Test-driven development

Test-Driven Development (TDD) can be implemented in JavaScript using the Mocha testing framework. Mocha is a popular JavaScript testing framework that allows you to write and run tests for your code. Here's a step-by-step guide on how to practice TDD using Mocha in JavaScript:

**Note:** Before you begin, make sure you have Node.js and npm (Node Package Manager) installed on your computer.

[Installation of Node.js on Windows - GeeksforGeeks](https://www.geeksforgeeks.org/installation-of-node-js-on-windows/)

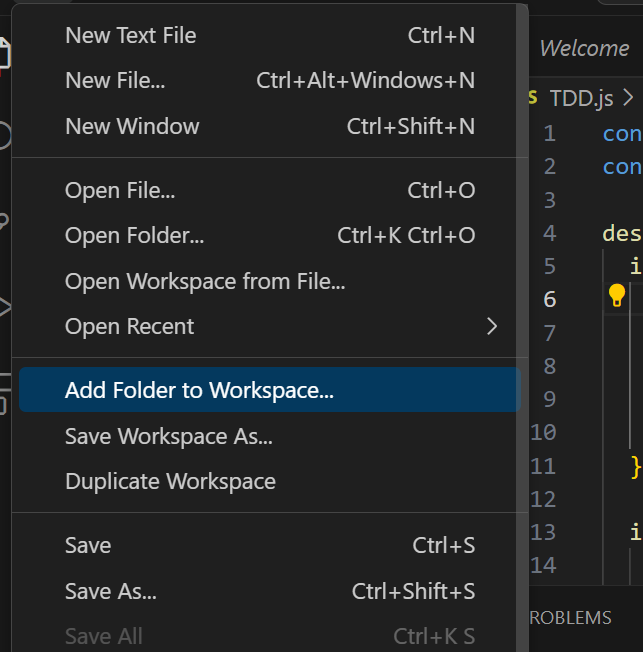
[What are the differences between npm and npx ? - GeeksforGeeks](https://www.geeksforgeeks.org/what-are-the-differences-between-npm-and-npx/)

Npm- Node package manager

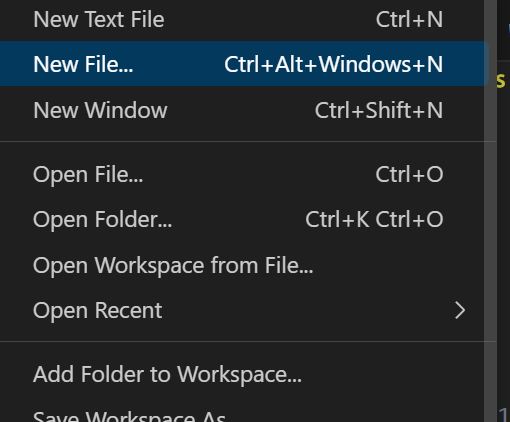
Npx- Node package execute.

Open Visual studio

**Initialize a JavaScript file:**



Add new File to that folder:



Save the file with name:

**Create a Function to Test:**

Create a JavaScript file TDD.js where you'll write the code you want to test. For example, let's create a simple function that adds two numbers, multiplication etc.

TDD.js

const chai = require('chai');

const expect = chai.expect;

describe('My Feature', () => {

  it('Test Case 1', () => {

    // Test code

    const res = 2 + 2;

    // Assertion

    expect(res).to.equal(4); // Test result

  });

  it('Test Case 2', () => {

    // Test code

    const res = 5 - 8;

    // Assertion

    expect(res).to.equal(-3);

  });

  it('Test Case 3', () => {

    // Test code

    const res = 3 \* 6;//functanality

    // Assertion

    expect(res).to.equal(18);

  });

  it('Test Case 4', () => {

    // Test code

    const res = 12 / 3;

    // Assertion

    expect(res).to.equal(4);

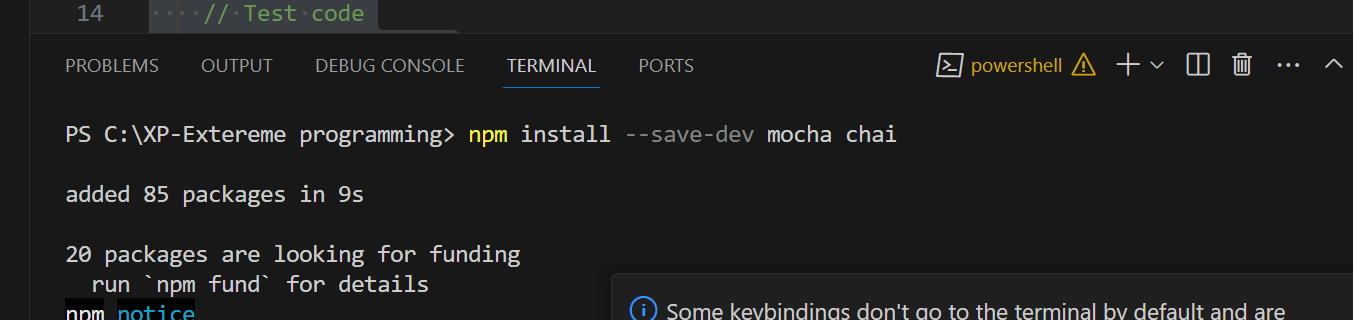
  });

