

DAY 7

Note: The names for my HTML, CSS, and JavaScript files (like DAYTHREE.html, DAYTHREE.css, DAYTHREE.js) are **not recommended** as they are considered **bad practice** in professional projects. However, since this is a **mini project**, I prefer using these names to **distinguish them from my other files**. For **larger or professional projects**, it's better to use standard and descriptive file names like index.html, style.css, and script.js.

DAYSEVEN.HTML

Gallery container

```
<div class="gallery">
```

This is the main box that holds all the image cards.

Image carousel

```
<div class="image-container" id="imageContainer">
```

This part creates a **rotating circle** of images.

Each span below rotates and shows:

- a **wrapper**
- a **caption (text)** above
- an **image**

Example:

```
<span style="--i:1">  
  <div class="wrapper">  
    <div class="caption">Illustration Design</div>  
      
  </div>  
</span>
```

You repeated this block 8 times for 8 different images and captions.

Navigation Buttons

```
<div class="btn-container">  
  <button class="btn" id="prev">Prev</button>  
  <button id="about" onclick="window.open('DAY 7.pdf', '_blank')">About</button>  
  <button class="btn" id="next">Next</button>  
</div>
```

- Prev and Next: Rotate the carousel left or right.

- About: Opens a **PDF file** in a new tab.
-

DAYSEVEN.CSS

```
body {
  margin: 0; /* Removes default space */
  display: flex; /* Use flexbox layout */
  flex-direction: column; /* Stack items vertically */
  align-items: center; /* Center items horizontally */
  background-color: black; /* Background color */
  font-family: Arial, sans-serif; /* Font style */
  height: 100vh; /* Full screen height */
  justify-content: center; /* Center items vertically */
  overflow: hidden; /* Prevent scrollbars */
}
```

.gallery — 3D effect setup

```
.gallery {
  perspective: 1000px; /* Enables 3D look */
}
```

.image-container — Rotating image box

```
.image-container {
  width: 200px;
  height: 250px;
  position: relative; /* For absolute children */
  transform-style: preserve-3d; /* Keep 3D rotation */
  transform: rotateY(0deg); /* Start position */
  transition: transform 0.7s; /* Smooth rotation */
}
```

.image-container span — Each image position

```
.image-container span {
  position: absolute;
  top: 0;
  left: 0;
  transform: rotateY(calc(var(--i) * 45deg)) translateZ(400px); /* Spread around in a circle */
  transition: transform 0.7s;
}
```

- --i is the index (1 to 8), so every image is placed at a different angle in the circle.

.wrapper — Groups caption + image

```
.wrapper {
  display: flex;
  flex-direction: column;
  align-items: center;
}
```

.caption – Text above image

```
.caption {
  color: white;
  margin-bottom: 8px; /* Space between text and image */
  font-size: 16px;
  font-weight: bold;
  text-align: center;
}
```

.wrapper img – Image styling

```
.wrapper img {
  width: 200px; /* Same size for all images */
  border-radius: 10px; /* Rounded corners */
  transition: transform 0.3s;
}
```

Zoom effect on hover

```
.wrapper:hover img {
  transform: scale(1.1); /* Image grows slightly */
}
```

.btn-container – Button area

```
.btn-container {
  display: flex;
  justify-content: center;
  gap: 20px;
  margin-top: 60px;
}
```

.btn – Prev/Next button style

```
.btn {
  background-color: darkorange;
  color: white;
  border: none;
  padding: 10px 20px;
  border-radius: 5px;
  cursor: pointer;
  font-weight: bold;
}
```

#about – About button style (same look)

```
#about {
  background-color: darkorange;
  color: white;
  font-weight: bold;
  padding: 10px 20px;
  border-radius: 5px;
  cursor: pointer;
}
```

```
}
```

Hover effect for all buttons

```
.btn:hover,  
#about:hover {  
  filter: brightness(1.3); /* Glow effect */  
}
```

DAYSEVEN.JS

Goal of this Code:

You have a rotating image gallery (carousel), and this code helps:

1. **Rotate images left/right using buttons**
2. **Rotate automatically every 4 seconds**
3. **Pause auto-rotate when mouse is over the images**

```
let imageContainer = document.getElementById("imageContainer");
```

This line **selects the image box** (<div> with id imageContainer) and stores it in a variable so we can **rotate it later**.

```
let angle = 0;
```

This sets the **starting angle** of rotation to 0.
(We will rotate the container by adding/subtracting 45 degrees.)

Next Button Click

```
document.getElementById("next").onclick = function () {  
  angle -= 45;  
  imageContainer.style.transform = `rotateY(${angle}deg)`;  
};
```

When the **Next** button is clicked:

- The angle decreases by 45 degrees (rotating to the **left**)
- The image container is rotated using rotateY() in CSS

Previous Button Click

```
document.getElementById("prev").onclick = function () {  
  angle += 45;  
  imageContainer.style.transform = `rotateY(${angle}deg)`;  
};
```

When the **Prev** button is clicked:

- The angle increases by 45 degrees (rotating to the **right**)
- Again, we update the container's rotation

Auto-Rotate Every 4 Seconds

```
let autoRotate = setInterval(() => {  
  angle += 45;  
  imageContainer.style.transform = `rotateY(${angle}deg)`;  
}, 4000);
```

This part says:

- Every **4000 milliseconds (4 seconds)**, rotate the gallery **left** by 45 degrees
- This keeps the images moving automatically

Stop Auto-Rotate on Mouse Hover

```
imageContainer.addEventListener("mouseover", () => clearInterval(autoRotate));
```

When the **mouse goes over the image**, stop the auto-rotate.

- It does this by **clearing the interval timer**

Resume Auto-Rotate on Mouse Out

```
imageContainer.addEventListener("mouseout", () => {  
  autoRotate = setInterval(() => {  
    angle += 45;  
    imageContainer.style.transform = `rotateY(${angle}deg)`;  
  }, 4000);  
});
```

When the **mouse leaves** the image area, start rotating again every 4 seconds.