntax:

.grid-container {

display: grid;

}





IV. **CSS Media Queries** allow you to apply different styles for different screen sizes or devices.

4. CSS FUNCTION

CSS functions allow you to perform calculations, manipulate values, or apply effects within your stylesheets.

1. **calc():** The calc() function is used for performing calculations in CSS. It allows you to combine different units and mathematical operations.

2. **var():**The var() function is a way to use custom properties (CSS variables) within your stylesheets. It allows you to define a variable and reuse it throughout your styles.

3. **rgb(), rgba(), hsl(), hsla():**These functions are used for defining colors. rgb() and rgba() are for defining colors using the RGB color model, while hsl() and hsla() are for the HSL color model.

4**. url():**The url() function is used to specify a reference to a resource, such as an image or a font.

5. **linear-gradient():**The linear-gradient() function creates a linear gradient, transitioning between two or more colors.

6. **cubic-bezier():**The cubic-bezier() function is used in CSS transitions and animations to define a cubic Bezier curve for timing functions.

7. **rotate(), scale(), translate(), skew():**These functions are used with the transform property to apply transformations to elements.

5. CSS ADVANCED TOPICS

I. **CSS Transition** allow you to smoothly change property values over a specified duration. They are a powerful tool for adding subtle animations and effects to elements on a webpage.

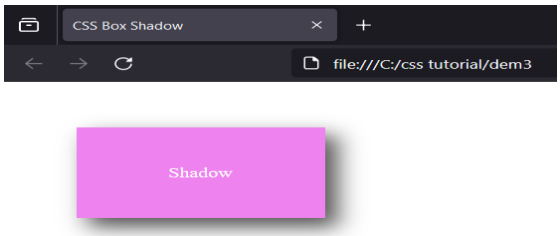
II. **CSS Border Images** allow you to use an image as a border for an element, giving you more flexibility and customization options than traditional border styles. The border-image property is used for this purpose.

III. **CSS Gradients** allow you to create smooth transitions between multiple colors. Gradients are often used as backgrounds or for styling elements where a smooth color transition is desired.

IV. **CSS Inherit** keyword in CSS is used to explicitly set a property to the computed value of its parent element. It allows an element to inherit a specific property from its parent, ensuring consistency in styling.

V. **CSS Shadows** are used to add shadow effects to elements, providing depth and dimensionality.

```
  <style>
    .box-with-shadow {
      width: 200px;
      height: 100px;
      background-color: violet;
      color: white;
      text-align: center;
      line-height: 100px;
      margin: 50px;
      box-shadow: 5px 5px 10px rgba(0, 0, 0, 0.5);
      transition: box-shadow 0.3s ease;
    }
    .box-with-shadow:hover {
      box-shadow: 10px 10px 20px rgba(0, 0, 0, 0.7);
    }
  </style>
</head>
<body>

  <div class="box-with-shadow">Shadow</div>
```

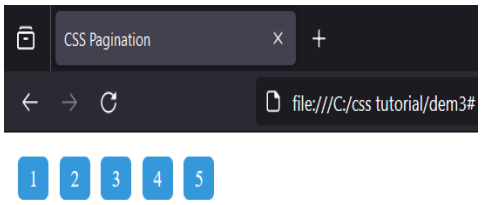


VI. **CSS ToolTip Text** is a technique to display additional information or a description when a user hovers over an element. It's often used to provide context or details about an element.

VII. **CSS Masking** is a technique to show or hide portions of an element based on a mask image or gradient. It is commonly used for creating interesting visual effects.

VIII. **CSS Pagination** is used to style and layout pagination elements, typically seen in sets of links for navigating through pages.

```
<style>
.pagination {
  display: flex;
  list-style: none;
  padding: 0;
}

.pagination li {
  margin: 0 5px;
}

.pagination li a {
  display: block;
  padding: 5px 10px;
  background-color: #3498db;
  color: white;
  text-decoration: none;
  border-radius: 5px;
}

.pagination li a:hover {
  background-color: #2980b9;
}
</style>
</head>
<body>

<ul class="pagination">
  <li><a href="#">1</a></li>
  <li><a href="#">2</a></li>
  <li><a href="#">3</a></li>
  <li><a href="#">4</a></li>
  <li><a href="#">5</a></li>
```



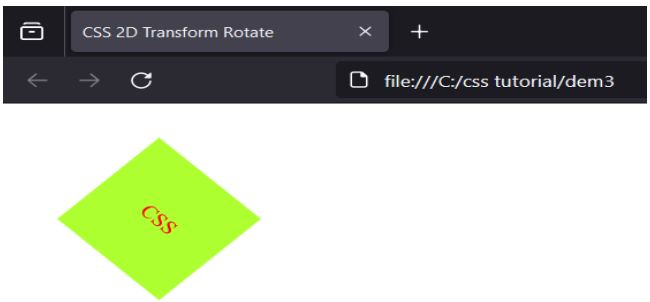
IX. **CSS Media Queries Advanced** involve creating responsive designs for various devices and screen sizes. Media queries can be nested, allowing for more complex conditions.

X. **CSS Animations** allow you to create dynamic and visually appealing effects on web elements. They involve transitioning between different states smoothly.

XI. **CSS 2D transform** allow you to apply various transformations to HTML elements, including rotation**.**

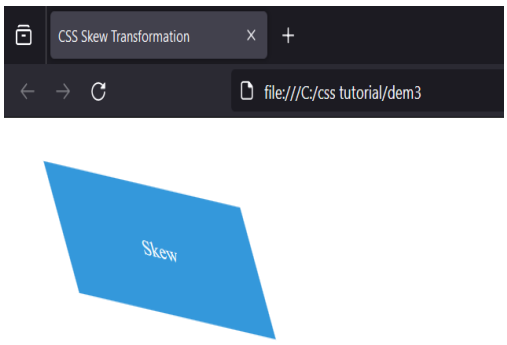
Rotate() function is used for rotating an element.

```
<style>
  .rotated-box {
    width: 100px;
    height: 100px;
    background-color: greenyellow;
    color: red;
    text-align: center;
    line-height: 100px;
    margin: 50px;
    transition: transform 0.5s ease;
  }
  .rotated-box:hover {
    transform: rotate(45deg);
  }
 </style>
/head>
body>

 <div class="rotated-box">CSS</div>
```



Skew() function in CSS is used to apply a skew transformation to an element. It skews an element by a given angle along both the X and Y axes.

```
  <style>
    .skewed-box {
      width: 200px;
      height: 100px;
      background-color: #3498db;
      color: white;
      text-align: center;
      line-height: 100px;
      margin: 50px;
      transition: transform 0.5s ease;
    }

    .skewed-box:hover {
      transform: skew(20deg, 10deg);
    }
  </style>
</head>
<body>

  <div class="skewed-box">Skew</div>
```



6. CONCLUSION

It's the language we use to make websites look good and work well. Imagine a webpage without CSS ,it would be plain and boring, lacking colors, styles, and structure. CSS allows us to give each element on a webpage a unique look, set the layout, and make it responsive to different devices. It's like the magic that turns a simple webpage into a visually appealing and user-friendly experience. With CSS, we can create consistency across pages, easily update styles, and even add cool animations. It's a crucial tool for web designers and developers, helping them make the internet a more beautiful and functional place for everyone.