1. Interactive Dropdown Menu

Description: Create a dropdown menu that opens on click.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Dropdown Menu</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
        .dropdown { position: relative; display: inline-block; }
        .dropdown-content { display: none; position: absolute; background-
color: #f9f9f9; min-width: 160px; box-shadow: 0px 8px 16px 0px
rgba(0,0,0,0.2); z-index: 1; }
        .dropdown-content a { color: black; padding: 12px 16px; text-
decoration: none; display: block; }
        .dropdown-content a:hover { background-color: #f1f1f1; }
        .show { display: block; }
    </style>
</head>
<body>
    <div class="dropdown">
        <button onclick="toggleDropdown()">Dropdown</button>
        <div id="myDropdown" class="dropdown-content">
            <a href="#">Link 1</a>
            <a href="#">Link 2</a>
            <a href="#">Link 3</a>
        </div>
    </div>

    <script>
        function toggleDropdown() {

document.getElementById('myDropdown').classList.toggle('show');
        }
        window.onclick = function(event) {
            if (!event.target.matches('button')) {
```

```
            var dropdowns =
document.getElementsByClassName("dropdown-content");
            for (var i = 0; i < dropdowns.length; i++) {
                var openDropdown = dropdowns[i];
                if (openDropdown.classList.contains('show')) {
                    openDropdown.classList.remove('show');
                }
            }
        }
    }
  </script>
</body>
</html>
```

2. Image Slider

Description: Create a simple image slider with previous and next buttons.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Image Slider</title>
  <style>
    .slider { position: relative; max-width: 600px; margin: auto; overflow:
hidden; }
    .slides { display: flex; transition: transform 0.5s ease; }
    .slides img { width: 100%; }
    .prev, .next { position: absolute; top: 50%; width: auto; padding:
16px; color: white; background-color: rgba(0,0,0,0.5); cursor: pointer; }
    .prev { left: 0; }
    .next { right: 0; }
  </style>
</head>
<body>
  <div class="slider">
    <div class="slides">
      <img src="image1.jpg" alt="Slide 1">
```

```
            <img src="image2.jpg" alt="Slide 2">
            <img src="image3.jpg" alt="Slide 3">
        </div>
        <a class="prev" onclick="prevSlide()">&#10094;</a>
        <a class="next" onclick="nextSlide()">&#10095;</a>
    </div>

    <script>
        let index = 0;
        function showSlide() {
            const slides = document.querySelector('.slides');
            const totalSlides = document.querySelectorAll('.slides
img').length;
            if (index >= totalSlides) index = 0;
            if (index < 0) index = totalSlides - 1;
            slides.style.transform = `translateX(${-index * 100}%)`;
        }
        function nextSlide() {
            index++;
            showSlide();
        }
        function prevSlide() {
            index--;
            showSlide();
        }
        showSlide();
    </script>
</body>
</html>
```

3. To-Do List

Description: Create a simple to-do list application with add and remove functionalities.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
```

```
    <title>To-Do List</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
        .container { max-width: 400px; margin: auto; padding: 2rem; }
        ul { list-style-type: none; padding: 0; }
        li { padding: 10px; background: #f4f4f4; margin: 5px 0; border-
radius: 5px; }
        button { background: #4CAF50; color: white; border: none; padding:
10px; cursor: pointer; border-radius: 5px; }
        button:hover { background: #45a049; }
    </style>
</head>
<body>
    <div class="container">
        <h1>To-Do List</h1>
        <input type="text" id="taskInput" placeholder="Add a task">
        <button onclick="addTask()">Add Task</button>
        <ul id="taskList"></ul>
    </div>

