Let's Program in Kotlin

Sebastian Aigner
for IEEE Student
Branch Passau

About me

- Hi, I‘m Sebastian!
- Education Advocate
- Love tinkering and toying with tech.
- 🥰 Kotlin

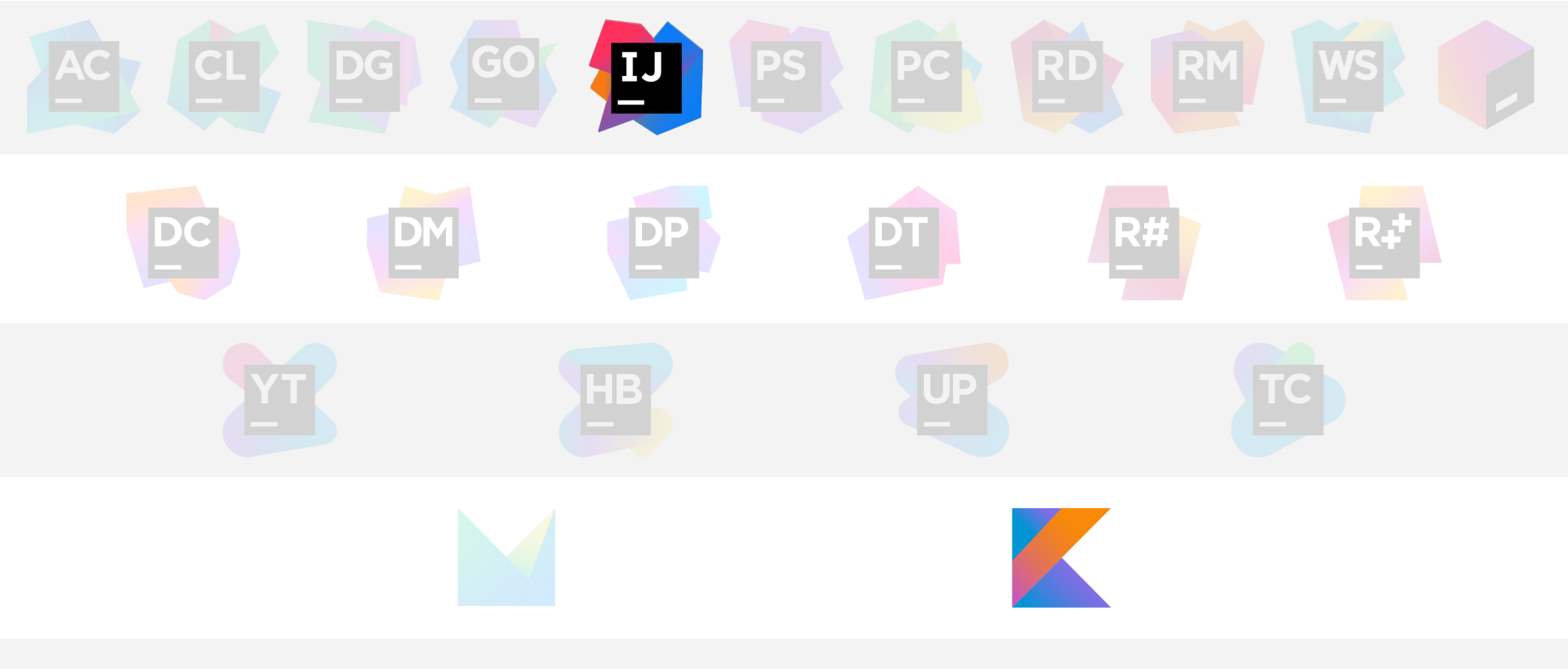
Fig. 1: me

twitter.com/TrueSebi • github.com/SebastianAigner • sebastian.aigner@jetbrains.com

About JetBrains

About JetBrains

About JetBrains



Kotlin?

Kotlin is a **multi-platform**, **multi-paradigm** programming language with an emphasis on **conciseness**.

Key Features

- Object-Oriented
- Functional
- Strongly Typed
- Null-Safe
- Concise

Object Oriented

- Classes
- Methods
- Interfaces
- Inheritance
- Generics

Object Oriented... and concise!

```
public class Customer {
    private int id;
    private String firstName;
    private String lastName;
    private double estimatedValue;

    public Customer(int id, String firstName, String lastName, double estimatedValue) {
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
        this.estimatedValue = estimatedValue;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }
```

Object Oriented... and concise!

```
data class Customer(
    var id: Int,
    var firstName: String?,
    var lastName: String?,
    var estimatedValue: Double
)
```

Type marked as nullable!

Null-Safe?

```
public class Api {

    public String getResponse() {
        if(everythingOkay) {
            return "Okay";
        } else {
            return null;
        }
    }
}
```

Will crash at runtime!

```
Api api = new Api();
String response = api.getResponse();
System.out.println(response.length());
```

Null-Safe!!

```
class KotlinApi {
    fun getResponse(): String? {
        return if (everythingOkay) {
            "Okay"
        } else {
            null
        }
    }
}
```

The compiler and IDE are your friends!

```
val a = KotlinApi()
a.getResponse() .length
```

Will compile!

