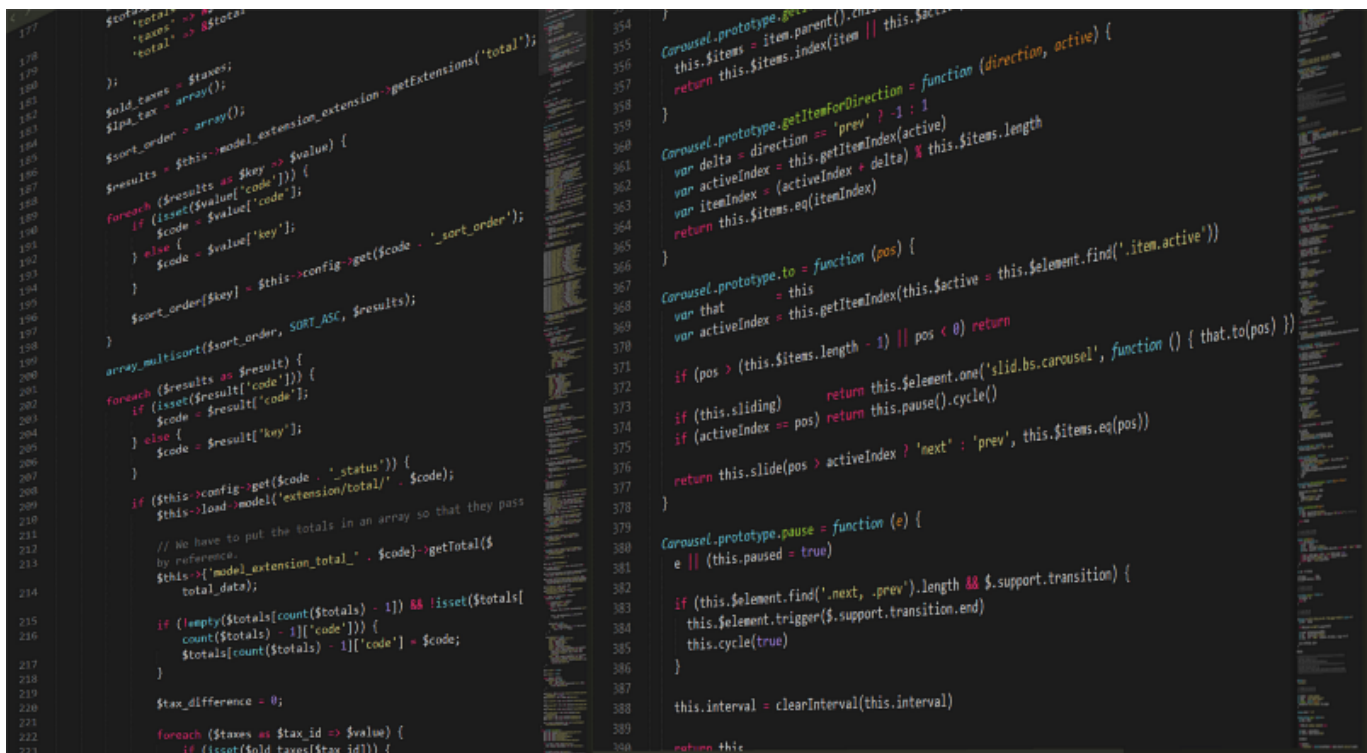


Stay safe, friends. Learn to code from home. Use our free 2,000 hour curriculum.

12 FEBRUARY 2019 / #PROGRAMMING

How to learn to code in Java: why you should and where to start



by John Selawsky

Define your career goals and choose a language. This is the most important advice for anyone who has zero experience in programming and aims to join the developers' community. But how can you make a choice having no idea which language suits your purposes the most?

Wouldn't it be easy if there was a "market leader" among all programming languages? Then all newcomers could always have a safe choice — learn this language and you'll surely be in high demand. Of course, it's impossible. Languages are "tools" for solving numerous tasks. But still, there's one which can be considered as a stellar choice — it's Java.

Java is a well-structured, object-oriented language, which can be considered easy for beginners. You can master it quite rapidly, as there are many processes that run automatically. You don't have to delve into "how the things work in there" too deep. Java is a cross-platform language. It allows a programmer to create an application, which can be deployed on any device. It's the language of choice for the Internet of Things and the right tool for building enterprise-level applications.

What's important for a "freshman", Java has one of the biggest global communities and high-quality documentation. If you have a problem, it's more than likely that the answer is ready there, waiting for you. Two more benefits of Java are vast collections and frameworks that cover most of the challenges you'll have to deal with as a Junior developer.

Finally, it is a language with a respectable background and a prominent future. Two latest versions, enhanced with new features, were released within six months of each other. This helps Java to stand on the front burner and keep the highest positions in respectable professional rankings.

But even when we're talking about a programming language with a low learning curve, there are plenty of hidden rocks for beginners.

mistakes to avoid? Let's go all the way through your learning experience.

