

## Do you want to join the QA team?

### Check what you should know as a Junior Tester:

- You can start by **reading about Software Testing** to help you understand what a tester actually does. I recommend two books written by well known experts in Testing world:

Radek Smilgin - "Zawód tester" wyd.II

Adam Roman - "Testowanie i Jakość"

- Have you ever heard about the **ISTQB FL certificate**? It is a certificate that summarizes test knowledge at a basic level. If you do not want to approach the exam right away, check out the syllabus which standardizes the concepts related to testing.

SJSI: <https://sjsi.org/ist-qb/do-pobrania/>

- **Soft skills** are a must-have in a tester's profession for a reason. It is you who often has to discuss the bug found with the creator of the code :)

About soft skills:

<https://techbeacon.com/app-dev-testing/7-soft-skills-every-qa-tester-needs>

- Have you heard something about **Scrum** or **Agile**? Knowledge of these work methodologies is very important in the IT profession, you never know which methodology you will work in.

Scrum: <https://link.do/scrum>

Waterfall: <https://link.do/waterfall>

- Some of the basic tester tasks are writing test plans, test cases and reporting bugs. One of the most popular **reporting tools** is the **Jira** software from Atlassian.

Trial version 7 days: <https://link.do/362YH>

- If you want to test web applications, it is good to know what information you can get from the **Dev Tools** browser . In order to complete your bug report, you need to be able to analyze HTTP requests and responses, read the console and have a basic understanding of **HTML5 / CSS3**

Dev tools: <https://link.do/CkbL7>

First step: <https://www.codecademy.com/learn/learn-html>

- If you want to test mobile applications, it is worth keeping up with the latest iOS / Android system versions, knowing how to read phone logs and having an idea how **Android Studio** and **Xcode** work.

Android Studio: <https://developer.android.com/studio>

Xcode: available on MacOS

- A tester often has to test using different environments. How? It is worth knowing how to create a **Virtual Machine** to facilitate your work.

<https://www.howtogeek.com/196060/beginner-geek-how-to-create-and-use-virtual-machines/>

- Do you know how to use a **terminal**? Do you know basic commands such as: `mkdir`, `ls` or `cd`? If so, great! ! If you haven't heard about this, learn about command lines. If the project hasn't been uploaded to the test environment yet, you can always test it locally. Knowledge of command lines will be needed.

Basic cmd: <https://link.do/cmd>

- The ability to use **Git** is a very useful skill. At the beginning of your career, you should know some basic commands, such as `git clone`.

Git: <https://git-scm.com/>

- You should understand the basics of **relational** and **non-relational databases** and know what the difference between them is.

SQL: <https://www.w3schools.com/sql/>

NoSQL: <https://www.mongodb.com/nosql-explained>

Want to know more? There are plenty of **blogs about testing**. You can check ranking here: <https://www.globalapptesting.com/blog/top-software-qa-blogs>

### **What to learn next after you have gained the basic skills?**

- Maybe it is not obvious, but yes, testers can also **code**. Certain repetitive tasks can be easily **automated using simple scripts**. You just have to know some basics to start.

The most useful to the tester are these three **programming languages** (this can change depending on the project or your preferences):

Python: <https://www.python.org/>

Java: [https://www.java.com/en/download/faq/whatis\\_java.xml](https://www.java.com/en/download/faq/whatis_java.xml)

JavaScript: <https://www.javascript.com/>

(Do not confuse Java with JavaScript! They are two different programming languages: <https://www.geeksforgeeks.org/difference-between-java-and-javascript/>)

- **Rest API** and **Postman** support. It will help you in backend tests. Also, knowledge of the **HTTP protocol** will be needed.

Rest API: <https://www.sitepoint.com/developers-rest-api/>

HTTP: <https://www.pickaweb.co.uk/kb/what-is-http/>

Postman: <https://www.getpostman.com/>

- Do you want to start writing **automatic tests**? Simple scripts will be useful if you are trying to repeat the same activities in the application's UI. There are a lot of frameworks that will make this job easier for you, such as **Selenium WebDriver**, **Cypress**, **TestCafe** or **Robot**. The choice of a tool depends on the specifics of the project, there is no one universal tool.
- Knowledge about **Unit tests**, **TDD** and **BDD**:

What is the difference? :

<https://codeutopia.net/blog/2015/03/01/unit-testing-tdd-and-bdd/>

- **Performance testing** has many functions. We can use a lot of tools to do them, the most popular is **Apache JMeter**, but again, it depends on the preferences and project specifics.

Here we have a ranking of performance testing tools:

<https://www.softwaretestinghelp.com/performance-testing-tools-load-testing-tools/>

- Knowledge about **CI/CD** - sounds like DevOps stuff? Not necessarily. It's worth it for the tester to know a little something about that.

<https://pl.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment>