Javascript-Else if and logical operators

Objective

In this exercise, you will learn how to use ternary operators in JavaScript. Ternary operators allow you to write concise conditional statements that can simplify your code. You will work with simple and nested ternary operators.

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 Variables for Comparison**

Declare two variables: `num1` and `num2`.

- **num1**: Declare a variable `num1` using the `var` keyword and assign it the value `10`.
- **num2**: Declare a variable `num2` using the `var` keyword and assign it the value `20`.

These variables will be used for the ternary operations.

2. **Using a Basic Ternary Operator**

You will use a simple ternary operator to compare `num1` and `num2`.

- **Condition**: Declare a variable with name `result` using `var` keyword and check if `num1` is greater than `num2`. If true, return "num1 is greater". Otherwise, return "num2 is greater" and save in it.

Once the condition is checked, you will use `console.log()` to print the result.

3. **Using a Nested Ternary Operator**

Now, you will use a **nested ternary operator** to perform a more complex check.

- **Condition 1**: Check if `num1` is greater than `num2`.
- **Condition 2**: If true, check if `num1` is equal to `10`. If true, return "num1 is 10". Otherwise, return "num1 is greater than num2".
- **If false**: Return "num2 is greater".

Declare a variable with name `comparison` using `var` keyword and store above result.
Once the conditions are checked, you will use `console.log()` to print the result.

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.**