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.

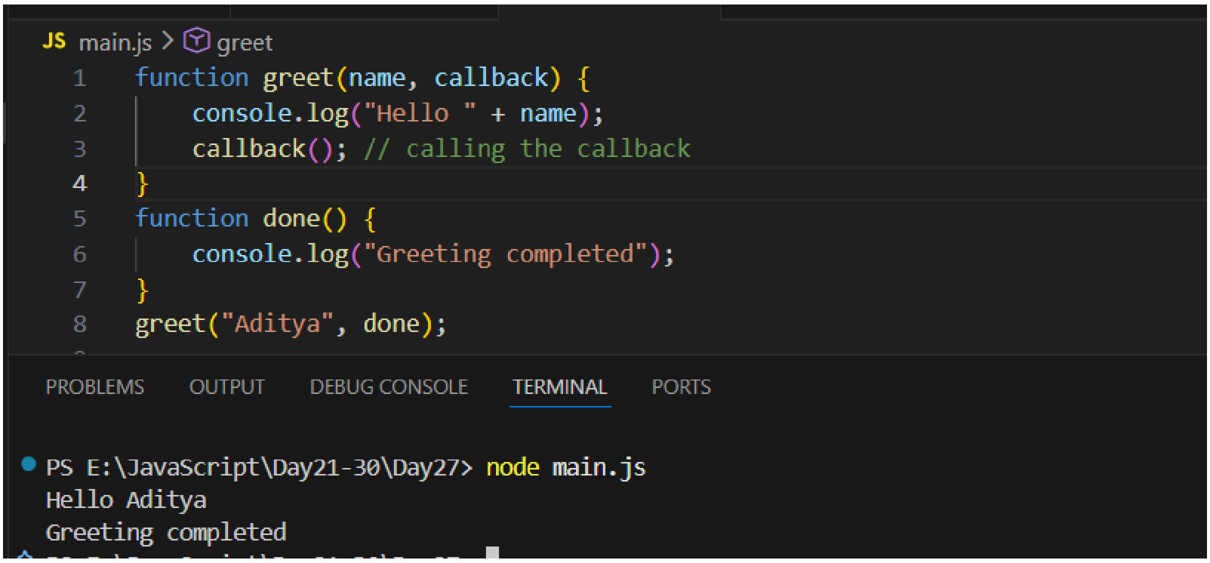
Why do we use Callbacks?

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.

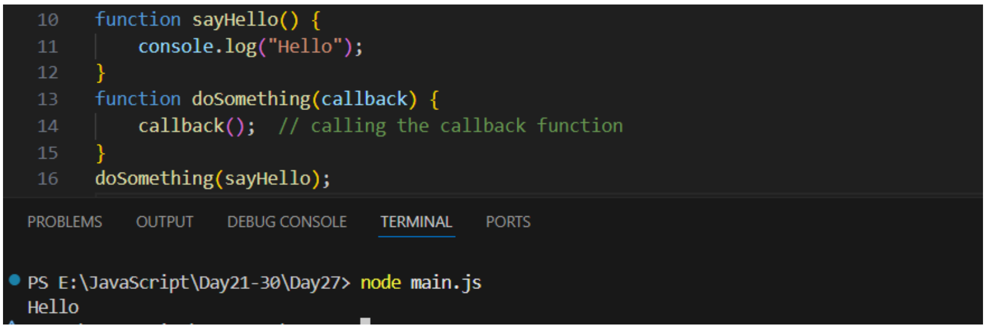
Example: a basic example of callback



```javascript
function greet(name, callback) {
    console.log("Hello " + name);
    callback(); // calling the callback
}
function done() {
    console.log("Greeting completed");
}
greet("Aditya", done);
```

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

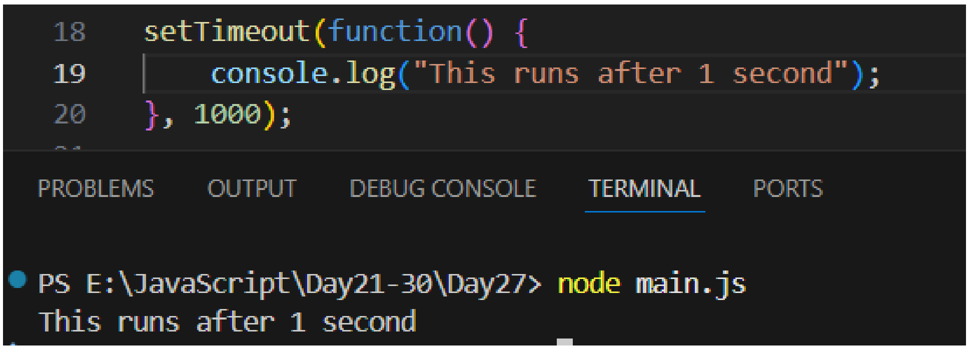
Example:



```javascript
function sayHello() {
    console.log("Hello");
}
function doSomething(callback) {
    callback();  // calling the callback function
}
doSomething(sayHello);
```

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

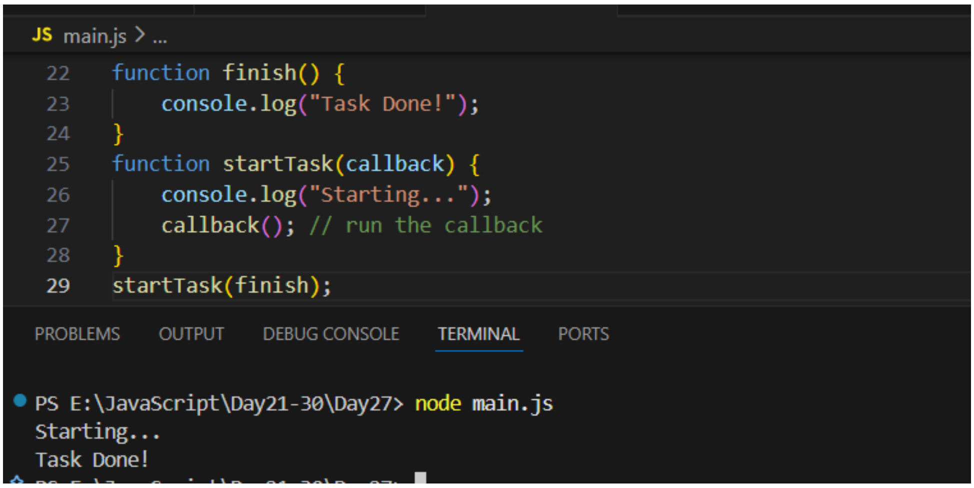
Example: The function inside setTimeout is a callback.



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

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

Example: callback with a message



```javascript
function finish() {
    console.log("Task Done!");
}
function startTask(callback) {
    console.log("Starting...");
    callback(); // run the callback
}
startTask(finish);
```

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