

Unit Testing in Frontend Development

irwansahrulsidik.com

github.com/irwansah



Objective of Unit Testing

- ★ Ensure code quality and reliability
- ★ Identify and fix bugs early in the development process
- ★ Facilitate refactoring and code maintenance
- ★ Enhance collaboration between developers



Common Testing Frameworks

Jasmine

[Jasmine Documentation](#)

Jest/Vitest

[Jest · !\[\]\(cbe80b694ebd74fcfe136a095b608235_img.jpg\) Delightful JavaScript Testing \(jestjs.io\)](#)

[Vitest | A blazing fast unit test framework powered by Vite](#)

Mocha

[Mocha - the fun, simple, flexible JavaScript test framework](#)

Karma

<https://karma-runner.github.io>



Testing Tools and Libraries

Tools

Cypress (JS) [JavaScript Component Testing and E2E Testing Framework | Cypress](#)

Playwright (JS, Python, Java, .NET) [Fast and reliable end-to-end testing for modern web apps | Playwright](#)

Libraries

Enzyme (for React)

Vue Test Utils (for Vue.js)

React Testing Library



Basic Structure of a Unit Test

Arrange: Set up the test environment and create necessary objects and data.

Act: Execute the code or function being tested.

Assert: Verify the expected outcomes and check if they match the actual results.



Writing a Unit Test

- **Use descriptive test names for clear understanding.**
- **Use test suites and test cases to organize tests.**
- **Cover edge cases and boundary conditions.**





Example

Best Practices for Unit Testing

- Keep tests independent and isolated from each other.
- Avoid testing implementation details; focus on behavior.
- Run tests frequently during development.
- Maintain a good test coverage percentage.
- Refactor tests as the codebase evolves.





Conclusion



Thanks