    <script>
        function addTask() {
            const input = document.getElementById('taskInput');
            const taskText = input.value.trim();
            if (taskText === '') return;

            const taskList = document.getElementById('taskList');
            const li = document.createElement('li');
            li.textContent = taskText;
            li.onclick = function() {
                this.remove();
            };
            taskList.appendChild(li);
            input.value = '';
        }
    </script>
</body>
</html>
```

4. Form Validation

Description: Validate a form with required fields and simple error messages.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Form Validation</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; display:
flex; justify-content: center; align-items: center; height: 100vh;
background: #f4f4f4; }
        .form-container { max-width: 400px; width: 100%; padding: 2rem;
background: #fff; border: 1px solid #ddd; border-radius: 5px; }
        input { width: 100%; padding: 10px; margin: 5px 0; border: 1px solid
#ddd; border-radius: 3px; }
        button { background: #4CAF50; color: white; border: none; padding:
10px; cursor: pointer; border-radius: 3px; }
        button:hover { background: #45a049; }
        .error { color: red; font-size: 0.875rem; }
    </style>
</head>
<body>
    <div class="form-container">
        <h1>Sign Up</h1>
        <form id="myForm">
            <input type="text" id="name" placeholder="Name" required>
            <input type="email" id="email" placeholder="Email" required>
            <button type="submit">Submit</button>
            <div id="error" class="error"></div>
        </form>
    </div>

    <script>
        document.getElementById('myForm').onsubmit = function(event) {
            event.preventDefault();
            const name = document.getElementById('name').value.trim();
            const email = document.getElementById('email').value.trim();
            const error = document.getElementById('error');
            error.textContent = '';
```

```
            if (name === '' || email === '') {
                error.textContent = 'All fields are required!';
                return;
            }

            alert('Form submitted successfully!');
        }
    </script>
</body>
</html>
```

5. Interactive Tab Component

Description: Create a tabbed interface with content switching.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Tab Component</title>
    <style>
        .tab { display: flex; cursor: pointer; margin: 0; padding: 0; list-style-
type: none; background: #f4f4f4; }
        .tab button { background: #ddd; border: none; padding: 10px;
cursor: pointer; }
        .tab button.active { background: #4CAF50; color: white; }
        .tab-content { display: none; padding: 10px; background: #fff;
border: 1px solid #ddd; }
        .tab-content.active { display: block; }
    </style>
</head>
<body>
    <div>
        <ul class="tab">
            <li><button class="tablink" onclick="openTab('home')">Home</
button></li>
            <li><button class="tablink"
onclick="openTab('services')">Services</button></li>
```

```
            <li><button class="tablink"
onclick="openTab('contact')">Contact</button></li>
        </ul>
        <div id="home" class="tab-content">
            <h2>Home</h2>
            <p>Welcome to the home page!</p>
        </div>
        <div id="services" class="tab-content">
            <h2>Services</h2>
            <p>Learn more about our services.</p>
        </div>
        <div id="contact" class="tab-content">
            <h2>Contact</h2>
            <p>Get in touch with us.</p>
        </div>
    </div>

    <script>
        function openTab(tabId) {
            const tabs = document.querySelectorAll('.tablink');
            const contents = document.querySelectorAll('.tab-content');

            tabs.forEach(tab => tab.classList.remove('active'));
            contents.forEach(content => content.classList.remove('active'));

            document.querySelector(`.tablink[onclick="openTab('$
{tabId}')"]`).classList.add('active');
            document.getElementById(tabId).classList.add('active');
        }
        openTab('home');
    </script>
</body>
</html>
```

6. Countdown Timer

Description: Create a simple countdown timer.

Code Snippet:

html

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

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Countdown Timer</title>
    <style>
        body { font-family: Arial, sans-serif; display: flex; justify-content:
center; align-items: center; height: 100vh; margin: 0; }
        .timer { font-size: 2rem; }
    </style>
</head>
<body>
    <div class="timer" id="timer">00:00:00</div>

    <script>
        function startTimer(duration) {
            let timer = document.getElementById('timer');
            let endTime = Date.now() + duration * 1000;

            function updateTimer() {
                let now = Date.now();
                let timeLeft = Math.max(0, endTime - now);
                let hours = Math.floor(timeLeft / (1000 * 60 * 60));
                let minutes = Math.floor((timeLeft % (1000 * 60 * 60)) / (1000 *
60));
                let seconds = Math.floor((timeLeft % (1000 * 60)) / 1000);

                timer.textContent = `${String(hours).padStart(2, '0')}:$
{String(minutes).padStart(2, '0')}:${String(seconds).padStart(2, '0')}`;

                if (timeLeft > 0) {
                    requestAnimationFrame(updateTimer);
                }
            }
            updateTimer();
        }
        startTimer(60); // Countdown for 60 seconds
    </script>
</body>
</html>
```

