



# BACKGROUND

- Started in 2010 by JetBrains
- We needed a language
  - Concise, Expressive, Toolable, Interoperable...
- **Pragmatic**
- Developed under Apache 2 OSS on GitHub

# WHAT IS KOTLIN

- Static Language
- Targeting JVM/JavaScript and Native

# CURRENT STATE

- 1.0 Release on 15th February 2016
- 40+ Developers at JetBrains
- 100+ Committers
- Used in close to 10 Products at JetBrains
- External Companies include **Expedia, NBC News Digital, Netflix, Amex** amongst others

# WHERE CAN I USE IT?

- Anywhere
- Android - Compatible with Java 6. Small runtime
- Similarity to Java/C#/JavaScript/Groovy allows quick ramp-up time
- Interoperability allows gradual adoption



# HOW CAN I USE IT?

- Command Line
- Maven
- Gradle
- Kobalt
- Ant
- IntelliJ IDEA
- Android Studio
- Eclipse
- NetBeans

LET'S SEE SOME CODE

# ANKO

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
  
    verticalLayout {  
        padding = dip(30)  
        editText {  
            hint = "Name"  
            textSize = 24f  
        }  
        editText {  
            hint = "Password"  
            textSize = 24f  
        }  
        button("Login") {  
            textSize = 26f  
        }  
    }  
}
```



# ANDROID EXTENSIONS

```
// Using R.layout.activity_main from the main source set
import kotlinx.android.synthetic.main.activity_main.*

public class MyActivity : Activity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        textView.setText("Hello, world!") // Instead of findViewById(R.id.textView) as
        TextView
    }
}
```

# SPRING BOOT

**SPRING INITIALIZR** bootstrap your application now

Generate a  with Spring Boot

### Project Metadata

Artifact coordinates

Group

Artifact

Name

Description

Package Name

Packaging

Java Version

Language

### Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Selected Starters

# KOBALT

```
import com.beust.kobalt.*
import com.beust.kobalt.plugin.java.*
import com.beust.kobalt.plugin.packaging.*
import com.beust.kobalt.plugin.publish.*

val jcommander = project {
    name = "jcommander"
    group = "com.beust"
    artifactId = name
    version = "1.52"

