

CSS – Day-3 Hands On – Chintha Ramakrishna

Problem 1

1. Problem Statement - Create two HTML pages (Home and Contact), link them using navigation links, and add a registration form with Name, Email, Password, Gender (radio buttons), Course (dropdown), and use required and placeholder attributes.

Code :-

```
<!DOCTYPE html>
<html>
<head>
    <title>Styled Webpage</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <h1>Welcome to My Website</h1>
    <h2>Learning CSS Styling</h2>

    <p>Learning CSS styling involves understanding how to use Cascading Style Sheets (CSS) to control the layout, colors, fonts, and overall visual appearance of a webpage, which is structured using HTML.</p>

    <section class="info-section">
        <h3>About CSS</h3>
        <p>Cascading Style Sheets is a style sheet language used for specifying the presentation and styling of a document written in a markup language, such as HTML or XML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript..</p>
    </section>

    <div class="box">
        Margin and padding are both used to create space around an HTML element's content, but they operate in different areas of the CSS Box Model.
        Padding is the space inside the element's border, between the content and the border. Its background color is visible in this area.
        Margin is the space outside the element's border, creating a gap between it and other nearby elements.
    </div>
```

```

<h3>Contact Form</h3>
<form>

    <label>Name:</label><br>
    <input type="text"><br><br>

    <label>Email:</label><br>
    <input type="email"><br><br>

    <label>Telephone:</label><br>
    <input type="tel"><br><br>

    <!-- Gender -->
    <label>Gender:</label><br>
    <input type="radio" name="gender" value="Male" required> Male
    <input type="radio" name="gender" value="Female"> Female
    <input type="radio" name="gender" value="Other"> Other
    <br><br>

    <!-- Course -->
    <label>Select Course:</label><br>
    <select required>
        <option value="">--Select Course--</option>
        <option>Python</option>
        <option>Java</option>
        <option>Web Development</option>
        <option>Data Science</option>
    </select>
    <br><br>
    <input type="submit" value="Submit">
</form>

</body>
</html>

```

CSS Code :-

```

/* Body Styling */
body {
    background-color: #f4f4f4;
    font-family: Arial, sans-serif;
}

```

```
/* Heading Styling */
h1 {
    color: darkblue;
    text-align: center;
}

h2 {
    color: green;
}

h3 {
    color: brown;
}

/* Paragraph Styling */
p {
    font-size: 18px;
    color: #333;
}

/* Section Styling */
.info-section {
    background-color: lightyellow;
    border: 2px solid orange;
    padding: 15px;
    margin: 20px 0;
}

/* Box Model Example */
.box {
    background-color: lightblue;
    border: 3px solid blue;
    margin: 30px;
    padding: 20px;
}

/* Form Styling */
form {
    background-color: white;
    padding: 20px;
    border: 2px solid gray;
    width: 300px;
```

```

}

input[type="text"] ,
input[type="email"] ,
input[type="tel"] {
    width: 80% ;
    padding: 8px ;
    margin-bottom: 8px ;
}

input[type="submit"] {
    background-color: darkblue ;
    color: white ;
    padding: 8px 15px ;
    border: none ;
    cursor: pointer ;
}

input[type="submit"]:hover {
    background-color: green ;
}

```

Output:-

Welcome to My Website

Learning CSS Styling

Learning CSS styling involves understanding how to use Cascading Style Sheets (CSS) to control the layout, colors, fonts, and overall visual appearance of a webpage, which is structured using HTML.

About CSS

Cascading Style Sheets is a style sheet language used for specifying the presentation and styling of a document written in a markup language, such as HTML or XML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript..

Margin and padding are both used to create space around an HTML element's content, but they operate in different areas of the CSS Box Model. Padding is the space inside the element's border, between the content and the border. Its background color is visible in this area. Margin is the space outside the element's border, creating a gap between it and other nearby elements.

Contact Form

Name:

Email:

Telephone:

Gender:

Male Female Other

Select Course:

Explanation :-

1. First, create a separate CSS file (style.css) and connect it to the HTML using the `<link>` tag inside the `<head>` so all styles apply externally.
2. Use CSS selectors like `h1`, `h2`, `p` to change text color, font size, font family, and alignment for headings and paragraphs.
3. Apply background color and border to a section using properties like `background-color` and `border` to make that area visually clear.
4. Show the box model by adding `margin` (space outside) and `padding` (space inside) to a div so content looks well-spaced.
5. Finally, style the form elements (input, button, label) using selectors to improve layout, colors, and user experience, resulting in a clean styled webpage.

