**TO CREATE A CV using HTML and CSS.**

**Objective:**

1. To understand the use of different HTML tags.
2. To learn and understand the use of CSS properties.

**Theory:**

**HTML:**

HTML (Hypertext Markup Language) is the standard language used to create and design web pages. It provides a structure for organizing content using tags, which define the elements of a webpage such as headings, paragraphs, images, links, and more. HTML documents are interpreted by web browsers to render the visual layout and content of a webpage. It is the backbone of the World Wide Web, forming the foundation for building interactive and multimedia-rich websites. HTML works in conjunction with CSS (Cascading Style Sheets) and JavaScript to create dynamic and visually appealing web experiences.

**HTML properties used in this project:**

1. Main Element: `<main>` serves as the primary content section of a web page, providing thecentral focus for users and search engines to understand the purpose and theme of the document.
2. Article Element: Enclosed within `<article>`, individual pieces of content gain semanticsignificance, aiding accessibility and SEO by delineating stand-alone sections such as blog posts, forum threads, or news articles.
3. Section Element: `<section>` organizes related content thematically, enhancing the structuralclarity of web pages, thus facilitating navigation and comprehension for both users and assistive technologies.
4. Anchor Tag: `<a>` facilitates seamless navigation within and between web pages, acting ashyperlinks that connect content across the internet, enhancing user experience and interconnectivity.
5. Heading Tags: From `<h1>` to `<h6>`, heading tags hierarchically structure content, aidingreadability and SEO by signaling the importance of text elements, thereby guiding both users and search engines through the page's information hierarchy.
6. P Tag: Wrapped within `<p>`, textual content gains semantic meaning as paragraphs, aiding inreadability, accessibility, and styling, while maintaining separation between distinct units of information.
7. Attributes (ID and Classes):
   * ID Attribute: Uniquely identifies individual elements within a webpage, facilitating targetedstyling and scripting, while also serving as anchor points for intra-page navigation.
   * Class Attribute: Groups elements sharing similar characteristics, streamlining CSS styling andJavaScript functionality, promoting maintainability and scalability of web projects.

**CSS:**

CSS (Cascading Style Sheets) is a style sheet language used to describe the presentation of a document written in HTML or XML. It controls the visual appearance of HTML elements by specifying colors, fonts, spacing, layout, and more. CSS enables web designers to separate the content of a webpage from its presentation, allowing for consistent styling across multiple pages. It enhances the user experience by providing flexibility, responsiveness, and aesthetic appeal to websites. CSS works alongside HTML and JavaScript to create engaging and visually appealing web designs.

**Some of the CSS properties used in this project:**

* + Color: Sets the color of text content or backgrounds, enhancing readability and aesthetics.
  + Background Color: Defines the background color of elements, adding visual emphasis ordifferentiation to sections of a webpage.
  + Margin: Controls the spacing around elements, allowing for precise control over layout andalignment.
  + Padding: Adjusts the space between the content of an element and its border, enhancing visualappeal and readability.
  + Line Height: Sets the vertical spacing between lines of text, improving legibility and overalldesign consistency.
  + Text Transform: Alters the capitalization of text, offering options like uppercase or lowercase forstylistic purposes or to enhance accessibility.
  + Width: Specifies the width of an element, enabling responsive design and precise layout control.- Height: Determines the height of an element, crucial for creating consistent and visually appealing interfaces, particularly in grid-based layouts.

**Methodology:**

**HTML code:**

<!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*="style.css">

    <title>Kushal Poudel | Flutter Developer</title>

</head>

<body>

    <div *class*="resume">

        <div *class*="left-side">

            <div *class*="profile">

                <div *class*="profile-img">

                    <img *src*="profile.jpg" *alt*="Profile Photo">

                </div>

                <h1 *class*="name">Kushal Poudel</h1>

                <p *class*="title">Student</p>

            </div>

            <div *class*="contact padding-top-bg">

                <h1 *class*="heading-primary-white">Contact</h1>

                <div *class*="info-box padding-top-sm">

                    <div *class*="info-title">Phone</div>

                    <div *class*="info">

                        <a *href*="+977-9846925453">+977-9846925453</a>

                    </div>

                </div>

                <div *class*="info-box padding-top-sm">

                    <div *class*="info-title">Email</div>

                    <div *class*="info">

                        <a *href*="kushalpoudel077@gmail.com">kushalpoudel077@gmail.com</a>

                    </div>

                </div>

                <div *class*="info-box padding-top-sm">

                    <div *class*="info-title">Address</div>

                    <div *class*="info">Archalboat, Pokhara</div>

                </div>

            </div>

            <div *class*="education padding-top-bg">

                <h1 *class*="heading-primary-white">Education</h1>

                <div *class*="info-box padding-top-sm">

                    <div *class*="info-title">Tribhuwan University</div>

                    <div *class*="info">Computer Science & Information technology,in progress.</div>

                </div>

            </div>

            <div *class*="expertise padding-top-bg">

                <h1 *class*="heading-primary-white">Expertise</h1>

                <div *class*="skill padding-top-sm">

                    <span></span>

                    <div *class*="skill-name">UI/UX</div>

                </div>

                <div *class*="skill padding-top-sm">

                    <span></span>

                    <div *class*="skill-name">Visual Design</div>

                </div>

                <div *class*="skill padding-top-sm">

                    <span></span>

                    <div *class*="skill-name">Flutter Development</div>

                </div>

            </div>

            <div *class*="language padding-top-bg">

                <h1 *class*="heading-primary-white">Language</h1>

                <div *class*="info-box padding-top-sm">

                    <div *class*="lang-box">

                        <p *class*="info-title">English</p>

                        <span *class*="percent"><div *style*="width: 70%;"></div></span>

                    </div>

                    <div *class*="lang-box">

                        <p *class*="info-title">Nepali</p>

                        <span *class*="percent"><div *style*="width: 100%;"></div></span>

                    </div>

                    <div *class*="lang-box">

                        <p *class*="info-title">Hindi</p>

                        <span *class*="percent"><div *style*="width: 40%;"></div></span>

                    </div>

                </div>

            </div>

        </div>

        <!-- *Right-side* -->

        <div *class*="right-side">

            <div *class*="about-me padding-top-bg">

                <h1 *class*="heading-primary-black">About Me</h1>

                <p *class*="discription">

Dedicated and enthusiastic BSc.CSIT student with a keen interest in mobile application development, particularly in FLutter. Posses a strong foundation in computer science concepts and programming languages. Proven ability to adapt quickly to new technlogies and work collaboratively in team enviroments.                </p>