});

Click on Terminal

**Install Mocha and Chai:**

Mocha is a testing framework, and Chai is an assertion library that pairs well with Mocha. You'll install both as development dependencies:



Use command : npm install --save-dev mocha chai

**Run Tests:**

Run your tests using Mocha by executing the following command in your terminal:

npx mocha TDD.js

My Feature

✔ Test Case 1

✔ Test Case 2

✔ Test Case 3

✔ Test Case 4

4 passing (20ms)

My Feature

1) Test Case 1

✔ Test Case 2

✔ Test Case 3

✔ Test Case 4

3 passing (24ms)

1 failing

1) My Feature

Test Case 1:

AssertionError: expected 5 to equal 4

+ expected - actual

-5

+4

at Context.<anonymous> (TDD.js:10:20)

at process.processImmediate (node:internal/timers:478:21)

---------------------------END--------

npm, which stands for "Node Package Manager," is a package manager for the Node.js JavaScript runtime environment. It is one of the largest package ecosystems in the world and is widely used by developers to manage and distribute JavaScript libraries and tools. npm serves several important purposes in the world of JavaScript development:

1. **Dependency Management:** npm allows developers to declare and manage project dependencies. You can specify which third-party libraries or modules your project relies on in a **package.json** file, and npm will automatically download and install these dependencies for you. This simplifies the process of managing and updating external code libraries.
2. **Package Distribution:** Developers can publish their own JavaScript libraries and packages to the npm registry. This makes it easy for others to discover, install, and use these packages in their own projects. npm makes it simple to share code with the broader developer community.
3. **Version Control:** npm provides versioning for packages, allowing you to specify which version of a package your project depends on. This ensures that your project's dependencies remain consistent and that you can upgrade packages when necessary without breaking existing functionality.
4. **Scripts and Automation:** npm allows you to define custom scripts in your **package.json** file. These scripts can automate various development tasks, such as running tests, building your project, or deploying it to a production server. This makes it easy to create a standardized development workflow for your projects.
5. **Global and Local Installation:** npm can install packages globally, making them available as command-line tools, or locally, making them available only within a specific project. This flexibility allows you to manage both project-specific dependencies and development tools.
6. **Security:** npm has security features that help identify and mitigate security vulnerabilities in packages. It scans packages for known vulnerabilities and provides tools for developers to update packages with security fixes.
7. **Scripts and Binaries:** npm allows package authors to define scripts and binaries that can be executed from the command line. This is useful for creating command-line utilities and tools that can be easily installed and used globally.
8. **Community and Collaboration:** npm fosters collaboration among developers by providing a centralized platform for sharing code. Developers can collaborate on open-source projects, report issues, and contribute to the improvement of packages hosted on the npm registry.

In summary, npm is a crucial tool for JavaScript developers that simplifies the management of dependencies, facilitates package distribution, and provides a range of tools for automating development tasks. It has played a significant role in the growth and popularity of the JavaScript ecosystem and has become an essential part of modern web and server-side development.

C:\XP-Extereme programming> npm install --save-dev mocha chai

added 85 packages in 9s

20 packages are looking for funding

run `npm fund` for details

npm notice

npm notice New major version of npm available! 9.6.6 -> 10.1.0

npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.1.0

npm notice Run npm install -g npm@10.1.0 to update!

npm notice

PS C:\XP-Extereme programming> npx mocha TDD.js

0 passing (2ms)

PS C:\XP-Extereme programming> npx mocha TDD.js

My Feature

✔ Test Case 1

✔ Test Case 2

✔ Test Case 3

✔ Test Case 4

4 passing (19ms)

PS C:\XP-Extereme programming> npx mocha TDD.js

My Feature

1) Test Case 1

✔ Test Case 2

✔ Test Case 3

✔ Test Case 4

3 passing (22ms)

1 failing

1) My Feature

Test Case 1:

AssertionError: expected 5 to equal 4

+ expected - actual

-5

+4

at Context.<anonymous> (TDD.js:10:20)

at process.processImmediate (node:internal/timers:478:21)

PS C:\XP-Extereme programming>

\* History restored

PS C:\XP-Extereme programming> npx mocha TDD.js

My Feature

✔ Test Case 1

✔ Test Case 2

✔ Test Case 3

✔ Test Case 4

4 passing (17ms)

PS C:\XP-Extereme programming>

\* History restored

PS C:\XP-Extereme programming>

\* History restored

PS C:\XP-Extereme programming>