Problem 2 :- Create a Personal Profile Card using only HTML and CSS that shows a user's name, role, short bio, and a contact button with proper text styling, spacing, borders, background color, and a hover effect on the button.

Code :-

Index.html:-

```
<!DOCTYPE html>
<html>
<head>
    <title>Personal Profile Card</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <div class="profile-card">
        <h1>Rama Krishna</h1>
        <h3 class="role">Aspiring Software Developer</h3>
        <p>
            Passionate about Python, Web Development, and AI.
            I enjoy building simple and clean user-friendly
            applications.
        </p>
        <a href="contact.html" class="contact-btn">Contact Me</a>
    </div>

</body>
</html>
```

Contact.html :-

```
<!DOCTYPE html>
<html>
<head>
    <title>Contact Page</title>
</head>
<body>

<!-- Navigation Bar -->
<nav>
    <a href="index.html">Home</a> |
    <a href="contact.html">Contact</a>
</nav>

<h1>Contact Us</h1>

<p>Email: support@example.com</p>
```

```
<p>Phone: +91 9876543210</p>

</body>
</html>
```

Css :-

```
/* Page Styling */
body {
    background-color: #f4f6f9;
    font-family: Arial, sans-serif;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

/* Profile Card */
.profile-card {
    background-color: white;
    width: 320px;

    /* Box Model */
    padding: 25px; /* Inside spacing */
    border: 2px solid #ddd;
    margin: 20px; /* Outside spacing */

    border-radius: 10px; /* Rounded corners */
    text-align: center;
}

/* Name Styling */
.profile-card h1 {
    font-size: 24px;
    font-weight: bold;
    letter-spacing: 1px;
    margin-bottom: 5px;
}

/* Role Styling */
.role {
    color: gray;
    font-size: 16px;
```

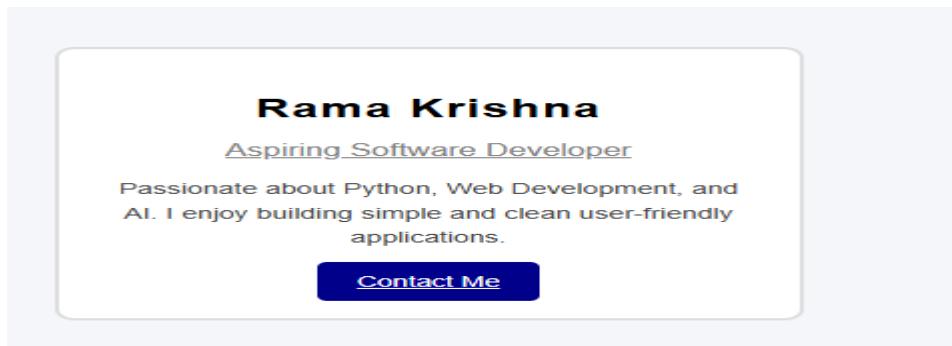
```
    font-weight: 500;
    margin-bottom: 15px;
    text-decoration: underline;
}

/* Description */
.profile-card p {
    font-size: 14px;
    line-height: 1.6;
    color: #444;
    margin-bottom: 20px;
}

/* Button Styling */
.contact-btn {
    background-color: darkblue;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    font-size: 14px;
    cursor: pointer;
    transition: 0.3s ease;
}

/* Hover Effect */
.contact-btn:hover {
    background-color: green;
    transform: scale(1.05);
}
```

Output -



Contact Us

Email: support@example.com

Phone: +91 9876543210

Explanation :-

1. This task helps you design a simple profile card layout by structuring content using HTML elements like headings, paragraphs, and a button.
2. You apply CSS typography such as font-family, font size, weight, alignment, line-height, and letter-spacing to make text clear and readable.
3. The CSS box model is used by adding padding inside the card, a border around it, and margin to create space from the page edges.
4. Visual styling like background color and border radius improves the card's appearance and gives it a modern look.
5. Adding a hover effect on the button makes the design interactive and helps you understand basic user interface styling.

Problem 3:-Create a Product Feature List Section using HTML and CSS that includes a heading and an unordered list of 4–5 features, styled with colors, fonts, borders, padding, and hover effects.

