JavaScript Switch Statement Guide

Objective

In this exercise, you will learn how to use the `switch` statement in JavaScript. A `switch` statement is used to perform different actions based on different conditions. It is a useful alternative to multiple `if/else` statements, especially when you need to compare a single variable to multiple values.

Instructions

Understanding the Code

You are provided with a blank `index.js` file. Your task is to fill in the file by completing the following steps.

Here's what you need to do:

1. **Declare the Variable for Day**

Declare a variable `day` using the `var` keyword and assign it the value `3`.

Declare a variable `dayName` using the `var` keyword and don't assign anything to it now.

2. **Using the `switch` Statement**

You will use the `switch` statement to determine which day it is based on the value of `day`.

- **Switch Structure**: Write a `switch` statement that checks the value of the variable `day`.
- **Cases**: Each `case` in the `switch` should correspond to a day of the week, and the `dayName` variable should be assigned the correct day name based on the value of `day` like `Monday`, `Tuesday` and so on. For default assign `Invalid day` to `dayName` variable.

You should use `case` for each day and a `default` case to handle invalid values for `day`.

3. **Output the Day Name**

After the `switch` statement, use `console.log()` to print the value of `dayName` to the console.

Mandatory Assessment Guidelines:

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to**
   Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.
3. **This editor Auto Saves the code.**
4. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**
5. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
6. **This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.**
   Note: The application will not run in the local browser
7. **You can follow series of command to setup Angular environment once you are in your project-name folder:**
   a. **npm install -> Will install all dependencies -> takes 10 to 15 min.**
   b. **node src/index.js -> To compile and run the index.js file.**
   c. **node src/test/custom-grader.js -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min.**
8. **Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on "Submit Assessment" after you are done with code.**

9. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**