



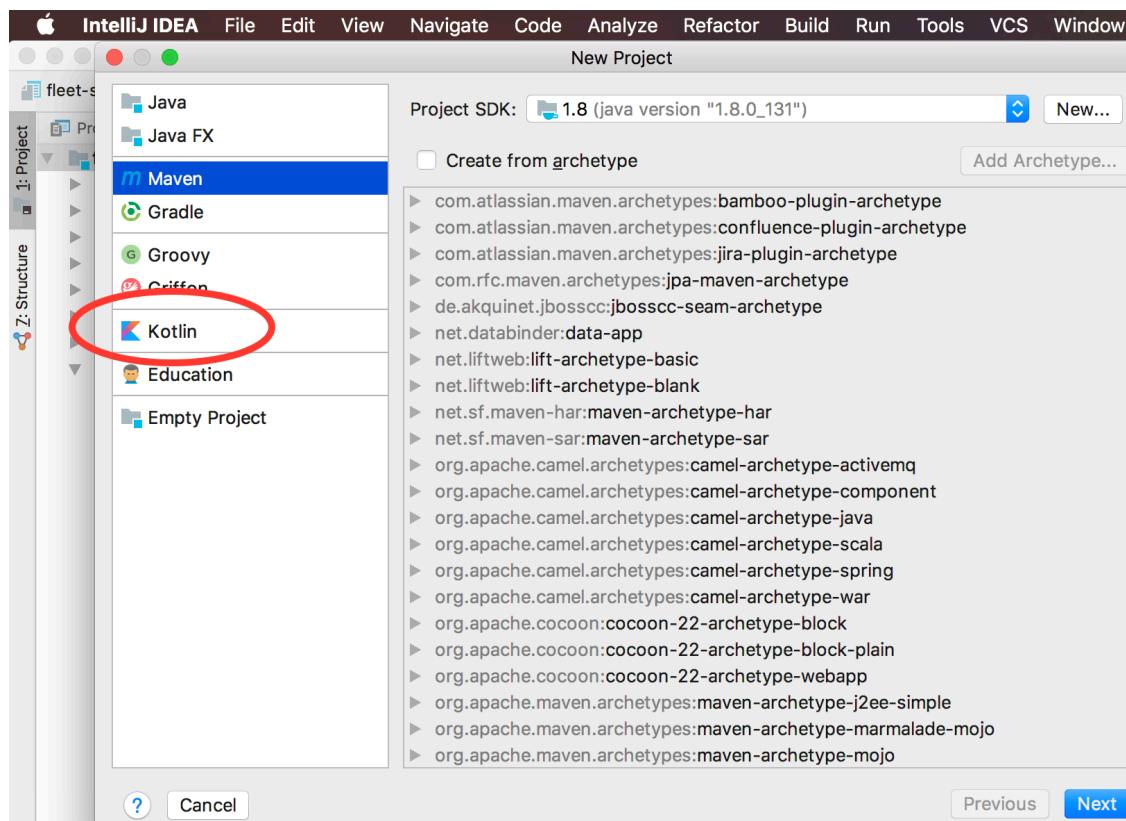
INTRODUCTION TO KOTLIN SYTAC - SLAB

Shaun Thomas
shaunthomas999@gmail.com
www.shaunthomas999.com

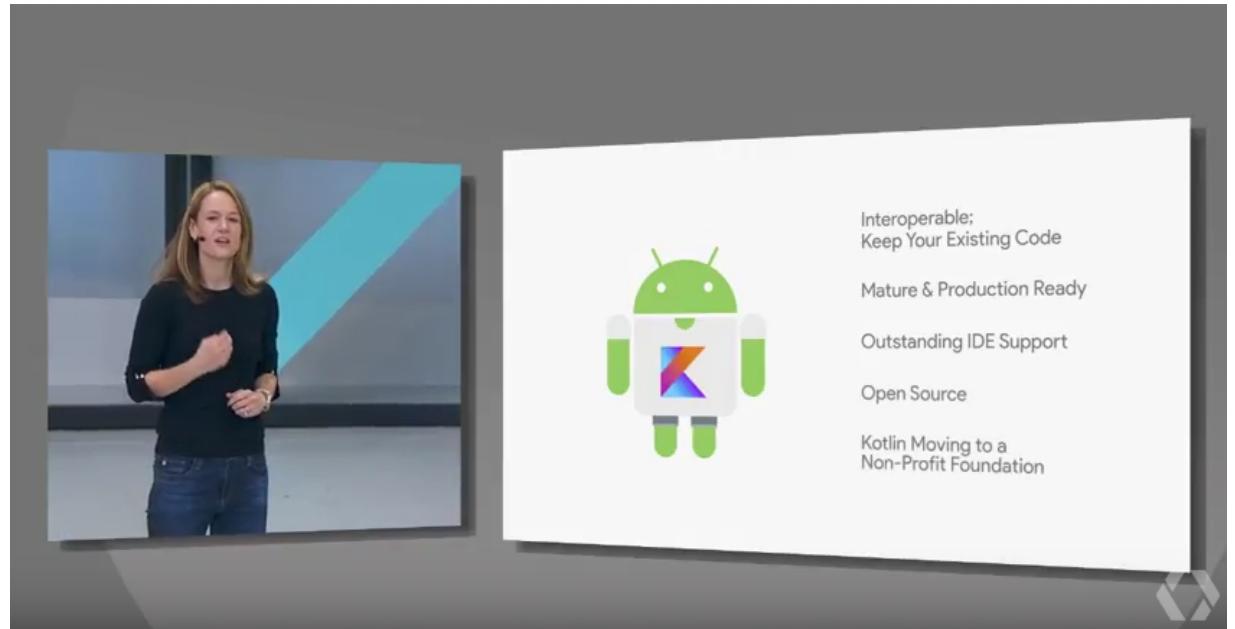
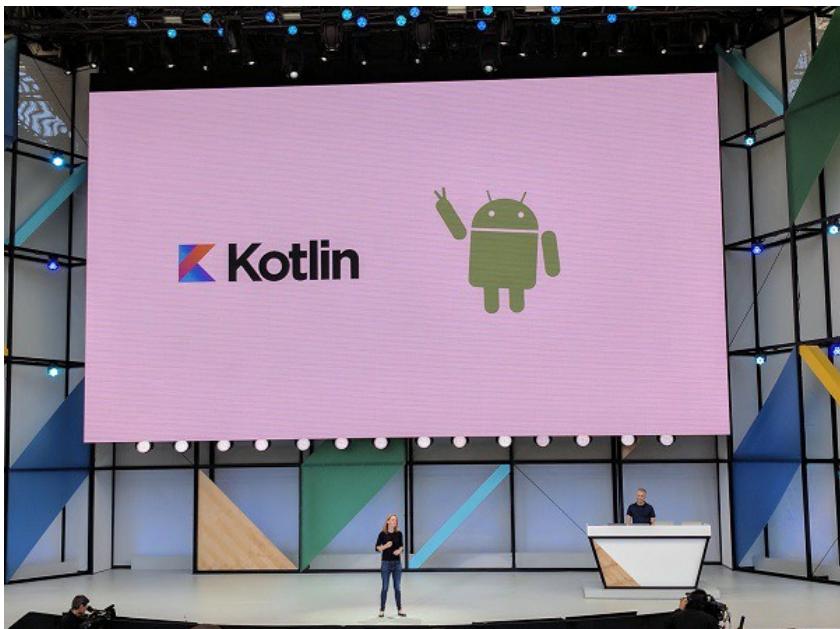
WHY ARE YOU HERE?



