JavaScript Variables, String Manipulation, and Arithmetic Guide

In this exercise, you will learn how to work with different types of variables, string manipulation techniques, and perform basic arithmetic operations in JavaScript. You will declare variables, use string concatenation and template literals, and output the results using `console.log()`.

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**

Declare variables using different types: `var`, `let`, and `const`.
- **Name**: Declare a variable `name` using the `var` keyword. Assign it the value "Alice".
- **Age**: Declare a variable `age` using the `let` keyword. Assign it the value `30`.
- **City**: Declare a constant variable `city` using the `const` keyword. Assign it the value "Wonderland".

2. **String Manipulation**

- **Greeting**: Create a variable `greeting` using the `let` keyword and use **string concatenation** to combine the following elements:
  - "Hello, my name is "
  - `name` (variable)
  - " and I am "
  - `age` (variable)
  - " years old."

- **Introduction**: Create a variable `introduction` using the `let` keyword and use **template literals** to create a string that says: "I live in [city]."

3. **Arithmetic Operations**

- **Doubling Age**: Create a variable `doubledAge` and use the `age` variable to calculate your doubled age.

- **Age Next Year**: Create a variable `ageNextYear` and add one year to your current age.

- **Age Plus 5**: Create a variable `sum` and add 5 to your current age.

4. **Log the Results**

Now, use `console.log()` to display the results of the variables and calculations.
- **Greeting**: Print the `greeting` variable.
  // This should print: "Hello, my name is Alice and I am 30 years old."

- **Introduction**: Print the `introduction` variable.
  // This should print: "I live in Wonderland."

- **Double Age**: Print the `doubledAge` variable.
  // This should print: "Double age: 60"

- **Next Year's Age**: Print the `ageNextYear` variable using a template literal.
// This should print: "Next year, I'll be 31 years old."

- **Age Plus 5**: Print the `sum` variable.
// This should print: "Age plus 5: 35"

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.