

# Batch: Roll No.: Experiment No.: 3

**Aim**: To apply basic CSS and flexible content layout using CSS3 to your website..



**Resources needed:** need to fill by students



# Theory:

**Cascading Style Sheets** (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language.HTML was NEVER intended to contain tags for formatting a web page.HTML was created to **describe the content** of a web page. CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. The file should not contain any html tags. The style sheet file must be saved with a

.css extension.

CSS has various levels and profiles like CSS1, CSS2, CSS3, CSS4. There are three ways of inserting a style sheet:

* External style sheet
* Internal style sheet
* Inline style

# External Style Sheet

Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section of html page. An external style sheet is ideal when the style is applied to many pages.

# Internal Style Sheet

An internal style sheet may be used if one single page has a unique style.Internal styles are defined within the <style> element, inside the<head> section of an HTML page

# Inline Styles

An inline style may be used to apply a unique style for a single element.To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

# CSS Rule-Set

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

Eg:-

p {

text-align: center; color: red;

}

Here p stands for paragraph element.

# CSS Colors and Backgrounds

Colors can be applied by using colors and background-color. Colors are displayed combining RED, GREEN, and BLUE light.The CSS background properties are used to define the background effects for elements.

# CSS Margin and Padding Properties.

The CSS margin properties are used to generate space around elements. Margin is the space outside something whereas padding is the space inside something. The margin properties set the size of the white space OUTSIDE the border.The CSS padding properties are used to generate space around content.The padding properties set the size of the white space between the element content and the element border.

# CSS Dimension Properties

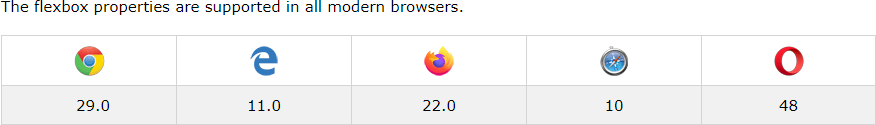
The CSS dimension properties allow you to control the height and width of an element.

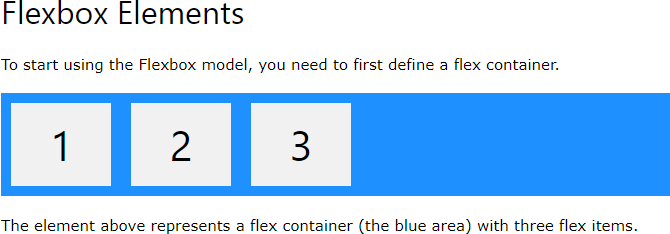
# Navigation Bars

Having easy-to-use navigation is important for any web site.With CSS you can transform boring HTML menus into good-looking navigation bars.

# CSS Flexbox Layout

The Flexible Box Layout Module makes it easier to design flexible responsive layout structure without using float or positioning.





Example

<!DOCTYPE html>

<html>

<head>

<style>

.flex-container { display: flex;

background-color: DodgerBlue;

}

.flex-container > div { background-color: #f1f1f1; margin: 10px;

padding: 20px; font-size: 30px;

}

</style>

</head>

<body>

<div class="flex-container">

<div>1</div>

<div>2</div>

<div>3</div>

</div>

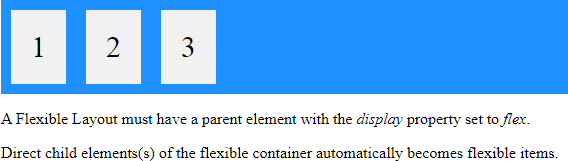
<p>A Flexible Layout must have a parent element with the <em>display</em> property set to <em>flex</em>.</p>

