1. Pick A Language

So you got an idea of the mobile app. Every great idea needs a tool to be made. In this section we are going to compare 2 tools: Java and Kotlin. You will see the strengths and weaknesses of each, and decide if Java or Kotlin will fit your goals.

The Kotlin mobile app development community is continuing to grow. Back in 2017, Google acknowledged Kotlin by making it the second official language of Android app development. Since then, the programming language has seen a monumental rise in demand in both the developer and enterprise community. With Google announcing that Kotlin is now its preferred language for Android app developers, the language is proving to be a pragmatic, modern, and intuitive programming language.

What is Java?

Java is an object-oriented programming language. It was released in 1995 by the Sun Microsystems. This company is the property of Oracle. Most Java elements are available in open-source.

A large part of the Android apps and the Android itself are based on Java. In 2020 Java is the 3rd most popular language on GitHub. Developers of all kinds use it to build the following:

- Android apps;
- Web apps;
- Embedded systems;
- Big Data tools and systems;
- Server apps, and more.

What Is Kotlin?

Guys from JetBrains launched Kotlin to make coding in Java more productive. Kotlin became an official programming language in 2018. It is the programming language that runs on JVM. JVM means Java Virtual Machine. Besides that, the Kotlin can be compiled into JavaScript and run in browsers. It is possible to code on Kotlin/Native. Android developers can use IDE to build cross-platform apps.

Most tech giants make their Android apps with Kotlin. Others are planning the migration. While we say enterprise leaders, we mean the companies like:

- Netflix;

- Trello;
- Pinterest;
- Uber;
- Twitter.

Kotlin vs. Java — An Ultimate Comparison With Examples

In this part of an article, we will compare Java and Kotlin in terms of the parameters important for the development. They include the following:

- Speed of Coding;
- Performance;
- Stability;
- Documentation;
- Popularity;
- Community;
- Talent Pool;
- Easiness to Learn.

Speed Of Coding

We can say that Java and Kotlin have the same speed of coding. Kotlin has more laconic constructions that allow a coder to type less. Yet, there is a trick. Finding a solution to a task on Kotlin needs more time than on Java. That means Kotlin has more cognitive load than Java. If you are a fantastic abstract thinker, Kotlin is your choice.

If you choose Java, you have to type more code than in Kotlin. We are talking about the process of the solution of the same task. Yet, you will spend less time thinking of the solution using Java than Kotlin.

This doesn’t make a noticeable difference. We can say that the speed of coding on Java and Kotlin is the same.

Verdict: draw

Kotlin vs. Java Performance

This parameter is too opinionated. Both Kotlin and Java compile to ByteCode that runs on JVM. It’s nearly impossible to compare Kotlin vs. Java when it comes to memory usage. So it's hard to measure, track, and compare their performance, but we will try.

Kotlin is more functional than Java. It has more extra features than Java. Kotlin is more convenient to work with multithreading apps. It's true because of the Coroutines tool Kotlin has. Yet, Kotlin compiles and executes a little slower than Java. It happens because of the vast number of features Kotlin has.

Java has less extra features than Kotlin and is a little simpler. But due to this fact, it compiles faster than Kotlin. It works a little quicker than Kotlin due to the absence of extra features.

But this difference is not generally noticeable and varies. We can say that Java and Kotlin have the same performance.

Verdict: draw

Stability

Stability is the parameter where we can see the difference. Let's start with Java. This language exists for quite a long time. It has versions with long-term support, like Java 8 and Java 11. What is long-term support? If anything goes wrong with these versions, devs will fix it with a patch. Following this link, you can download different versions of Java with a free license.

We can call Kotlin quite an established language, but it is younger than Java. There are no versions with long-term support. In general, we can call both Java and Kotlin stable languages. But if stability is your main priority, it's best to choose Java.

Verdict: Java

Kotlin vs. Java Documentation (With Examples)

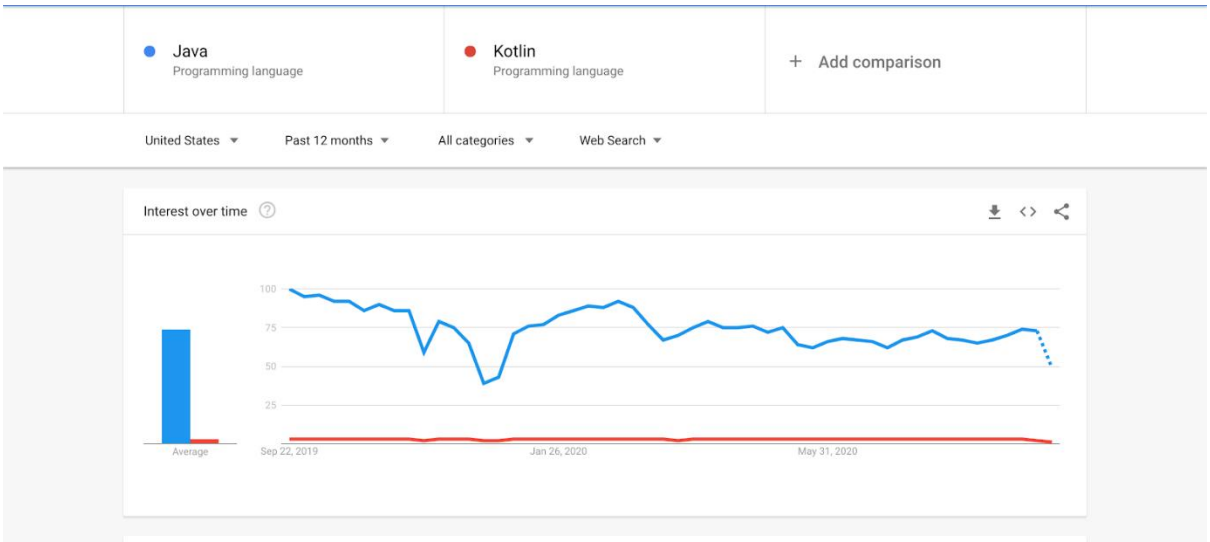
Both Java and Kotlin have enough documentation to learn. Java exists longer than Kotlin, so it's easy to find a tutorial or a book. Developers consider Java as the most straightforward programming language to learn. Here you can find some official Java documentation from Oracle.

Kotlin is comparatively a young language, yet, it has enough good official docs. It would help if you dug a little deeper to find some trusted sources on Kotlin. But that's it, no more difference.

Verdict: draw

Kotlin vs. Java Popularity Comparison (With Examples)

According to Google Trends, Java awakes much more interest than Kotlin. Java has been through some ups and downs. But Kotlin is not even close to the popularity of Java.



Source: Google Trends.

We can assume this happens because Java exists for longer. Another reason is that Java needs less cognitive investments in learning.

What Java has that kotlin does not?	What Kotlin has that Java does not?
Static members	String templates
Primitive types that are not classes	Singletons
Non-private fields	Null-safety
Wildcard	Extension functions
Checked exceptions	Smart casts

Source: DZone

Java and Kotlin are similar and different at the same time. And Java is here for more time. That's why the difference in their popularity is so visible.

Verdict: Java

Community

Since Kotlin comes from Russia, many Russian developers switched to this language. Russian people form the core of the Kotlin community. Look at Google Trends above. You'll see Kotlin's squad is around 20 times smaller than Java's.

The Java community has many people from India. Java is quite simple to learn, and India got many junior devs. So they start with learning Java. Its squad is large. So if you face any issue, there is a high chance someone resolved it before.

Verdict: Java

Talent pool

The latest Dev Survey shows that Kotlin is within the most popular techs. (Source: StackOverflow Dev Survey). 8% of pro developers choose this language. Java is not on this list, despite high interest on Google Trends.

But Java is on the list of the most loved technologies. 44.1% of devs adore it. Kotlin is on this list, above Java. 62.9% of devs love Kotlin. Kotlin brings more money to the devs than Java. Kotlin brings $55K per year, while Java brings $50K. (Source: StackOverflow Dev Survey).

You don't need a developer in your office to make a cool product. So we take global rankings data for this part of an article.

Verdict: Kotlin

Kotlin vs. Java: Which One Is Easy To Learn?

Java is a little easier to learn than Kotlin. You need to write more code in Java than in Kotlin to resolve the task. But Java exists longer, so you will easier find tutorials. It's easy to find proper docs and ready-made solutions on Java.

Kotlin needs more cognitive investments in learning than Java. But it allows for resolving the same problems with less code than Java. If you are a good abstract thinker, learning Kotlin won't be an issue for you.

Verdict: Java

Brief Comparison Table Of Java vs. Kotlin

Parameter	Java	Kotlin
Static members	Almost the same. Thinking of the solution is the most time-consuming part.	Almost the same. Thinking of the solution is the most time-consuming part.
Performance	Almost the same. Both compile to ByteCode.	Almost the same. Both compile to ByteCode.
Stability	Has stable versions with long-term maintenance.	Almost the same. Both compile to ByteCode.
Documentation	Good, easy to find.	Good, a little bit harder to find.
Popularity	Extremely popular worldwide.	Not so popular worldwide.
Community	Mostly Indian, very broad.	Mostly Russian, comparatively little.
Talent pool	Not in the top-list.	In the list of the most popular technologies 2020 according to StackOverflow Dev Survey.
Easiness to learn	Easy to learn.	Can be tricky to learn if you are not a good abstract thinker.

Kotlin vs. Java. When to Use & Examples:

Let's switch to the practical part of Java and Kotlin's usage. Those languages are similar, but devs use them for different tasks. Below you will find out what technology is best for your project. And the answer to the main question: Java vs. Kotlin for Android development — what is better?

What is Java good for? (+ a few examples)

Java is a universal language. Ten years ago, people used it to build some Android apps. But now it has found a use for BigData, eCommerce, and enterprise systems. There are popular instruments for BigData like Hadoop and HBase. Hadoop contains MapReduce and HDFS, as well as tech giants, use Java for their high-loaded servers.

Companies use Java to develop enterprise systems. There are BPM and Rule Engines. JakartaEE offers many services for enterprises; also, e-commerce companies use Java.

Examples:

- Netflix;
- Apple TV;
- Hybris;
- ATG.

What is Kotlin good for? (+ a few examples)

Companies and indie developers use Kotlin to develop Android apps. Almost every app in the Google Play Market has Kotlin as a base. Kotlin serves as a base of writing microservices. So Kotlin is better for Android when it comes to modern mobile app development. In this case, it is combined with Spring Boot, Quarkus, Micronaut.

Examples:

- Pinterest;
- Coursera;
- Trello;
- Evernote.

Kotlin 2019: Where Is The Language Now?

Since Google I/O 2017, Kotlin has seen explosive growth in Android development and after hosting Kotlin Conference only twice, the conference has become something of an institution in the developer community.

Accessibility on all platforms has always been a primary objective for Kotlin, but multi-platform programming is only the premise to a much more innovative outlook: sharing code between all platforms. With the release of Kotlin 1.3, improvements to Kotlin/Native are

advancing the concept of multi-platform convenience. Finally, Android developers can use one integrated development environment (IDE) to develop with Kotlin on all platforms. The newest release makes mobile app scalability more attainable by supporting the invaluable benefit of code reuse, saving time and effort for more challenging tasks.

Kotlin/Native uses the compiler technology LLMV to compile Kotlin sources into stand-alone binaries for multiple operating systems and CPU architectures like iOS, Linux, Windows, Mac, and Web assembly.

In 2019, more enterprise leaders are migrating to Kotlin or planning to do so. Mobile products like Pinterest, Twitter, Netflix, Uber, AirBnB, Trello, and Evernote are all switching to Kotlin for Android applications. While the adoption of cross-platform Kotlin development hasn't been explosive, major industry players are taking note of the many benefits Kotlin has to offer.

Is This The End Of Java?

There are mixed opinions from developers.