7. Simple Calculator

Description: Create a basic calculator with addition, subtraction,

multiplication, and division.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Simple Calculator</title>
    <style>
        .calculator { max-width: 200px; margin: auto; padding: 1rem;
border: 1px solid #ddd; border-radius: 5px; background: #fff; }
        .calculator input { width: 100%; padding: 10px; margin-bottom:
10px; text-align: right; }
        .calculator button { width: 23%; padding: 10px; margin: 1%;
background: #4CAF50; color: white; border: none; border-radius: 5px;
cursor: pointer; }
        .calculator button:hover { background: #45a049; }
    </style>
</head>
<body>
    <div class="calculator">
        <input type="text" id="display" readonly>
        <div>
            <button onclick="appendNumber('7')">7</button>
            <button onclick="appendNumber('8')">8</button>
            <button onclick="appendNumber('9')">9</button>
            <button onclick="setOperation('/')">/</button>
        </div>
        <div>
            <button onclick="appendNumber('4')">4</button>
            <button onclick="appendNumber('5')">5</button>
            <button onclick="appendNumber('6')">6</button>
            <button onclick="setOperation('*')">*</button>
        </div>
        <div>
            <button onclick="appendNumber('1')">1</button>
            <button onclick="appendNumber('2')">2</button>
            <button onclick="appendNumber('3')">3</button>
```

```
            <button onclick="setOperation('-')">-</button>
        </div>
        <div>
            <button onclick="appendNumber('0')">0</button>
            <button onclick="appendNumber('.')">.</button>
            <button onclick="calculate()">=</button>
            <button onclick="setOperation('+')">+</button>
        </div>
    </div>

    <script>
        let display = document.getElementById('display');
        let currentOperation = '';
        let operator = '';
        let firstNumber = '';

        function appendNumber(number) {
            display.value += number;
        }

        function setOperation(op) {
            firstNumber = display.value;
            operator = op;
            display.value = '';
        }

        function calculate() {
            let secondNumber = display.value;
            let result;
            switch (operator) {
                case '+': result = parseFloat(firstNumber) +
parseFloat(secondNumber); break;
                case '-': result = parseFloat(firstNumber) -
parseFloat(secondNumber); break;
                case '*': result = parseFloat(firstNumber) *
parseFloat(secondNumber); break;
                case '/': result = parseFloat(firstNumber) /
parseFloat(secondNumber); break;
            }
            display.value = result;
            operator = '';
        }
    </script>
```

```
</body>
</html>
```

8. Simple Modal

Description: Create a modal that opens and closes with buttons.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Simple Modal</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
        .modal { display: none; position: fixed; z-index: 1; left: 0; top: 0;
width: 100%; height: 100%; overflow: auto; background-color: rgb(0,0,0);
background-color: rgba(0,0,0,0.4); }
        .modal-content { background-color: #fefefe; margin: 15% auto;
padding: 20px; border: 1px solid #888; width: 80%; }
        .close { color: #aaa; float: right; font-size: 28px; font-weight: bold; }
        .close:hover, .close:focus { color: black; text-decoration: none;
cursor: pointer; }
    </style>
</head>
<body>
    <button id="openModal">Open Modal</button>

    <div id="myModal" class="modal">
        <div class="modal-content">
            <span class="close">&times;</span>
            <p>This is a simple modal!</p>
        </div>
    </div>

    <script>
        const modal = document.getElementById('myModal');
        const btn = document.getElementById('openModal');
        const span = document.getElementsByClassName('close')[0];
```

```
        btn.onclick = function() {
            modal.style.display = 'block';
        }

        span.onclick = function() {
            modal.style.display = 'none';
        }

        window.onclick = function(event) {
            if (event.target === modal) {
                modal.style.display = 'none';
            }
        }
    </script>
</body>
</html>
```

