✅ Day 5: Timers in JavaScript – `setTimeout`, `setInterval`, `clearInterval`, `clearTimeout`

🧠 Why Timers?

Timers allow us to **delay** or **repeat** certain actions in JS, which is useful in:

- UI animations

- Auto-refresh

- Countdown timers

- Debouncing (later topic)

- Async patterns

🕰️ 1. `setTimeout()`

Runs a function **after a delay** (once only).

js
CopyEdit
```js
setTimeout(() => {
  console.log("Executed after 2 seconds");
}, 2000);
```

- First parameter: callback function

- Second parameter: delay in **milliseconds** (1000ms = 1s)

✅ You can assign it to a variable to cancel it:

js
CopyEdit
```js
const timer = setTimeout(() => {
  console.log("This will not run");
}, 3000);

clearTimeout(timer); // cancels the timeout
```

🔁 2. `setInterval()`

Repeats a function at regular intervals.

js
CopyEdit
```js
setInterval(() => {
  console.log("This runs every 2 seconds");
}, 2000);
```

✅ Stop it using `clearInterval()`:

js
CopyEdit
```js
const myInterval = setInterval(() => {
  console.log("Repeating...");
}, 1000);

setTimeout(() => {
  clearInterval(myInterval);
  console.log("Stopped interval after 5 seconds");
}, 5000);
```

🚫 3. `clearTimeout()` and `clearInterval()`

Used to **stop** the actions triggered by `setTimeout` and `setInterval`.

- `clearTimeout(timerID)`

- `clearInterval(intervalID)`

💡 Real Example: Countdown Timer

js
CopyEdit
```js
let count = 5;
const countdown = setInterval(() => {
  console.log(count);
  count--;

  if (count === 0) {
    clearInterval(countdown);
    console.log("Countdown finished!");
  }
}, 1000);
```

🧪 Tasks for Practice

◆ Task 1: Delay Greeting

Make a function `sayHello()` that logs "Hello User" after 3 seconds using `setTimeout`.

◆ Task 2: Auto Increment

Make a counter that logs numbers from 1 to 10, one number per second using `setInterval`.

◆ Task 3: Stop Interval

Create a timer that prints "Running..." every 2 seconds, and stops after 6 seconds.

🔹 Bonus 🔥: Interactive Timer Function

js
CopyEdit
```js
function startTimer(seconds) {
  let remaining = seconds;
  const timer = setInterval(() => {
    console.log(`${remaining} sec left`);
    remaining--;
    if (remaining < 0) {
      clearInterval(timer);
      console.log("Timer complete!");
    }
  }, 1000);
}
startTimer(5); // try with different numbers
```

📌 Summary Table

Function	Use Case
`setTimeout()`	Run once after delay
`setInterval()`	Repeat at interval
`clearTimeout()`	Stop timeout
`clearInterval()`	Stop interval