Gmail Bikash Adhikari <virtual.bikash@gmail.com>

Task 1.2: Simulate Multiple Async Tasks with Differ...

1 message

Bikash <bikash.caim@gmail.com> Sat, Mar 8, 2025 at 6:27 PM
To: virtual.bikash@gmail.com

JavaScript

Task 1.2: Simulate Multiple Async Tasks with Different Delays

Create a function `simulateMultipleTasks()` that starts three asynchronous tasks with different delays (1 second, 2 seconds, and 3 seconds).

Each task should log "Task [n] finished" where [n] is the task number. Ensure the tasks run asynchronously.

```
function simulateMultipleTasks() {
  console.log("Multiple tasks started");

  setTimeout(() => {
    console.log("Task 1 finished");
  }, 1000); // 1 second

  setTimeout(() => {
    console.log("Task 2 finished");
  }, 2000); // 2 seconds

  setTimeout(() => {
    console.log("Task 3 finished");
  }, 3000); // 3 seconds

  console.log("All tasks scheduled");
}

// Example usage:
simulateMultipleTasks();
```

Explanation:

1. `simulateMultipleTasks()` **Function:**
   - This function simulates starting three asynchronous tasks.
   - It begins by logging "Multiple tasks started."
   - It then uses three `setTimeout()` calls to schedule the execution of three separate tasks.
2. **Different Delays:**
   - Each `setTimeout()` call has a different delay: 1000 milliseconds (1 second), 2000 milliseconds (2 seconds), and 3000 milliseconds (3 seconds). This ensures that the tasks finish in the order of their delays.

3. **Task Completion Messages:**
    - Inside each `setTimeout()` callback function, a message is logged to the console indicating which task has finished (e.g., "Task 1 finished").
4. **Asynchronous Execution:**
    - The `setTimeout()` function is non-blocking. This means that the program doesn't wait for each task to finish before scheduling the next one.
    - The "All tasks scheduled" message will print almost immediately after the multiple tasks are started, before any of the tasks finish.
    - The tasks will then complete asynchronously, in the order of their delays.