9. Image Hover Effect

Description: Apply a hover effect to an image to change its appearance.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Image Hover Effect</title>
    <style>
        .container { max-width: 300px; margin: auto; }
        .image { width: 100%; transition: transform 0.3s ease; }
        .image:hover { transform: scale(1.1); }
    </style>
</head>
<body>
    <div class="container">
        <img src="image.jpg" alt="Image" class="image">
    </div>
</body>
</html>
```

10. Sticky Header

Description: Create a header that stays at the top of the page when scrolling.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Sticky Header</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 0; padding: 0; }
        .header { background: #333; color: white; padding: 10px; position:
-webkit-sticky; position: sticky; top: 0; }
        .content { padding: 20px; }
    </style>
</head>
<body>
    <div class="header">
        <h1>Sticky Header</h1>
    </div>
    <div class="content">
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer
nec odio. Praesent libero. Sed cursus ante dapibus diam.</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer
nec odio. Praesent libero. Sed cursus ante dapibus diam.</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer
nec odio. Praesent libero. Sed cursus ante dapibus diam.</p>
        <!-- Add more content to test scrolling -->
    </div>
</body>
</html>
```

11. Collapsible Section

Description: Create a section that can be expanded or collapsed.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Collapsible Section</title>
    <style>
        .collapsible { cursor: pointer; padding: 10px; text-align: left;
background: #f4f4f4; border: 1px solid #ddd; }
        .content { display: none; padding: 10px; background: #eee; border:
1px solid #ddd; }
    </style>
</head>
<body>
    <div class="collapsible">Click to Expand</div>
    <div class="content">
        <p>This is a collapsible content section.</p>
    </div>

    <script>
        document.querySelector('.collapsible').onclick = function() {
            const content = document.querySelector('.content');
            content.style.display = content.style.display === 'block' ? 'none' :
'block';
        }
    </script>
</body>
</html>
```

12. Responsive Navigation Menu

Description: Create a responsive navigation menu that changes layout on smaller screens.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Responsive Navigation Menu</title>
    <style>
        body { margin: 0; padding: 0; }
        .navbar { display: flex; justify-content: space-between; background:
#333; padding: 10px; }
        .navbar a { color: white; text-decoration: none; padding: 10px; }
        .menu { display: flex; }
        .menu a { padding: 0; margin: 0 5px; }
        @media (max-width: 600px) {
            .menu { flex-direction: column; display: none; }
            .menu.active { display: flex; }
        }
        .menu-toggle { display: none; }
        @media (max-width: 600px) {
            .menu-toggle { display: block; cursor: pointer; }
        }
    </style>
</head>
<body>
    <div class="navbar">
        <div class="menu-toggle" onclick="toggleMenu()">Menu</div>
        <div class="menu" id="menu">
            <a href="#">Home</a>
            <a href="#">About</a>
            <a href="#">Services</a>
            <a href="#">Contact</a>
        </div>
    </div>

    <script>
        function toggleMenu() {
            const menu = document.getElementById('menu');
            menu.classList.toggle('active');
        }
    </script>
</body>
</html>
```

13. Progress Bar

Description: Create a progress bar that fills up on button click.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Progress Bar</title>
    <style>
        .progress-container { width: 100%; background: #f3f3f3; border-
radius: 5px; }
        .progress-bar { width: 0; height: 30px; background: #4CAF50; text-
align: center; color: white; line-height: 30px; border-radius: 5px; }
        button { margin-top: 10px; padding: 10px 20px; background:
#4CAF50; color: white; border: none; cursor: pointer; }
        button:hover { background: #45a049; }
    </style>
</head>
<body>
    <div class="progress-container">
        <div class="progress-bar" id="progressBar">0%</div>
    </div>
    <button onclick="updateProgress()">Increase Progress</button>

    <script>
        let progress = 0;
        function updateProgress() {
            progress += 10;
            if (progress > 100) progress = 100;
            document.getElementById('progressBar').style.width = progress
+ '%';
            document.getElementById('progressBar').textContent = progress
+ '%';
        }
    </script>
</body>
</html>
```

14. Simple FAQ Accordion

Description: Create a simple FAQ accordion component.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>FAQ Accordion</title>
    <style>
        .accordion { background: #eee; border: 1px solid #ddd; padding:
10px; cursor: pointer; }
        .panel { display: none; padding: 10px; border: 1px solid #ddd;
border-top: none; background: #f9f9f9; }
    </style>
</head>
<body>
    <div class="accordion">Question 1</div>
    <div class="panel">
        <p>Answer to question 1.</p>
    </div>

    <div class="accordion">Question 2</div>
    <div class="panel">
        <p>Answer to question 2.</p>
    </div>

    <script>
        const accordions = document.querySelectorAll('.accordion');
        accordions.forEach(acc => acc.onclick = function() {
            const panel = this.nextElementSibling;
            panel.style.display = panel.style.display === 'block' ? 'none' :
'block';
        });
    </script>
</body>
</html>
```