Java is a reputable programming language with vast open-source tools and libraries to help developers. With that said, no language is without fault and even Java is subject to complications that can make a developer's job tedious. If anything, Kotlin will introduce solutions to common programming headaches and improve the Java ecosystem as a whole.

In two years, Kotlin has become a more stable and congruous development option for Android Studio. Some developers seem to

believe that Kotlin will oust Java for Android development in the coming years. Other experts see Kotlin and Java coexisting without one outweighing the other.

For most, Kotlin's strengths outweigh the language's setbacks. There are definite limitations within Java that impede Android API design. Kotlin is inherently lightweight, clean and far less verbose, especially in terms of writing callbacks, data classes, and getters/setters. In other words, Kotlin is specifically designed to improve existing Java models by offering solutions to API design deficiencies.

Kotlin addresses a number of Java's weaknesses:

Brevity

A lot of developers praise Kotlin for being concise. This is a quality Java is not known for; however, readability should always take priority over concision. Yes, the succinct nature of Kotlin simplifies a developer's job and mitigates the risk for error, but Kotlin doesn't practice concision for concision's sake. Boilerplate code is a problem to read and leads to more bugs and wasted time trying to identify them.

Below is a simple calculator function written in Java.

```
public class ClearBridge {

public static double calculate (double a, String op, double b)
throws Exception {
```

```
        switch (op) {
                case "add":
                        return a + b;
                case "subtract":
                        return a - b;
                case "multiply":
                        return a * b;
                case "divide":
                        return a / b;
                default:
                        throw new Exception();
                }
        }

}
```

For comparison, here is the same calculator in Kotlin:

```
fun calculate (a: Double, op: String, b: Double): Double {
        when (op) {
                "add" -> return a + b
                "subtract" -> return a - b
                "multiply" -> return a * b
                "divide" - > return a / b
                else -> throw Exception()
        }
}
```

It may not seem like much, but the Kotlin version of this calculator is written in half the lines of code it took to program the function in Java. Brevity is a crucial factor in productivity. Writing large projects becomes easier when a developer is given more power for every line of code. A key observation here is Kotlin does not overlook comprehension for the sake of brevity. The syntax is concise, readable and still substantial.

Interoperability

Interoperability is Kotlin's core purpose. From the beginning, the project's intention has been to use existing knowledge and expertise to make every library available to Kotlin programmers. Developers can simply write modules in Kotlin that work flawlessly within existing Java code. By emitting the Bytecode, a Kotlin compiler allows the two languages to work in unison in the same project.

Inbuilt Null Safety

Kotlin's type system has inbuilt null safety. The infamous NullPointerException is largely responsible for Android development mistakes. Android relies on null to represent the absence of a value, but null can easily destroy an app. Kotlin solves this problem by incorporating inherent null safety. This addition saves developers from writing extra code to work around the issue.

No Raw Types

Before generics came into play, raw types were used quite frequently. Raw types allow for backward compatibility, but raw types can throw a CastClassException and the error will occur during execution and not the compiling stage. Kotlin does not allow raw types, and as a result, produces a more type-safe code.

No Checked Exceptions

The checked exception feature in Java can be problematic. Checked exceptions are when the compiler forces the caller of a function to catch or (re-throw) an exception. The fact is, checked exceptions are often unnecessary and cause empty catch blocks. Non-existent checked exceptions are annoying for developers because empty catch blocks force developers to weed through the code to identify a nonexistent exception. As a solution, Kotlin removes them entirely, which minimizes verbosity and improves type-safety.

Is It Necessary For Developers To Learn Kotlin?

Again, there are mixed opinions.

While it is not entirely necessary for developers to make the switch to Kotlin, they're going to encounter the language eventually. If you're already familiar with Java, learning Kotlin will be simple. The language is poised to impact app development on a massive scale, so it doesn't hurt to learn the basics of the language. Additionally, growing with modern techniques and development styles will aid the growth of any developer's skill set.

We hope that the above article helped you in choosing the programming language.

Note – We have assumed that you have chosen Java and made the application according to it.