So, you're new to programming. Congratulations: as you can make use of other programmers' experience and avoid the common mistakes in a self-education. Here are some frequent lapses.

Learning without setting a specific goal



To estimate the scale and desired level of knowledge in a certain programming language, first of all, you should answer a simple question: why do you need programming?

If you're serious about coding and intend to become a Java developer, you'll need to study every day for at least 4–5 hours. But if you simply want to enlarge your scope, you can choose a more relaxed tempo. Anyway, try not to extend your education for many years to come,

or development tool to get a job, it becomes outdated.

Reading “anything useful” that pops up isn’t the best strategy of learning. It’s okay to make adjustments to the general plan. However, not following it at all will inevitably demotivate you.

An attempt to learn too many technologies at once

This a subsequent mistake, which leads to learning without an end. Don’t try to stuff your educational plan with too many technologies and tools to avoid burnout. You’ll need time to process the information and see how everything works on practice.

Too much research and theory without practice



You'll need a serious background in theory, but as long as you only read (or watch) lectures and courses, you can't really code. Think about it. Nobody can learn how to swim without trying it. You need explanations, but above all, you need action.

Only plenty of practice will teach you how to deal with everyday programmers' problems. For example, you'll get a habit to check your code regularly and not let the connected problems to pile up. You will also start with the easiest solutions. Then move to more exquisite, constantly working on the improvement of your code. And the most important, you'll get used to coding on a daily basis.

Learning to code in isolation

Beginners often hesitate, whether they should join online programming communities or attend events, mainly because of imposter syndrome. But remember: nobody wants to bring you down. The programming community is collaborative. This is how numerous open data sources appear. They're used to work in a team, and surely, all of them were beginners at a certain moment of life.

Take a look: there are plenty of forums and web sources on programming. Developers eagerly share their knowledge and are open to collaborations. You can start learning online, get a useful hint from more experienced colleagues and later, find your first coding project.

Being sure that one day you will fully understand all aspects of

programming language, he'd be the richest person on the planet (sorry, Jeff Bezos). But until then, there's no certain end in your education, if you've chosen to be a programmer. Even when you have a job, never stop learning. Things change rapidly in a programming world.

On the other side, when you're still learning how to code, try to accept some facts as they are. Do not dig into every subject, as its background might be too "heavy".

It's easier to achieve your aim step by step. Now, when you know the common mistakes, you can avoid them. Here are five steps you can take to learn Java.

1. Set a goal and choose a language.

You can do pretty much anything with Java. You can build application servers, desktop, and mobile applications, enterprise applications, and run unit tests. But of course, a programming language isn't everything you need to know to become a pro. Try to think of specific activity areas.

Today's trends are building cloud-based applications (you'll also need to learn cloud computing basics and specific tools for cloud development), working with the Internet of Things, performing Big data analysis, creating games, etc.

2. Create an educational plan.

Are you ready to make Java programming your profession? Then dedicate at least 2–3 (on the workdays) to 5 hours (on the weekends)

prepare for a Junior developer position.

In general, your learning process will be divided into a few stages:

- Installation of JDK (Java Development Kit) from Oracle website and Installation of Java runtime environment (IntelliJ IDEA, Eclipse or NetBeans);
- Learning Java Syntax;
- Learning Java Core;
- Learning Java Collections;
- Learning popular libraries and frameworks;
- Exploring frequently used APIs (for example, servlets, JSP, JDBC, JUnit);
- Mastering Git;
- And so on.

Try to follow your plan and keep the balance between theory and practice. Programming is a skill that needs training every day. If you finish studying a certain subject, continue with a few coding tasks (3 to 10, depending on their difficulty).

You can easily adjust the well-known Pareto principle (80/20 rule) to Java learning. The “20” will stand for research and “80” for practice. This is not an exaggeration: you’ll need four times more coding than reading to memorize what you’ve learned and developed your coding skills. Unfortunately, if you won’t write your code at once, you will forget everything you’ve read on the next day. Luckily, you don’t need to invent your own training program, if you have the right sources.

3. Choose the right tools and join programming communities.

Let's be honest: a perfect Java course doesn't exist. Some will give you a profound basic knowledge, some will provide with a good deal of practice. Choose a few platforms for education, communication, and practice. Choose one online course which combines a plain explanation of theory with the real coding.

Join the communities, where you can ask any question about the aspects of coding on Java, share your thoughts and find like-minded people. The top communities for Java developers are listed below.

You will need lots of tools to accomplish your goals. Here's a set of "tools" for you to succeed.

1. Strictly practice: [CodeAcademy](#)

This is probably one of the best-known online platforms for learning numerous languages. Of course, there's a section for Java learners, too. Here you start coding from the first lesson. There are pretty useful hints, which you can get if a certain task seems to be tough (which is normal when you are a beginner). The logic is simple: read the task, write code and run it to see what happens as a result, and then move to the next step. The tasks gradually become more complicated, until you start to write your code from a clean sheet. The FAQ section is pretty useful and you have an option to discuss each task with other students.

