Web Development

Day : 10

CSS Properties: Border-radius

Used to make rounded corners on elements.

Can make boxes look soft, or even make circles.

```
.box {
  width: 100px;
  height: 100px;
  background: tomato;
  border-radius: 50%; /* Perfect circle */
}
```

Box-shadow (for depth and glow effects)

- Adds shadows around elements for a 3D effect.
- .card {
- width: 200px;
- height: 100px;
- background: white;
- box-shadow: 4px 4px 15px rgba(0, 0, 0, 0.3);
- }
- Makes buttons, cards, and images look modern

Cursor (change mouse pointer)

Fun property to show different cursor styles.

```
.link {
  cursor: pointer;   /* looks like a hand */
}

.text {
  cursor: text;      /* looks like text selection cursor */
}
```

You can immediately see results when hovering.

Sticky position

Position: sticky (sticky navbar effect)
 Keeps elements fixed while scrolling.

```
.navbar {
  position: sticky;
  top: 0;
  background: darkblue;
  color: white;
  padding: 10px;
}
```

Text-shadow (glow & depth for text)

- Adds shadow effects behind text. You can set horizontal offset, vertical offset, blur, and color.
- Syntax:
- text-shadow: horizontal vertical blur color;

Example 1: Simple text shadow

- Code:
- h1 {
- font-size: 40px;
- color: navy;
- text-shadow: 2px 2px 5px gray;
- }
- Creates a soft gray shadow behind the text.

Example 2: Glowing effect

- Code:
- h1 {
- font-size: 40px;
- color: white;
- background: black;
- text-shadow: 0 0 10px cyan, 0 0 20px blue;
- }
- Makes the text glow like neon lights.

Example 3: Multiple shadows for style

- Code:
- h1 {
- font-size: 40px;
- color: crimson;
- text-shadow: 2px 2px 0px black, 4px 4px 0px gray;
- }
- Adds layered shadows for a 3D look.

Align-items (Flexbox property)

- Controls vertical alignment of items in a flex container.
- Values:
- flex-start → top
- flex-end → bottom
- center → middle
- stretch → stretch to container height
- baseline → align text baselines
- .container {
- display: flex;
- align-items: center;
- }

Justify-content (Flexbox property)

- Controls horizontal alignment of items in a flex container.
- Values:
- flex-start → align items to the left (default)
- flex-end → align items to the right
- center → align items in the middle
- space-between → equal space between items
- space-around → equal space around items
- space-evenly → equal space everywhere
- .container {
- display: flex;
- justify-content: space-between;
- }

!important

- Used to give highest priority to a CSS rule.
- Overrides inline styles and other rules (use carefully).
- P {
- color: red !important; /* This will always apply */
- }

Clip-path

- Creates shapes by “clipping” part of an element.
- Can make circles, polygons, stars, etc.
- Img {
- clip-path: circle(50%);
- clip-path: polygon(50% 0%, 0% 100%, 100% 100%);
- }

Transform

- Used to move, rotate, scale, or skew elements.
- Code:
- .box {
- transform: rotate(45deg) scale(1.2);
- }

Navbar Exercise

- Create a modern navigation bar with these features:
- Should stick to top when scroll
- A logo/brand name on the left.
- Navigation links (Home, About, Services, Contact) in the center.
- A search bar with a search icon (Font Awesome) on the right.
- Use Flexbox for layout (justify-content, align-items).
- Add hover effects for navigation links.
- Use border-radius and transform for stylish effects.
- Use clip-path for a cool navbar background.

What is a Transition?

- Used for smooth changes between property values
- Triggered by user interaction (hover, click, focus) or property change
- Syntax:
- transition: property duration timing-function delay;

Transition Properties

- property → which property to animate (color, width, all)
- duration → how long it takes (2s, 500ms)
- timing-function → speed curve (ease, linear, etc.)
- delay → wait time before start

Transition code for buttons

```
button {
  background: teal;
  color: white;
  padding: 10px 20px;
  transition: background 0.5s ease, transform 0.5s;
}
button:hover {
  background: orange;
  transform: scale(1.1);
}
```

What are Keyframes?

- Used for complex or continuous animations
- Can define multiple steps
- Works without user interaction
- Syntax:
- @keyframes name {
- 0% { … }
- 50% { … }
- 100% { … }
- }

Animation Properties

- animation-name → name of keyframes
- animation-duration → how long it runs
- animation-iteration-count → number of repeats (1, infinite)
- animation-direction → normal, reverse, alternate
- animation-fill-mode → none, forwards, backwards, both

Keyframes example

```
.box {
  width: 100px;
  height: 100px;
  background: red;
  animation: moveBox 3s infinite alternate ease-in-out;
}
@keyframes moveBox {
  0%   { left: 0px; top: 0px; }
  50%  { left: 200px; background: orange; }
  100% { left: 200px; top: 200px; background: teal; }
}
```

Transition vs animation

- Transitions
- Triggered only when a property changes (hover, click, focus, etc.)
- Work between two states only (start → end)
- Short and simple animations (e.g., button hover effects)
- No built-in looping support
- Easier to write and use

