

JavaScript Math Object and Arithmetic Operations Guide

In this exercise, you will explore the JavaScript `Math` object and perform various arithmetic operations. You will learn to use `Math` methods to perform operations like square roots, powers, random numbers, and rounding. Additionally, you will perform basic arithmetic operations such as addition, subtraction, multiplication, and division.

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****

- ****num1****: Declare a variable `num1` using the `var` keyword. Assign it the value `16`.
- ****num2****: Declare a variable `num2` using the `var` keyword. Assign it the value `4`.

2. ****Using Math Methods****

- ****Square Root****: Declare a variable `sqrt` using the `var` keyword and use the `Math.sqrt()` method to calculate the square root of `num1` and save in it.
- ****Exponentiation****: Declare a variable `power` using the `var` keyword and use the `Math.pow()` method to raise `num2` to the power of `2` and save in it.
- ****Random Number****: Declare a variable `randomNum` using the `var` keyword and use the `Math.random()` method to generate a random number between `0` and `1` and save in it.
- ****Rounded Random Number****: Declare a variable `rounded` using the `var` keyword and multiply the `randomNum` by `100` and use the `Math.round()` method to round the result and save in it.

3. ****Arithmetic Operations****

- ****Addition****: Declare a variable `addition` using the `var` keyword and add `num1` and `num2` together and save in it.

- ****Subtraction****: Declare a variable ``subtraction`` using the ``var`` keyword and subtract ``num2`` from ``num1`` and save in it.
- ****Multiplication****: Declare a variable ``multiplication`` using the ``var`` keyword and multiply ``num1`` by ``num2`` and save in it.
- ****Division****: Declare a variable ``division`` using the ``var`` keyword and divide ``num1`` by ``num2`` and save in it.
- ****Modulus****: Declare a variable ``modulus`` using the ``var`` keyword and calculate the modulus of ``num1`` and ``num2`` (the remainder when ``num1`` is divided by ``num2``) and save in it.
- ****Exponentiation****: Declare a variable ``exponentiation`` using the ``var`` keyword and use the ``Math.pow()`` method to raise ``num2`` to the power of ``3`` and save in it.

4. ****Log the Results****

Now, use ``console.log()`` to display the results of the variables and calculations.

- ****Square Root****: Print the square root of ``num1``.
`// This should print: "Square root of 16 is 4"`
- ****Squared Number****: Print ``num2`` squared.
`// This should print: "4 squared is 16"`
- ****Random Number****: Print the random number generated.
`// This should print: "Random number between 0 and 1: [random number]"`
- ****Rounded Random Number****: Print the rounded random number.
`// This should print: "Random number rounded to 100: [rounded number]"`
- ****Addition****: Print the result of adding ``num1`` and ``num2``.
`// This should print: "16 + 4 = 20"`
- ****Subtraction****: Print the result of subtracting ``num2`` from ``num1``.
`// This should print: "16 - 4 = 12"`
- ****Multiplication****: Print the result of multiplying ``num1`` by ``num2``.
`// This should print: "16 * 4 = 64"`
- ****Division****: Print the result of dividing ``num1`` by ``num2``.
`// This should print: "16 / 4 = 4"`

- ****Modulus****: Print the modulus of `num1` and `num2`.

// This should print: "16 % 4 = 0"

- ****Exponentiation****: Print the result of raising `num2` to the power of 3.

// This should print: "4 raised to the power of 3 is 64"

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.