Functional

- Functions as first class citizens
- Higher-Order Functions
- Extension functions

Functions as first class citizens

```
val increment: (Int) -> Int = { a: Int -> a + 1 }

val x = increment(5)
```

Higher-Order Functions

```
fun repeat(repetitions: Int, action: (Int) -> Unit) {
    for(i in 0 until repetitions) {
        action(i)
    }
}

repeat(10) {
    println(it)
}
```

Trailing Lambda!

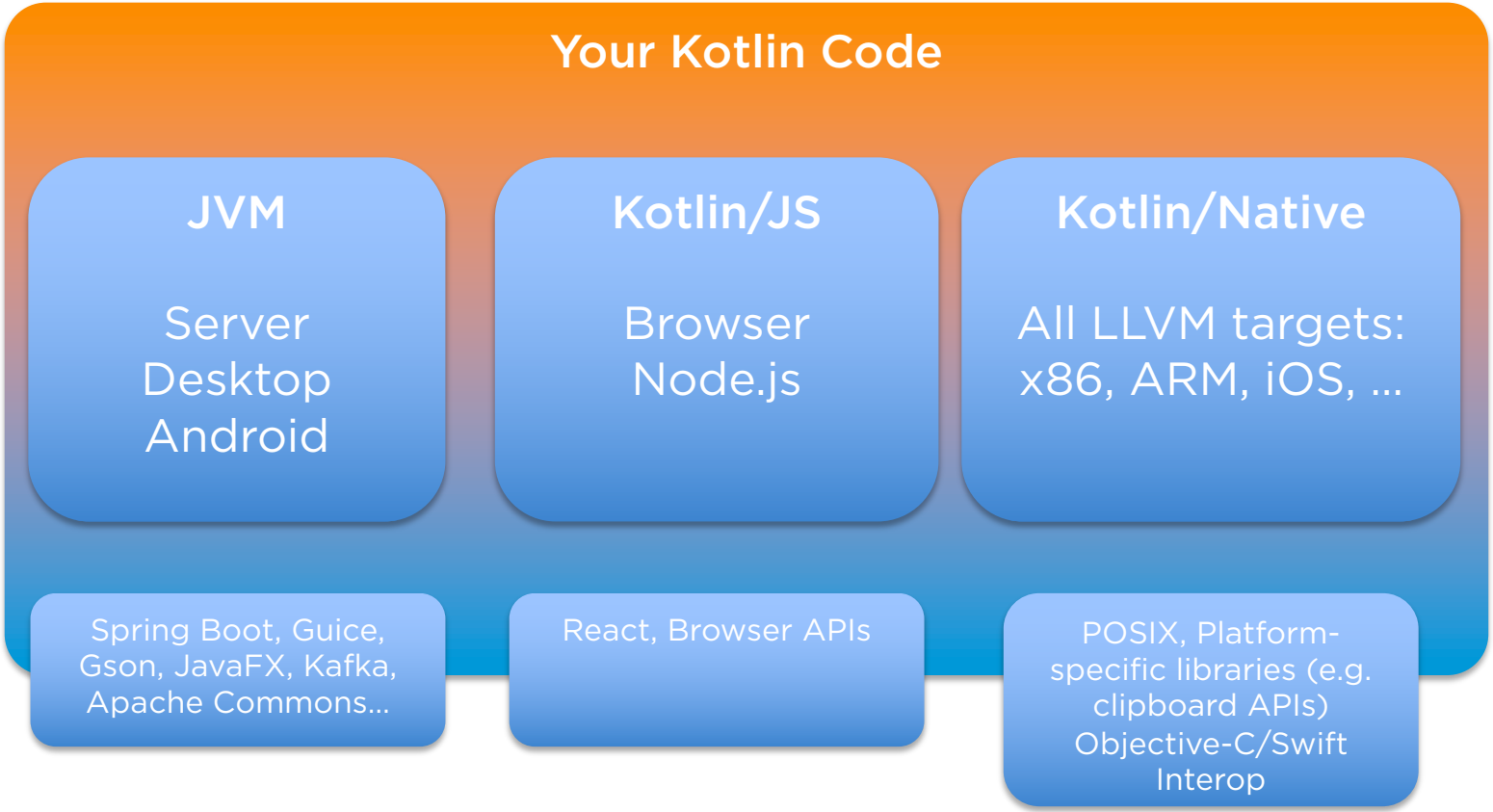
Kotlin!

Kotlin is a **multi-platform**, multi-paradigm programming language with an emphasis on conciseness.

Multi-Platform

- **JVM**
  - Run on server, desktop, Android
  - 100% interoperable with existing Java code
- **JavaScript (Kotlin/JS)**
  - Run in the browser/Node.JS
- **Native (Kotlin/Native)**
  - Create native binaries through LLVM
  - Run on all LLVM targets, ARM/iOS.