15. Simple Slider

Description: Create a simple text slider.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Text Slider</title>
    <style>
        .slider { position: relative; max-width: 300px; margin: auto; overflow:
hidden; }
        .slides { display: flex; transition: transform 0.5s ease; }
        .slides div { min-width: 100%; box-sizing: border-box; }
        .prev, .next { position: absolute; top: 50%; width: auto; padding:
10px; color: white; background-color: rgba(0,0,0,0.5); cursor: pointer; }
        .prev { left: 0; }
        .next { right: 0; }
    </style>
</head>
<body>
    <div class="slider">
        <div class="slides">
            <div>Slide 1</div>
            <div>Slide 2</div>
            <div>Slide 3</div>
        </div>
        <a class="prev" onclick="prevSlide()">&#10094;</a>
        <a class="next" onclick="nextSlide()">&#10095;</a>
    </div>

    <script>
        let index = 0;
        function showSlide() {
            const slides = document.querySelector('.slides');
            const totalSlides = document.querySelectorAll('.slides
div').length;
            if (index >= totalSlides) index = 0;
            if (index < 0) index = totalSlides - 1;
            slides.style.transform = `translateX(${-index * 100}%)`;
        }
```

```
        function nextSlide() {
            index++;
            showSlide();
        }
        function prevSlide() {
            index--;
            showSlide();
        }
        setInterval(nextSlide, 3000); // Auto-slide every 3 seconds
    </script>
</body>
</html>
```

16. Toggle Dark Mode

Description: Create a button to toggle between light and dark themes.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Toggle Dark Mode</title>
    <style>
        body { font-family: Arial, sans-serif; transition: background 0.3s; }
        .dark-mode { background: #333; color: #fff; }
        .button { padding: 10px 20px; cursor: pointer; }
    </style>
</head>
<body>
    <button class="button" onclick="toggleDarkMode()">Toggle Dark Mode</button>

    <script>
        function toggleDarkMode() {
            document.body.classList.toggle('dark-mode');
        }
    </script>
</body>
```

</html>

17. Image Carousel

Description: Create a simple image carousel that cycles through images.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Image Carousel</title>
    <style>
        .carousel { position: relative; max-width: 600px; margin: auto;
overflow: hidden; }
        .carousel img { width: 100%; }
        .prev, .next { position: absolute; top: 50%; width: auto; padding:
10px; color: white; background: rgba(0,0,0,0.5); cursor: pointer; }
        .prev { left: 0; }
        .next { right: 0; }
    </style>
</head>
<body>
    <div class="carousel">
        <img src="image1.jpg" alt="Image 1">
        <img src="image2.jpg" alt="Image 2">
        <img src="image3.jpg" alt="Image 3">
        <a class="prev" onclick="prevSlide()">&#10094;</a>
        <a class="next" onclick="nextSlide()">&#10095;</a>
    </div>

    <script>
        let slideIndex = 0;
        function showSlides() {
            const slides = document.querySelectorAll('.carousel img');
            slides.forEach((slide, index) => slide.style.display = (index ===
slideIndex ? 'block' : 'none'));
        }
        function nextSlide() {
```