            </div>

            <div *class*="experience padding-top-bg">

                <h1 *class*="heading-primary-black">Programming Skills</h1>

                <div *class*="xp-box padding-top-sm">

                    <p *class*="discription">

                        - Intermediate Dart Programming <br>

                        - Basic Knowledge of HTML CSS & JS <br>

                        - Basic C++ Programming <br>

                        - Intermediate C programming

                    </p>

                </div>

            </div>

            <div *class*="experience padding-top-bg">

                <h1 *class*="heading-primary-black">Hobbies</h1>

                <div *class*="xp-box padding-top-sm">

                    <p *class*="discription">

                        - Football <br>

                        - Fitness ( GYM ) <br>

                        - Reading <br>

                        - Basketball

                    </p>

                </div>

            </div>

            <div *class*="experience padding-top-bg">

                <h1 *class*="heading-primary-black">Projects</h1>

                <div *class*="xp-box padding-top-sm">

                    <p *class*="discription">

                        You can find projects created by me on following link. <br>

                        Github Repository Link:

<a *href*="https://github.com/kushalchh07">Click Me.</a>

                    </p>

                </div>

            </div>

        </div>

    </div>

</body>

</html>

**CSS code:**

@*import* url('https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&display=swap');

/\* *Basic Constraction* \*/

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

html {

    font-size: 62.5%;

}

body {

    display: flex;

    justify-content: center;

    align-items: center;

    font-family: 'lato', sans-serif;

    background-color: #586783;

}

.*resume* {

    margin: 10rem;

    display: grid;

    grid-template-columns: 1fr 2fr;

}

.*resume* .*left-side* {

    padding: 4rem;

    background-color: #323b4c;

    overflow: hidden;

}

.*resume* .*left-side* ::*selection* {

    color: #000;

    background-color: #fff;

}

.*resume* .*right-side* {

    padding: 4rem 3rem;

    background-color: #fff;

    overflow: hidden;

}

.*resume* .*right-side* ::*selection* {

    color: #fff;

    background-color: #323b4c;

}

/\* *Reusable Componant* \*/

.*padding-top-bg* {

    padding-top: 4rem !important;

}

.*padding-top-sm* {

    padding-top: 2rem !important;

}

.*heading-primary-white*,

.*heading-primary-black* {

    font-size: 2.4rem;

    font-weight: bold;

    letter-spacing: 2px;

    margin-bottom: 1rem;

    position: relative;

}

.*heading-primary-white* {color: #fff;}

.*heading-primary-black* {color: #000;}

.*heading-primary-white*::*after*,

.*heading-primary-black*::*after* {

    content: '';

    height: 2px;

    position: absolute;

    left: 0;

    top: 3.5rem;

}

.*heading-primary-white*::*after* {background-color: #fff; width: 120%;}

.*heading-primary-black*::*after* {background-color: #000; width: 100%;}

.*info-box* .*info-title* {

    color: #fff;

    font-size: 1.8rem;

    font-weight: bold;

    margin-bottom: 5px;

}

.*info-box* .*info*,

.*info-box* .*info* a {

    color: #ddd;

    font-size: 1.4rem;

    text-decoration: none;

    margin-bottom: 5px;

}

.*discription* {

    font-size: 1.6rem;

    color: #555;

    padding: 1.5rem 0;

    text-align: justify;

    line-height: 1.5;

}

/\* *Profile* \*/

.*profile* {

    display: flex;

    flex-direction: column;

    align-items: center;

}

.*profile* .*name* {

    font-size: 2.4rem;

    color: #fff;

    margin-bottom: 2px;

}

.*profile* .*title* {

    font-size: 1.6rem;

    color: #ddd;

    letter-spacing: 3px;

}

.*profile* .*profile-img* {

    background-color: #fff;

    width: 20rem;

    height: 20rem;

    border-radius: 50%;

    overflow: hidden;

    display: flex;

    justify-content: center;

    align-items: center;

    border: 2px solid #fff;

    margin: 0 auto;

    margin-bottom: 2rem;

}

.*resume* .*left-side* .*profile-img* img {

    width: 100%;

}

/\* *Expertise* \*/

.*skill* {

    display: flex;

    align-items: center;

    gap: 2rem;

}

.*skill* span {

    display: block;

    height: 3px;

    width: 3px;

    border-radius: 50%;

    background-color: #fff;

}

.*skill* .*skill-name* {

    color: #fff;

    font-size: 1.6rem;

}

/\* *Language* \*/

.*lang-box* {

    display: flex;

    justify-content: space-between;

    align-items: center;

    gap: 2rem;

    padding-bottom: 1rem;

}

.*lang-box* p {

    width: 40%;

}

.*lang-box* .*percent* {

    position: relative;

    display: block;

    height: 5px;

    width: 60%;

    background-color: #fff;

}

.*lang-box* .*percent* div {

    position: absolute;

    background-color: #2264ff;

    height: 5px;

}

/\* *Experience* \*/

.*xp-box* {

    position: relative;

    padding-left: 3rem;

}

.*xp-box*::*before* {

    content: '';

    position: absolute;

    top: 2rem;

    left: 0;

    background-color: #555;

    height: 100%;

    width: 2px;

}

.*xp-box*:*last-child*:*before* {

    height: 91%;

}

.*xp-box*::*after* {

    content: '';

    position: absolute;

    top: 2rem;

    left: -7px;

    background-color: #fff;

    height: 1.2rem;

    width: 1.2rem;

    border-radius: 50%;

    border: 2px solid #333;

}

.*xp-box* .*xp-year* {

    color: #333;

    font-size: 1.4rem;

    font-weight: bold;

    margin-bottom: 5px;

    letter-spacing: 2px;

}

.*xp-box* .*xp-company* {

    color: #555;

    font-size: 1.6rem;

    margin-bottom: 5px;

}

.*xp-box* .*xp-position* {

    color: #333;

    font-size: 1.8rem;

    font-weight: bold;

}

.*xp-box* .*padding-fix* {

    padding-bottom: 0;

}

/\* *Responsive* \*/

@*media* screen and (max-width: 1000px) {

    .*resume* {

        grid-template-columns: repeat(1, 1fr);

    }

}

@*media* screen and (max-width: 700px) {

    .*refer-box* {display: block;}

    .*refer*:*first-child* {margin-bottom: 3rem;}

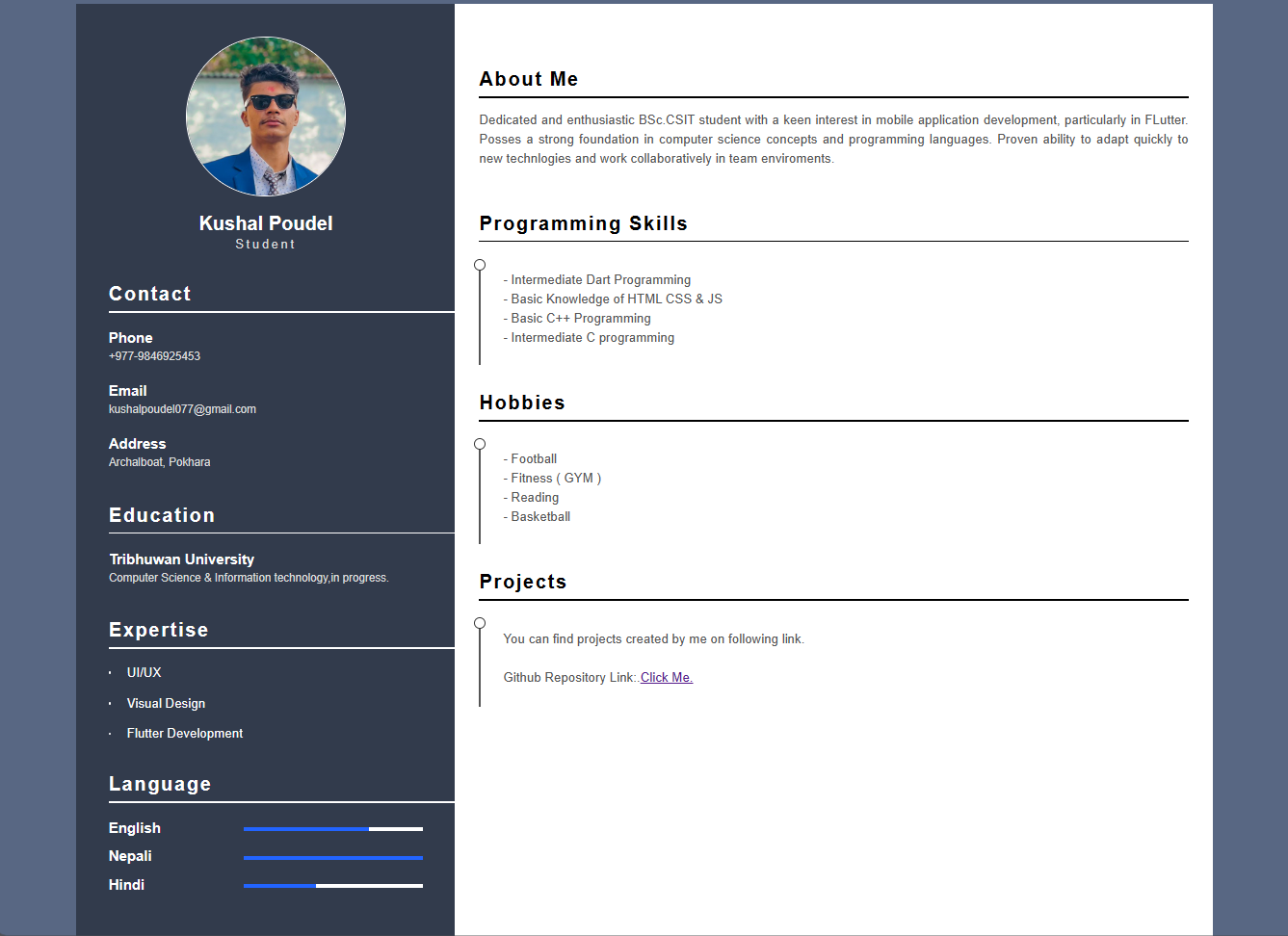
    .*resume* {margin: 0;}

}

The code can be cloned or downloaded using this link.

<https://github.com/kushalchh07/HTML.git>

**Output:**



**Conclusion:**

HTML's structural elements and CSS's styling properties, crafting a compelling Curriculum Vitae (CV) becomes an intuitive endeavor. Employing CSS properties like margin and padding allows for precise layout control, ensuring information is well-organized and visually appealing. Properties such as color and background-color add vibrancy and emphasis to key sections, while texttransform enhances readability. Utilizing width and height attributes ensures consistency and balance, while line-height promotes legibility.

**Working With Different HTML Elements**

**Objective:**

1. To understand the use of different HTML tags.

**Theory:**

**HTML:**

HTML (Hypertext Markup Language) is the standard language used to create and design web pages. It provides a structure for organizing content using tags, which define the elements of a webpage such as headings, paragraphs, images, links, and more. HTML documents are interpreted by web browsers to render the visual layout and content of a webpage. It is the backbone of the World Wide Web, forming the foundation for building interactive and multimedia-rich websites. HTML works in conjunction with CSS (Cascading Style Sheets) and JavaScript to create dynamic and visually appealing web experiences.

**Experiment 1)**

**Create an HTML document that includes both ordered and unordered lists. The ordered list should contain the steps for preparing a cup of tea. The unordered list should include five of your favorite movies.**

**HTML CODE:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Tea Preparation and Favorite Movies</title>

</head>

<body>

    <h1>How to Prepare a Cup of Tea</h1>

    <ol>

        <li>Boil water in a kettle.</li>

        <li>Place a tea bag in a cup.</li>

        <li>Pour the boiling water into the cup over the tea bag.</li>

        <li>Let the tea steep for 3-5 minutes.</li>

        <li>Remove the tea bag and discard it.</li>

        <li>Add sugar, milk, or lemon to taste, if desired.</li>

        <li>Stir and enjoy your tea.</li>

    </ol>

    <h1>My Favorite Movies</h1>

    <ul>

        <li>3 Idiots</li>

        <li>Avengers: Endgame</li>

        <li>Your Name</li>

        <li>I Want to Eat Your Pancreas</li>

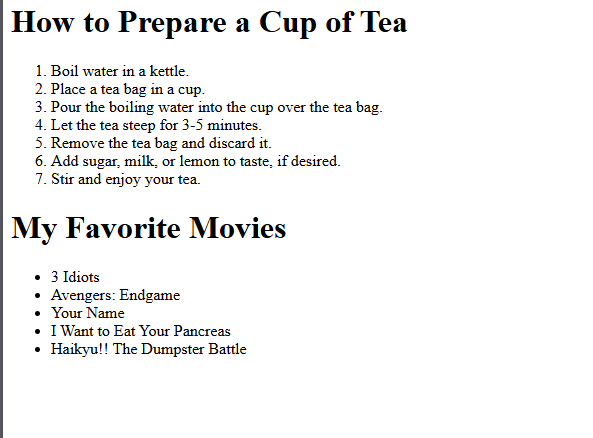
        <li>Haikyu!! The Dumpster Battle</li>

    </ul>

</body>

</html>

**Output:**



**Experiment 2) Working With Html Tables**

1. **Create any table structure using HTML using cellspacing , cellpadding,border,rowspan,**

**And colspan)**

1. **Create your regular class routine using HTML table.**

**HTML Code:**

**a)**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>HTML Table Example</title>

    <style>

        table {

            border-collapse: collapse;

        }

        table, th, td {

            border: 1px solid black;

        }

        th, td {

            padding: 10px;

        }

    </style>

</head>

<body>

    <h1>Sample Table with Various Attributes</h1>

    <table border="1" cellspacing="5" cellpadding="10">

        <tr>

            <th rowspan="2">Header 1</th>

            <th colspan="2">Header 2</th>

            <th>Header 3</th>

        </tr>

        <tr>

            <th>Subheader 2.1</th>

            <th>Subheader 2.2</th>

            <th>Header 4</th>

        </tr>

        <tr>

            <td>Row 1, Cell 1</td>

            <td>Row 1, Cell 2</td>

            <td>Row 1, Cell 3</td>

            <td rowspan="2">Row 1-2, Cell 4</td>

        </tr>

        <tr>

            <td>Row 2, Cell 1</td>

            <td colspan="2">Row 2, Cells 2-3</td>

        </tr>

        <tr>

            <td colspan="3">Row 3, Cells 1-3</td>

            <td>Row 3, Cell 4</td>

        </tr>

    </table>

</body>

</html>

**b)**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Class Routine</title>

    <style>

        table {

            width: 100%;

            border-collapse: collapse;

