

Selenium Bootcamp 1: The First Things You Need to Know

Selenium Bootcamp is a series of articles prepared by Selenium guru Dave Haeffner for Sauce Labs. Dave also authors the [Elemental Selenium website](#), which includes tips for using Selenium, and where you can sign up for his weekly email on the topic of Selenium testing.

- [Define a Test Strategy](#)
- [Pick a Programming Language](#)
- [Choose an Editor](#)

Selenium is really good at a specific set of things. If you know what those are and stick to them then you will be able to easily write reliable, scalable, and maintainable tests that you and your team can trust.

But before we dig in, there are a few things you'll want to know before you write your first test.

Define a Test Strategy

A great way to increase your chances of automated web testing success is to focus your efforts by mapping out a testing strategy. The best way to do that is to answer four questions:

1. How does your business make money (or generate value for the end-user)?
2. How do your users use your application?
3. What browsers are your users using?
4. What things have broken in the application before?

After answering these, you will have a good understanding of the functionality and browsers that matter most for the application you are testing. This will help you narrow down your initial efforts to the things that matter most.

From the answers you should be able to build a prioritized list (or backlog) of critical business functionality, a short list of the browsers to focus on, and include the risky parts of your application to watch out for. This prioritized list will help you make sure you're on the right track (e.g., focusing on things that matter for the business and its users).

Pick a Programming Language

In order to work well with Selenium, you need to choose a programming language to write your acceptance tests in. Conventional wisdom will tell you to choose the same language as what the application is written in. That way if you get stuck you can ask the developers on your team for help. But if you're not proficient in this language (or new to development), then your progress will be slow and you'll likely end up asking for more developer help than they have time for -- hindering your automation efforts and setting you up for failure.

A great way to determine which language to go with is to answer one simple question: Who will own the automated tests?

Also, as you are considering which language to go with, consider what open source frameworks already exist for the languages you're eyeing. Going with one will save you a lot of time and give you a host of functionality out of the box that you would otherwise have to build and maintain yourself -- and it's FREE.

You can see a list of available open source Selenium frameworks [here](#).

Choosing a programming language for automated testing is not a decision that should be taken lightly. If you're just starting out (or looking to port your tests) then considering and discussing these things will help position you for long term success.

For this course we'll be using the Java programming language. If you need help installing Java, then check out one of the following links:

- [Linux](#)
- [Mac](#)
- [Windows](#)

Choose an Editor

In order to be productive when writing Java code, you will want to use an integrated development environment (IDE). Here are some of the more popular ones:

- [Eclipse](#)
- [IntelliJ](#)

Next, we'll be diving into how to decompose a web app and writing our first test.