```
            slideIndex = (slideIndex + 1) %
document.querySelectorAll('.carousel img').length;
            showSlides();
        }
        function prevSlide() {
            slideIndex = (slideIndex - 1 +
document.querySelectorAll('.carousel img').length) %
document.querySelectorAll('.carousel img').length;
            showSlides();
        }
        showSlides();
        setInterval(nextSlide, 3000); // Auto-slide every 3 seconds
    </script>
</body>
</html>
```

18. Simple To-Do List

Description: Create a simple to-do list where users can add and remove tasks.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>To-Do List</title>
    <style>
        body { font-family: Arial, sans-serif; }
        .todo-list { max-width: 400px; margin: auto; padding: 10px; border:
1px solid #ddd; border-radius: 5px; }
        .todo-item { display: flex; justify-content: space-between; padding:
5px; border-bottom: 1px solid #ddd; }
        button { background: red; color: white; border: none; padding: 5px
10px; cursor: pointer; }
    </style>
</head>
<body>
    <div class="todo-list">
        <input type="text" id="taskInput" placeholder="Add a task">
```

```
        <button onclick="addTask()">Add</button>
        <div id="tasks"></div>
    </div>

    <script>
        function addTask() {
            const taskInput = document.getElementById('taskInput');
            const taskText = taskInput.value;
            if (taskText === '') return;

            const taskDiv = document.createElement('div');
            taskDiv.className = 'todo-item';
            taskDiv.innerHTML = `${taskText} <button
onclick="removeTask(this)">Remove</button>`;
            document.getElementById('tasks').appendChild(taskDiv);
            taskInput.value = '';
        }

        function removeTask(button) {
            button.parentElement.remove();
        }
    </script>
</body>
</html>
```

19. Contact Form with Validation

Description: Create a contact form with basic validation.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Contact Form</title>
    <style>
        body { font-family: Arial, sans-serif; }
        .form-container { max-width: 400px; margin: auto; padding: 20px;
border: 1px solid #ddd; border-radius: 5px; }
        .form-container input, .form-container textarea { width: 100%;
```

```
padding: 10px; margin: 5px 0; }
        .form-container button { padding: 10px 20px; background:
#4CAF50; color: white; border: none; cursor: pointer; }
        .form-container button:hover { background: #45a049; }
    </style>
</head>
<body>
    <div class="form-container">
        <h2>Contact Us</h2>
        <form onsubmit="return validateForm()">
            <input type="text" id="name" placeholder="Name" required>
            <input type="email" id="email" placeholder="Email" required>
            <textarea id="message" rows="4" placeholder="Message"
required></textarea>
            <button type="submit">Send</button>
        </form>
    </div>

    <script>
        function validateForm() {
            const name = document.getElementById('name').value;
            const email = document.getElementById('email').value;
            const message = document.getElementById('message').value;

            if (name === '' || email === '' || message === '') {
                alert('All fields are required!');
                return false;
            }

            if (!validateEmail(email)) {
                alert('Please enter a valid email address.');
                return false;
            }

            alert('Form submitted successfully!');
            return false; // Prevent form submission for demo purposes
        }

        function validateEmail(email) {
            const re = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
            return re.test(email);
        }
    </script>
```

```
</body>
</html>
```

20. Sticky Footer

Description: Create a footer that stays at the bottom of the page.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Sticky Footer</title>
    <style>
        html, body { height: 100%; margin: 0; }
        .container { min-height: 100%; display: flex; flex-direction: column; }
        .content { flex: 1; }
        .footer { background: #333; color: white; text-align: center; padding:
10px; }
    </style>
</head>
<body>
    <div class="container">
        <div class="content">
            <h1>Content Goes Here</h1>
            <p>This is some content above the footer.</p>
        </div>
        <div class="footer">
            <p>Sticky Footer</p>
        </div>
    </div>
</body>
</html>
```

21. Simple Tooltip

Description: Create a tooltip that appears when hovering over an element.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Tooltip</title>
    <style>
        .tooltip { position: relative; display: inline-block; }
        .tooltip .tooltiptext { visibility: hidden; width: 120px; background-
color: #555; color: #fff; text-align: center; border-radius: 5px; padding:
5px 0; position: absolute; z-index: 1; bottom: 125%; left: 50%; margin-
left: -60px; opacity: 0; transition: opacity 0.3s; }
        .tooltip:hover .tooltiptext { visibility: visible; opacity: 1; }
    </style>
</head>
<body>
    <div class="tooltip">Hover over me
        <span class="tooltiptext">Tooltip text</span>
    </div>
</body>
</html>
```

22. Card Layout

Description: Create a card layout with image, title, and description.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Card Layout</title>
    <style>
        .card { max-width: 300px; border: 1px solid #ddd; border-radius:
5px; overflow: hidden; margin: 20px auto; }
        .card img { width: 100%; }
        .card-content { padding: 20px; }
```

```
        .card-title { font-size: 1.5em; margin: 0; }
        .card-description { font-size: 1em; color: #555; }
    </style>
</head>
<body>
    <div class="card">
        <img src="image.jpg" alt="Card Image">
        <div class="card-content">
            <h2 class="card-title">Card Title</h2>
            <p class="card-description">This is a brief description of the card
content.</p>
        </div>
    </div>
</body>
</html>
```