        }

        th, td {

            border: 1px solid black;

            padding: 8px;

            text-align: left;

        }

        th {

            background-color: #f2f2f2;

        }

    </style>

</head>

<body>

    <h1>Class Routine</h1>

    <table>

        <tr>

            <th>Subject</th>

            <th>Time</th>

            <th>Teacher</th>

        </tr>

        <tr>

            <td>Design and Analysis of Algorithm</td>

            <td>6:00 - 6:50</td>

            <td>Prithivi Raj Paneru</td>

        </tr>

        <tr>

            <td>Web Technology</td>

            <td>6:50 - 7:40</td>

            <td>Gyaneshwor Dhungana</td>

        </tr>

        <tr>

            <td>System Analysis and Design</td>

            <td>7:40 - 8:30</td>

            <td>Surya Poudel</td>

        </tr>

        <tr>

            <td>Cryptography</td>

            <td>8:30 - 9:20</td>

            <td>Devilal Timilsina</td>

        </tr>

        <tr>

            <td>Simulation and Modeling</td>

            <td>9:20 - 10:10</td>

            <td>Amrita Thapa</td>

        </tr>

        <tr>

            <td>Multimedia</td>

            <td>10:10 - 11:00</td>

            <td>Thakur Prasad Upadhya</td>

        </tr>

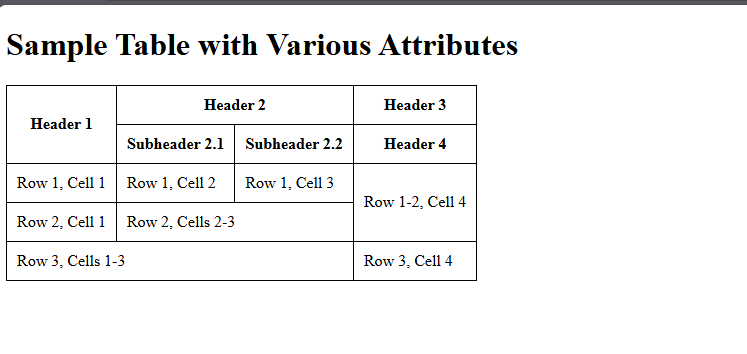
    </table>

</body>

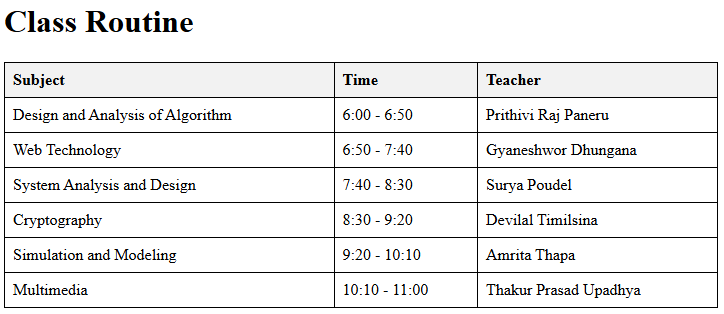
</html>

**Output:**

**a)**

****

**b)**

****

**Experiment 3) Working with Forms in HTML**

1. Create A user Login Form
2. Create a user Registration Form

**Html Code:**

**A)**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Login</title>

</head>

<body>

    <h1>Login</h1>

    <form action="/login" method="post">

        <label for="login-email">Email:</label><br>

        <input type="email" id="login-email" name="login-email" required><br><br>

        <label for="login-password">Password:</label><br>

        <input type="password" id="login-password" name="login-password" required><br><br>

        <input type="submit" value="Login">

    </form>

</body>

</html>

**B)**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Registration</title>

</head>

<body>

    <h1>Registration</h1>

    <form action="/register" method="post">

        <label for="register-name">Name:</label><br>

        <input type="text" id="register-name" name="register-name" required><br><br>

        <label for="register-phone">Phone Number:</label><br>

        <input type="tel" id="register-phone" name="register-phone" required><br><br>

        <label for="register-email">Email:</label><br>

        <input type="email" id="register-email" name="register-email" required><br><br>

        <label for="register-password">Password:</label><br>

        <input type="password" id="register-password" name="register-password" required><br><br>

        <input type="submit" value="Register">

    </form>

</body>

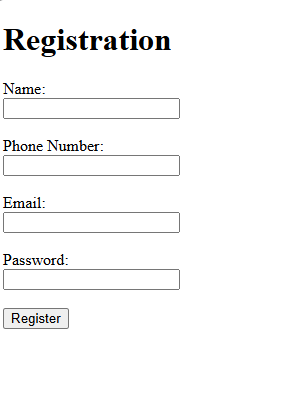
</html>

**Output:**

**A)**

****

**B)**

****

**Experiment 4)**

**- Working With Hyperlink in HTML**

**HTML Code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Simple Webpage</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

        }

        nav {

            background-color: #333;

            color: white;

            padding: 10px;

        }

        nav a {

            color: white;

            margin: 0 15px;

            text-decoration: none;

            font-weight: bold;

        }

        nav a:hover {

            color: #ff6347;

        }

        section {

            padding: 20px;

            margin: 20px 0;

            display: none; /\* Hide all sections by default \*/

        }

        section.active {

            display: block; /\* Show the active section \*/

        }

        a {

            color: #0066cc;

            text-decoration: none;

        }

        a:hover {

            color: #ff6347;

            text-decoration: underline;

        }

    </style>

    <script>

        function showSection(id) {

            // Hide all sections

            const sections = document.querySelectorAll('section');

            sections.forEach(section => section.classList.remove('active'));

            // Show the selected section

            const activeSection = document.getElementById(id);

            activeSection.classList.add('active');

        }

        // Ensure the first section (home) is displayed by default

        window.onload = function() {

            showSection('home');

        }

    </script>

</head>

<body>

    <nav>

        <a href="javascript:void(0)" onclick="showSection('home')">Home</a>

        <a href="javascript:void(0)" onclick="showSection('about')">About</a>

        <a href="javascript:void(0)" onclick="showSection('contact')">Contact</a>

        <a href="javascript:void(0)" onclick="showSection('external')">External Link</a>

        <a href="javascript:void(0)" onclick="showSection('download')">Download</a>

    </nav>

    <section id="home" class="active">

        <h1>Home</h1>

        <p>Welcome to our simple webpage! Use the navigation links to explore different sections.</p>

    </section>

    <section id="about">

        <h1>About</h1>

        <p>This is the about section. Here you can write some information about the website or yourself.</p>

    </section>

    <section id="contact">

        <h1>Contact</h1>

        <p>Feel free to contact us at <a href="mailto:example@example.com">example@example.com</a>.</p>

