Node Modules and Node Server - Practice Exercises

Questions on Node Modules

1. Create a module called "**myFirst**":
   a. Inside the "**myFirst**" module, log a simple text that says "**My first module**" on the console.
   b. Run the module and see the output of the code you just wrote in this module
   c. Write a function called "**myMultiplier**" inside of your "**myFirst**" module. This function takes a single number as a parameter and returns a value that is 2 times the parameter
   d. Execute the function inside the module. Pass the number 4 to the function
   e. Save the returned value in a variable and log the variable on the console. Now, run your module on the terminal to see the printed output
   f. Export your "**myMultiplier**" function so that other modules can use it

2. Create another module called "**mySecond**":
   a. Add another function in this module with the same name as above (**myMultiplier**). This function takes a number as a parameter and returns the value that is 3 times the parameter
   b. Execute the function inside the module. Pass the number 4 to the function
   c. Save the returned value in a variable and log the variable on the console. Now, run your module on the terminal to see the printed output
   d. Export your "**myMultiplier**" function so that other modules can use it

3. Create another module called "**myCollector**":
   a. Import both functions from "**myFirst**" and "**mySecond**" modules inside of your "**myCollector**" module
   b. Pass the value 5 to both functions that are imported from "**myFirst**" and "**mySecond**" modules
   c. Run the "**myCollector**" module on your terminal to display the outputs on your console

4. While you are in your "**myCollector**" module:
    a. Write a script inside of your "**myCollector**" module that passes the number 14 to your "**myMultiplier**" function that you imported from "**myFirst**" module and writes the returned value on a file called "**results.txt**". The result you write on the file should read like this: "*The value of 14 when passed through the myMultiplier function is ( )*."
        - **Hint**: You will need to find the core Node module that will allow you to create the "**results.txt**" file and write the result on this file
    b. Write another script inside of your "**myCollector**" module that passes the number 14 to your "**myMultiplier**" function that you imported from "**mySecond**" module and writes the returning value on the same file, the "**results.txt**" on a new line. The result you write on the file should read like this:
    "*The value of 14 when passed through the myMultiplier function is (   )*".
        - **Note**: Make sure not to replace/remove what you wrote on your "**results.txt**" file previously.
        - **Note**: Also, make sure to add the new result on a new line, right below the result written previously.

Questions on Node Server

Use the "HTTP" module for the questions from 5 to 8

5. Create a module called my "**randomNumber**".
    a. The "**randomNumber**" module has a function called random(). The random function just returns a random number when it gets executed
    b. Execute the function inside the module
    c. Save the returned value in a variable and log the variable on the console. Now, run your module on the terminal to see the printed output
    d. Export your module so that it is accessible to other module

6. Create "**myWebServer**" module
    a. Inside of your myWebServer module, create a web server which listens to requests on port 1234
        - Note: It is a good practice to write a custom message such as **console.log("Server running")** in your server listener you create to check if your server is running and listening to requests
    b. Run your "**myWebServer**" module on your terminal to check if your server is listening to requests at port 1234
    c. Inside of your "**myWebServer**" module, write your request listener function as a callback inside of the server you created above. This function should return the following text message "*Request received and processed*" to the browser.
    d. Run your "**myWebServer**" module on your terminal and go to your browser and type "localhost:1234" to check if your browser displays the "*Request received and processed* " message sent from your server

7. Inside your "**myWebServer**" module
    a. Import your "**randomNumber**" module inside of your "**myWebServer**" module. Now, generate a random number using the random() function from the "**randomNumber**" module and return the generated random number to the client browser when request is sent to port 1234.
        - Don't forget to run your module on your terminal to keep your server running and
        - Go to your browser and type "localhost:1234" to see the random number generated

8. Create a new folder called "**static**"
    a. Inside the "**static**" folder, save the **"apple html css replica"** folder by downloading and extracting it
    b. Add a sample "**about.html**" page inside of your "**apple html css replica"** folder you just downloaded. Open your "**about.html**" file and add the text '*This is coming from my "about page* " '

c. Modify your request listener function in a way it would serve the "**about.html**" page when users request it on the browser

   - **Hint**: You will need to import additional node modules to display your "**about.html**" page when users request it on the browser

d. Don't forget to run your module on your terminal to keep your server running. Now, go to your browser and type "localhost:1234" to see the ‘*This is coming from my "about page* " ’ text sent to your browser

e. Now, modify your listener function in a way that it serves <u>any of the pages</u> inside of your "**static**" folder when requested

Use the "Express" module for question 9

9. Solve questions 6 - 8 using the Express module

<u>Hint on how to organize your file/folder and use VSC terminal</u>

- ### For Node modules questions
  - By now, you must have downloaded and installed Node on your computer. To check if you have installed Node, open the integrated terminal in VSC and type the following command
    - node -v
    - If you have installed Node, it will tell you the version you have
  - Create a folder called "**nodeModulesProject**" in your Evangadi folder
  - It is inside your "**nodeModulesProject**" folder, you will create the following Node module files:
    - myFirst.js
    - mySecond.js
    - myCollector.js
  - While writing code in response to the questions, you will need to run each module to see the output of the code you write. To do that:

- Open your “**nodeModulesProject**” folder with VSC and right click on one of your modules (Example: myFirst.js) and select “**Open in Integrated Terminal**” option from the list
- Run your module to see the output on the terminal. For example, if you want to see the output of the code you wrote in your myFirst.js module, type the below command on your VSC terminal and hit enter:
    - node myFirst.js

- **For Node server questions (HTTP and Express web server building)**
    - Create a folder called “**nodeServerProject**” in your Evangadi folder. It is in this folder that you will create the following 2 node modules and download/save the “static” folder:
        - randomNumber.js
        - myWebServer.js
        - “static” folder
    - Open your “**nodeServerProject**” folder with VSC
    - Right click on your “**nodeServerProject**” folder and open it in your integrated terminal. Make sure to run “**npm init**” on your terminal to set up a new npm package
    - **Note**: Do not forget to import/require all the core modules you need for your project (such as HTTP module)
    - **Note**: For the question on Express, you will need to install Express module and require/import in the module that will use Express
        - Express module is not one of the core/ built-in modules in Node. Meaning, it needs to be installed as it is not readily available to your Node projects like HTTP module. Please run the following command on your terminal to install and save Express in your project:
            - npm install express --save
        - Once you installed Express, make sure to import/require it in your module

- **General note on exporting and importing**
    - Just creating a new module is not enough. In order for other modules to use any module you write, you will need to export the module
    - To use an exported module in your other modules, make sure you import it

Happy coding 😀