DAY 1

General Body Styling

```
body {
  margin: 0;
```

- Removes default space around the page.

```
  background-color: black;
```

- Sets the background color of the page to black.

```
  font-family: sans-serif;
```

- Uses a clean, simple font style.

```
  display: flex;
  flex-direction: column;
```

- Arranges elements vertically using Flexbox.

```
  align-items: center;
```

- Centers items horizontally on the page.

```
  height: 100vh;
}
```

- Makes the body take up the full height of the screen (100% of the viewport height).

Heading Styling

```
h1 {
  color: white;
```

- Makes the heading text white.

```
  margin-top: 20px;
```

- Adds space above the heading.

```
  text-decoration: underline white;
}
```

- Underlines the heading with a white line.

Calculator Container

```
.calculator {
  width: 320px;
```

- Sets the calculator's width to 320 pixels.

```
  margin-top: 20px;
```

- Adds space above the calculator.

```
  border: 1PX solid lightgrey;
}
```

- Gives the calculator a light grey border (note: `PX` should be lowercase: `1px`).

Display Screen Styling

```
.display {
  background-color: black;
```

- Sets the display area’s background to black.

```
  color: white;
```

- Makes the text on the display white.

```
  font-size: 48px;
```

- Makes the numbers on the display large.

```
  text-align: right;
```

- Aligns the numbers to the right side.

```
  padding: 20px;
```

- Adds space inside the display area.

```
  box-sizing: border-box;
```

- Ensures padding is included in the width.

```
  border-bottom: 1px solid #333;
}
```

- Adds a dark line below the display area.

Buttons Container

```
.buttons {
  display: grid;
```

- Arranges buttons in a grid layout.

```
 grid-template-columns: repeat(4, 1fr);
```

- Creates 4 equal columns.

```
 gap: 10px;
```

- Adds space between buttons.

```
 padding: 10px;
}
```

- Adds space inside the button container.

Button Styling

```
button {
 height: 60px;
```

- Each button is 60 pixels tall.

```
 border: none;
```

- Removes the default border.

```
 border-radius: 30px;
```

- Makes buttons rounded (like circles).

```
 font-size: 24px;
```

- Makes button text fairly large.

```
 cursor: pointer;
```

- Changes the mouse to a pointer when hovering.

```
 transition: 0.2s;
}
```

- Adds a smooth animation effect (e.g., when hovering).

Button Types

```
.digit { background-color: #333; color: white; }
```

- Number buttons are dark gray with white text.

```
.symbol { background-color: orange; color: white; }
```

- Operation buttons (like +, –, ×) are orange with white text.

```
.function { background-color: #aaa; color: black; }
```

- Special function buttons (like AC or %) are light gray with black text.

```
.wide { grid-column: span 2; }
```

- Makes some buttons (like 0) take up two columns in width.

JAVASCRIPT

Accessing the Display Element

```
const display = document.getElementById('display');
```

- This gets the HTML element with the `id="display"` and stores it in a variable called `display`.
- We'll use this to show numbers and results on the calculator screen.

Variable to Store Input

```
let currentInput = '';
```

- This keeps track of what the user has typed (numbers, +, -, etc.).
- It starts out as an empty string.

Function to Add Numbers or Symbols

```
function appendValue(val) {
```

- Starts a function named `appendValue` that takes one value (`val`), like `1`, `+`, or `.`.

```
 if (display.innerText === '0' && val !== '.') {
```

- If the screen shows just `0` and the user presses anything **except** the decimal point ....

```
   currentInput = val;
```

- Replace `0` with the new input (e.g., user presses `5`, display becomes `5`).

```
 } else {
   currentInput += val;
```

- Otherwise, add the new value to the end of the existing input (like building `45+2` step by step).

```
 display.innerText = currentInput;
```

```
}
```

- Update the calculator screen to show the current input.

Function to Clear the Screen

```
function clearDisplay() {
```

- This function resets everything when you press the **C** or **AC** button.

```
  currentInput = '';
```

- Clears the current input.

```
  display.innerText = '0';
}
```

- Shows `0` on the screen again.

Function to Toggle Sign (±)

```
function toggleSign() {
```

- This function changes a number from positive to negative or vice versa.

```
  if (currentInput) {
```

- Only do this if there's something typed (not empty).

```
    currentInput = String(eval(currentInput) * -1);
```

- Convert the input into a number, multiply by `-1`, then turn it back into a string.

```
    display.innerText = currentInput;
  }
}
```

- Show the new (positive or negative) number on the screen.

Function to Calculate the Answer (=)

```
function calculate() {
```

- This runs when the **=** button is pressed.

```
  try {
```

- Try to safely calculate the result.

```
    currentInput = String(eval(currentInput));
```

- `eval` solves the math (like turning `2+3` into `5`), then stores it as a string.

```
    display.innerText = currentInput;
```

- Show the result on the screen.

```
  } catch {
```

- If something goes wrong (like invalid input)...

```
    display.innerText = 'Error';
    currentInput = '';
  }
}
```

- Show "Error" on the screen and reset the input.