

Brief Introduction

Discover why it's a great time to master Kotlin and learn this programming language, who this course is for, and what you'll learn in this course.

WE'LL COVER THE FOLLOWING



- What is Kotlin?
 - What Can I Use Kotlin For?
- Who's Using Kotlin?
- Why Learn Kotlin?
- What This Course Covers
- Summary

What is Kotlin?

Kotlin is an open-source, statically typed programming language. It can transpile to Java bytecode to run on the Java Virtual Machine (JVM) or Android. Interestingly, it can also transpile to JavaScript code or native bytecode, allowing you to use it for virtually anything from web applications to iOS apps to embedded devices and more. This course focuses on Kotlin for the JVM.

Note: The term “transpile” refers to translating a piece of code between languages or representations. For instance, transpiling Kotlin code to Java bytecode means that a piece of code in Kotlin syntax is translated to its corresponding representation in Java bytecode. Thus, the syntax changes but the semantics remains the same.

Kotlin is developed by JetBrains, the company behind the IntelliJ IDE and other development tools. **They designed Kotlin to be 100% interoperable with Java, meaning that you can use any Java libraries and frameworks in**

Kotlin, and vice versa. Keep this in mind, as we will make use of this interoperability throughout the course.

What Can I Use Kotlin For?

As previously mentioned, **you can use Kotlin on any platform.** Currently, it's most prominent for Android app development. However, its usage for developing server-side applications for the JVM is increasing.

Additionally, it can be used to write web applications by transpiling to JavaScript. Kotlin then lets you use popular libraries and frameworks such as React and VueJS.

It's useful to know about the corresponding sub-projects in case you encounter them:

- **Kotlin/JVM:** lets you write JVM applications by transpiling to Java bytecode (e.g., Android development, Spring services, desktop applications)
- **Kotlin/JS:** lets you write web applications by transpiling to JavaScript (e.g., React apps, and Vanilla JS apps)
- **Kotlin/Native:** lets you write applications for any target platform such as embedded devices or iOS (e.g., iOS apps, microcontroller programming)

As a side note, Kotlin supports multi-platform application development, which enables you to share code between apps for different platforms. The most obvious use case is to share core functionality between your Kotlin Android app and your Kotlin iOS app.

Who's Using Kotlin?

By now, Kotlin is used by too many large companies to name them all. But just to convince you that Kotlin is used in production by some of the largest tech companies in the world, here's a selection of them:

- Google
- Netflix

- Uber
- Slack
- Udacity
- Evernote
- Trello
- Slack
- Lyft
- ...

Many of these companies use Kotlin on Android, but also on server side and other platforms.

Why Learn Kotlin?

Personally, learning Kotlin has taught me a lot about language design and some of the shortcomings of Java that I wasn't aware of previously. This was back in 2015 before Kotlin version 1.0 was out. Therefore, it can be a great way to learn about modern programming language features that you will encounter in other languages as well, such as TypeScript, Scala, Go or Rust.

Apart from this, there are many **companies actively looking for Kotlin developers, especially in the Android development space**. So having experience in Kotlin can give you an edge in the interview process.

But even without the goal of landing a Kotlin developer job, learning the language and its concepts will allow you to **quickly master other modern languages such as TypeScript, Scala, and Swift** because they share many language concepts.

What This Course Covers

This course covers the basics of the Kotlin language:

- Variables and data types, including type inference and one of my favorite Kotlin features: nullable types
- Conditions using `if` and `when`, why `when` is more powerful than `switch` in other languages, and how to use conditions as expressions
- Collection types from lists to sets to maps (and arrays), with notes about

the importance of immutability

- Loops for iterative control flow and how Kotlin's `for` differs from `for` loops you may be familiar with
- Functions, including Kotlin's shorthand notation for simple functions, how to use default values and named parameters, Kotlin's extension functions, and how to use infix functions and custom operators to improve your code's readability

Each lesson ends with a quiz to review the main takeaways and includes at least one hands-on programming exercise. Doing the exercises is the most important part of the course since these will give you the most value in terms of mastering Kotlin.

Summary

Kotlin is a modern language that made its way into the mainstream over the last few years, with Google I/O 2017 being one major accelerator of this development.

- It can run on any platform by transpiling to Java bytecode, JavaScript or native bytecode
- Large companies use Kotlin and continue to have a positive experience with it, proving again that it's ready for adoption at larger scale
- Learning Kotlin can land you a job and helps you master other modern languages faster
- This course is the entry point into your Kotlin journey, teaching you all you need to write your first simple applications

In the next lesson, you will learn about Kotlin's principles and goals and how they guide the design of the language.