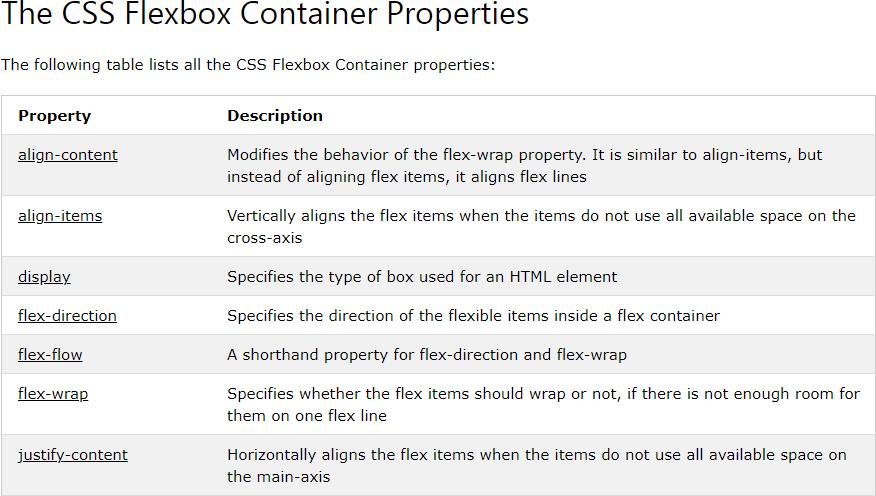
<p>Direct child elements(s) of the flexible container automatically becomes flexible items.</p>

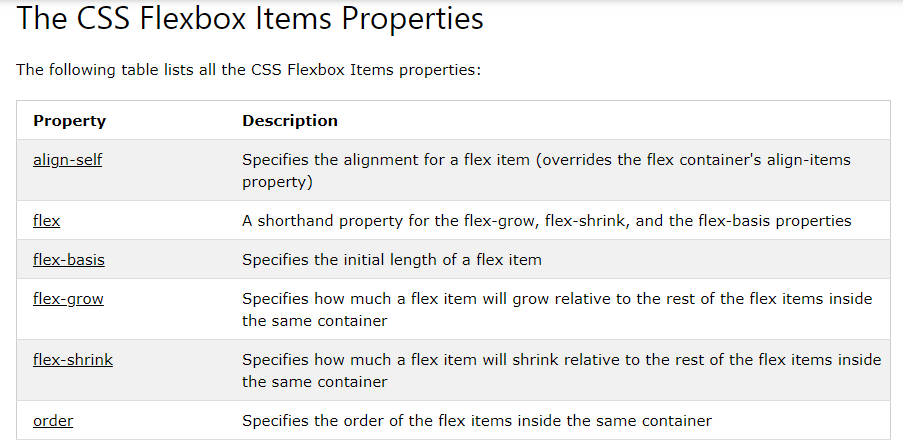
</body>

</html>

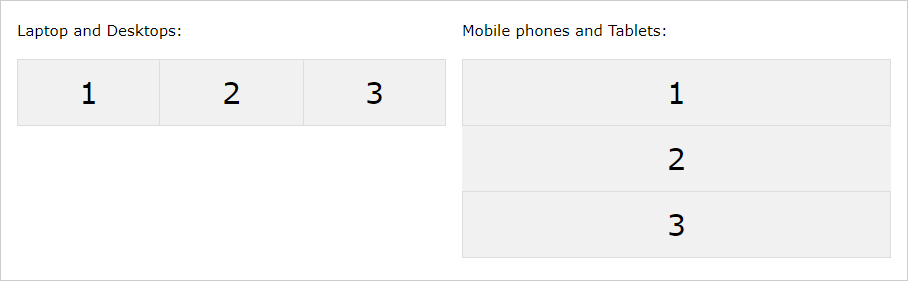
Output



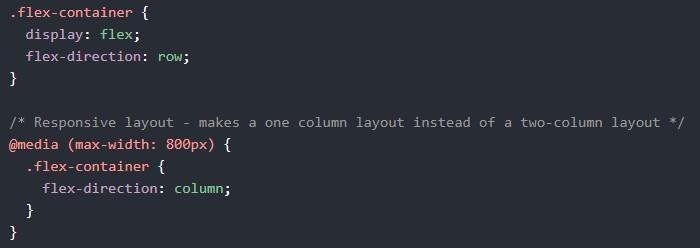




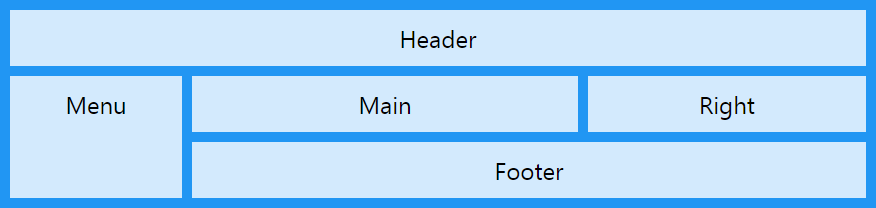
# Responsive Flexbox



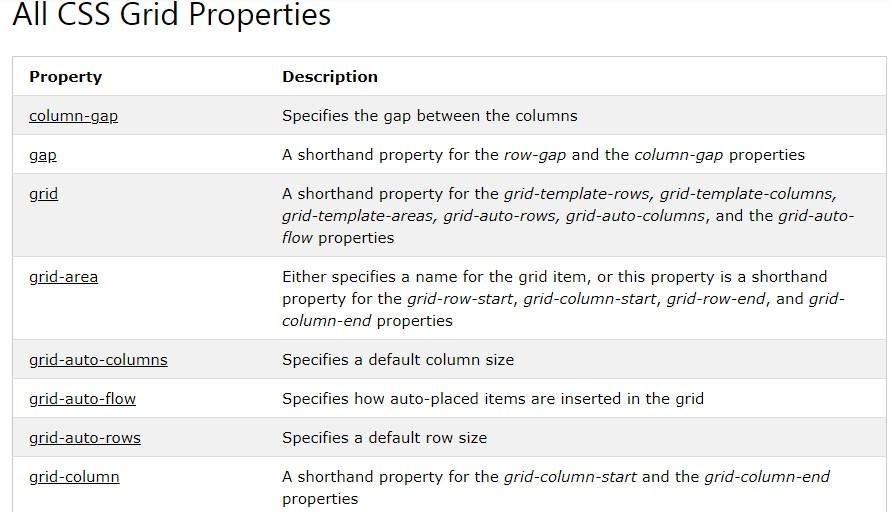
For example, if you want to create a two-column layout for most screen sizes, and a one- column layout for small screen sizes (such as phones and tablets), you can change the flex- direction from row to column at a specific breakpoint (800px in the example below.

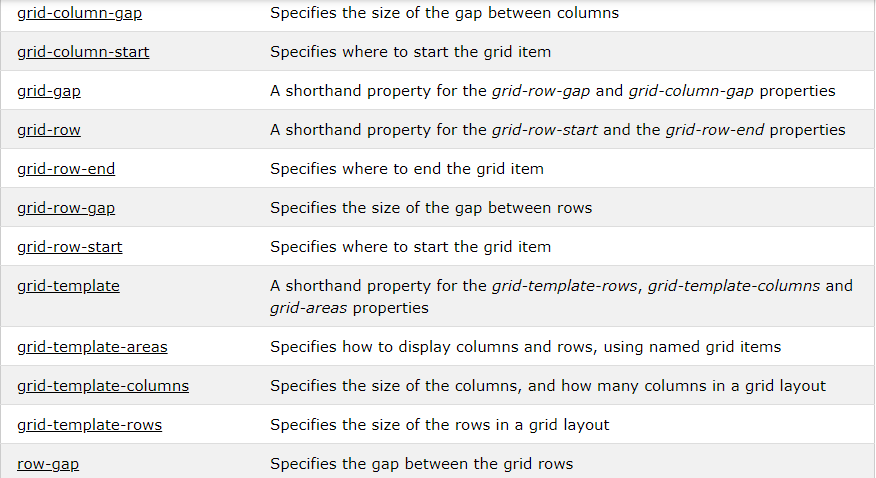


# CSS Grid Layout Module



The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.





# Grid Container

To make an HTML element behave as a grid container, you have to set the display property to grid or inline-grid.

Grid containers consist of grid items, placed inside columns and rows.

# A grid container contains grid items.

By default, a container has one grid item for each column, in each row, but you can style the grid items so that they will span multiple columns and/or rows.



# Activity:

Design a website pages using CSS properties like Colors and Background-color,Lengths and percentages,Margin and padding,Borders,Navigation bars, flexbox layot, flexbox container and item properties, responsive flexbox ,CSS Grid ,Grid Container ,A grid container contains grid items.