Multi-Platform



...and we got a lot more!

Coroutines and structured concurrency

Infix operators

One-click migrations

Solid standard library

Type safe Domain Specific Languages

Smart Casting

Auto-convert from Java

Operator Overloading

Lambdas with Receivers

Inline functions

Contracts

Serialization

Tail recursive function unrolling

Kotlin is loved by many!

Basecamp

Evernote

Uber

Square

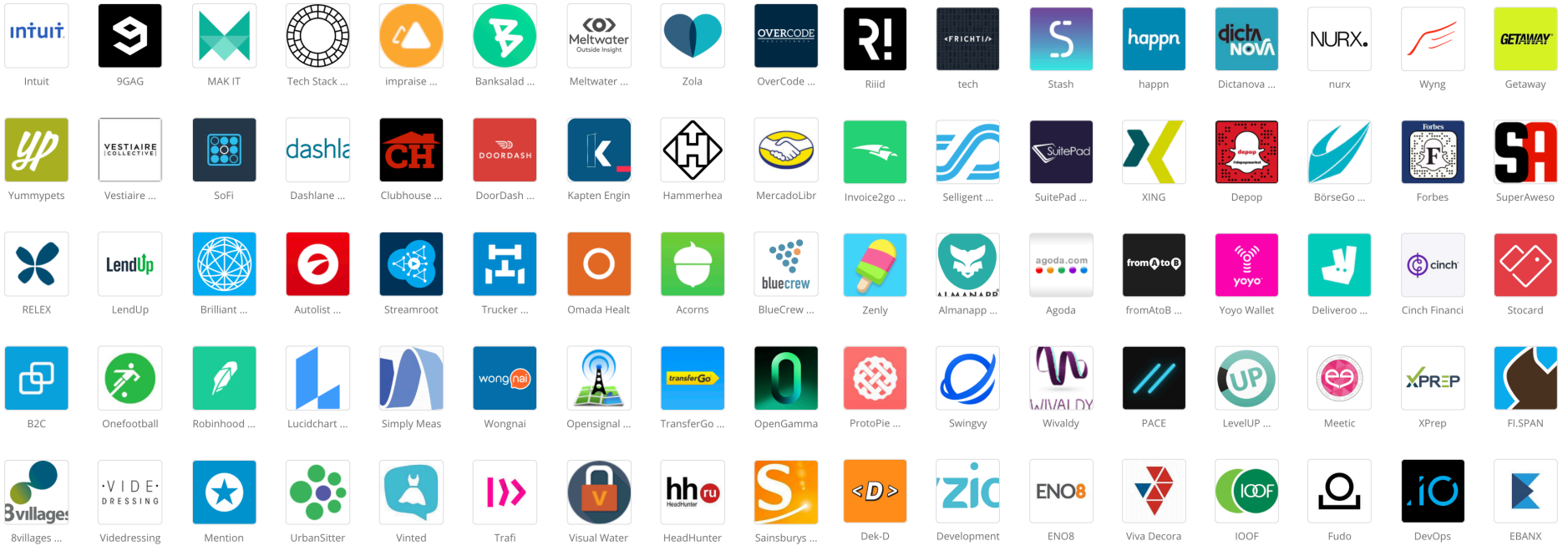
coursera

Trello

Pinterest

Google

Kotlin is loved by many!



(source: stackshare.io)

Let’s pass an interview in Kotlin!

FizzBuzz

The "Fizz-Buzz test" is an interview question designed to help filter out the 99.5% of programming job candidates who can't seem to program their way out of a wet paper bag.

http://wiki.c2.com/?FizzBuzzTest

Write a program that prints the numbers from 1 to 100.

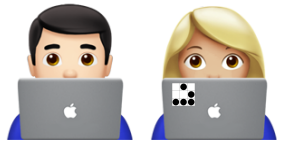
But for multiples of **three** print "**Fizz**" instead of the number and for multiples of **five** print "**Buzz**".

For numbers which are multiples of both **three and five** print "**FizzBuzz**".

FizzBuzz in Kotlin

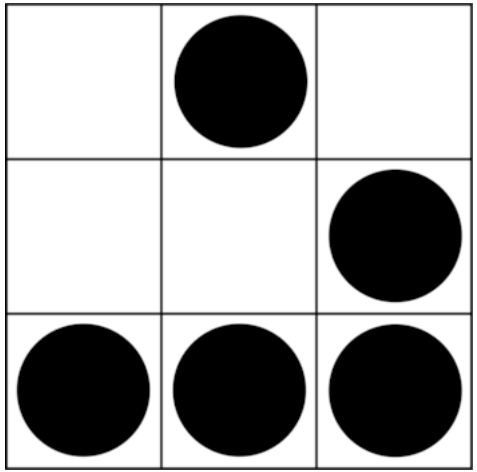
IntelliJ: Your favorite IDE

- Cross-platform Java & Kotlin IDE
- **Ultimate Edition**
  - Free for students
- **Community Edition**
  - Open source & free forever
  - Full support for Kotlin development



”Congratulations, you’ve moved on to the next stage. Wow us!”