23. Image Gallery

Description: Create a responsive image gallery.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Image Gallery</title>
    <style>
        .gallery { display: flex; flex-wrap: wrap; gap: 10px; }
        .gallery img { width: 100%; height: auto; max-width: calc(33% -
10px); }
        @media (max-width: 768px) {
            .gallery img { max-width: calc(50% - 10px); }
        }
        @media (max-width: 480px) {
            .gallery img { max-width: 100%; }
        }
    </style>
</head>
<body>
    <div class="gallery">
```

```
        <img src="image1.jpg" alt="Image 1">
        <img src="image2.jpg" alt="Image 2">
        <img src="image3.jpg" alt="Image 3">
        <img src="image4.jpg" alt="Image 4">
        <img src="image5.jpg" alt="Image 5">
        <img src="image6.jpg" alt="Image 6">
    </div>
</body>
</html>
```

24. Simple Countdown Timer

Description: Create a countdown timer that counts down to a specific date.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Countdown Timer</title>
    <style>
        .countdown { font-size: 2em; text-align: center; }
    </style>
</head>
<body>
    <div class="countdown" id="timer"></div>

    <script>
        const countdownDate = new Date("Dec 31, 2024 23:59:59").getTime();
        const timer = document.getElementById('timer');

        function updateCountdown() {
            const now = new Date().getTime();
            const distance = countdownDate - now;

            const days = Math.floor(distance / (1000 * 60 * 60 * 24));
            const hours = Math.floor((distance % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
```

```
        const minutes = Math.floor((distance % (1000 * 60 * 60)) / (1000
* 60));
        const seconds = Math.floor((distance % (1000 * 60)) / 1000);

        timer.innerHTML = `${days}d ${hours}h ${minutes}m ${seconds}
s`;

        if (distance < 0) {
            clearInterval(countdownInterval);
            timer.innerHTML = "EXPIRED";
        }
    }

    const countdownInterval = setInterval(updateCountdown, 1000);
    updateCountdown();
  </script>
</body>
</html>
```

25. Simple Calculator

Description: Create a basic calculator with HTML, CSS, and JavaScript.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Simple Calculator</title>
  <style>
    .calculator { max-width: 200px; margin: auto; padding: 10px; border:
1px solid #ddd; border-radius: 5px; }
    .calculator input { width: 100%; padding: 10px; margin: 5px 0; font-
size: 1.5em; }
    .calculator button { width: 25%; padding: 10px; font-size: 1.2em; }
    .buttons { display: flex; flex-wrap: wrap; }
    .buttons button { flex: 1; }
  </style>
</head>
```

```
<body>
    <div class="calculator">
        <input type="text" id="display" disabled>
        <div class="buttons">
            <button onclick="clearDisplay()">C</button>
            <button onclick="appendToDisplay('/')">/</button>
            <button onclick="appendToDisplay('*')">*</button>
            <button onclick="appendToDisplay('-')">-</button>
            <button onclick="appendToDisplay('7')">7</button>
            <button onclick="appendToDisplay('8')">8</button>
            <button onclick="appendToDisplay('9')">9</button>
            <button onclick="appendToDisplay('+')">+</button>
            <button onclick="appendToDisplay('4')">4</button>
            <button onclick="appendToDisplay('5')">5</button>
            <button onclick="appendToDisplay('6')">6</button>
            <button onclick="calculate()">=</button>
            <button onclick="appendToDisplay('1')">1</button>
            <button onclick="appendToDisplay('2')">2</button>
            <button onclick="appendToDisplay('3')">3</button>
            <button onclick="appendToDisplay('0')">0</button>
        </div>
    </div>