    </section>

    <section id="external">

        <h1>External Link</h1>

        <p>Visit our favorite website: <a href="https://www.facebook.com" target="\_blank">Example Website</a>.</p>

    </section>

    <section id="download">

        <h1>Download</h1>

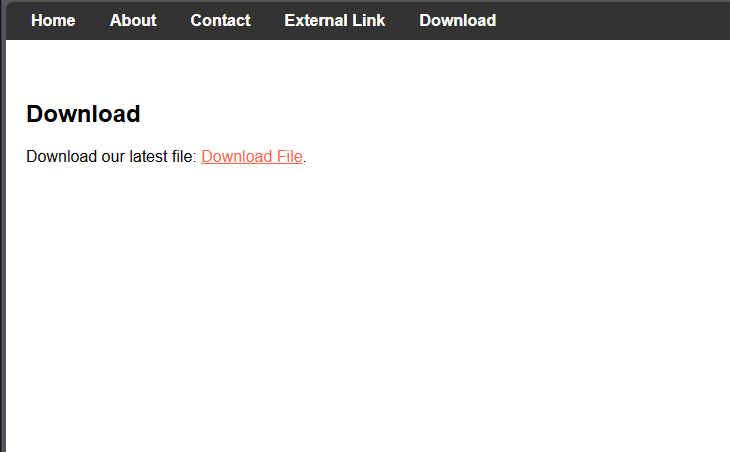
        <p>Download our latest file: <a href="path/to/your/file.zip" download>Download File</a>.</p>

    </section>

</body>

</html>

**Output:**

****

**Experiment 5)**

**- Working with frames in html(frameset & iframe)**

**Html code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Embed YouTube Videos and Auto-Play Audio</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 20px;

        }

        .video {

            margin-bottom: 20px;

        }

        audio {

            display: none; /\* Hide the audio controls \*/

        }

    </style>

</head>

<body>

    <h1>Embedded YouTube Videos</h1>

    <div class="video">

        <h2>Video 1</h2>

        <iframe width="560" height="315" src="https://www.youtube.com/embed/dQw4w9WgXcQ" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

    </div>

    <div class="video">

        <h2>Video 2</h2>

        <iframe width="560" height="315" src="https://www.youtube.com/embed/3JZ\_D3ELwOQ" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

    </div>

    <div class="video">

        <h2>Video 3</h2>

        <iframe width="560" height="315" src="https://www.youtube.com/embed/VYOjWnS4cMY" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

    </div>

    <audio autoplay>

        <source src="path/to/your/audiofile.mp3" type="audio/mpeg">

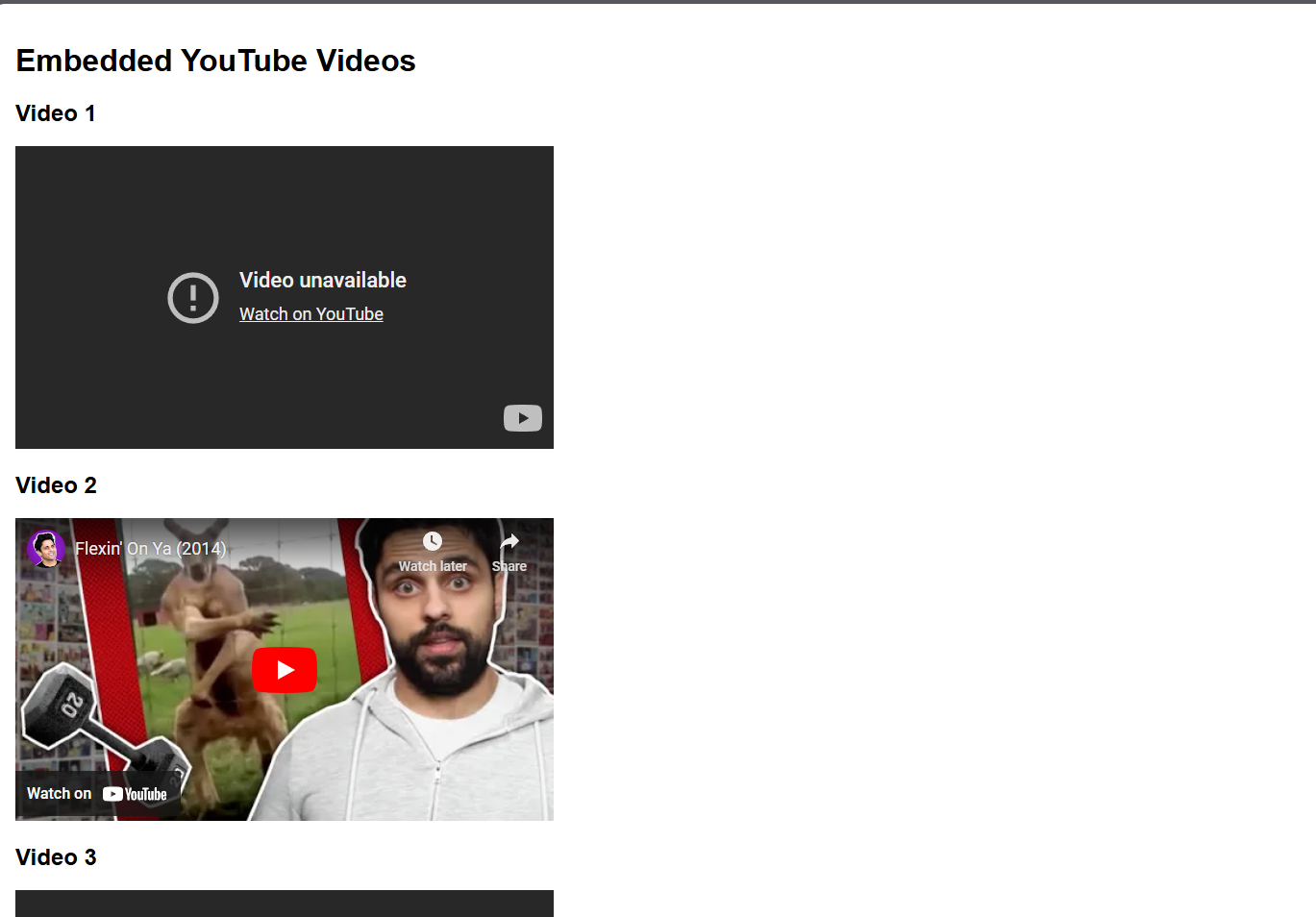
        Your browser does not support the audio element.

    </audio>

</body>

</html>

**Output:**

****

Note: Code can be cloned and downloaded using following link:

https://github.com/kushalchh07/HTML.git

**Conclusion:**

After Doing All of above experiment I have become able to understand how HTML works. How different elements can be used when, working with lists, tables, hyperlinks, forms and iframes In html.