- Keyframes
- Can run automatically (no user interaction needed)
- Allow multiple steps (0%, 25%, 50%, 100%, etc.)
- Suitable for complex animations (moving objects, loaders, banners)
- Support looping and repetition (infinite)
- More customizable (timing, direction, delays, chaining effects)

Moving Box

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS Animation</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
  <h2>CSS Animation Example</h2>
  <div class="box"></div>
</body>
</html>
```

```
    .box {
    width: 100px;
    height: 100px;
    background: tomato;
    position: relative;
    animation: moveBox 3s infinite alternate ease-in-out;
  }

  @keyframes moveBox {
    from { left: 0; }
    to { left: 200px; }
  }
```

Color Changing Text

- <!DOCTYPE html>
- <html lang="en">
- <head>
- <meta charset="UTF-8">
- <title>Color Changing Text</title>
- <link rel="stylesheet" href="style.css">
- </head>
- <body>
- <h1>CSS is Fun 🎨</h1>
- </body>
- </html>

- h1 {
- font-size: 50px;
- text-align: center;
- animation: colorChange 3s infinite alternate;
- }
-
- @keyframes colorChange {
- 0% { color: tomato; }
- 50% { color: seagreen; }
- 100% { color: royalblue; }
- }

Rotating box

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Rotating Box</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="box"></div>
</body>
</html>
```

```
.box {
  width: 100px;
  height: 100px;
  background: royalblue;
  margin: 100px auto;
  animation: spin 3s linear infinite;
}

@keyframes spin {
  from { transform: rotate(0deg); }
  to   { transform: rotate(360deg); }
}
```

Exercise

- Create a simple personal webpage using HTML and CSS with the following requirements:
- The webpage should have a header with your name and a navigation bar containing links: Home, About, Contact.
- The page should include a hero section with a background color and a short welcome message.
- Add an “About Me” section with a heading, an image (use any placeholder), and a short paragraph.
- Add a “Contact Me” section with a simple form containing:
- Name input field
- Email input field
- Message textarea
- A submit button

Exercise

- Style the page using CSS:
- Use a nice font (like Google Fonts).
- Make the navigation bar horizontal with hover effects.
- Center the hero message.
- Add some spacing and background colors to make it look clean.
- Add a footer with the text: © 2025 My Website

Exercise code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
  <title>My Personal Webpage</title>
  <!– Google Font →
  <link href=
https://fonts.googleapis.com/css2?family=Ro
boto:wght@400;600&display=swap
rel="stylesheet">
```

```
<style>
 /* General styling */
 body {
   margin: 0;
   font-family: ‘Roboto’, sans-serif;
   line-height: 1.6;
   background-color: #f4f4f9;
 }
```

Exercise code

```
/* Header and nav */
header {
  background: #333;
  color: #fff;
  padding: 10px 0;
  text-align: center;
}
```

```
nav ul {
  list-style: none;
  padding: 0;
  margin: 0;
}
nav ul li {
  display: inline;
  margin: 0 15px;
}
```

Exercise code

```
nav ul li a {
  color: white;
  text-decoration: none;
  font-weight: 600;
}
nav ul li a:hover {
  color: #ffd700;
}

/* Hero section */
.hero {
  background: #5dade2;
  color: white;
  padding: 50px 20px;
  text-align: center;
}
```

```
/* About section */
.about {
  padding: 40px 20px;
  text-align: center;
}
.about img {
  width: 150px;
  border-radius: 50%;
  margin: 20px 0;
}

/* Contact section */
.contact {
  background: #fff;
  padding: 40px 20px;
}
```

Exercise code

```
form {
  max-width: 400px;
  margin: auto;
  display: flex;
  flex-direction: column;
}
form input, form textarea, form button {
  margin: 10px 0;
  padding: 10px;
  font-size: 16px;
}
form button {
  background: #333;
  color: white;
  border: none;
  cursor: pointer;
}
form button:hover {
  background: #5dade2;
}

/* Footer */
footer {
  background: #333;
  color: white;
  text-align: center;
  padding: 10px;
  margin-top: 20px;
}
</style>
```

Exercise code

```
</head>
<body>
 <!– Header 
 <header>
  <h1>My Personal Webpage</h1>
  <nav>
   <ul>
    <li><a href=“#”>Home</a></li>
    <li><a href=“#about”>About</a></li>
    <li><a href=“#contact”>Contact</a></li>
   </ul>
  </nav>
 </header>
 <!– Hero Section 
 <section class=“hero”>
  <h2>Welcome to My Website</h2>
  <p>I am learning HTML and CSS to build amazing websites!</p>
 </section>
 <!– About Section 
 <section class=“about” id=“about”>
  <h2>About Me</h2>
  <img src=https://via.placeholder.com/150 alt=“My Photo”>
  <p>Hello! My name is [Your Name]. I love coding, designing, and creating useful web applications.</p>
 </section>
 <!– Contact Section 
 <section class=“contact” id=“contact”>
  <h2>Contact Me</h2>
  <form>
   <input type=“text” placeholder=“Your Name” required>
   <input type=“email” placeholder=“Your Email” required>
   <textarea placeholder=“Your Message” rows=“5” required></textarea>
   <button type=“submit”>Send</button>
  </form>
 </section>
```

Exercise code

```
  <!– Footer 
  <footer>
    <p>© 2025 My Website</p>
  </footer>
</body>
</html>
```