SUPPORT IN INTELLIJ IDEA PROGRAMMING LANGUAGE FROM JETBRAINS



OFFICIAL SUPPORT FOR ANDROID DEVELOPMENT



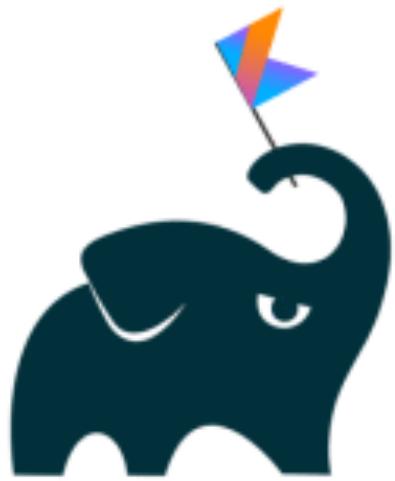
Google I/O 2017

SPRING SUPPORT

The screenshot shows a blog post on the Spring.io website. The header includes links for DOCS, GUIDES, PROJECTS, BLOG (which is highlighted in green), and QUESTIONS, along with a search icon. The post title is "Introducing Kotlin support in Spring Framework 5.0". Below the title are author information (SEBASTIEN DELEUZE), publication date (JANUARY 04, 2017), and comment count (47 COMMENTS). A sidebar on the left lists categories like ENGINEERING, RECOMMENDED, and NEWS. The main content discusses the introduction of Kotlin support in Spring Framework 5.0, mentioning its interoperability with Java libraries and additional Kotlin dedicated features.

<https://spring.io/blog/2017/01/04/introducing-kotlin-support-in-spring-framework-5-0>

The screenshot shows the Spring Initializr web application. It features a sidebar with links for ALL POSTS, ENGINEERING, RECOMMENDED, and NEWS. The main area has a dark header with the text "SPRING INITIALIZR bootstrap your application now". Below this, there are two main sections: "Generate a [Maven Project] with [Java, Kotlin, Groovy] and Spring Boot [1.5.6]" and "Project Metadata" (Artifact coordinates: Group com.example, Artifact demo). To the right, there is a "Dependencies" section with a "Selected Dependencies" dropdown containing "Web, Security, JPA, Actuator, Devtools...". At the bottom, there is a "Generate Project" button and a note about switching to the full version.



GRADLE SUPPORT

 Gradle Blog

Kotlin Meets Gradle

May 18, 2016 Chris Beams New Features

Many readers will be familiar with JetBrains' excellent [Kotlin](#) programming language. It's been under development since 2010, had its first public release in 2012, and went 1.0 GA earlier this year.

We've been watching Kotlin over the years, and have been increasingly impressed with what the language has to offer, as well as with its considerable uptake—particularly in the Android community.

Late last year, Hans sat down with a few folks from the JetBrains team, and they wondered together: what might it look like to have a Kotlin-based approach to writing Gradle build scripts and plugins? How might it help teams—especially big ones—work faster and write better structured, more maintainable builds?

The possibilities were enticing.

<https://blog.gradle.org/kotlin-meets-gradle>

<https://github.com/gradle/kotlin-dsl>

- Use Kotlin for writing
 - Gradle Script Kotlin (`build.gradle.kts`)
 - Plugins
- Benefits
 - auto-completion and content assist
 - quick documentation
 - navigation to source
 - refactoring

INTERESTING FACTS

Started development in 2010 at Jetbrains

Open sourced in 2012

First official stable release v1 - Feb 15, 2016

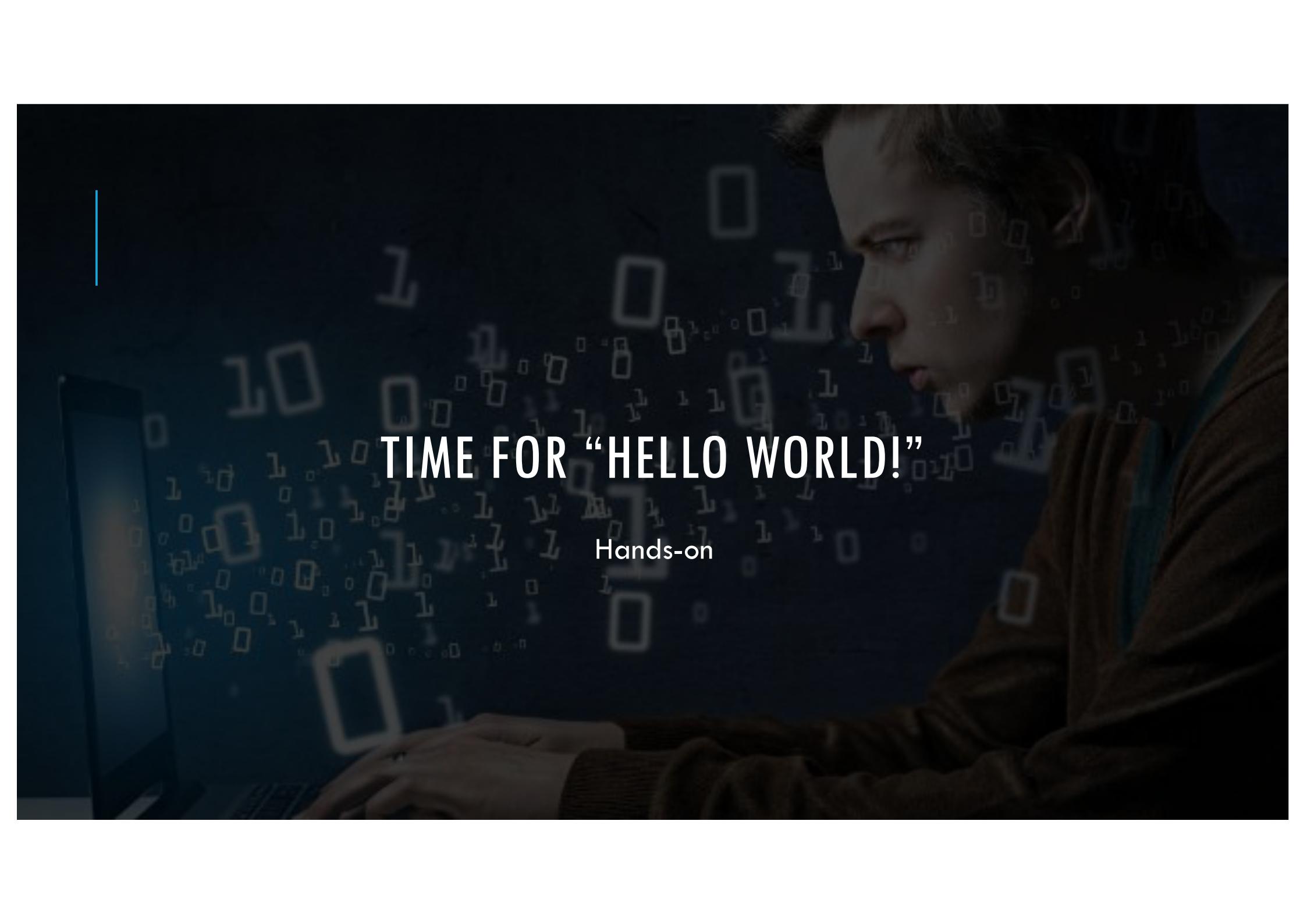
(current release v1.1.4 – Aug 15, 2017)

Used in all Jetbrains IDEs

100% interoperable with Java

Syntax is similar to Swift

Influenced by Java, Scala, Groovy, C#, Gosu, JavaScript, Swift

A person with short hair is sitting at a desk, looking down at a laptop screen. Binary code (0s and 1s) is visible on the screen and floating around the person's head, suggesting a digital or programming environment.

TIME FOR “HELLO WORLD!”

Hands-on

DEFAULT IMPORTS

<https://kotlinlang.org/docs/reference/packages.html>

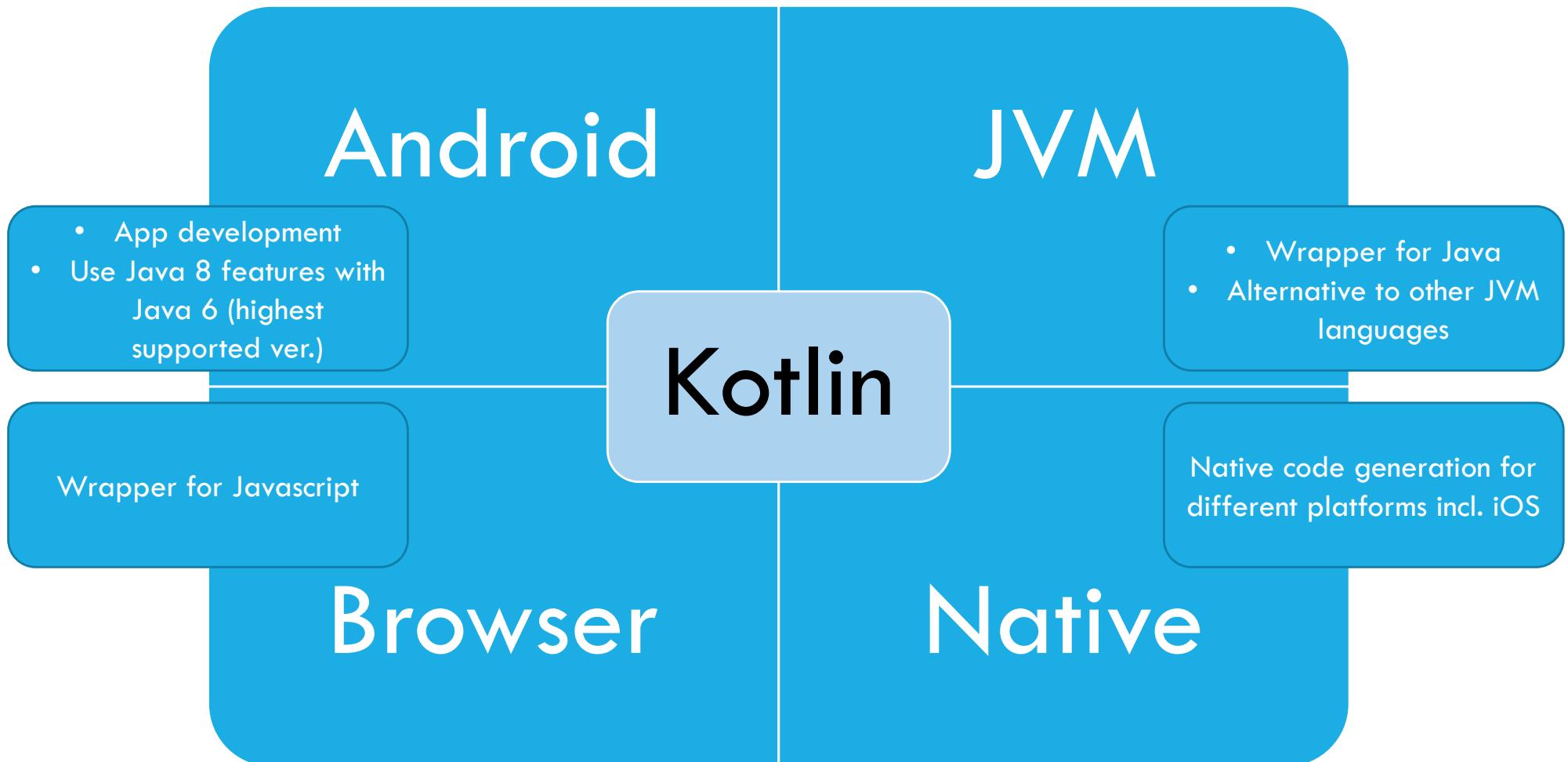
A number of packages are imported into every Kotlin file by default:

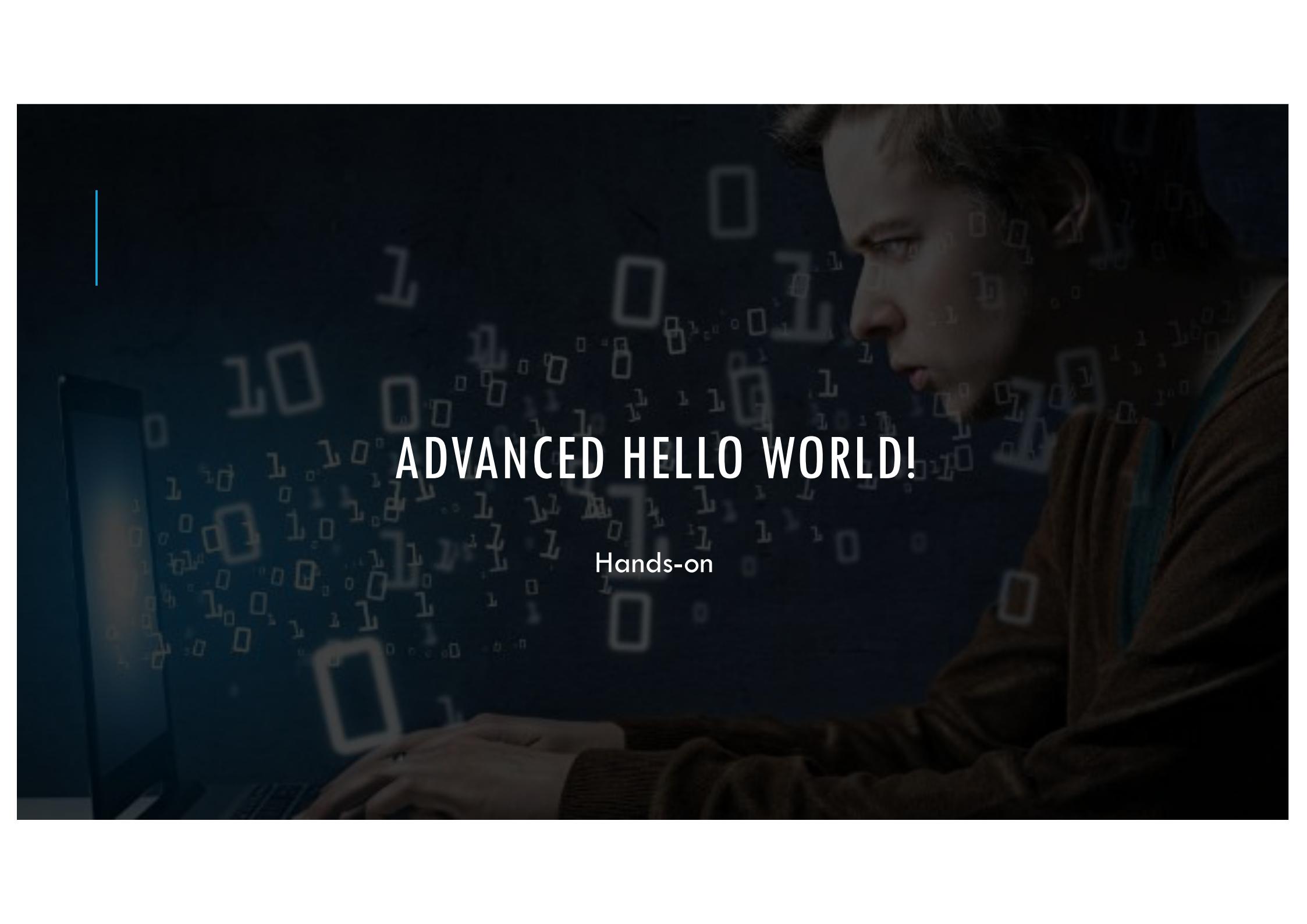
- [kotlin.*](#)
- [kotlin.annotation.*](#)
- [kotlin.collections.*](#)
- [kotlin.comparisons.*](#) (since 1.1)
- [kotlin.io.*](#)
- [kotlin.ranges.*](#)
- [kotlin.sequences.*](#)
- [kotlin.text.*](#)

Additional packages are imported depending on the target platform:

- JVM:
 - `java.lang.*`
 - [kotlin.jvm.*](#)
- JS:
 - [kotlin.js.*](#)

LANDSCAPE



A person with short hair, wearing a brown jacket over a blue shirt, is sitting at a desk and working on a laptop. Binary code (0s and 1s) is visible as a floating overlay around their head and hands, suggesting digital interaction or programming. A vertical blue bar is on the left side.

ADVANCED HELLO WORLD!

Hands-on

VISIBILITY MODIFIERS

Top level declarations (classes, functions, properties ...)

Modifier	Visibility
Public (default)	world
Private	file
Internal	module

Members inside a class

Modifier	Visibility
Public (default)	world
Private	class
Protected	class and sub-class
Internal	module

A dark, moody photograph of a person from the side, wearing a hoodie, looking down at a laptop keyboard. Floating around the person are large, semi-transparent binary digits (0s and 1s) and square shapes, suggesting digital data or code.

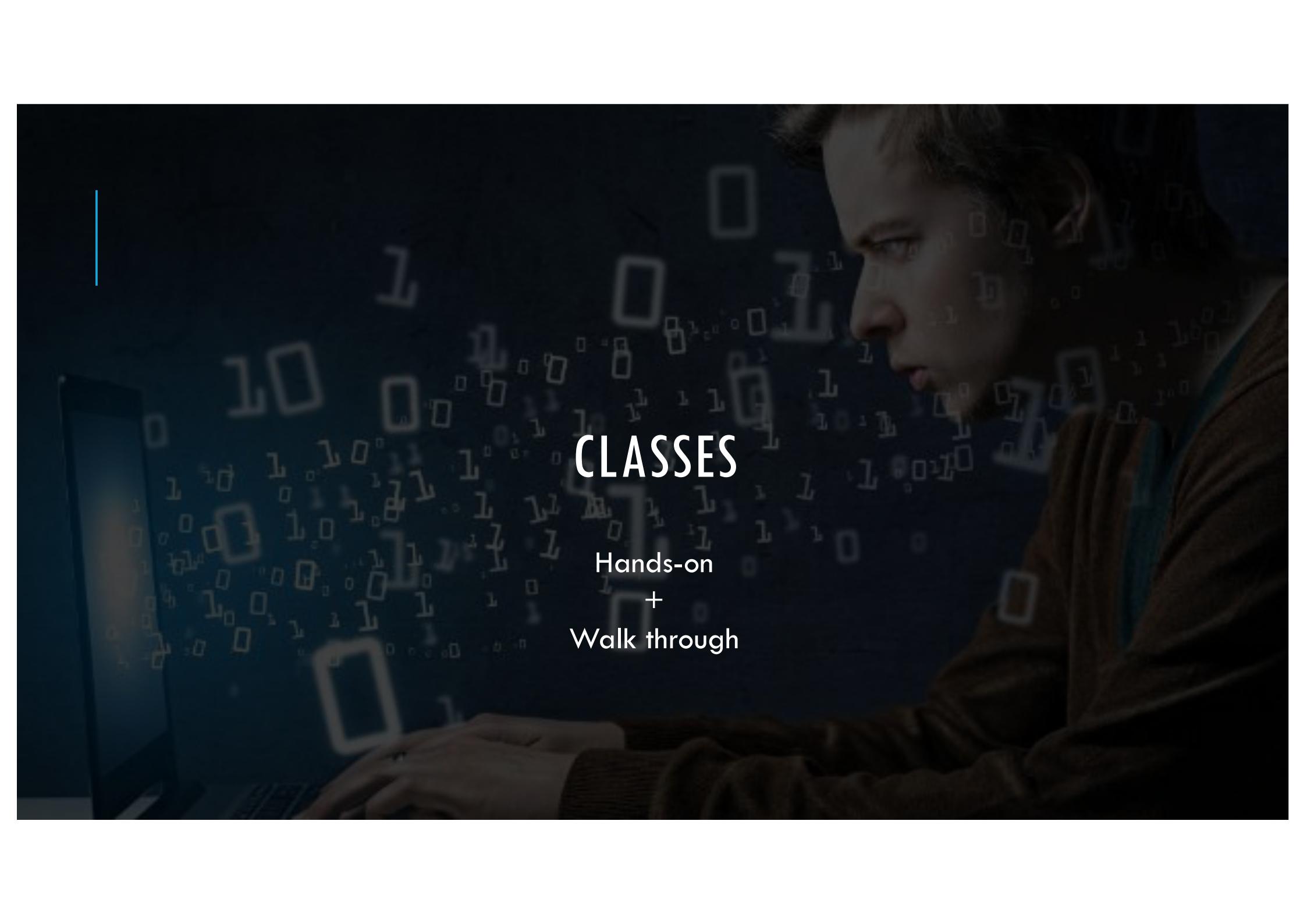
INTEROP WITH JAVA

Hands-on

A dark-themed photograph of a person from the side, wearing a brown jacket, working on a laptop. Numerous binary digits (0s and 1s) are floating in the air around their head, suggesting digital thought or data processing.

DATA TYPES & CONTROL FLOWS

Walk through

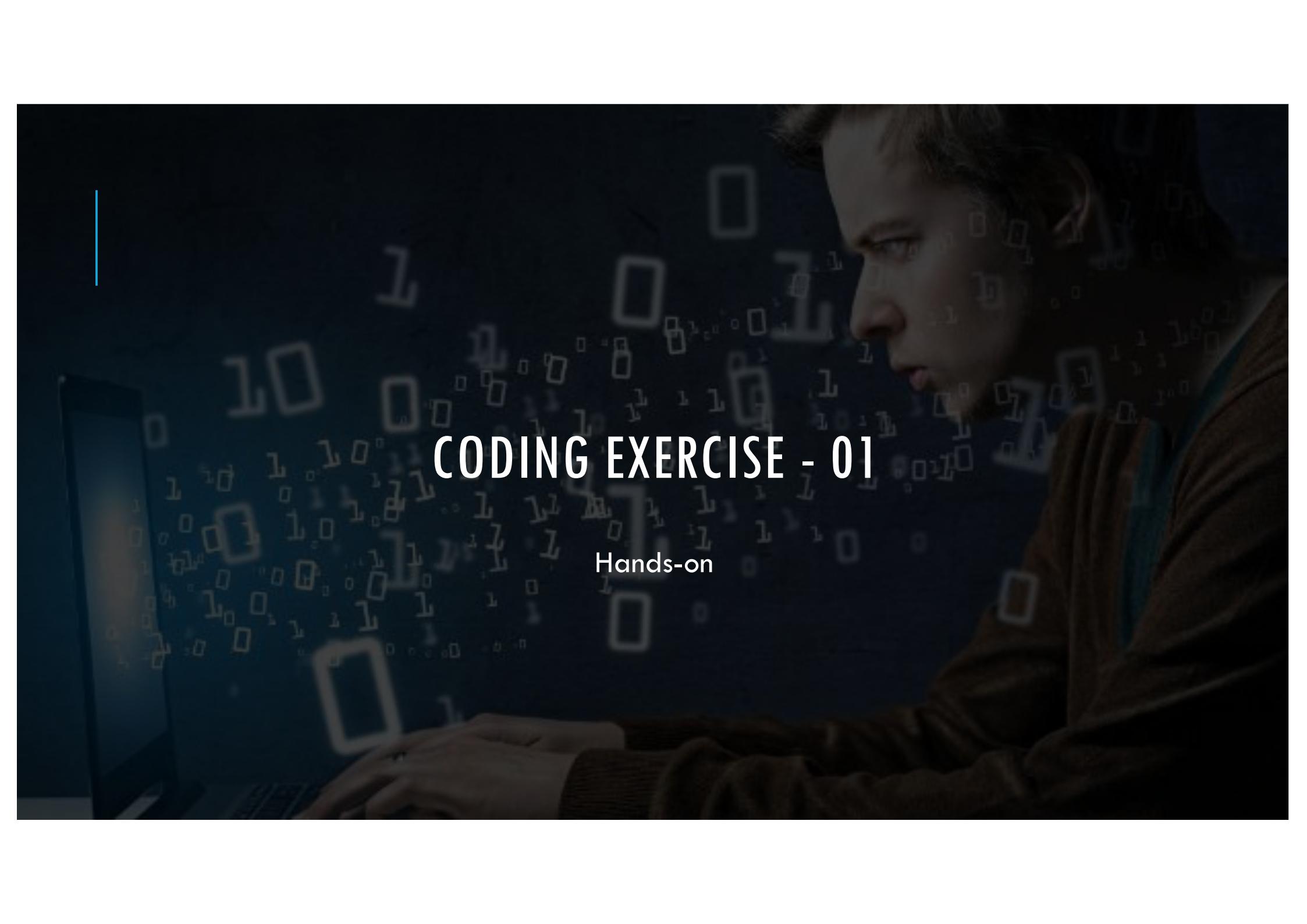
A dark, atmospheric photograph of a person wearing a virtual reality headset. Binary code (0s and 1s) is visible as floating digital particles around their head. The person's hands are visible at the bottom, suggesting they are interacting with a VR interface.

CLASSES

Hands-on
+
Walk through

SAMPLE CODE GITHUB REPO

<https://github.com/shaunthomas999/kotlin-workshop>

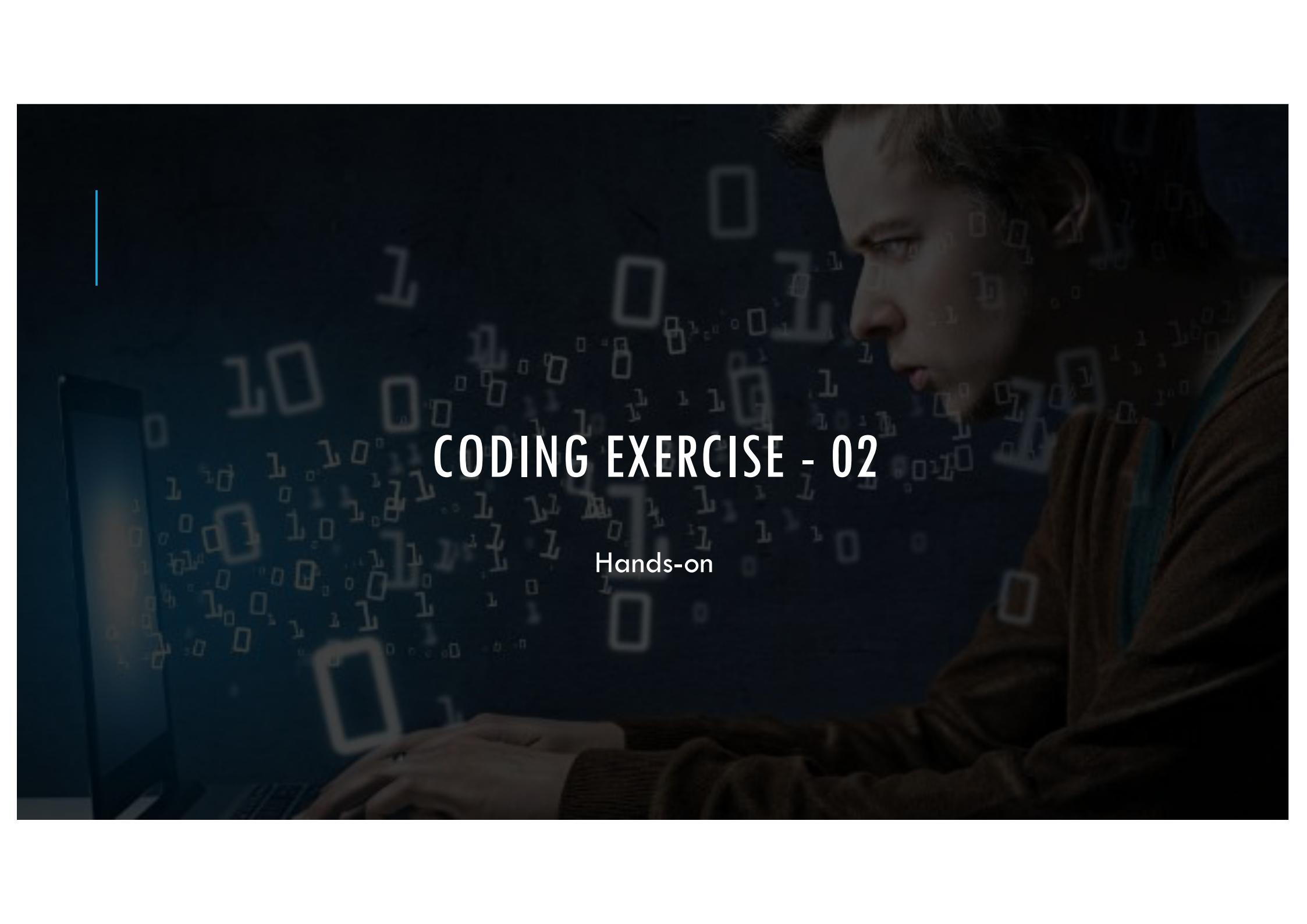
A person is shown from the side, wearing a brown jacket, working on a laptop. Binary code (0s and 1s) is depicted as floating 3D cubes in various colors (blue, green, yellow) around the person's head and hands, suggesting a digital or coding environment.

CODING EXERCISE - 01

Hands-on

FUNCTIONS

Walk through

A person is shown from the side, wearing a brown jacket, working on a laptop. Binary code (0s and 1s) is depicted as floating 3D cubes in the air around their hands and the laptop screen.

CODING EXERCISE - 02

Hands-on

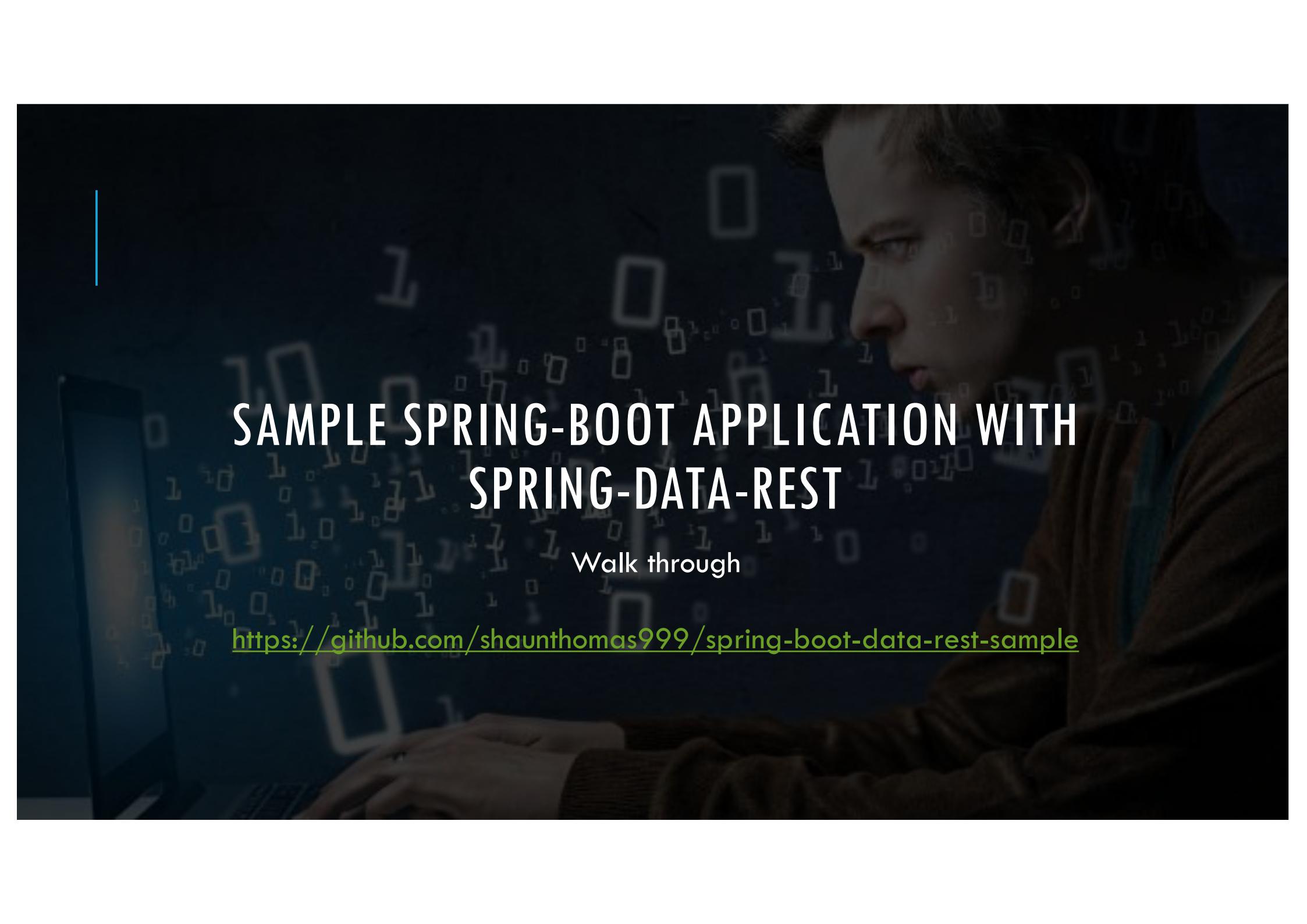
DATA CLASSES

Hands-on

A dark, moody photograph of a person from the side, wearing a hoodie, looking down at a laptop screen. Floating around the person are large, semi-transparent binary digits (0s and 1s) and square shapes, suggesting digital data or code.

CODING EXERCISE - 03

Hands-on

A dark-themed background featuring a person from the side, looking down at a laptop keyboard. Numerous binary digits (0s and 1s) are floating in the air around the person's head, suggesting digital work or data processing.

SAMPLE SPRING-BOOT APPLICATION WITH SPRING-DATA-REST

Walk through

<https://github.com/shaunthomas999/spring-boot-data-rest-sample>

CREATE PROJECT

Go to <http://start.spring.io/>

SPRING INITIALIZR bootstrap your application now

Generate a with and Spring Boot

Project Metadata

Artifact coordinates

Group

Artifact

Dependencies

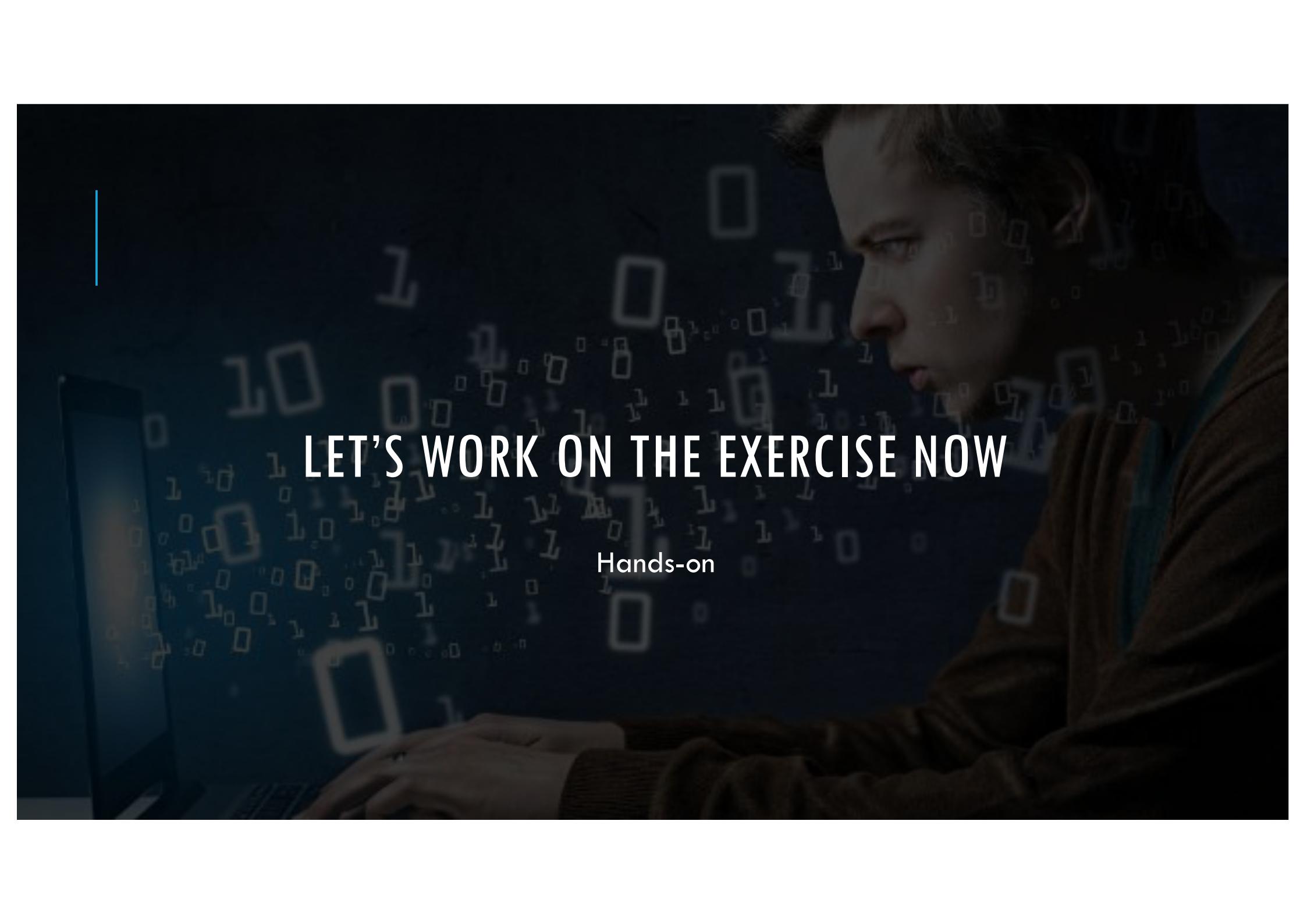
Add Spring Boot Starters and dependencies to your application

Search for dependencies

Selected Dependencies

⌘ + ↵

Don't know what to look for? Want more options? [Switch to the full version.](#)

A person with blonde hair is sitting at a desk, looking down at a laptop screen. Binary code (0s and 1s) is floating around their head in a 3D space, suggesting digital data or programming. The background is dark.

LET'S WORK ON THE EXERCISE NOW

Hands-on

EXTRA KOTLIN-MAVEN-PLUGIN CONFIGURATION REQUIRED

```
<build>
    <sourceDirectory>${project.basedir}/src/main/kotlin</sourceDirectory>
    <testSourceDirectory>${project.basedir}/src/test/kotlin</testSourceDirectory>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
        <plugin>
            <artifactId>kotlin-maven-plugin</artifactId>
            <groupId>org.jetbrains.kotlin</groupId>
            <version>${kotlin.version}</version>
            <configuration>
                <compilerPlugins>
                    <plugin>spring</plugin>
                    <plugin>jpa</plugin> ←
                </compilerPlugins>
                <jvmTarget>1.8</jvmTarget>
            </configuration>
            <executions...>
            <dependencies>
                <dependency...>
                <dependency>
                    <groupId>org.jetbrains.kotlin</groupId>
                    <artifactId>kotlin-maven-noarg</artifactId> ←
                    <version>${kotlin.version}</version>
                </dependency>
            </dependencies>
        </plugin>
    </plugins>
</build>
```



- General syntax

`<keyword> <name>: <Type>`

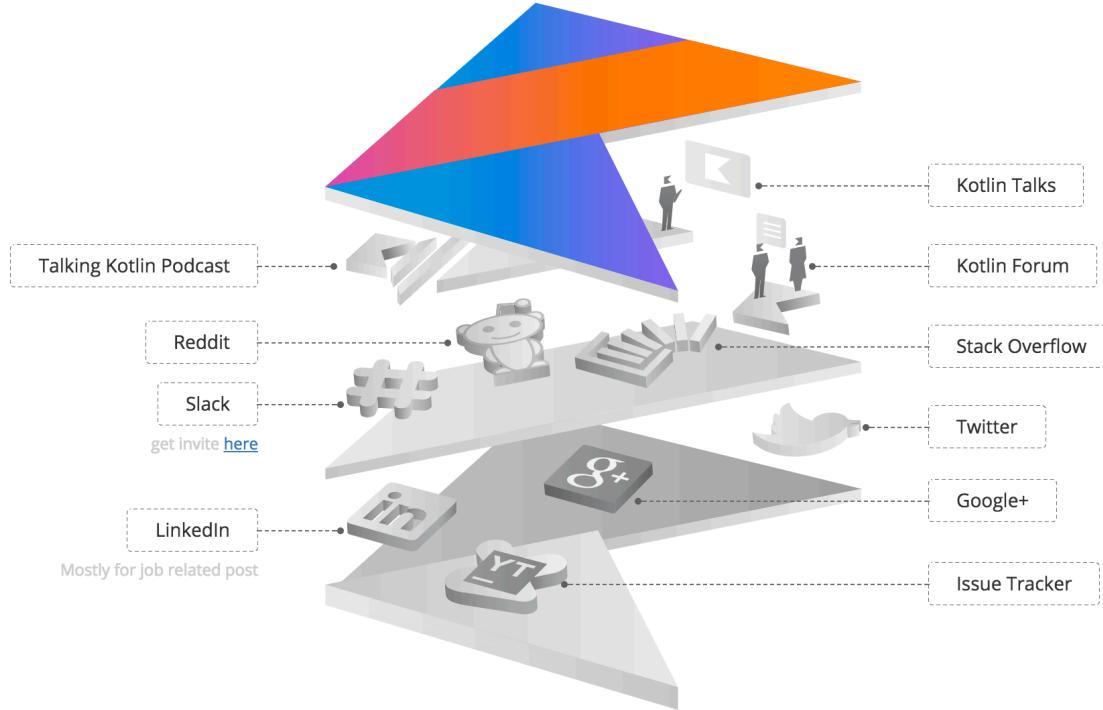
- There is no 'new' operator in Kotlin
- By default Kotlin classes, properties and functions are public and final
- Everything is an object in Kotlin
- Numbers, characters and boolean can have primitive values at runtime
- `main()` should not be inside a class

RESOURCES

<https://kotlinlang.org/community/>

<https://kotlinlang.org/docs/reference/>

Kotlin Online Resources



Community-Maintained Resources

1. [Kotlin.link](#) — almost a complete list of links to Kotlin projects, libraries and other resources
2. [Kotlin Weekly](#) — a weekly mailing list containing latest notable news and articles
3. [The Daily Kotlin](#) — Daily Kotlin news and tips

THANK YOU

Shaun Thomas

shaunthomas999@gmail.com

www.shaunthomas999.com