

Day 27



Introduction to Callbacks:

What is a Callback?

A callback is a function that you pass as an argument to another function so that it can run later.

In simple words:

A callback = a function you give to another function to call when it is needed.

Why do we use Callbacks?

Because JavaScript does not wait for things like:

- setTimeout
- file loading
- API requests
- animations
- user actions

So, we use callbacks to run code after something finishes.

Example: a basic example of callback

The screenshot shows a code editor window with a dark theme. The code in main.js is as follows:

```
JS main.js > ⚡ greet
1  function greet(name, callback) {
2    console.log("Hello " + name);
3    callback(); // calling the callback
4  }
5  function done() {
6    console.log("Greeting completed");
7  }
8  greet("Aditya", done);
```

Below the code editor, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the output of running the script:

- PS E:\JavaScript\Day21-30\Day27> node main.js
- Hello Aditya
- Greeting completed

Example:

```
10  function sayHello() {
11    console.log("Hello");
12  }
13  function doSomething(callback) {
14    callback(); // calling the callback function
15  }
16  doSomething(sayHello);

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS E:\JavaScript\Day21-30\Day27> node main.js
Hello
```

Example: The function inside setTimeout is a callback.

```
18  setTimeout(function() {
19    console.log("This runs after 1 second");
20  }, 1000);

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS E:\JavaScript\Day21-30\Day27> node main.js
This runs after 1 second
```

Example: callback with a message

```
JS main.js > ...
22  function finish() {
23    console.log("Task Done!");
24  }
25  function startTask(callback) {
26    console.log("Starting...");
27    callback(); // run the callback
28  }
29  startTask(finish);

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS E:\JavaScript\Day21-30\Day27> node main.js
Starting...
Task Done!
```