Task 8: Using Callbacks for Asynchronous Operations

Objective:

Implement a function that accepts a callback and executes it after a specified delay. Understand how callbacks are used in JavaScript to manage asynchronous operations.

Pre-requisites:

- Basic JavaScript (variables, functions)
- Understanding of asynchronous operations

Concepts Covered:

- Callbacks
- Asynchronous operations
- `setTimeout`

Setup:

Install Node.js:

- Ensure Node.js is installed on your machine. You can download it from nodejs.org.

Tasks:

1. **Define `delayedExecution` Function:**
    - **Task:**
        - Define a function named `delayedExecution` that takes two arguments:
            - `callback` : a function to be executed after the delay.
            - `delay` : the time in milliseconds to wait before executing the callback.
        - Inside the `delayedExecution` function, use `setTimeout` to call the callback function after the specified delay.
        - Ensure the callback function can accept parameters and those parameters are passed correctly.
    - **Outcome:**
        - Ensure the function correctly handles the callback and delay, and the callback function is executed after the specified time with the correct parameters.

Instructions:

- **Perform the following tasks:**
    - Write the required code in `index.js` .
    - Run the file using Node.js to ensure the code executes without errors and demonstrates the use of callbacks for asynchronous operations.

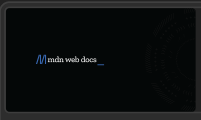
Example Input:

1. **Set 1:**

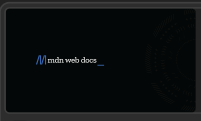
- Input:
  - `callback` : A function that logs "This is a delayed message."
  - `delay` : 1000 milliseconds
  - `message` : "This is a delayed message."
- Expected Output (after 1 second):
  - "This is a delayed message."

2. **Set 2:**
   - Input:
     - `callback` : A function that logs the sum of two numbers.
     - `delay` : 3000 milliseconds
     - `numbers` : 5 and 10
   - Expected Output (after 3 seconds):
     - "15"
3. **Set 3:**
   - Input:
     - `callback` : A function that logs the current time.
     - `delay` : 5000 milliseconds
   - Expected Output (after 5 seconds):
     - Current time in HH:MM format

Resources:

- 

  Callback function - MDN Web Docs Glossary: Definitions of Web-related term...
  A callback function is a function passed into another function as an argument, which is the...
  https://developer.mozilla.org/en-US/docs/Glossary/Callback_function
- 

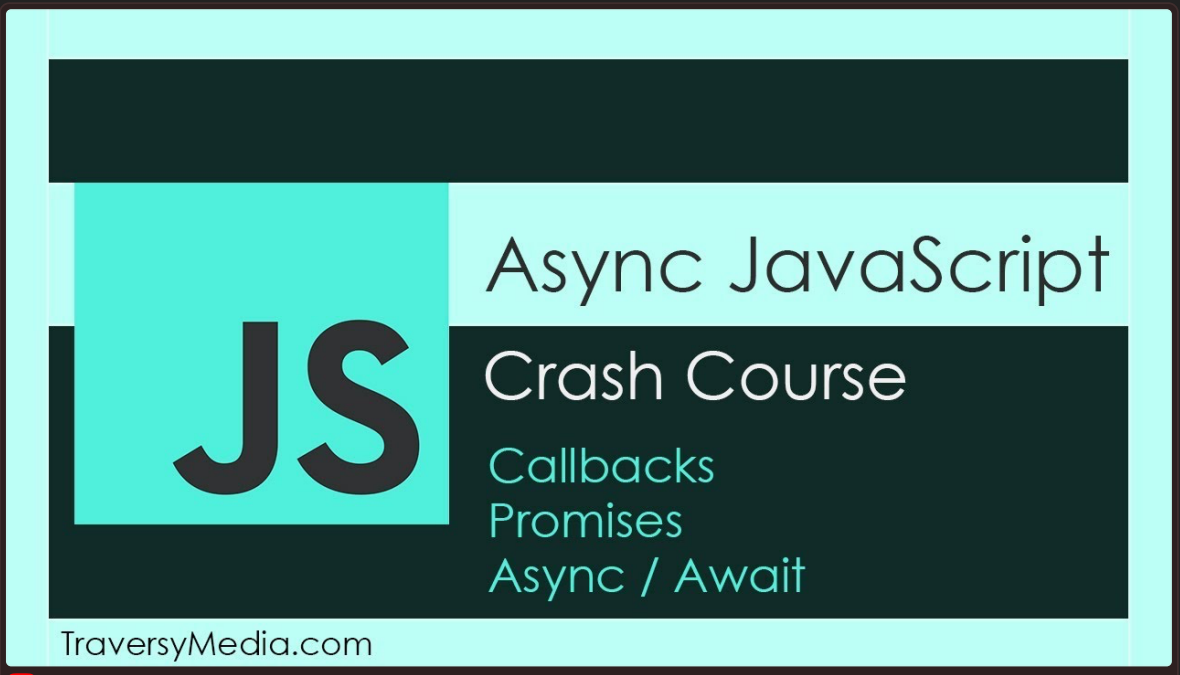
  Asynchronous JavaScript - Learn web development | MDN
  In this module, we take a look at asynchronous JavaScript, why it is important, and how it c...
  https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous
- 

  setTimeout() global function - Web APIs | MDN
  The global setTimeout() method sets a timer which executes a function or specified piece...
  https://developer.mozilla.org/en-US/docs/Web/API/WindowOrWorkerGlobalScope/setTimeout

Videos:

- 

  Akshay Saini 23:26 min 569,917 Views 19,031 Likes

- 

  Traversy Media 24:31 min 1,437,758 Views 37,284 Likes

- 

GitHub Instructions:

1. **Open in Visual Studio Code:**
   - After clicking on the "Open in Visual Studio Code" button from the GitHub Classroom confirmation page, Visual Studio Code (VSCode) will open the repository directly.
   - If prompted, select "Open" or "Allow" to open the repository in VSCode.
2. **Open the Terminal in VSCode:**
   - In VSCode, open a terminal by selecting Terminal > New Terminal from the top menu.
3. **Complete the Task:**
   - In VSCode, write your solution in the `index.js` file.
4. **Run and Test Your Code:**
   - In the VSCode terminal, navigate to the directory containing `index.js` .
   - Run your code to ensure it works correctly. Use the following commands:

     ```
     node index.js
     ```

5. **Commit Your Changes:**
   - In the VSCode terminal, add your changes to git:

     ```
     git add index.js
     ```

   - Commit your changes with a meaningful message:

```
git commit -m "Completed task 8"
```

6. **Push Your Changes to Your Repository:**
   - Push your changes to your forked repository:

```
git push origin main
```

7. **Create a Pull Request:**
   - Go to your repository on GitHub.
   - Click on the "Pull Requests" tab.
   - Click the "New Pull Request" button.
   - Ensure the base repository is the original template repository and the base branch is `main`.
   - Ensure the head repository is your forked repository and the compare branch is `main`.
   - Click "Create Pull Request".
   - Add a title and description for your pull request and submit it.

Summary of Commands:

```
# Open in Visual Studio Code

# Open terminal in VSCode

# Complete the task by editing index.js

# Navigate to the directory containing index.js
cd path/to/your/index.js

# Run your code
node index.js

# Add, commit, and push your changes
git add index.js
git commit -m "Completed task 8"
git push origin main

# Create a pull request on GitHub
```

Need Help?

Task 8 Solution

Code Hints — index.js: — Ensure your JavaScript file correctly defines the delayedExecution f...

w3o NFThing Last Edited 7/3/2024