**Different ways of Inserting CSS to HTML**

**Objectives:**

1. Learning about CSS.
2. Using different ways of inserting CSS to HTML.

**Theory:**

CSS stands for Cascading Style Sheets. It is a stylesheet language used for describing the presentation of a document written in HTML or XML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

Here are some key points about CSS:

1. **Separation of Content and Presentation**: CSS is used to separate the content of a web page (which is written in HTML) from its presentation (how it looks).
2. **Styles**: CSS allows you to apply styles to web pages. This includes setting colors, fonts, layout, and various visual effects.
3. **Selectors and Rules**: CSS works by associating rules with HTML elements. These rules are specified in "selectors", which match elements in the document tree, and "declarations", which define the styles to be applied.
4. **Cascading and Inheritance**: CSS follows a "cascading" model, meaning that styles can fall back from more specific to less specific rules. Styles can also be inherited from parent elements to child elements.
5. **Responsive Design**: CSS supports responsive web design techniques, allowing web pages to adapt their layout and appearance based on the size and orientation of the screen or device.
6. **Syntax**: CSS has a specific syntax where selectors are used to target HTML elements, and properties and values are defined to apply styles.

**Experiment 1)**

**a)Using different ways of inserting CSS to HTML**

**- Inline CSS**

**- Internal CSS**

**- External CSS**

**HTML code**

**a)Inline CSS**

<!-- Inline CSS example -->

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Inline CSS Example</title>

</head>

<body>

    <h1 style="color: blue; font-size: 24px;">This is an inline CSS example</h1>

    <p style="color: green; font-size: 16px;">Inline CSS applies styles directly to the HTML element.</p>

</body>

</html>

**Output:**

**b)Internal CSS**

<!-- Internal CSS Example -->

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Internal CSS Example</title>

    <style>

        h1 {

            color: blue;

            font-size: 24px;

        }

        p {

            color: green;

            font-size: 16px;

        }

    </style>

</head>

<body>

    <h1>This is an internal CSS example</h1>

    <p>Internal CSS is defined within a &lt;style&gt; tag in the head section.</p>

</body>

</html>

**Output:**

1. **External CSs**

**Using external css include using external file which extension is .css (styles.css)**

**We need to provide the link to the html file as :**

    <link rel="stylesheet" href="styles.css">

**HTML code:**

<!-- External css example -->

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>External CSS Example</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <h1>This is an external CSS example</h1>

    <p>External CSS is defined in a separate file.</p>

</body>

</html>

**Styles.css**

h1 {

    color: blue;

    font-size: 24px;

}

p {

    color: green;

    font-size: 16px;

}

**Output:**

**Experiment 2)**

**Using different css selectors (Universal Selector , element selector, id and class selector, attribute selector, pseudo class selector, psuedo element selector, descendant selector, child selector, adjacent sibling selector, general sibling selector, grouping selectors using comma).**

**HTML Code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>CSS Selectors Example</title>

    <style>

        /\* Universal Selector \*/

        \* {

            box-sizing: border-box;

        }

        /\* Element Selector \*/

        p {

            color: blue;

        }

        /\* ID Selector \*/

        #unique {

            font-size: 20px;

            font-weight: bold;

        }

        /\* Class Selector \*/

        .highlight {

            background-color: yellow;

        }

        /\* Attribute Selector \*/

        [type="text"] {

            border: 1px solid gray;

        }

        /\* Pseudo-class Selector \*/

        a:hover {

            color: red;

        }

        /\* Pseudo-element Selector \*/

        p::first-line {

            font-weight: bold;

        }

        /\* Descendant Selector \*/

        div p {

            color: green;

        }

        /\* Child Selector \*/

        ul > li {

            list-style-type: square;

        }

        /\* Adjacent Sibling Selector \*/

        h1 + p {

            margin-top: 0;

        }

        /\* General Sibling Selector \*/

        h1 ~ p {

            color: gray;

        }

        /\* Grouping Selectors \*/

        h1, h2, h3 {

            font-family: Arial, sans-serif;

        }

    </style>

</head>

<body>

    <h1>Heading 1</h1>

    <p>This is a paragraph styled with element selector.</p>

    <p class="highlight">This paragraph has a highlight class applied.</p>

    <input type="text" placeholder="Text input styled with attribute selector">

    <input type="password" placeholder="Password input not styled with attribute selector">

    <div>

        <p>This paragraph inside a div is styled with descendant selector.</p>

    </div>

    <ul>

        <li>First item (styled with child selector)</li>

        <li>Second item (styled with child selector)</li>

    </ul>

    <h2>Heading 2</h2>

    <p>This paragraph is adjacent to an h2 and styled with adjacent sibling selector.</p>

    <p>This paragraph is a general sibling of an h1 and styled with general sibling selector.</p>