# Results: Display the designed web pages along with the code

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>My Portfolio</title>

  <style>

    body {

      font-family: 'Arial', sans-serif;

      margin: 0;

      padding: 0;

      background-color: #f9f9f9;

      color: #333;

    }

    header {

      background-color: #3498db;

      color: #fff;

      padding: 20px;

      text-align: center;

    }

    main {

      max-width: 800px;

      margin: 20px auto;

      padding: 20px;

      background-color: #fff;

      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

      border-radius: 8px;

    }

    section {

      margin-bottom: 20px;

    }

    h1 {

      color: #333;

    }

    .project {

      margin-bottom: 20px;

    }

    form {

      background-color: #ecf0f1;

      padding: 20px;

      border-radius: 8px;

    }

    label {

      display: block;

      margin-bottom: 10px;

      font-weight: bold;

    }

    input, textarea {

      width: 100%;

      padding: 10px;

      margin-bottom: 10px;

      border: 1px solid #ddd;

      border-radius: 5px;

      box-sizing: border-box;

    }

    button {

      background-color: #3498db;

      color: #fff;

      padding: 10px 15px;

      border: none;

      border-radius: 5px;

      cursor: pointer;

    }

  </style>

</head>

<body>

  <header>

    <h1>My Portfolio</h1>

    <p>Welcome to my showcase of projects and skills.</p>

  </header>

  <main>

    <section>

      <h2>About Me</h2>

      <p>

        I am a passionate web developer with a keen interest in creating engaging and interactive websites.

        My skills include HTML, CSS, JavaScript, and various web development frameworks.

      </p>

    </section>

    <section>

      <h2>Projects</h2>

      <div class="project">

        <h3>Project 1</h3>

        <p>Description of Project 1.</p>

      </div>

      <div class="project">

        <h3>Project 2</h3>

        <p>Description of Project 2.</p>

      </div>

    </section>

    <section>

      <h2>Contact Me</h2>

      <form>

        <label for="name">Your Name:</label>

        <input type="text" id="name" name="name" required>

        <label for="email">Your Email:</label>

        <input type="email" id="email" name="email" required>

        <label for="message">Your Message:</label>

        <textarea id="message" name="message" rows="4" required></textarea>

        <button type="submit">Send Message</button>

      </form>

    </section>

  </main>

</body>

</html>

# 

# 

# Outcomes:

# Design webpages using HTML5 and CSS3.



**Questions:**

1. How many axis does a flex-box layout contain?

A flexbox layout contains two axes: the main axis and the cross axis. The main axis is defined by the `flex-direction` property, and it determines the direction in which flex items are placed. The cross axis is perpendicular to the main axis. The combination of these axes allows for flexible and responsive positioning of items within a flex container, providing powerful control over the layout. Flexbox is a one-dimensional layout model, and its simplicity and versatility make it widely used in modern web development for creating dynamic and responsive designs.

1. What is the difference between auto-fill and auto-fit properties when defining a grid- template.

The `auto-fill` and `auto-fit` properties in a grid-template define how the grid tracks are generated based on the available space. `auto-fill` creates as many tracks as possible while filling the container, even if they remain empty. This may result in unevenly sized tracks. On the other hand, `auto-fit` also creates as many tracks as possible but then collapses the empty tracks, distributing the space evenly among the non-empty ones. Unlike `auto-fill`, `auto-fit` ensures a more balanced distribution of space and is often preferred for responsive grid layouts. Both properties are valuable tools for creating flexible and adaptive grid structures in CSS.

1. What is the meaning of the fr unit in the grid layout?

The "fr" unit in the grid layout stands for "fractional unit." It is a flexible unit of measurement that allows the grid container to distribute available space among its columns or rows proportionally. When used in a grid definition, the "fr" unit allocates space based on the specified fractions. For example, if a column is set to "1fr" and another to "2fr," the second column would take up twice as much space as the first. This unit is particularly useful for creating responsive and dynamic grid layouts where columns or rows adjust proportionally based on the available space in the grid container.



# Conclusion: (Conclusion to be based on the outcomes achieved)

**The provided code illustrates the implementation of a simple, responsive portfolio website using HTML and CSS. The structure includes a header section with a welcome message, a main content area with sections for "About Me," "Projects," and a "Contact Me" form. The styling focuses on a clean and professional design, utilizing various CSS properties for layout, colors, and form elements.**

**Key aspects covered in the code include responsive design practices, the use of flexbox for layout control, and the incorporation of grid-based structures for organizing content. The website's design promotes readability and user engagement, with a consistent color scheme and clear section divisions.**



**Grade: AA / AB / BB / BC / CC / CD /DD**

Signature of faculty in-charge with date



# References:

**Books/ Journals/ Websites:**

* + "HTML5: Black Book", Dreamtech Publication.
  + "Web Technologies: Black Book", Dreamtech Publication.
  + [http://www.w3schools.com](http://www.w3schools.com/)