Code :-

```
<!DOCTYPE html>
<html>
<head>
    <title>Product Features</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <section class="features-section">
        <h2>Our Product Features</h2>
        <ul>
            <li>Fast Performance</li>
            <li>User Friendly Interface</li>
            <li>Secure Data Protection</li>
            <li>Cloud Integration</li>
        </ul>
    </section>
</body>
</html>
```

```
        <li>24/7 Customer Support</li>
    </ul>
</section>

</body>
</html>
```

Css:-

```
/* Element Selector */
body {
    font-family: Verdana, sans-serif;
    background-color: #f0f4f8;
}

/* Class Selector */
.features-section {
    width: 400px;
    margin: 50px auto;
    padding: 20px;
    background-color: #ffffff;
    border: 2px solid #ccc;
    border-radius: 8px;
}

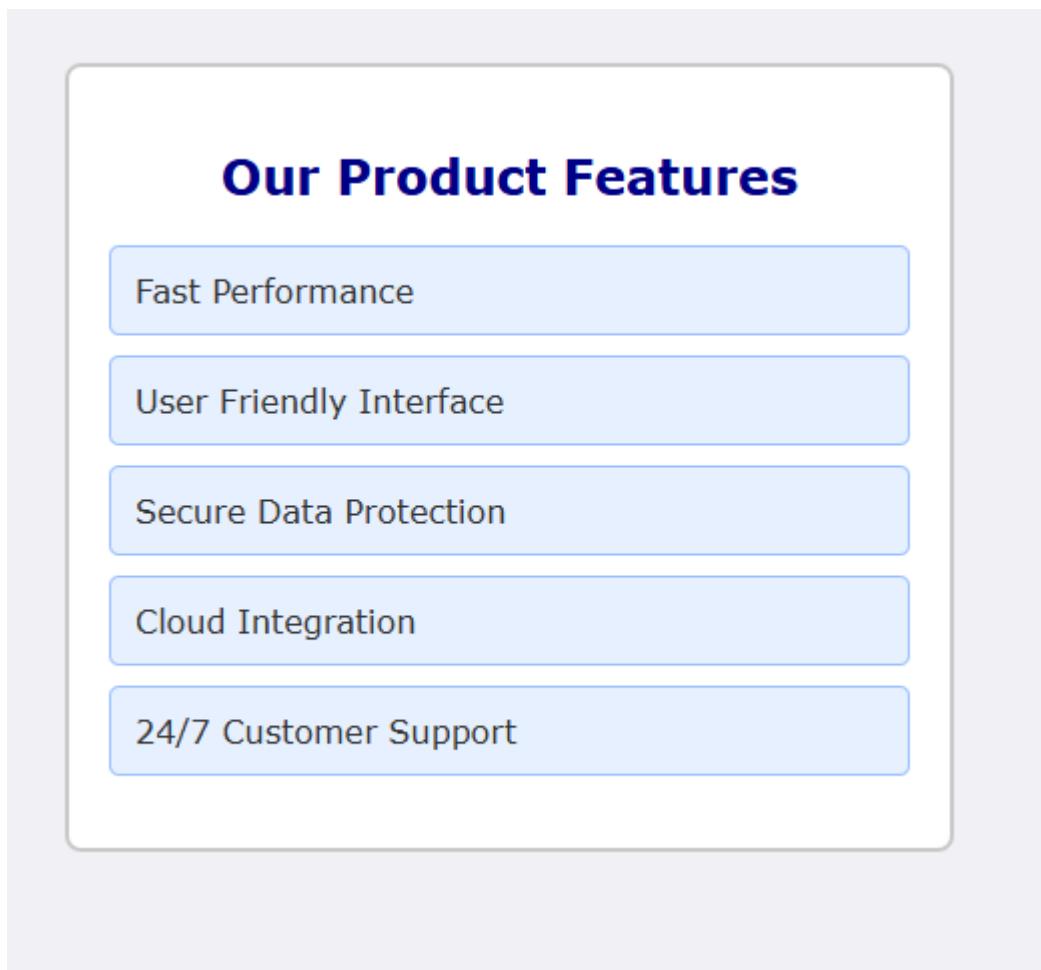
/* Descendant Selector */
.features-section h2 {
    text-align: center;
    color: darkblue;
    margin-bottom: 20px;
}

/* Descendant Selector */
.features-section ul {
    list-style-type: none;
    padding: 0;
}

/* Styling Each Feature */
.features-section ul li {
    background-color: #e6f2ff;
    color: #333;
    padding: 12px;
}
```

```
margin-bottom: 10px;  
border: 1px solid #99c2ff;  
border-radius: 5px;  
transition: 0.3s ease;  
}  
  
/* Hover Effect */  
.features-section ul li:hover {  
background-color: #3399ff;  
color: white;  
transform: translateX(5px);  
}
```

Output :-



Explanation :-

1. This task helps you practice CSS selectors like element, class, and descendant selectors to style different parts of the section.
2. You style each list item by adding custom fonts, text color, background color, border, and padding to make features look like separate cards.

