### DAY-7 (Imp)

**🕓 1. setTimeout() — "Kuch der baad kaam kar"**

**🧠 Matlab:**

“Ye function bolta hai — *is code ko itne time baad chalao*.”

**💡 Syntax:**

setTimeout(callback, delay);

* callback: wo function jo baad me chalana hai
* delay: kitni der baad (in milliseconds)  
  👉 1000 ms = 1 second

**🎯 Example:**

setTimeout(() => {

console.log("Done");

}, 2000);

🗣️ Aam bhaasha me:

Ye bolta hai — “2 second ruk ja, phir ‘Done’ print kar dena.”

🕐 Output after 2 seconds:

Done

**⏰ 2. setInterval() — "Har thodi der me repeat kar"**

**🧠 Matlab:**

“Ye function har fixed interval pe same code baar-baar chalata hai.”

**💡 Syntax:**

setInterval(callback, delay);

**🎯 Example:**

setInterval(() => {

console.log("Hello");

}, 1000);

🗣️ Aam bhaasha me:

Har 1 second me “Hello” print hota rahega, jab tak code chal raha hai.

(Stop karne ke liye: clearInterval(intervalID))

**🌈 3. Promise — “Future me milne wala result”**

**🧠 Matlab:**

Promise ek object hai jo future me **resolve (success)** ya **reject (fail)** hota hai.  
Jaise — “Main 1 second me tujhe result dunga.”

**💡 Syntax:**

let promise = new Promise((resolve, reject) => {

// koi kaam karo

// agar sab thik raha -> resolve(value)

// agar error aayi -> reject(error)

});

**🎯 Example (tumhara hi question 👇)**

let promise = new Promise((resolve, reject) => {

setTimeout(() => {

resolve("Success");

}, 1000);

});

promise.then((result) => {

console.log(result);

});

🗣️ Aam bhaasha me:

“Ek promise banaya jo 1 second me poora hoga.  
Jab poora hoga (resolve), tab .then() ke andar wala code chalega.”

🕐 Output after 1 sec:

Success

**⚙️ 4. async / await — “Promise ka easy version”**

**🧠 Matlab:**

Async-await kaam ko **asynchronous (baad me hone wala)** banata hai,  
lekin likhne ka style **simple aur step-by-step** lagta hai.

**💡 Syntax:**

async function myFunc() {

let result = await promise;

console.log(result);

}

myFunc();

🗣️ Aam bhaasha me:

await ka matlab hota hai — “is promise ke poore hone tak ruk ja.”  
async function me hi await likh sakte ho.

**🎯 Example:**

let promise = new Promise((resolve) => {

setTimeout(() => resolve("Success"), 1000);

});

async function callPromise() {

console.log("Waiting...");

let result = await promise; // wait until resolved

console.log(result);

}

callPromise();

🕐 Output:

Waiting...

Success

🗣️ Matlab:

Pehle “Waiting...” print hoga,  
fir 1 second baad “Success” print hoga.

**⚡ QUICK SUMMARY TABLE:**

| **Concept** | **Meaning** | **Example** |
| --- | --- | --- |
| setTimeout() | Code ko kuch der baad chalata hai | setTimeout(()=>console.log("Hi"),2000) |
| setInterval() | Code ko har fixed time pe repeat karta hai | setInterval(()=>console.log("Hi"),1000) |
| Promise | Future me result milega | new Promise((res)=>setTimeout(()=>res("Done"),1000)) |
| async/await | Promise ko easy tarike se handle karta hai | await promise |

**DAY-7 PRACTICE SET** 👇

## 🧩 ****1️⃣ setTimeout – simple delay****

👉 2 second baad “Done” print karwao

setTimeout(() => {

console.log("Done");

}, 2000);

🗣️ Output (2 sec baad):

Done

## 🧩 ****2️⃣ setInterval – repeat after delay****

👉 Har 1 second me “Tick” print ho,  
aur 5 second baad band ho jaaye

let count = 0;

let timer = setInterval(() => {

console.log("Tick");

count++;

if (count === 5) {

clearInterval(timer);

console.log("Stopped!");

}

}, 1000);

🗣️ Output:

Tick

Tick

Tick

Tick

Tick

Stopped!

## 🧩 ****3️⃣ Promise – resolve after 1 second****

👉 Ek promise banao jo 1 sec baad “Success” return kare

let promise = new Promise((resolve) => {

setTimeout(() => {

resolve("Success");

}, 1000);

});

promise.then((result) => {

console.log(result);

});

🗣️ Output (1 sec baad):

Success

## 🧩 ****4️⃣ Promise – reject case samjho****

👉 Agar kuch galat ho gaya, to “Error occurred!” print karo

let promise = new Promise((resolve, reject) => {

let error = true;

setTimeout(() => {

if (error) reject("Error occurred!");

else resolve("Success!");

}, 1000);

});

promise

.then((result) => console.log(result))

.catch((err) => console.log(err));

🗣️ Output:

Error occurred!

## 🧩 ****5️⃣ async/await – promise ka easy use****

👉 Promise se value lo aur “Got it!” print karo

let promise = new Promise((resolve) => {

setTimeout(() => {

resolve("Got it!");

}, 1500);

});

async function getData() {

console.log("Waiting...");

let result = await promise;

console.log(result);

}

getData();

🗣️ Output:

Waiting...

Got it!

## 🧩 ****6️⃣ Challenge: Mix all together 😎****

👉 1 sec baad ek promise resolve ho,  
fir har 2 sec me “Running...” print ho,  
aur 6 sec baad “Stopped!” likh ke interval band ho jaaye.

async function startProcess() {

let promise = new Promise((resolve) => {

setTimeout(() => resolve("Started!"), 1000);

});

let msg = await promise;

console.log(msg);

let count = 0;

let interval = setInterval(() => {

console.log("Running...");

count++;

if (count === 3) {

clearInterval(interval);

console.log("Stopped!");

}

}, 2000);

}

startProcess();

🗣️ Output:

Started!

Running...

Running...

Running...

Stopped!

✅ **Summary line bhai yaad rakh:**

setTimeout – ek baar delay ke baad  
setInterval – baar-baar repeat  
Promise – future result (success/error)  
async/await – promise ko easy tarike se handle karna