

JavaScript Alert, Prompt, and Confirm Guide

In this exercise, you will learn how to use the `alert()`, `prompt()`, and `confirm()` methods in JavaScript. These methods allow you to interact with users by displaying messages, getting user input, and asking for confirmations.

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. **Using `alert()` to Display a Message**

You should use `alert()` to display the following message:

- **Message**: "Welcome to the JavaScript World!"

2. **Using `prompt()` to Get User Input**

- **Question to ask the user**: "What is your name?"

You need to store the response in a variable called `userName` declared using `var` keyword.

When the user types in their name and clicks "OK", it will be stored in the `userName` variable.

3. **Using `confirm()` to Get Yes/No Confirmation**

In this case, you should ask the user whether they are a student.

- **Question to ask the user**: "Are you a student?"

You need to store the response in a variable called `isStudent` declared using `var` keyword.

If the user clicks "OK", `isStudent` will be `true`; if they click "Cancel", `isStudent` will be `false`.

4. ****Display the Results in the Console****

After gathering the information, you will display the results in the console.

- ****User's Name****: Print the ``userName`` variable to the console to show the user's name.
 - This should print: ``User's Name: [name]``
- ****Is User a Student?****: Print the ``isStudent`` variable to the console to show whether the user is a student.
 - This will should: ``Is user a student? true/false``

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 --no-bin-links --unsafe-perm ->` 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.