This course can be a great complementary tool in learning Java. As told before, you need to code to master Java programming.

2. Play and code from the beginning: [CodeGym.cc](#)

gamification. Nowadays it's common for online courses to adopt gamified elements, as they bring instant motivation, a sense of constant achievements and make it easier to reach your goal.

At CodeGym's you start from level 0 in the futuristic universe with space travel, robots, and other cool things. Your aim is to pass four educational quests to level up a character — a robot. Each quest includes 10 levels, and each level has up to 10–12 lessons, taught by crew members of a galactic ship. A lesson consists of a minimum theory and a set of tasks with instant verification and tips on how to improve your code. You start coding from the beginning, solving very simple tasks, until you're ready for much more complicated and massive coding projects.

The whole story allows you not to get stuck on a certain subject or task, move forward and return later to the “twisters”. All in all, there are impressive 1200+ tasks in a course. An amount that converts in at least 300–500 hours of coding.

3. Challenge programmers: Codewars

This is another online platform with tons of practice. Though it's more suitable for learners with at least a small background in programming. Codewars empower you to sharpen skills in a certain language by training katas.

You can write your code in a browser and use test cases to check it step by step. You can also compare your solution with others to broaden your knowledge and discuss the best practices. Each kata has a rank, and each kata is a challenge with other programmers. When you solve more difficult tasks, you can get more complicated

others is that it allows to study and practice in a group.

4. Listen to the lectures of a “pro” level: EdX

EdX is an educational platform founded by Harvard University and MIT. It covers many fields of study, including Java programming and many related subjects. There are plenty of verified introductory courses and certificate programs, provided by universities.

Each course has a defined time frame and includes video lectures with a text transcript. It's easy to return to previous lessons and revise the material. There are dozens of different courses, some of them are free. But still, this is a source strictly for learning Java theory. Be sure to strengthen those courses with hours of coding.

Want to stay tuned to the latest news, reviews and updated in Java world? That's the right thing to do. Java is constantly evolving, upgrading and bringing new features. Be sure to have a couple of media you can read daily. One of the largest are:

- Javaworld (news, useful tutorials, “how to's” and everything else to keep you up to date with Java programming);
- Java Code Geeks (a resource center for those who learn Java with educational tutorials, articles, and tips on how to get a job as a Java programmer);
- DZone (plenty of articles and guides for beginners and experienced developers);
- Javarevisited (a daily updated blog which covers numerous Java learning topics).

Do you remember the advice to join a community to make your

[Overflow's](#) Java section or [Coderanch](#). Also, there are at least three large communities on Reddit, like [learnprogramming.java](#) and [learnjava](#), where you will find useful tips on learning Java and programming in general. If you need more specific information, for example, a hint on improving your code in the certain tasks, try the special [help section](#) at CodeGym, where other students and “coding coaches” will help you with any problem. And finally, Oracle has a vast [Java Community](#), open for both the newcomers and experienced developers.



Being a programmer means constant improvement and gaining new knowledge. If you develop a knack for self-education from the start, you'll succeed by all means. Are you ready to try?

If this article was helpful, [tweet it.](#)

helped more than 40,000 people get jobs as developers.

[Get started](#)

freeCodeCamp is a donor-supported tax-exempt 501(c)(3) nonprofit organization (United States Federal Tax Identification Number: 82-0779546)

Our mission: to help people learn to code for free. We accomplish this by creating thousands of videos, articles, and interactive coding lessons - all freely available to the public. We also have thousands of freeCodeCamp study groups around the world.

Donations to freeCodeCamp go toward our education initiatives, and help pay for servers, services, and staff.

You can [make a tax-deductible donation here](#).

Our Nonprofit

[About](#)

[Alumni Network](#)

[Open Source](#)

[Shop](#)

[Support](#)

[Sponsors](#)

[Academic Honesty](#)

[Code of Conduct](#)

[Privacy Policy](#)

[Terms of Service](#)

[Copyright Policy](#)

Trending Guides

[2019 Web Developer Roadmap](#)

[Python Tutorial](#)

[CSS Flexbox Guide](#)

[JavaScript Tutorial](#)

[Python Example](#)

[HTML Tutorial](#)

[Linux Command Line Guide](#)

[JavaScript Example](#)

[Git Tutorial](#)

[React Tutorial](#)

[Java Tutorial](#)

[Linux Tutorial](#)

[Donate](#)

[jQuery Example](#)

[SQL Tutorial](#)

[CSS Example](#)

[React Example](#)

[Angular Tutorial](#)

[Bootstrap Example](#)

[How to Set Up SSH Keys](#)

[WordPress Tutorial](#)

[PHP Example](#)