3. Using an unordered list () teaches how to present product features in a clear and structured way.
4. Applying hover effects on feature items makes the UI interactive and improves user experience.
5. Overall, this improves your ability to design clean feature sections and maintain consistent styling.

Problem 4 :- Create a Responsive Blog Layout using HTML and CSS with a header and multiple blog cards arranged using Flexbox or Grid, and make it responsive with media queries.

Code:-

```
<!DOCTYPE html>
<html>
<head>
    <title>Responsive Blog Layout</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <!-- Header -->
    <header class="header">
        <h1>My Blog</h1>
    </header>

    <!-- Main Content -->
    <main class="blog-container">

        <div class="blog-card">
            <h2>Understanding Flexbox</h2>
            <p>Learn how Flexbox helps create responsive layouts easily and efficiently.</p>
            <a href="#" class="read-btn">Read More</a>
        </div>

        <div class="blog-card">
            <h2>CSS Grid Basics</h2>
            <p>Explore how CSS Grid can structure complex layouts in a simple way.</p>
            <a href="#" class="read-btn">Read More</a>
        </div>

        <div class="blog-card">
            <h2>Responsive Design</h2>
```

```

        <p>Understand how media queries help websites adapt to
different screen sizes.</p>
        <a href="#" class="read-btn">Read More</a>
    </div>

    <div class="blog-card">
        <h2>Modern UI Principles</h2>
        <p>Improve user experience with better spacing, typography,
and alignment.</p>
        <a href="#" class="read-btn">Read More</a>
    </div>

</main>

</body>
</html>

```

Css :-

```

/* Global Styles */
body {
    margin: 0;
    font-family: Arial, sans-serif;
    background-color: #f4f6f9;
}

/* Header */
.header {
    background-color: #1e3a8a;
    color: white;
    text-align: center;
    padding: 2%;
}

/* Blog Container using Flexbox */
.blog-container {
    display: flex;
    flex-wrap: wrap;
    gap: 2%;
    padding: 3%;
    box-sizing: border-box;
}

```

```
/* Blog Card */
.blog-card {
    background-color: white;
    flex: 1 1 22%;
    padding: 2%;
    border: 1px solid #ddd;
    border-radius: 8px;
    box-sizing: border-box;
}

/* Card Title */
.blog-card h2 {
    font-size: 1.2rem;
    margin-bottom: 10px;
    letter-spacing: 0.5px;
}

/* Card Description */
.blog-card p {
    font-size: 0.9rem;
    line-height: 1.6;
    margin-bottom: 15px;
}

/* Read More Button */
.read-btn {
    display: inline-block;
    text-decoration: none;
    background-color: #2563eb;
    color: white;
    padding: 8px 14px;
    border-radius: 4px;
    transition: 0.3s ease;
}

/* Hover Effect */
.read-btn:hover {
    background-color: #1e40af;
}

/* Tablet View */
@media (max-width: 992px) {
    .blog-card {
```

```

        flex: 1 1 45%;
    }

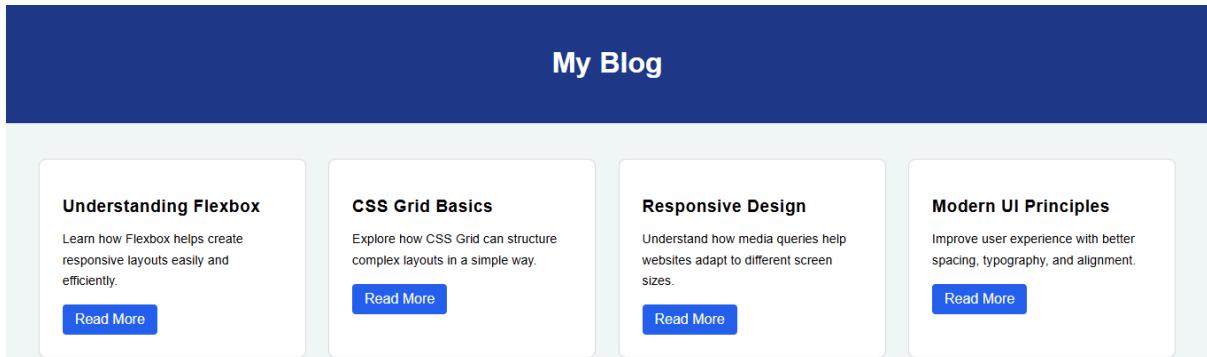
}

/* Mobile View */
@media (max-width: 600px) {
    .blog-container {
        flex-direction: column;
    }

    .blog-card {
        flex: 1 1 100%;
    }
}