    dependenciesTest {
        compile("org.testng:testng:6.9.5")
    }

    assemble {
        mavenJars {
        }
    }

    bintray {
        publish = false
    }
}
```



# COMING IN 1.1

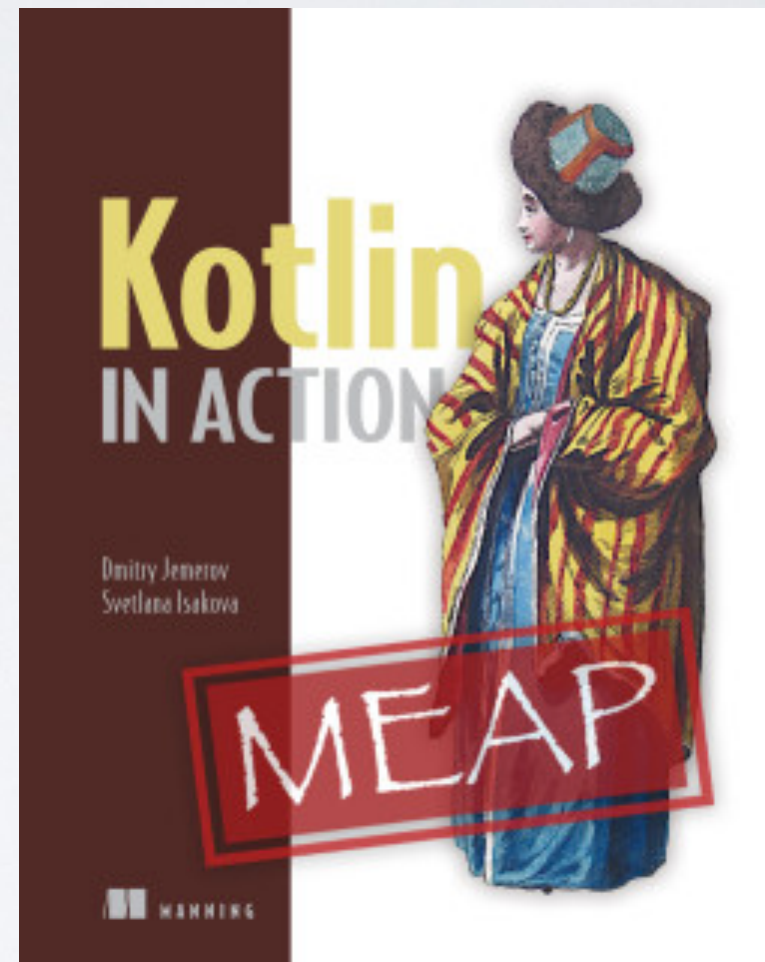
- Coroutines (async/await)
- Data Class Hierarchies
- Type Aliases
- Deconstructing in Lambdas
- Bound Method References
- Local Delegated Properties
- Java 8/9 Support
- JavaScript Support



# NEXT STEPS

- [kotlinlang.org](https://kotlinlang.org)
- [try.kotlinlang.org](https://try.kotlinlang.org)
- Kotlin Koans (online/offline)

# BOOKS



# O'REILLY COURSE

Introduction to Kotlin (5 Hours)  
Advanced Kotlin Programming (3 Hours)

# COMMUNITY

**#general**  
4364 members | Welcome, have fun


Yesterday

**max** 16:22  
I mean safe call `?.`

**Kirill Rakhman** 18:21  
`@yoavst` you should probably rewrite your code to to use the expression form of try  
`val foo = try { bar() } catch (e: Exception) { baz() }`

**Yoav Sternberg** 18:49  
Yea, done that.

**Dmitry Kandalov** 19:35  
How should I name unused parameter? I'd use `_` but it's reserved in Kotlin.  
E.g. `val (x, value) = someFunction()` where I don't care about `x`.



**Kotlin**  
`@kotlin`

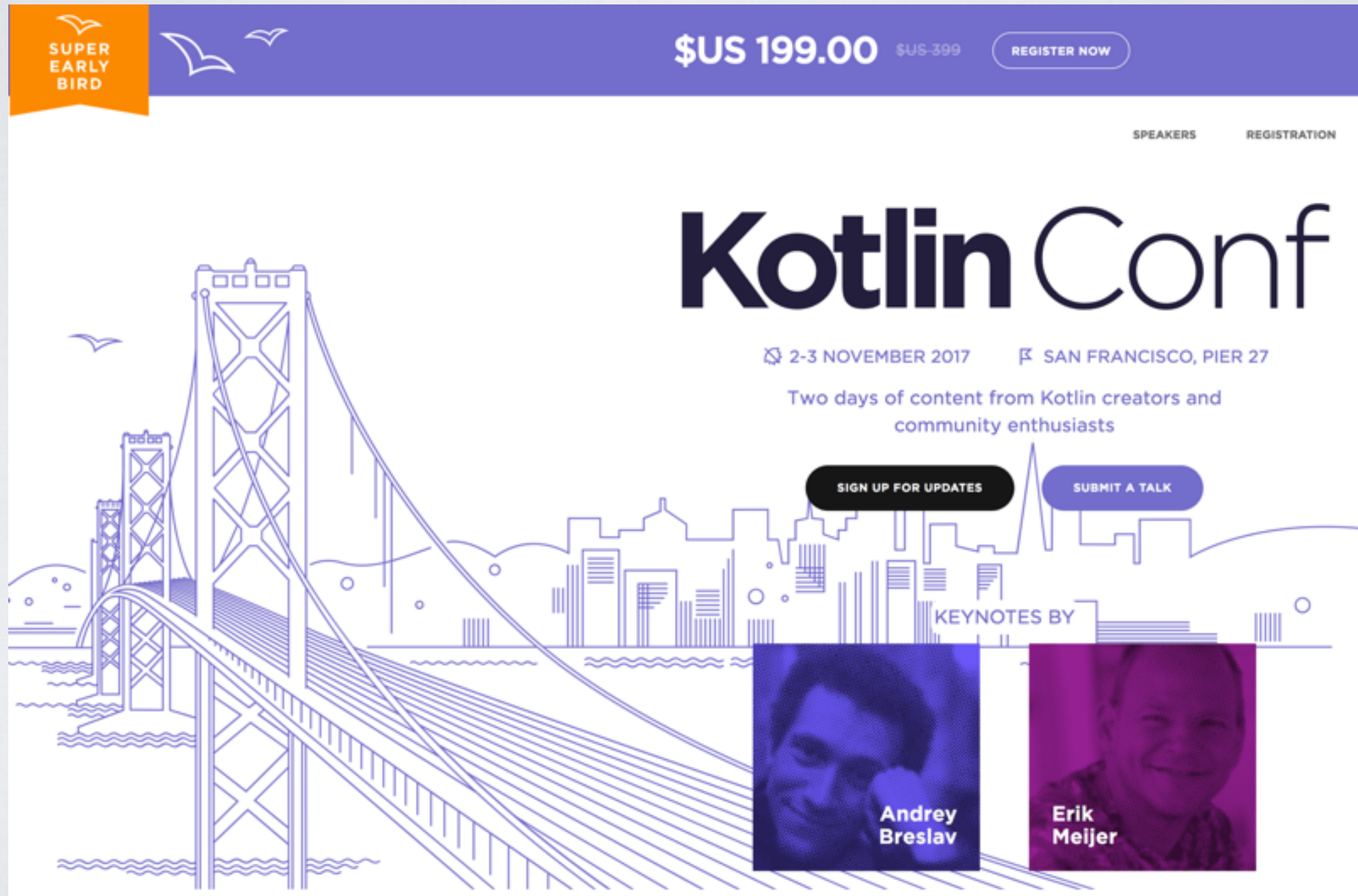
Statically typed Programming Language  
targeting JVM and JavaScript -  
Sponsored and Developed by  
`@JetBrains` Join us on Stack:  
`kotlinstackin.herokuapp.com`

`kotlinlang.org`  
Joined July 2011

Tweet to Kotlin



# KOTLINCONF



**SUPER EARLY BIRD**

**\$US 199.00** \$US 399 [REGISTER NOW](#)

[SPEAKERS](#) [REGISTRATION](#)

# Kotlin Conf

📅 2-3 NOVEMBER 2017 📍 SAN FRANCISCO, PIER 27

Two days of content from Kotlin creators and community enthusiasts

[SIGN UP FOR UPDATES](#) [SUBMIT A TALK](#)

KEYNOTES BY

**Andrey Breslav**

**Erik Meijer**



# SUMMARY

- Pragmatic Language
- Easy Learning Curve
- Interoperability provides low-risk adoption
- Enjoyable
- Kotlin is here to stay



**Brett Miller**  
@WrathZA



 **Follow**

Researched Kotlin last night and then spending the morning looking at mountains of boilerplate Java in Android Studio. I see why they did it

8:35 AM - 16 Feb 2016



THANK YOU

@hhariri - hadi@jetbrains.com