

QA

ADVANCED

4 DAYS

End-to-End Testing with Cypress (JavaScript)

Build a strong foundation in end to end test automation using Cypress by mastering its core capabilities and advanced testing techniques. Learn how to design maintainable test suites, integrate Cypress with CI/CD pipelines, optimize execution, and troubleshoot tests for scalable and reliable quality assurance.

Why Cypress (JavaScript)?

Faster E2E feedback

Automate critical user journeys in Cypress with JavaScript.

API + UI coverage

Stub networks and validate full flows.

Stable selectors and waits

Reduce flake with modern Cypress patterns.

CI-ready suites

Publish Cypress reports in pipelines.

What you'll be able to do

- ✓ Gain a solid foundation in the principles, benefits, and importance of end-to-end testing.
- ✓ Acquire a comprehensive knowledge of Cypress and its capabilities as a JavaScript-based testing framework.
- ✓ Discover strategies for organizing and structuring test suites to ensure maintainability and reusability.
- ✓ Dive into advanced topics such as handling asynchronous behavior, interacting with external APIs, performing cross-browser testing, and optimizing test execution speed.
- ✓ Understand how to integrate Cypress into continuous integration and deployment (CI/CD) pipelines.
- ✓ Develop troubleshooting skills for efficient test debugging.
- ✓ Understand test maintenance and scalability strategies.

Who this is for

- Software Developers
- Quality Assurance Engineers
- Solution Architects
- Test Automation Specialists

Curriculum

01 Cypress Foundations

- Apply cypress foundations using Cypress (JavaScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

02 Project Setup

- Apply project setup using Cypress (JavaScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

03 Selectors and Commands

- Apply selectors and commands using Cypress (JavaScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

04 Assertions

- Apply assertions using Cypress (JavaScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

05 Network Stubbing

- Apply network stubbing using Cypress (JavaScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

06 Page Objects

- Apply page objects using Cypress (JavaScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

07 API Testing in Cypress

- Apply api testing in cypress using Cypress (JavaScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

08 Fixtures and Data

- Apply fixtures and data using Cypress (JavaScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

09 CI Reporting

- Apply ci reporting using Cypress (JavaScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

10 Flake Control

- Apply flake control using Cypress (JavaScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

11 Best Practices

- Apply best practices using Cypress (JavaScript) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks