

Day 4: Advanced CSS Styling

Knowledge:

- **CSS Transitions:**

- Understand the concept of CSS transitions for smooth state changes.
- Learn about properties like `transition-property`, `transition-duration`, and `transition-timing-function`.

- **CSS Animations:**

- Explore CSS animations for creating dynamic effects.
- Understand properties like `@keyframes`, `animation-name`, `animation-duration`, and `animation-timing-function`.

Task:

1. **Transitions:**

- Implement CSS transitions to create smooth effects on hover or other state changes.
- Apply transitions to elements such as links, buttons, or background colors.

2. **Animations:**

- Explore CSS animations to add dynamic effects to your webpage.
- Create a simple animation, such as a fade-in effect or a bouncing element.

3. **Advanced Styling:**

- Experiment with advanced styling techniques like gradients, shadows, and rounded corners.
- Apply these styles to different sections of your webpage.

```
<!DOCTYPE html>
```

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

```
<head>
```

```
  <meta charset="UTF-8">
```

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

```
  <title>HTML Semantics and Accessibility</title>
```

```
<style>
```

```
  body {
```

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

margin: 0;

padding: 0;

background-color: #f8f8f8;

color: #333;

}
```

```
header {

background-color: #333;

color: #fff;

text-align: center;

padding: 20px;

}
```

```
nav {

background-color: #4CAF50;

padding: 10px;

text-align: center;

}
```

```
nav ul {

list-style: none;

padding: 0;

margin: 0;

}
```

```
nav li {

display: inline;

margin: 0 10px;

}
```

```
nav a {  
    text-decoration: none;  
    color: #fff;  
    font-weight: bold;  
}
```

```
section {  
    margin: 20px;  
    padding: 20px;  
    background-color: #fff;  
    border-radius: 8px;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}
```

```
article {  
    margin-bottom: 20px;  
}
```

```
h2 {  
    color: #4CAF50;  
}
```

```
footer {  
    background-color: #333;  
    color: #fff;  
    text-align: center;  
    padding: 10px;  
    position: fixed;  
    bottom: 0;  
    width: 100%;  
}
```

```
</style>
</head>
<body>
  <!-- Header Section -->
  <header>
    <h1>Web Development Blog</h1>
    <p>Welcome to our blog where we explore the world of web development!</p>
  </header>

  <!-- Navigation Section -->
  <nav>
    <ul>
      <li><a href="#home">Home</a></li>
      <li><a href="#articles">Articles</a></li>
      <li><a href="#about">About Us</a></li>
      <li><a href="#contact">Contact</a></li>
    </ul>
  </nav>

  <!-- Main Content Section -->
  <section>
    <h2 id="home">Home</h2>
    <p>This is the main landing page of our blog. Explore the latest web development trends and topics.</p>
  </section>

  <!-- Articles Section -->
  <section>
    <h2 id="articles">Latest Articles</h2>
    <!-- Article 1 -->
    <article>
      <h3>Building Responsive Websites</h3>
```

```
<p>Learn the essentials of creating responsive websites that adapt to different screen sizes.</p>
</article>
<!-- Article 2 -->
<article>
  <h3>Introduction to Accessibility</h3>
  <p>Understand the importance of making your websites accessible to everyone.</p>
</article>
</section>

<!-- About Us Section -->
<section>
  <h2 id="about">About Us</h2>
  <p>Discover more about our team and the mission behind our web development blog.</p>
</section>

<!-- Contact Section -->
<section>
  <h2 id="contact">Contact Us</h2>
  <p>Have questions or suggestions? Reach out to us via email or social media.</p>
</section>

<!-- Footer Section -->
<footer>
  <p>&copy; 2023 Web Development Blog. All rights reserved.</p>
</footer>
</body>
</html>
```

Additional Challenge (Optional):

- Implement responsive design by using media queries to adapt your styles based on different screen sizes.
- Explore and apply flexbox or grid layout for a more sophisticated layout.