    <script>
        function clearDisplay() {
            document.getElementById('display').value = '';
        }
        function appendToDisplay(value) {
            document.getElementById('display').value += value;
        }
        function calculate() {
            try {
                document.getElementById('display').value =
eval(document.getElementById('display').value);
            } catch (error) {
                document.getElementById('display').value = 'Error';
            }
        }
    </script>
</body>
</html>
```

26. Hover Effect on Buttons

Description: Create buttons with hover effects.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Button Hover Effect</title>
    <style>
        .button { padding: 10px 20px; font-size: 1em; border: none; cursor:
pointer; transition: background 0.3s, transform 0.3s; }
        .button:hover { background: #4CAF50; color: white; transform:
scale(1.1); }
    </style>
</head>
<body>
    <button class="button">Hover me!</button>
</body>
</html>
```

27. Form with Input Masks

Description: Create a form with input masks for phone numbers.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Input Mask</title>
    <style>
        body { font-family: Arial, sans-serif; }
        .form-container { max-width: 400px; margin: auto; padding: 20px;
border: 1px solid #ddd; border-radius: 5px; }
        .form-container input { width: 100%; padding: 10px; margin: 5px 0; }
```

```
        .form-container button { padding: 10px 20px; background:
#4CAF50; color: white; border: none; cursor: pointer; }
        .form-container button:hover { background: #45a049; }
    </style>
</head>
<body>
    <div class="form-container">
        <h2>Contact Form</h2>
        <form>
            <input type="text" id="phone" placeholder="Phone Number"
oninput="maskPhoneNumber(this)">
            <button type="submit">Submit</button>
        </form>
    </div>

    <script>
        function maskPhoneNumber(input) {
            let value = input.value.replace(/\D/g, '');
            if (value.length > 3 && value.length <= 6) {
                value = `${value.slice(0, 3)}-${value.slice(3)}`;
            } else if (value.length > 6) {
                value = `${value.slice(0, 3)}-${value.slice(3, 6)}-${value.slice(6,
10)}`;
            }
            input.value = value;
        }
    </script>
</body>
</html>
```

28. Animated Progress Bar

Description: Create a progress bar with animation.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Progress Bar</title>
```

```
    <style>
        .progress-container { width: 100%; background: #ddd; border-
radius: 5px; overflow: hidden; }
        .progress-bar { width: 0; height: 30px; background: #4CAF50; text-
align: center; color: white; line-height: 30px; transition: width 0.5s; }
    </style>
</head>
<body>
    <div class="progress-container">
        <div class="progress-bar" id="progressBar">0%</div>
    </div>
    <button onclick="increaseProgress()">Increase Progress</button>

    <script>
        let width = 0;
        function increaseProgress() {
            if (width < 100) {
                width += 10;
                document.getElementById('progressBar').style.width = `$
{width}%`;
                document.getElementById('progressBar').textContent = `$
{width}%`;
            }
        }
    </script>
</body>
</html>
```

29. Dropdown Menu

Description: Create a dropdown menu that appears on hover.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Dropdown Menu</title>
    <style>
        .dropdown { position: relative; display: inline-block; }
```

```
        .dropdown-content { display: none; position: absolute; background-color: #f9f9f9; min-width: 160px; box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2); z-index: 1; }
        .dropdown-content a { color: black; padding: 12px 16px; text-decoration: none; display: block; }
        .dropdown-content a:hover { background-color: #f1f1f1; }
        .dropdown:hover .dropdown-content { display: block; }
    </style>
</head>
<body>
    <div class="dropdown">
        <button>Dropdown</button>
        <div class="dropdown-content">
            <a href="#">Link 1</a>
            <a href="#">Link 2</a>
            <a href="#">Link 3</a>
        </div>
    </div>
</body>
</html>
```

30. Date Picker

Description: Create a date picker with JavaScript.

Code Snippet:

html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Date Picker</title>
    <style>
        body { font-family: Arial, sans-serif; }
        .date-picker { max-width: 200px; margin: auto; padding: 10px; }
        .date-picker input { width: 100%; padding: 10px; }
    </style>
</head>
<body>
    <div class="date-picker">
        <label for="date">Select a date:</label>
```

```
        <input type="date" id="date">
    </div>
</body>
</html>
```