</body>

</html>

**Output:**

**Experiment 3)**

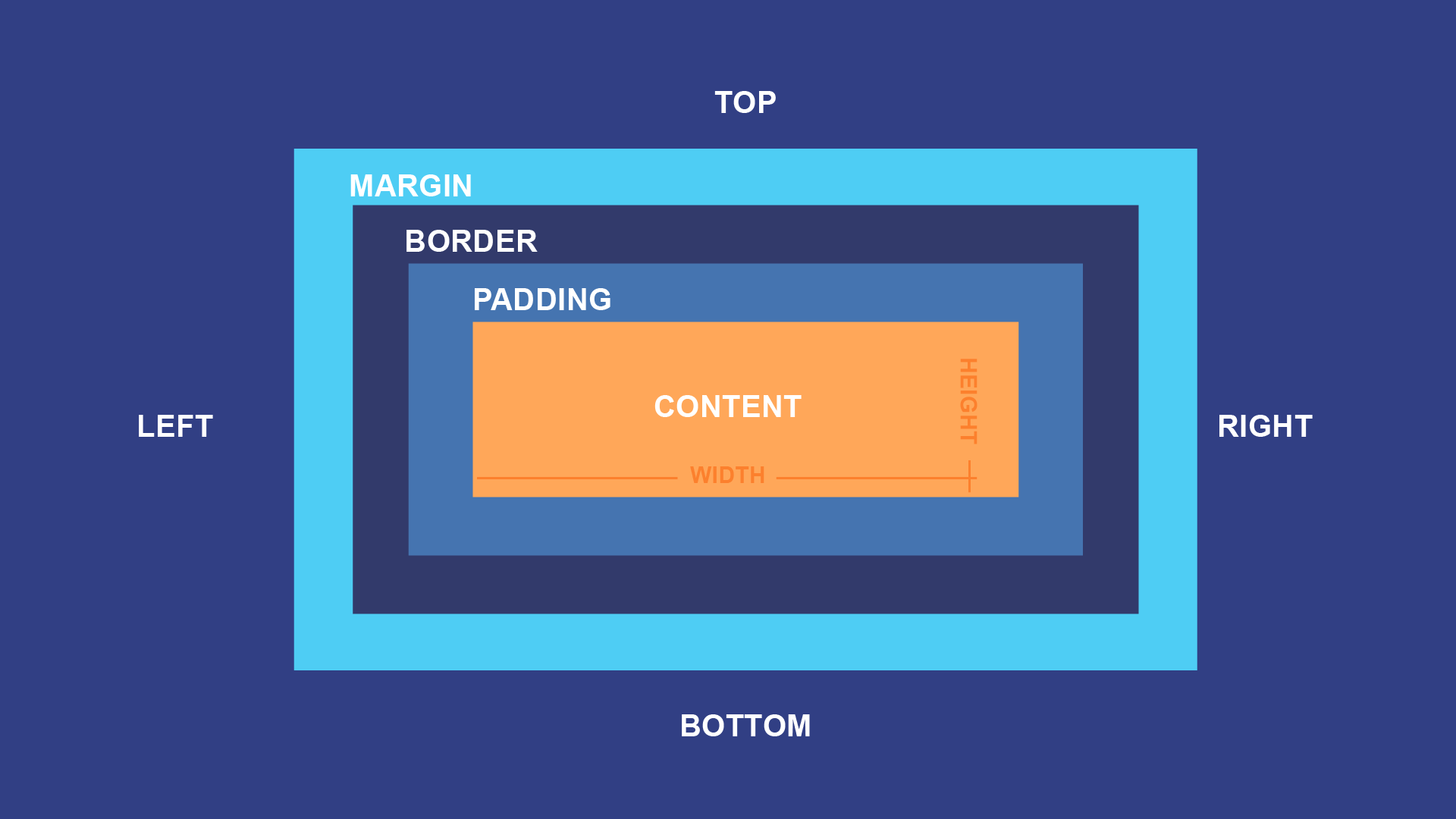
**Understanding the CSS Box Model.**

The CSS box model is a fundamental concept in web design that describes how elements are rendered and how their dimensions are calculated in a web page. It defines the structure of HTML elements as rectangular boxes. Each box consists of four main components, layered from the inside out:

1. Content: The innermost part where the actual content (text, images, etc.) is displayed.
2. Padding: The space between the content and the border.
3. Border: A line that surrounds the padding and content.
4. Margin: The outermost layer, creating space between the element and other elements.

Key points about the CSS box model:

* It applies to all HTML elements, though it's most noticeable with block-level elements.
* It helps in controlling layout and spacing of elements on a web page.
* The total space an element occupies is the sum of its content width/height, padding, border, and margin.
* The box-sizing property can alter how the total width and height of an element are calculated.

****

**Example that will help to understand the CSS box model.**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>CSS Box Model </title>

    <style>

        .box {

            width: 300px;

            height: 200px;

            padding: 20px;

            border: 5px solid #333;

            margin: 30px;

            background-color: #f0f0f0;

        }

    </style>

</head>

<body>

    <div class="box">

        This is the content of the box.

    </div>

</body>

</html>

**Output:**

**Experiment 4) Styling Fonts & texts in CSS.**

**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Text and Fonts Styling Example</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

            background-color: #f4f4f4;

            color: #333;

        }

        .heading {

            font-size: 32px;

            font-weight: bold;

            color: #2c3e50;

            text-align: center;

            margin-top: 20px;

        }

        .subheading {

            font-size: 24px;

            color: #16a085;

            text-align: center;

            margin: 10px 0;

        }

        .paragraph {

            font-size: 18px;

            line-height: 1.6;

            text-align: center;

            margin: 15px auto;

            max-width: 800px;

            padding: 0 20px;

        }

        .highlight {

            background-color: #f39c12;

            color: white;

            padding: 5px;

            border-radius: 3px;

        }

    </style>

</head>

<body>

    <h1 class="heading">Hello, I'm Kushal Poudel</h1>

    <h2 class="subheading">Flutter Developer at Tuki Soft Pvt Ltd</h2>

    <p class="paragraph">

        A passionate learner and tech enthusiast. I love to play football and am a gym freak. Styling text and fonts in CSS allows you to control how text appears on your web page.

        For example, the heading, subheading, and paragraph elements are styled differently to emphasize their importance.

    </p>

    <p class="paragraph">

        Here is a <span class="highlight">highlighted section</span> to draw attention to important content.

    </p>

</body>

</html>

**Output:**

**Experiment 5)**

**Styling border , Background and shadows in css.**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Border, Background, and Shadow Styling Example</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

            background-color: #e0e0e0;

        }

        .box {

            width: 300px;

            padding: 20px;

            margin: 20px auto;

            background-color: #ffffff;

            border: 5px solid #3498db;

            border-radius: 10px;

            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

            text-align: center;

        }

        .highlight {

            background-color: #f39c12;

            color: white;

            padding: 10px;

            border-radius: 5px;

        }

    </style>

</head>

<body>

    <div class="box">

        <h1>Hello, I'm Kushal Poudel</h1>

        <p>A passionate learner and tech enthusiast. I love to play football and am a gym freak.</p>

        <p class="highlight">This is a highlighted section with a different background.</p>

    </div>

</body>

</html>

**Output:**

**Experiment 6)**

**Using Float and clear properties in css**

**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Float and Clear Properties Example</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

            background-color: #f4f4f4;

        }

        .container {

            width: 80%;

            margin: 0 auto;

        }

        .box {

            width: 30%;

            padding: 20px;

            background-color: #ffffff;

            border: 2px solid #3498db;

            border-radius: 5px;

            margin: 10px;

            float: left;

        }

        .clear {

            clear: both;

        }

        .clearfix::after {

            content: "";

            display: table;

            clear: both;

        }

        .clearfix {

            background-color: #e0e0e0;

            padding: 20px;

            border: 2px solid #34495e;

        }

    </style>

</head>

<body>

    <div class="container">

        <div class="box">Box 1</div>

        <div class="box">Box 2</div>

        <div class="box">Box 3</div>

        <div class="clear"></div>

        <div class="clearfix">

            This is a clearfix container to ensure it contains its floated children.

        </div>

    </div>

</body>

</html>

**Output:**

**Experiment 7) Using Different types of positioning in CSS.(Static, Relative, Absolute, Fixed, Sticky)**

**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>CSS Positioning Example</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

            background-color: #f4f4f4;

            height: 2000px;

        }

        .static {

            background-color: #3498db;

            color: white;

            padding: 10px;

            margin: 10px;

            position: static;

        }

        .relative {

            background-color: #2ecc71;

            color: white;

            padding: 10px;

            margin: 10px;

            position: relative;

            top: 20px;

            left: 20px;

        }

        .absolute {

            background-color: #e74c3c;

            color: white;

            padding: 10px;

            position: absolute;

            top: 50px;

            left: 50px;

        }

        .fixed {

            background-color: #f39c12;

            color: white;

            padding: 10px;

            position: fixed;

            bottom: 20px;

            right: 20px;

        }

        .sticky {

            background-color: #9b59b6;

            color: white;

            padding: 10px;

            position: sticky;

            top: 0;

        }

        .container {

            height: 300px;

            border: 2px dashed #34495e;

            margin: 20px;

            position: relative;