```

Output:-



Explanation :-

1. This task helps you design a blog homepage where blog cards show a title, short description, and Read More button.
2. You use Flexbox or CSS Grid to arrange cards in rows and apply the box model (margin, padding, borders) for proper spacing.
3. Typography, background colors, and borders improve readability and visual hierarchy of each blog card.
4. With media queries, the layout shows multiple columns on desktop and changes to a single column on mobile for better viewing.
5. Overall, it teaches responsive design concepts and how to build real-world webpage layouts using only HTML and CSS.

Problem 5:- Write a program that stores a number in a variable, checks whether it is positive or negative using the ternary operator, checks even or odd using if-else, and prints numbers from 1 to the given number using a loop.

Code:-

```

<!DOCTYPE html>
<html>

```

```
<head>
    <title>Responsive Dashboard</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <div class="dashboard">

        <!-- Sidebar -->
        <aside class="sidebar">
            <h2>Admin Panel</h2>
            <nav>
                <ul>
                    <li>Dashboard</li>
                    <li>Users</li>
                    <li>Reports</li>
                    <li>Settings</li>
                </ul>
            </nav>
        </aside>

        <!-- Main Content -->
        <main class="main-content">
            <h1>Dashboard Overview</h1>

            <div class="card-container">
                <div class="card">
                    <h3>Total Users</h3>
                    <p>1,250</p>
                </div>

                <div class="card">
                    <h3>Revenue</h3>
                    <p>$8,400</p>
                </div>

                <div class="card">
                    <h3>New Orders</h3>
                    <p>320</p>
                </div>

                <div class="card">
                    <h3>Support Tickets</h3>
                </div>
            </div>
        </main>
    </div>
</body>
```

```
        <p>45</p>
    </div>
</div>

</main>

</div>

</body>
</html>
```

Css:-

```
/* Reset */
body {
    margin: 0;
    font-family: Arial, sans-serif;
    background-color: #f4f6f9;
}

/* Main Layout */
.dashboard {
    display: flex;
    min-height: 100vh;
}

/* Sidebar */
.sidebar {
    background-color: #1e3a8a;
    color: white;
    padding: 20px;
    width: 20%;
    box-sizing: border-box;
}

.sidebar h2 {
    margin-top: 0;
}

.sidebar ul {
    list-style: none;
    padding: 0;
}
```

```
.sidebar ul li {
    padding: 10px 0;
}

/* Main Content */
.main-content {
    flex: 1;
    padding: 20px;
    box-sizing: border-box;
}

.main-content h1 {
    margin-top: 0;
}

/* Card Container */
.card-container {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
}

/* Cards */
.card {
    background-color: white;
    flex: 1 1 45%;
    padding: 20px;
    border: 1px solid #ddd;
    border-radius: 8px;
    box-sizing: border-box;
}

/* Responsive - Tablet */
@media (max-width: 900px) {
    .card {
        flex: 1 1 100%;
    }
}

/* Responsive - Mobile */
@media (max-width: 600px) {
```

```

.dashboard {
    flex-direction: column;
}

.sidebar {
    width: 100%;
    text-align: center;
}

.card-container {
    flex-direction: column;
}

.card {
    flex: 1 1 100%;
}
}

```

Output :-

Dashboard Overview	
Total Users	Revenue
1,250	\$8,400
New Orders	Support Tickets
320	45

Explanation :-

1. This task helps you practice storing data in a variable and applying different control flow methods in one program.
2. The ternary operator is used for a quick check to decide whether the number is positive or negative.
3. An if–else statement is used to determine if the number is even or odd based on division by 2.
4. A loop (for or while) prints all numbers from 1 up to the given number, showing how iteration works.
5. Overall, it teaches how to combine conditions, operators, and loops to build a multi-step logical program.