

# DAY 16 (VERSION 2.0)

## Index.html

### **<head> section:**

This is where we include info *about* the page (not what's shown visually):

- <title> — sets the tab title in your browser to “Live Battery Indicator”.
- Google Fonts link — imports the **Orbitron** font for a futuristic look.
- style1.css — links to your custom CSS file to style the page.
- <link rel="icon" ...> — adds a small battery icon to the browser tab.

### **<body>:**

This is where the **actual visible content** is.

- div.battery-card — A box/card that holds all the battery details.

Inside it:

- <h2> — A heading saying "Battery Level".
- <div id="battery-percent"> — Shows the numeric battery level (e.g., 72%).
- <div class="battery-shell"> — A shell-shaped battery frame.
  - Inside it, <div id="battery-fill"> — This part fills up like a progress bar.
- <div id="battery-status"> — Displays if the device is **charging** or **not**.

## Style.css

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}
```

- Removes all default spacing and makes layouts more predictable.

```
body {  
  background-color: #121212;  
  font-family: 'Segoe UI', sans-serif;  
  height: 100vh;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  flex-direction: column;  
}
```

- Dark background for a clean look.

- Uses a modern sans-serif font.
- Centers everything in the middle of the screen (both horizontally and vertically).
- Puts content in a **column layout**.

```
.battery-card {
  background-color: #1e1e1e;
  padding: 40px 30px;
  border-radius: 30px;
  text-align: center;
  color: white;
  animation: glowPulse 1s infinite alternate;
}
```

- A rounded rectangle that holds battery content.
- Has a **glow animation** using the glowPulse keyframes below.

## Glow Animation

```
@keyframes glowPulse {
  0% {
    box-shadow: 0 0 10px rgba(30, 240, 30, 0.4),
      0 0 20px rgba(30, 240, 30, 0.3);
  }
  100% {
    box-shadow: 0 0 30px rgba(30, 240, 30, 0.9),
      0 0 50px rgba(30, 240, 30, 0.6);
  }
}
```

- Creates a **pulsing green glow** effect around the battery card.
- It animates between light and strong shadows.

```
.battery-card h2 {
  font-family: 'Orbitron', sans-serif;
  font-size: 22px;
  color: #c5c5c5;
  margin-bottom: 10px;
}
```

- Uses the Orbitron font to make it look techy.
- Lighter gray color for contrast.

```
.percentage {
  font-size: 48px;
  font-weight: bold;
  color: white;
  margin-bottom: 20px;
}
```

- Shows the big % number.
- Large and bold to grab attention.

```
.battery-shell {
```

```

width: 100px;
height: 180px;
background-color: #101010;
border: 2px solid #333;
border-radius: 50px;
margin: auto;
display: flex;
align-items: flex-end;
justify-content: center;
padding: 10px;
position: relative;
}

```

- A **vertical battery container**.
- Rounded like a capsule.
- Content aligns to the bottom (so the fill grows from the bottom up).

```

width: 100%;
border-radius: 40px;
transition: height 0.5s ease-in-out, background 0.5s ease-in-out;
}

```

- This bar changes **height and color** based on battery level.
- Smooth transitions when values update.

```

.battery-status {
margin-top: 20px;
font-size: 18px;
display: flex;
justify-content: center;
align-items: center;
gap: 10px;
}

```

- Displays text like "**Charging...**" or "**Not Charging**".
- Uses spacing between icons/text if any.

## Script.js

```
document.addEventListener("DOMContentLoaded", () => {
```

- Waits until your web page is **fully loaded** before running anything.

```

const batteryFill = document.getElementById("battery-fill");
const batteryPercent = document.getElementById("battery-percent");
const batteryStatus = document.getElementById("battery-status");

```

- Finds and stores the **parts of the battery design**:
  - batteryFill = the green fill that shows battery level.
  - batteryPercent = the number like "80%".
  - batteryStatus = the text below (e.g., "Full battery").

```
if ('getBattery' in navigator) {
```

- Checks if your **browser supports** the battery feature.
- If yes, it moves forward. If not, it shows an alert.

```
navigator.getBattery().then(battery => {
```

- Asks the browser for the **battery info**.
- battery gives:
  - battery.level: how full the battery is (0 to 1).
  - battery.charging: whether it's charging or not.

## Updates the Battery Display:

```
function updateBatteryUI() {
  const level = Math.floor(battery.level * 100);
```

- Converts battery level from 0.88 (for 88%) to a clean number like 88.

```
batteryPercent.textContent = `${level}%`;
batteryFill.style.height = `${level}%`;
```

- Changes the percentage number you see.
- Adjusts the height of the green battery bar to match that percent.

## Sets Color and Message Based on Level:

```
if (level <= 20) {
  batteryFill.style.background = 'red gradient';
  batteryStatus.innerHTML = 'Low battery 🔋';
}
```

- If battery is **20% or less**: red color, warning message.

```
else if (level <= 50) {
  batteryFill.style.background = 'yellow gradient';
  batteryStatus.innerHTML = 'Medium battery 🔋';
}
```

- If battery is **between 21–50%**: orange/yellow color, medium status.

```
else {
  batteryFill.style.background = 'green gradient';
  batteryStatus.innerHTML = 'Full battery 🔋';
}
```

- If battery is **above 50%**: green color, "Full battery" message.

## Keeps It Updating Live:

```
updateBatteryUI();
battery.addEventListener('levelchange', updateBatteryUI);
battery.addEventListener('chargingchange', updateBatteryUI);
```

- Shows the current status **right away**.
- Then, **listens for changes** in battery level or charging — and updates instantly.

### **If Not Supported:**

```
} else {  
  alert("Battery API not supported in this browser.");  
}
```

- If your browser doesn't allow battery checking, shows a message.