These are the slides for the Kotlin for Java Developers on Udemy. They are provided free of charge to all students.

More information about the course: https://lpa.dev/u1kfjd

If you have any questions of queries, please add your feedback in the Q&A section of the course on Udemy.

Best regards,

Tim Buchalka
Learn Programming Academy

Kotlin for Java Developers Slides

Main Course Slides.

What We'll Cover

- What is Kotlin
- Quick Differences
- Data Types
- Object Oriented Topics
- Loops and ranges
- Lambdas
- Generics (variance)
- I/O
- Java Interoperability
- Challenges

Kotlin

- Runs anywhere the JVM can run
- Statically typed
- Object-oriented
- Functional programming

Guiding Principles

- Conciseness
- Safety
- Pragmatism
- Interoperability

Hello World

Let's take a moment to talk about what happens when we compile a Kotlin application. The Kotlin compiler (kotlinc) takes files with the .kt extension as input and generates bytecode as .class files. At this point, the .class files are equivalent to Java .class files, and the JVM can run them.

However, when running Kotlin applications, you need the Kotlin runtime library, in addition to the JRE. So, the Kotlin compiler compiles .kt files into .class files, which contain bytecode. The .class files can be on the JVM, exactly like class files generated from Java code. When distributing a Kotlin application, you have to distribute the Kotlin runtime library and the JRE. Otherwise the application won't run.

Default Imports

- kotlin.*
- kotlin.annotation.*
- kotlin.collections.*
- kotlin.comparisons.*
- kotlin.io.*
- kotlin.ranges.*
- kotlin.sequences.*
- kotlin.text.*
- kotlin.jvm.* (JVM only)
- java.lang.* (JVM only)

Top-Level Items

Access Modifier	Kotlin	Java
private	Visible within the same file	Can't be used
protected	Can't be used	Can't be used
public	Visible everywhere	Visible everywhere
internal	Visible within the same module	N/A

Kotlin for Java Developers Slides

Challenge Slides.

Step 1

- Create a new Kotlin project called Challenge1
- Add package academy.learnprogramming.challenge1 to the project
- Add a file called Challenge1 to the package
- Add the main function to the file