Springboard

Asynchronous JavaScript Exercise

Hello, future software developer! In this exercise, you'll practice asynchronous JavaScript. ⏳

As the lead engineer 👷 at Intergalactic Espresso Explorers 💫, you'll run the operations of a space mission:

- 🔍 Continuously monitoring the space station conditions.
- 🗓️ Scheduling several one-time tasks.
- 🚀 Executing a thrilling countdown to a rocket launch.

Your tasks are:

1. **Declare the Task Array and the Interval ID**: Begin by declaring an array to hold your one-time tasks (`oneTimeTasks`) and variables for any interval IDs you'll need for continuous tasks (`monitoringTaskId`).
2. **Add One-Time Task Function**: Write a function named `addOneTimeTask` that accepts a function (`func`) and a delay (`delay`) as parameters. This function should add an object containing both parameters into the `oneTimeTasks` array.
3. **Run One-Time Tasks Function**: Create a function named `runOneTimeTasks` that iterates over the `oneTimeTasks` array and uses `setTimeout` to schedule each task according to its delay.
4. **Start Monitoring Function**: Write a function named `startMonitoring` that uses `setInterval` to simulate continuous monitoring. This function should print a message every few seconds and store the interval ID in `monitoringTaskId` .
5. **Stop Monitoring Function**: Implement a function named `stopMonitoring` that stops the continuous monitoring by using `clearInterval` on `monitoringTaskId` .

6. **Start Countdown Function**: Create a function named `startCountdown` that takes a duration parameter. Use `setInterval` to decrease the countdown every second and print the remaining time. Use `clearInterval` to stop the countdown when it reaches zero, printing a "Liftoff!" message.
7. **Schedule Pre-Launch Activities and Launch**: Use the functions you've created to schedule the pre-launch system check, start and stop monitoring, and execute the countdown. Make sure to adjust the delays appropriately to simulate a real mission timeline.
8. **Execute Your Script**: Run your script and watch your space mission come to life!

In the starter code, we left TODOs for you. Those are the places where you'll implement the above tasks.

Starter Code

Starter Code 2.zip 1.8KB

Happy coding!

▼ After you complete the exercise, please take a look at our solution:

Solution.zip 2.0KB