Task 3: Understanding Node.js Modules

Objective:

Create custom modules and import them using `require`. Ensure the main script correctly imports and uses the functionality from custom modules.

Prerequisites:

- Basic understanding of JavaScript.
- Node.js installation.

Concepts:

- **Creating Modules:**
  - Modules are reusable pieces of code that can be exported and imported in other files.
  - In Node.js, you can define a module by exporting objects, functions, or variables using `module.exports` or `exports`.

  Example:

```
// mathModule.js
const add = (a, b) => a + b;
const subtract = (a, b) => a - b;
module.exports = { add, subtract };
```

- **Importing Modules:**
  - To use a module in another file, use the `require` function.
  - The `require` function takes the path to the module and returns the exported object.

  Example:

```
// main.js
const math = require('./mathModule');
console.log(`Addition: ${math.add(5, 3)}`); // Output: Addition: 8
console.log(`Subtraction: ${math.subtract(5, 3)}`); // Output: Subtraction: 2
```

Setup:

1. **Install Node.js:**

   Ensure Node.js is installed on your machine. You can access the instructions here:

   - 

     Detailed Instructions for Installing Node.js and NPM and integration...

     For Windows: — Download Node.js Installer: — Visit the official Node.js website:...

     w3o NFThing Last Edited 7/3/2024

Tasks:

1. **Creating a Math Module:**

- **Task:**
  - Create a file named `mathModule.js` and write your own code that:
    - Defines functions to add, subtract, multiply, and divide two numbers.
    - Exports these functions using `module.exports`.
- **Outcome:**
  - Ensure the module exports the functions correctly.

2. **Creating a String Utilities Module:**
   - **Task:**
     - Create a file named `stringUtils.js` and write your own code that:
       - Defines functions to convert a string to lowercase, uppercase, and to reverse a string.
       - Exports these functions using `exports`.
   - **Outcome:**
     - Ensure the module exports the functions correctly.
3. **Importing and Using Modules:**
   - **Task:**
     - Create a file named `main.js` and write your own code that:
       - Imports the `mathModule` and `stringUtils` modules.
       - Uses the imported functions to perform some operations and logs the results.
   - **Outcome:**
     - Ensure the main script correctly imports and uses the functionality from custom modules.

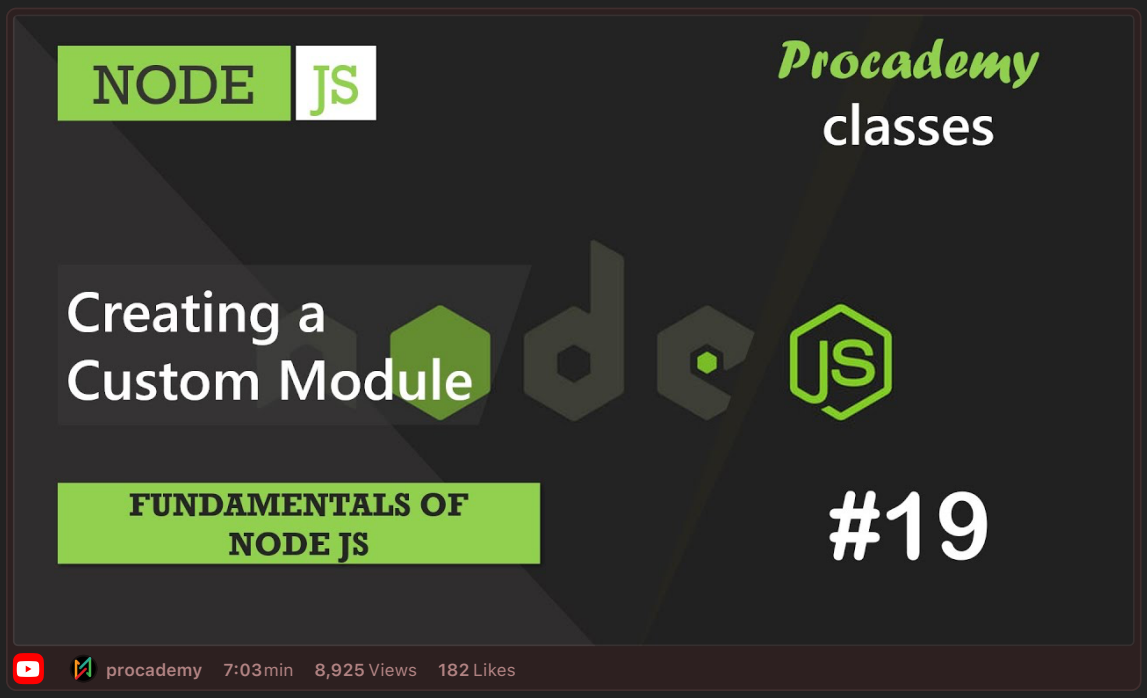
Instructions:

