

SuperSQA

QA AUTOMATION ENGINEER

Your Complete **Roadmap** & Execution Plan

The 8-Step Execution Plan

1

Learn a Programming Language

Master the fundamentals of a language like Python.

2

Learn Testing Basics

Understand the core concepts and terminology of QA.

3

Learn Web Testing Tools

Automate browsers with industry-standard tools like Selenium.

4

Learn SQL

Query and validate data directly from the database.

5

Learn API Testing

Test the backend of applications for faster, more stable checks.

6

Learn Docker

Understand modern engineering workflows and create consistent environments.

7

Learn CI/CD Tools

Automate your test execution with tools like Jenkins or GitHub Actions.

8

Keep Up-Skilling

Continuously learn new tools and build projects to advance your career.

1

Learn a Programming Language

This is the non-negotiable foundation of your entire career. Without a solid grasp of a programming language, you cannot build robust and maintainable automation.

What to Learn:

Py Python (Recommended) Java JavaScript

- **Python is the easiest language to start with.** Its simple syntax allows you to focus on automation concepts rather than complex language rules.
- It's growing fast and is used everywhere, not just in QA. This opens up future career paths.

Key Takeaway:

Don't get stuck in "language analysis paralysis." Pick one (preferably Python), and commit to learning its fundamentals deeply. This is the highest-leverage skill you will ever acquire.

2

Learn Testing Basics

You don't need to become a master of manual testing theory, but you need to speak the language of QA to be effective in your role and to pass interviews.

Key Terminology:

- | | |
|-----------------|-------------------------|
| - Test Case | - Severity vs. Priority |
| - Bug Lifecycle | - Regression Testing |
| - Test Plan | - Smoke & Sanity Tests |

- You only need the basics to get started. This is not a deep dive.
- Understanding these concepts will make the purpose of your automation tools much clearer.

Pro Tip:

Don't waste time here. Your goal is to spend a maximum of 2-3 days on this topic. The real learning happens when you start building with a testing tool.

3

Learn Web Testing Tools

This is where you apply your programming skills to control a browser and automate the frontend of a web application. It's often the first type of automation a new SDET will write.

What to Learn:

Se

Selenium (Recommended)

Playwright

Cypress

- **Selenium still dominates the job market, so it's my top recommendation.**
- The tool itself is not difficult; your mastery of the programming language is what makes you effective.

Key Takeaway:

Focus on understanding locators (e.g., CSS Selectors, XPath) and synchronization (waits). These are the two areas where most beginners struggle with flaky tests.

4

Learn SQL (Database)

Nearly every application you test will have a database behind it. Being able to directly query that database is a superpower for a tester.

Core Commands to Master:

- | | |
|-------------------|------------------|
| - SELECT ... FROM | - WHERE |
| - INSERT INTO | - UPDATE ... SET |
| - DELETE FROM | - INNER JOIN |

- SQL is used by both manual and automation testers.
- It allows you to set up test data, verify results, and debug issues far more efficiently than using the UI alone.

Pro Tip:

Focus on writing queries that answer a specific testing question, like "Did the user's record get created correctly after I submitted the form?"

5

Learn API Testing

Testing the backend directly via APIs is often more important than UI testing. It's faster, more stable, and allows you to catch bugs earlier in the development cycle.

Tools & Libraries:



Postman



Python (Requests, Pytest)

- API testing often makes up more than 50% of automation in a real job.
- Learning this skill early gives you a massive advantage over other junior candidates.

Key Takeaway:

API tests are less brittle than UI tests because they aren't affected by changes in web page design. This makes them a cornerstone of a good automation strategy.

6

Learn Docker

Docker is a tool that allows you to package your application and its dependencies into a standardized unit called a container. It solves the classic "it works on my machine" problem.

Core Concepts:

- Images vs. Containers
- Dockerfile
- docker-compose
- Volumes

- Docker is not a QA tool—it's a modern engineering tool.
- Learning it shows that you understand real-world development and deployment workflows.

Pro Tip:

Start by learning how to run your application and its database in Docker containers on your own laptop. This is a foundational skill.

Continuous Integration/Continuous Deployment (CI/CD) is the practice of automatically building, testing, and deploying code changes. This is where your automation provides its greatest value.

Popular Tools:



Jenkins (Recommended)

GitHub Actions

- Knowing how to configure a pipeline to run your tests automatically makes you a "full-stack" automation engineer.
- Most jobs expect at least some familiarity with CI/CD concepts.

Key Takeaway:

Your goal is to create a simple pipeline that checks out your code from Git, builds the environment (using Docker), and runs your tests automatically.

Once you have mastered the first seven steps, you are fully qualified for a junior-to-mid-level SDET role. But the learning never stops. The tech world is always evolving.

Potential Next Paths:

- Mobile Testing (Appium)
- Performance Testing (JMeter)
- BDD Frameworks (Gherkin)
- A second language (Java)

- Keep learning based on your interests or specific job requirements.

The Most Important Pro Tip:

The single most important thing you can do to advance your career is to ****keep building projects.**** A strong portfolio of real-world projects is what gets you hired and promoted.

Ready to Accelerate Your Journey?

This roadmap gives you the path. Now, it's time to start walking it. Here are the resources I've built to help you succeed at every stage.

Free QA Starter Course

My complete Software Testing Fundamentals course. The perfect place to start.

SuperSQA All-Access

Get all my courses, our private community, and monthly live Q&A calls.

1-on-1 Private Coaching

The fastest path. A direct partnership with me to accelerate your career.