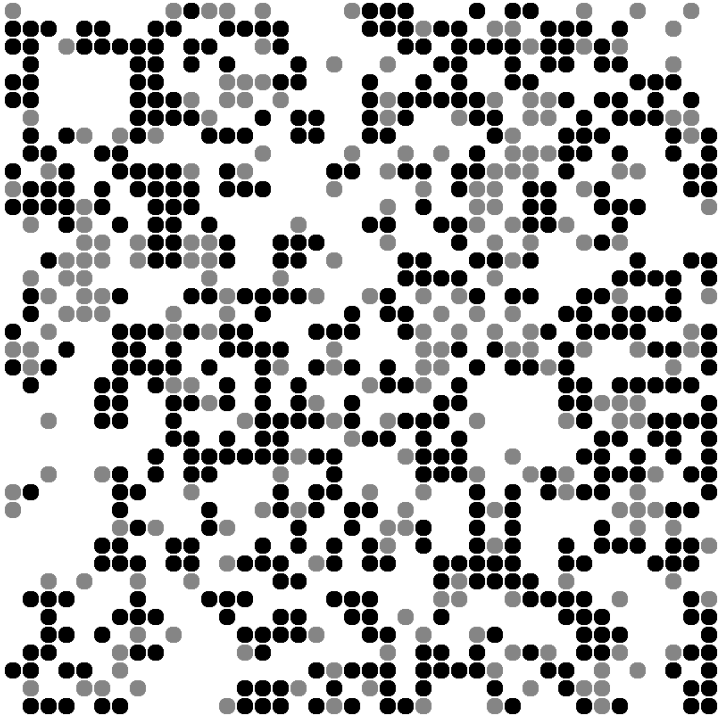
Hacker Glider



hack-er[1] *n. Informal* An intelligent, creative, and open-minded individual who enjoys problem-solving, learning, and the **sharing of knowledge above all else.**

From https://danielmiessler.com/blog/the-hacker-glider/

Conway‘s Game of Life



Conway‘s Game of Life

- Cellular automaton
- Created by John Horton Conway in 1970
- Simple set of rules generate complex patterns

John Conway

Basic Concept

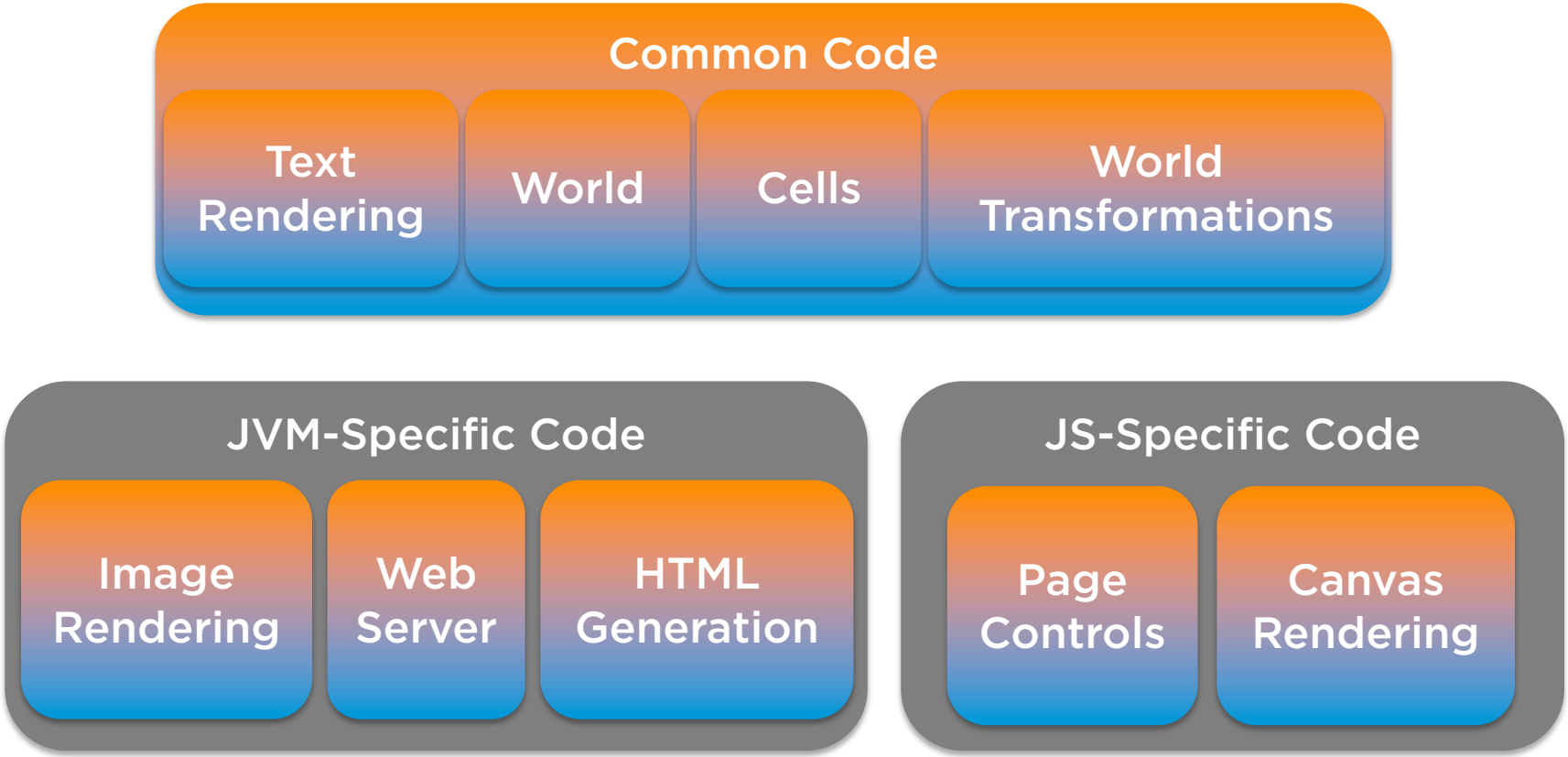
- Rectangular world
- Cells are either alive or dead
- The world is iterated in discrete time steps

Rules of the Game

- **Underpopulation:** Any live cell with fewer than two live neighbours dies. Cells are either alive or dead
- **Just Right:** Any live cell with two or three live neighbours lives on to the next generation.
- **Overpopulation:** Any live cell with more than three live neighbours dies.
- **Reproduction:** Any dead cell with exactly three live neighbours becomes a live cell.

”Wow us!”

Our Challenge

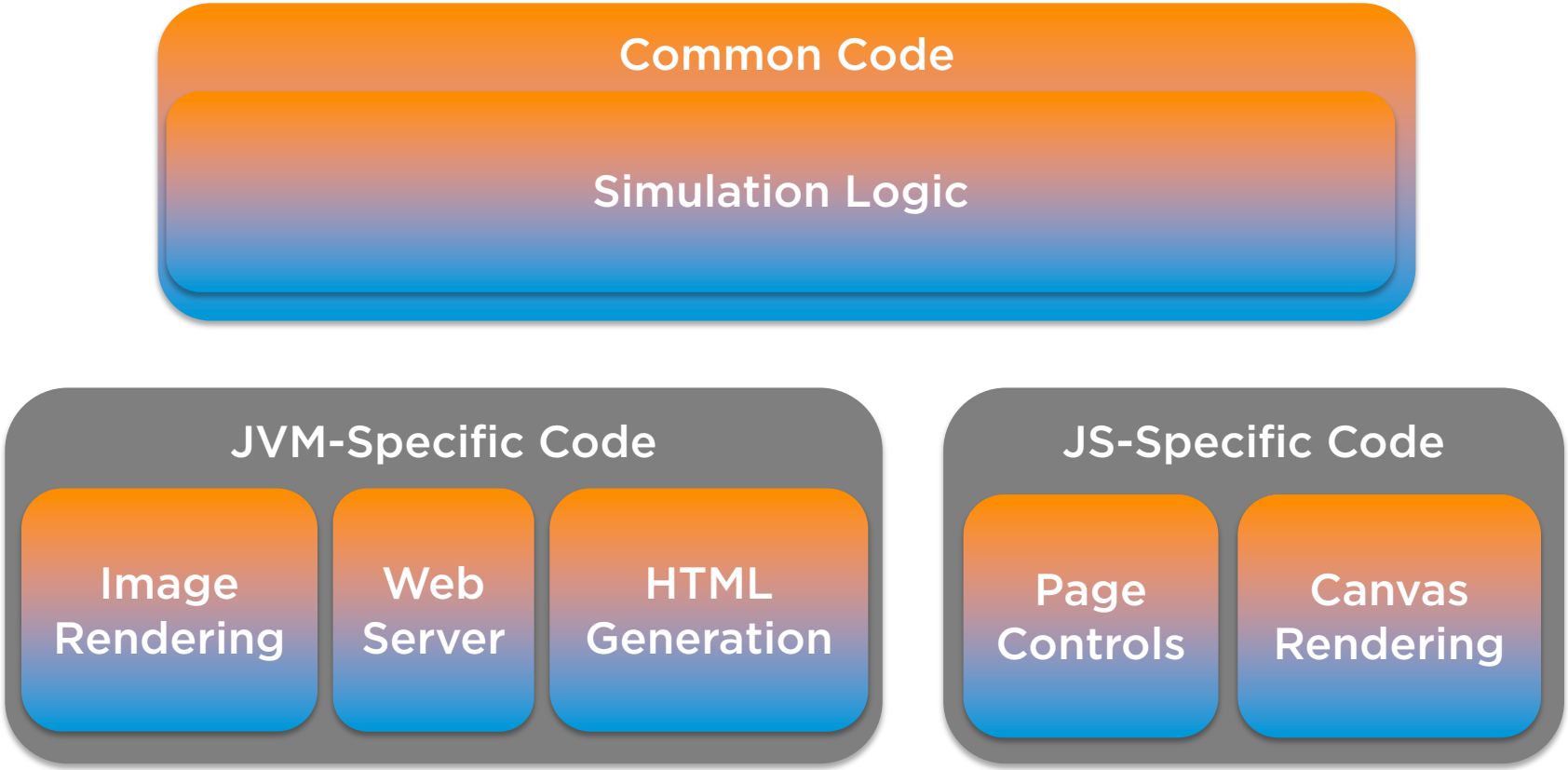


Let’s write some code!

Reusing Code for Multiple Platforms

- Supported by IntelliJ and templates
- One project, multiple platforms
- No need to write adapters
- Write platform-specific code in Kotlin

Our Challenge



Kotlin/JVM: Ktor

- Connected application framework
- Driven by coroutines
- Developed & supported by JetBrains
- Open Source

Compare to: Flask, Ruby, Express

Domain Specific Languages

- Clever combination of functional constructs
- Express your intent instead of manipulating the language.
- No hacks
- Example: Typesafe HTML

Other examples: CSS, K8s, Ktor routing

Let’s write some JVM code!

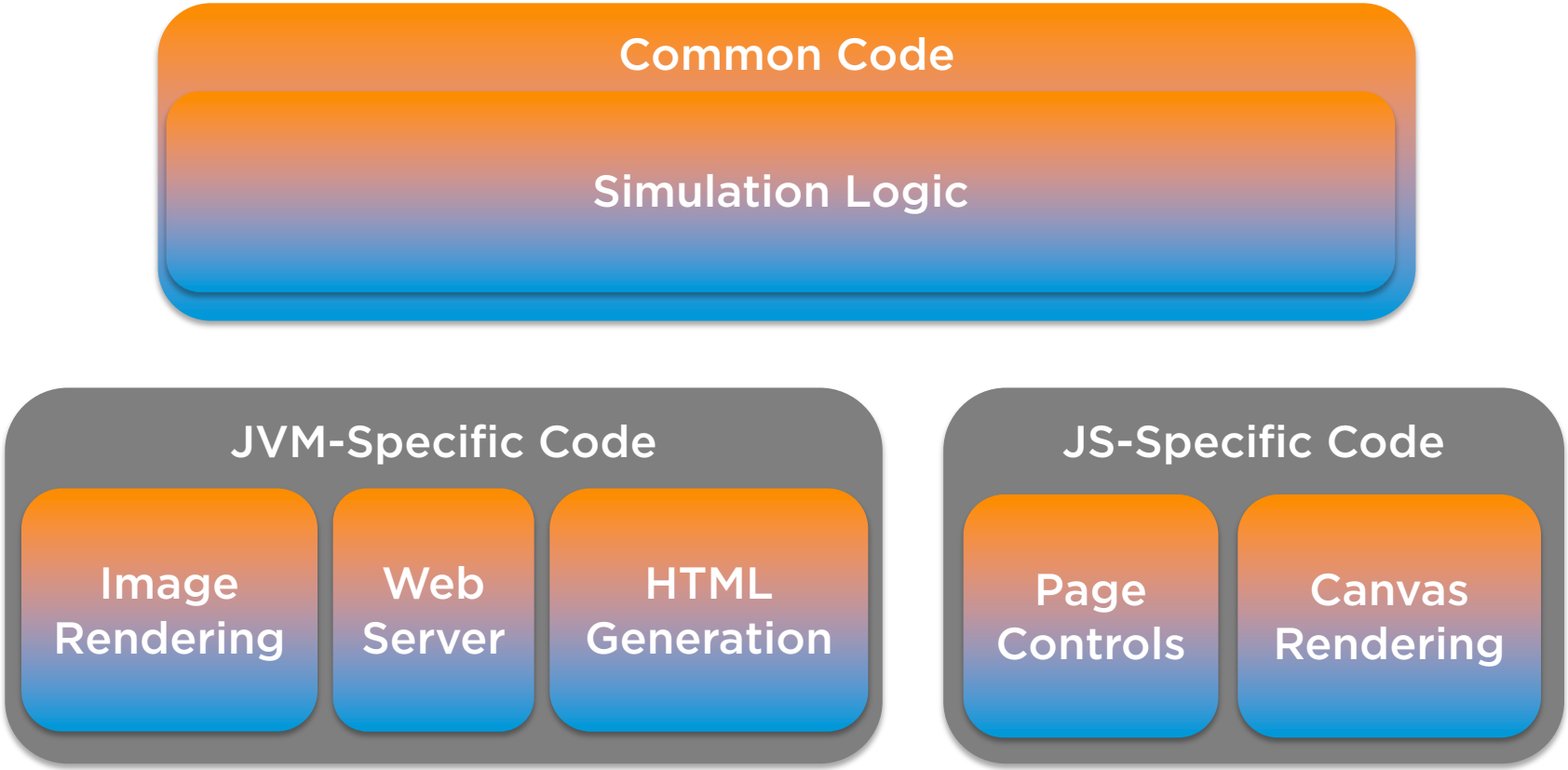
The Elvis Operator

source: dobsondev.com

```
call.parameters["step"]?.toInt() ?: return@get
```

- Compiler knows that if the `toInt` cast fails, code below will not be executed
- Therefore, code below this statement can assume that the cast succeeded!
  → type is `Int` instead of `Int?`

Our Challenge

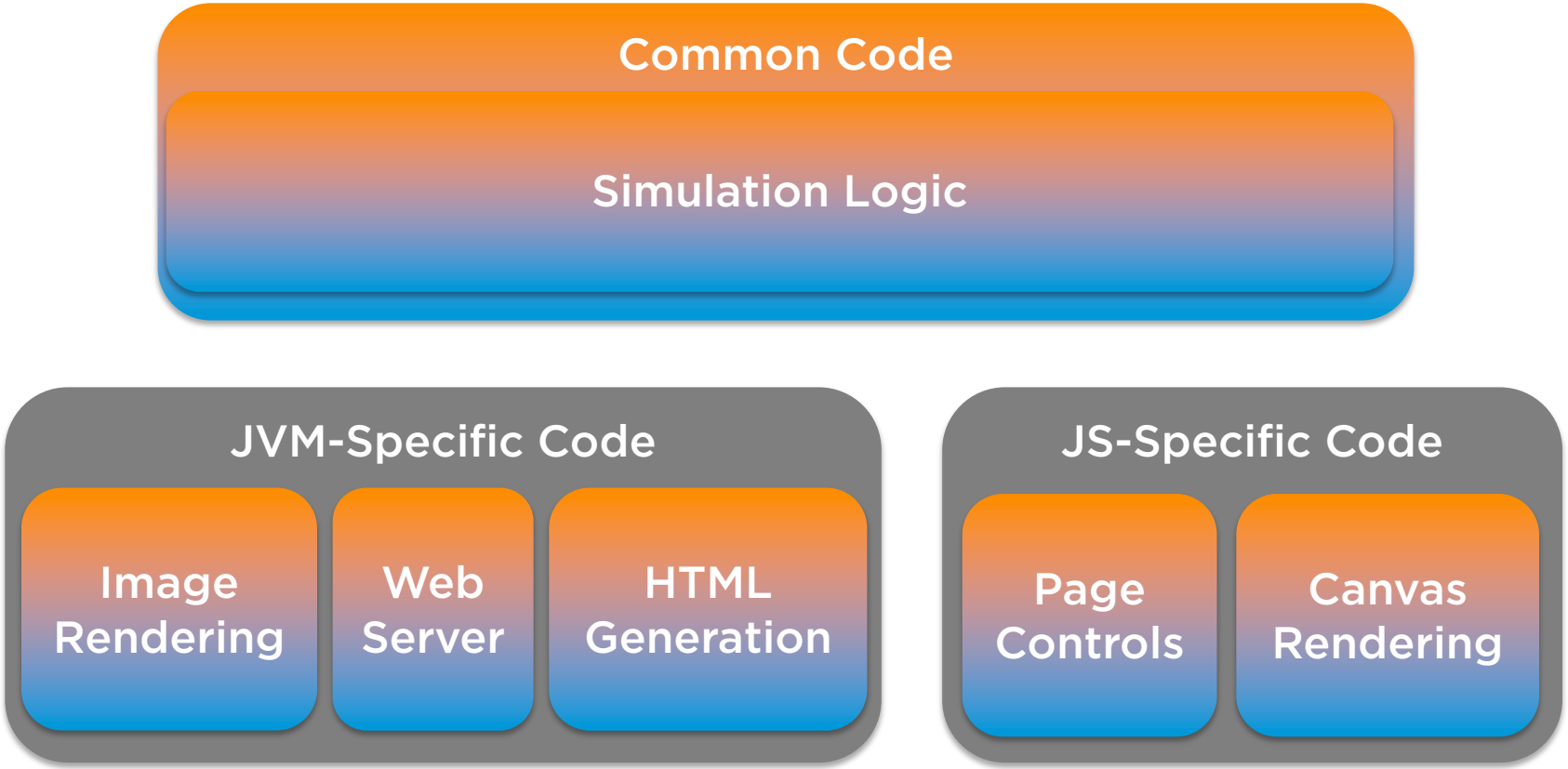


Kotlin/JS: Browser APIs

- Typesafe wrappers for standard APIs
- Same naming conventions
- Expose Kotlin/JS functions to JavaScript, vice-versa

Let’s write some JS code!

Done!


Common Code
Simulation Logic
JVM-Specific Code
Image Rendering
Web Server
HTML Generation
JS-Specific Code
Page Controls
Canvas Rendering

What could be next?

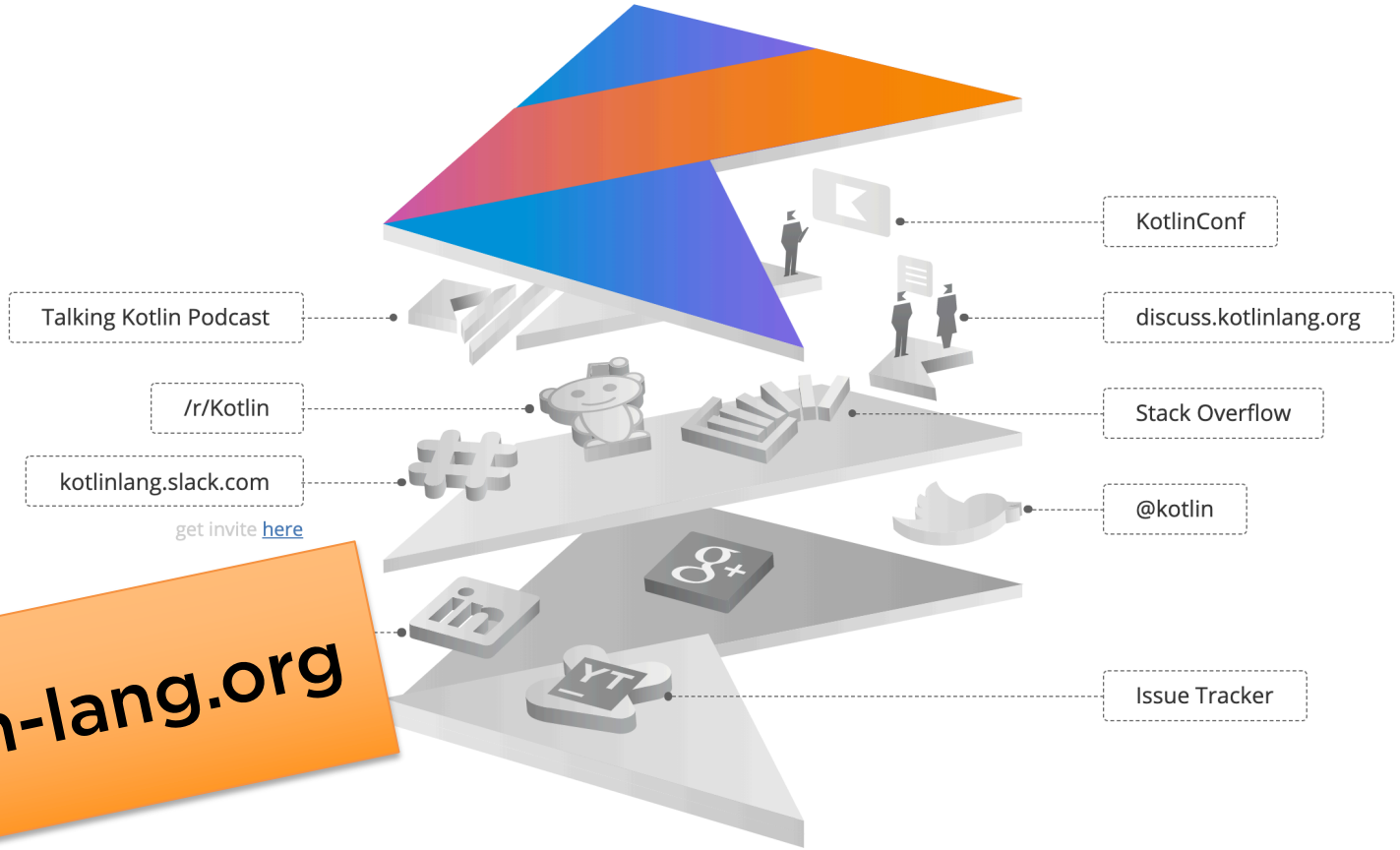
- Include more platforms:
  - Android
  - iOS
  - Native x86
- Leverage more frameworks
  - E.g. Typesafe React for the Frontend

”Wow!”

What have we learned?

- Kotlin is
  - Concise.
  - Safe.
  - Truly Multiplatform.
  - Adopted by many.
  - Always evolving.
  - Fun to write!

Continue your learning journey



kotlin-lang.org

Grow your local community

- JetBrains supports user groups
  - Maybe soon: Kotlin User Group Passau?
- JetBrains supports universities
  - Want to learn Kotlin in your university course?