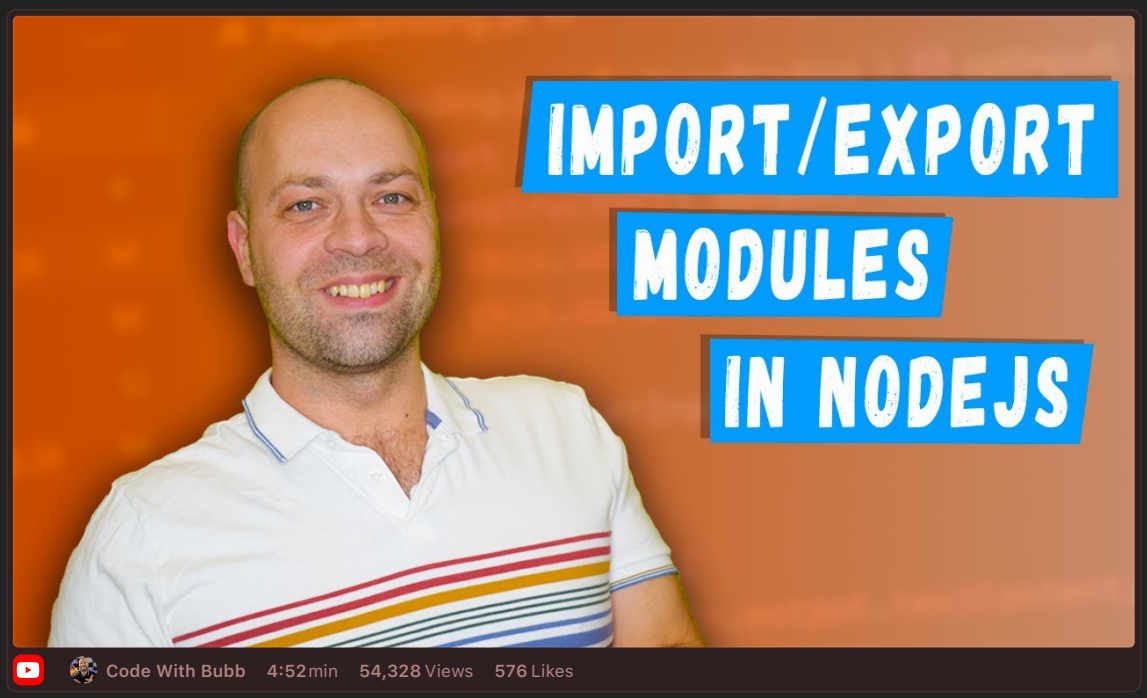
- **Perform the following tasks:**
  - Write the required code in separate files (`mathModule.js`, `stringUtils.js`, `main.js`).
  - Run each file using Node.js to ensure the code executes without errors and demonstrates the use of Node.js modules.

Resources:

- 
  Modules: CommonJS modules | Node.js v22.4.0 Documentation
  CommonJS modules are the original way to package JavaScript code for Node.js. — Node.js also sup...
  https://nodejs.org/api/modules.html
- 
  Index | Node.js v22.3.0 Documentation
  https://nodejs.org/en/docs/

Videos:

- 

- 

GitHub Instructions:

1. **Open in Visual Studio Code:**
   - After clicking on the "Open in Visual Studio Code" button from the GitHub Classroom confirmation page, Visual Studio Code (VSCode) will open the repository directly.
   - If prompted, select "Open" or "Allow" to open the repository in VSCode.
2. **Complete the Task:**
   - Write your solution in the respective files (`mathModule.js` , `stringUtils.js` , `main.js`).
3. **Run and Test Your Code:**

- Run your code to ensure it works correctly. Use the following command for each file:

```
node <filename>.js
```

4. **Commit Your Changes:**
   - Commit your changes with a meaningful message:

```
git commit -m "Completed Node.js Modules task"
```

5. **Push Your Changes to Your Forked Repository:**
   - Push your changes to your forked repository:

```
git push origin main
```

6. **Create a Pull Request:**
   - Go to your forked repository on GitHub.
   - Click on the "Pull Requests" tab.
   - Click the "New Pull Request" button.
   - Ensure the base repository is the original template repository and the base branch is `main`.
   - Ensure the head repository is your forked repository and the compare branch is `main`.
   - Click "Create Pull Request".
   - Add a title and description for your pull request and submit it.

Summary of Commands:

```
# Fork the repository on GitHub

# Clone the forked repository
git clone https://github.com/your-github-username/repository-name.git
cd repository-name

# Complete the task by writing and running the code in the specified files

# Add, commit, and push your changes
git commit -m "Completed Node.js Modules task"
git push origin main

# Create a pull request on GitHub
```