DAY 8

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.

DAYEIGHT.HTML

```
<link rel="stylesheet" href="font-awesome.css">
```

- Adds **Font Awesome icons**

<body> section

This part shows the **content on your web page**.

```
<div class="container">
```

- This is the **main wrapper** for your progress tracker.

.progress-container

Holds the **progress bar and steps**.

```
<div class="progress-bar">
  <div class="progress" id="progress"></div>
</div>
```

- The **gray background bar**.
- The green .progress line moves forward when clicking “Next”.

.steps

Shows **each step of the delivery**:

```
<div class="step checked">
  <div class="circle"><i class="fas fa-file-alt"></i></div>
  <p>PROCESSING</p>
</div>
```

Each .step has:

- A **circle** with an icon (like a file, box, truck, or home).
- A **label** (e.g., "PROCESSING", "SHIPPED", etc.).

- The first step has the class `checked`, meaning it's **already completed** (green).

Buttons

```
<div class="buttons">
  <button id="prev"> ← </button>
  <button id="next"> → </button>
</div>
```

- These buttons **move backward and forward** in the steps.
- The "Previous" button is **disabled at the start**.

DAYEIGHT.CSS

`body` — the whole page

```
body {
  margin: 0;
  padding: 0;
  background-color: #f0f0f0;
  font-family: Arial, sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}
```

What it does:

- Removes default space around the page (`margin: 0; padding: 0`)
- Sets a **light gray background**
- Uses **Arial** font
- Centers everything **horizontally and vertically** using Flexbox
- `height: 100vh` means the page takes **full screen height**

`.container` — main box for everything

```
.container {
  width: 90%;
  max-width: 800px;
  text-align: center;
}
```

- Makes the container take up to **90% of screen**, but **never more than 800px**
- Centers all text inside

`.progress-container` — area holding the progress bar

```
.progress-container {
```

```
  position: relative;
  margin-bottom: 40px;
  padding: 40px 0 20px;
}
```

- Allows placing the progress line **behind icons** using position: relative
- Adds space **above and below** with padding
- margin-bottom keeps space between this and buttons

.progress-bar — the gray line

```
.progress-bar {
  position: absolute;
  top: 73px;
  left: 0;
  right: 0;
  height: 4px;
  background-color: #ccc;
  z-index: 1;
}
```

- The **gray progress line**
- top: 73px places it right under the icons
- z-index: 1 puts it **behind the icons**

.progress — the green filling part

```
.progress {
  height: 4px;
  background-color: green;
  width: 0%;
  transition: width 0.4s ease;
}
```

- Starts with **0% width**, but grows as you click “Next”
- transition makes the growth **smooth**

.steps — holds all the steps/icons

```
].steps {
  display: flex;
  justify-content: space-between;
  position: relative;
  z-index: 2;
}
```

- Uses **Flexbox** to put steps side by side
- space-between spreads them evenly

- z-index: 2 puts steps **above the gray line**

.step — each individual step (icon + label)

```
.step {
  display: flex;
  flex-direction: column;
  align-items: center;
  width: 5%;
}
```

- Puts the **icon on top** and **text below**
- Centers them
- width: 5% makes the step small (you can increase if spacing is tight)

.circle — the icon box

```
.circle {
  width: 60px;
  height: 60px;
  background-color: white;
  border: 4px solid lightgray;
  border-radius: 15px;
  display: flex;
  align-items: center;
  justify-content: center;
  color: lightgray;
  font-size: 22px;
  transition: all 0.3s ease;
}
```

- A **rounded box** (like a square with curved edges)
- Light gray border and icon color
- Font size = 22px
- transition makes it animate smoothly when it changes

.step.checked .circle — when the step is active (green)

```
.step.checked .circle {
  background-color: green;
  color: white;
  border-color: green;
  box-shadow: 0 0 10px rgba(0, 128, 0, 0.6);
}
```

- Turns the icon **green with white color**
- Adds a **green glow**

Step label below the icon

```
.step p {
  margin-top: 10px;
  font-size: 14px;
  color: #999;
}
```

- Adds space above the text
- Makes text **gray** and small

When step is complete:

```
.step.checked p {
  color: green;
  font-weight: bold;
}
```

- Text becomes **green and bold**

.buttons — space for the buttons

```
.buttons {
  margin-top: 20px;
}
```

- Adds space **above the buttons**

.btn — style for “Next” and “Previous”

```
.btn {
  background-color: #4CAF50;
  color: white;
  padding: 10px 20px;
  margin: 0 10px;
  border: none;
  border-radius: 5px;
  font-size: 16px;
  cursor: pointer;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);
}
```

- Green buttons with white text
- Rounded corners, spacing, and shadow
- `cursor: pointer` changes to hand on hover

When the button is disabled:

```
.btn:disabled {
  background-color: lightgray;
  cursor: not-allowed;
}
```

- Turns gray and shows a **“not allowed” cursor**

DAYEIGHT.JS

What This Code Does:

This JavaScript code controls a **step progress bar** (like for order tracking). It updates the visual progress when you:

- Click the **Next** button
- Click the **Previous** button
- Press the **Right Arrow** key
- Press the **Left Arrow** key

```
const nextBtn = document.getElementById("next");
const prevBtn = document.getElementById("prev");
const progressBar = document.querySelector(".progress");
const steps = document.querySelectorAll(".step");
```

These lines **get the elements** from your HTML:

- The **Next button**
- The **Previous button**
- The **green progress bar**
- All the **steps** (icons like □, □, □)

```
let currentStep = 1;
```

This keeps track of **which step you're on right now**. It starts at step 1 (first one).

When You Click the "Next" Button:

```
nextBtn.addEventListener("click", () => {
  currentStep++;
  if (currentStep > steps.length) currentStep = steps.length;
  updateProgress();
});
```

If you click **Next**, it moves one step forward.

But it also checks:

- If you're already at the last step, it **won't go past** it.

Then it calls `updateProgress()` to visually update the bar.

When You Click the "Previous" Button:

```
prevBtn.addEventListener("click", () => {
```

```
  currentStep--;
  if (currentStep < 1) currentStep = 1;
  updateProgress();
});
```

This works the same way — but goes **one step back** when you click **Previous**.

It stops if you're already at the first step.

When You Press Arrow Keys:

```
document.addEventListener("keydown", function (event) {
  if (event.key === "ArrowRight") {
    currentStep++;
    if (currentStep > steps.length) currentStep = steps.length;
    updateProgress();
  } else if (event.key === "ArrowLeft") {
    currentStep--;
    if (currentStep < 1) currentStep = 1;
    updateProgress();
  }
});
```

This lets you use the **keyboard arrow keys** to move:

- **Right Arrow** = go forward
- **Left Arrow** = go backward

Same logic as the buttons.

The Main Function — updateProgress()

```
function updateProgress() {
  steps.forEach((step, index) => {
    step.classList.toggle("checked", index < currentStep);
  });
```

This part **adds or removes a green highlight** (checked class) for steps that have been reached.

```
  const progressPercent = ((currentStep - 1) / (steps.length - 1)) * 100;
  progressBar.style.width = `${progressPercent}%`;
```

This calculates how much of the green bar should be filled — based on the current step.

For example:

- Step 1 → 0%
- Step 2 → 33%
- Step 3 → 66%
- Step 4 → 100%

```
  prevBtn.disabled = currentStep === 1;
  nextBtn.disabled = currentStep === steps.length;
}
```

This part disables the **Next** or **Previous** button when you’re at the first or last step — so users can't go too far.