        }

    </style>

</head>

<body>

    <div class="static">Static Position</div>

    <div class="relative">Relative Position</div>

    <div class="container">

        <div class="absolute">Absolute Position</div>

    </div>

    <div class="fixed">Fixed Position</div>

    <div class="sticky">Sticky Position</div>

</body>

</html>

**Output:**

**Experiment 8)**

**Using CSS media queries for responsive web design.**

**Code:**

**Index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Responsive Design Example</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <header>

        <h1>Hello, I'm Kushal Poudel</h1>

        <h2>Flutter Developer at Tuki Soft Pvt Ltd</h2>

    </header>

    <main>

        <section class="container">

            <h2>Welcome!</h2>

            <p>A passionate learner and tech enthusiast. I love to play football and am a gym freak. This is a simple example of responsive design using CSS media queries.</p>

        </section>

        <section class="box">

            <p>Resize the browser window to see the responsive design in action.</p>

        </section>

    </main>

    <footer>

        <p>&copy; 2024 Kushal Poudel</p>

    </footer>

</body>

</html>

**Styles.css:**

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    background-color: #f4f4f4;

}

header {

    background-color: #3498db;

    color: white;

    padding: 20px;

    text-align: center;

}

main {

    padding: 20px;

}

.container {

    background-color: #ffffff;

    border: 2px solid #3498db;

    border-radius: 5px;

    padding: 20px;

    margin: 10px;

}

.box {

    background-color: #e74c3c;

    color: white;

    padding: 20px;

    margin: 10px 0;

    border-radius: 5px;

}

footer {

    background-color: #2ecc71;

    color: white;

    padding: 10px;

    text-align: center;

}

@media (max-width: 768px) {

    .container, .box {

        padding: 15px;

        margin: 5px;

    }

}

@media (max-width: 480px) {

    header {

        font-size: 18px;

        padding: 10px;

    }

    .container, .box {

        padding: 10px;

        margin: 5px;

        font-size: 14px;

    }

    footer {

        font-size: 14px;

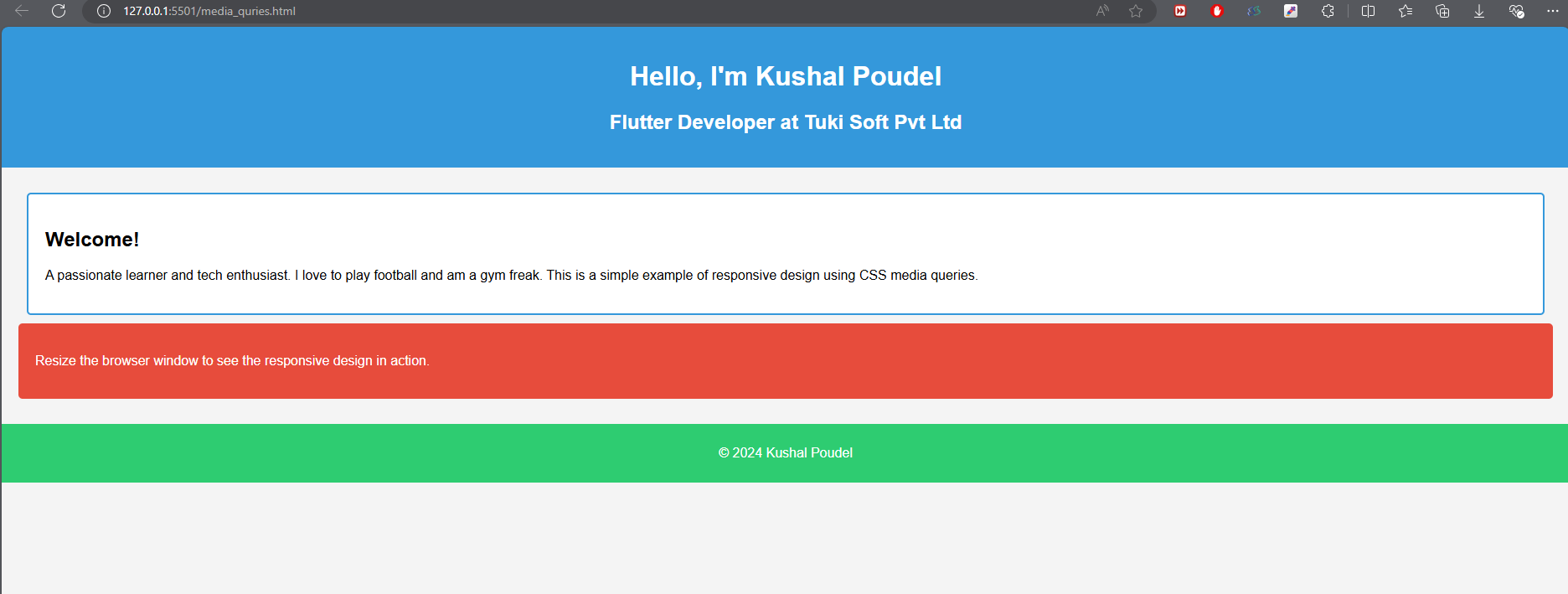
        padding: 5px;

    }

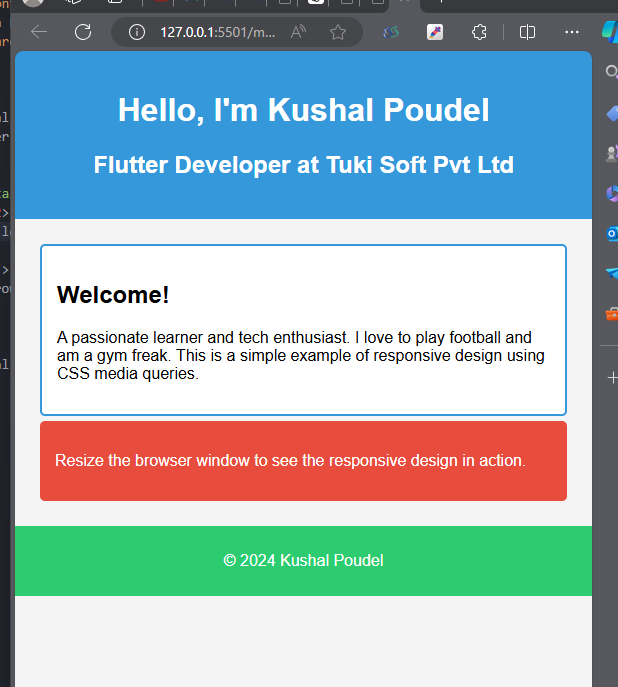
}

**Output:**

**BigScreen:**

****

**Small Screen:**

****

**Experiment 9)**

**Creating a clone of a popular website.**

**Code:**

**Output:**

**Conclusion:**

Through the provided CSS examples, we've covered essential styling techniques and layout principles. You’ve learned to apply CSS via inline, internal, and external methods and utilize various selectors to target HTML elements effectively. We explored the CSS box model, which defines how elements are sized and spaced. Styling borders, backgrounds, and shadows adds depth and visual appeal. We examined float and clear properties for layout management and different positioning methods like static, relative, absolute, fixed, and sticky. Finally, using media queries, you can create responsive designs that adapt to different screen sizes, ensuring a consistent user experience across devices.