

Kotlin for Java Developers

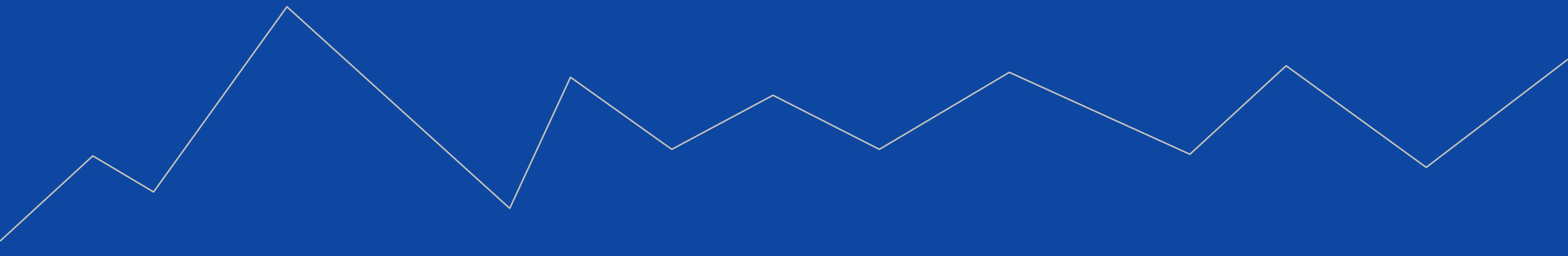


Ruslan Ibragimov

- Belarus Kotlin User Group Leader
- Java Professionals BY Leader
- FullStack Developer at ObjectStyle
- Kotliner :)



Language



Hello, Kotlin

```
fun main(args: Array<String>) {  
    println("Hi, JUG Latvia!")  
}
```

Hello, Kotlin

```
object Application {  
    @JvmStatic  
    fun main(args: Array<String>) {  
        println("Hi, JUG Latvia!")  
    }  
}
```

```
class Application {  
    companion object {  
        @JvmStatic  
        fun main(args: Array<String>) {  
            println("Hi, JUG Latvia!")  
        }  
    }  
}
```

Write some fun

```
fun sum(a: Int, b: Int): Int {  
    return a + b  
}
```

```
fun sum(a: Int, b: Int) = a + b
```

```
fun main(args: Array<String>) {  
    println("Hi, JUG Latvia!")  
}
```

```
fun main(args: Array<String>): Unit {  
    println("Hi, JUG Latvia!")  
}
```



Write some fun

```
fun sum(a: Int, b: Int = 42) = a + b  
sum(42)
```

```
sum(a = 12, b = 42)
```

Local variables

```
val final = "Hello"
```

```
var nonFinal = "Hello"
```

```
var nullable: String?
```

```
var nullable: String? = null
```

Properties

```
class Service {  
    @Autowired  
    lateinit var JpaRepository: String  
}
```

Strings

```
val text = "Length of $name is ${name.length}"
```

```
val text = """  
|Tell me and I forget.  
|Teach me and I remember.  
|Involve me and I learn.  
|(Benjamin Franklin)  
""".trimMargin()
```

Null?

String



Only String

String?



String or null

// Java, NullPointerException at Runtime

String nullable = **null**;

nullable.length();

// Compile Error: Only safe (?.) or non-null...

var nullable: String? = **null**

nullable.**length**

Null?

```
var nullable: String? = null
```

```
val len = nullable?.length // null
```

```
val len = nullable!!.length  
// KotlinNullPointerException
```

```
val len = nullable?.length ?: 42  
// 42
```

```
val len = nullable?.let { 42 }  
// 42
```

OOP

```
class Person(var name: String)
```

```
class Person(val firstName: String,  
             val lastName: String) {  
    fun getFullName() = "$firstName + $lastName"  
}
```

```
val person = Person("Ruslan", "Ibragimov")  
person.getFullName()
```

Data Class

```
data class Person(val name: String)
```

- properties
- equals/hashCode
- toString
- copy
- component

OOP

```
interface Base {  
    fun print()  
    fun default() {  
        // ...  
    }  
}
```

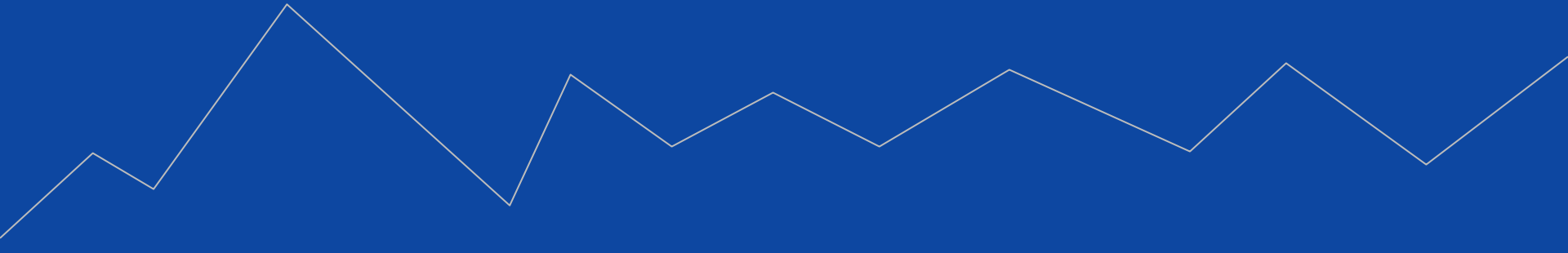
```
class Person(val firstName: String) : Base {  
    override fun print() {  
        // ...  
    }  
}
```


Build

```
buildscript {  
    ext.kotlin_version = '1.0.5-2'  
    repositories {  
        jcenter()  
    }  
    dependencies {  
        classpath("org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version")  
    }  
}  
  
apply plugin: "kotlin"  
  
repositories {  
    jcenter()  
}  
  
dependencies {  
    compile("org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version")  
}
```

IntelliJ Idea: Configurare Kotlin In Project

Language “Advanced”



Lambdas

```
val lambda = {}
```

```
val lambda = { -> }
```

```
val lambda: () -> Unit = {}
```

```
val λ: (Int) -> Int = { num -> num * 2 }
```

```
λ(2) // 4
```

```
val λ: (Int) -> Int = { it * 2 }
```

```
λ(2) // 4
```

Extension Functions

// File: Extensions.kt

```
fun Any?.print() = println(this)
```

```
person.print()
```

// Java

```
ExtensionsKt.print(object);
```

Extension Properties

```
val String.bd: BigDecimal  
  get() = BigDecimal(this)
```

```
"1.0".bd
```

Expressions, Expressions Everywhere!

```
val max = if (a > b) {  
  print("Choose a")  
  a  
} else {  
  print("Choose b")  
  b  
}
```

```
val hasPrefix = when (s) {  
  is String -> s.startsWith("prefix")  
  else -> false  
}
```

When

```
val hasPrefix = when (s) {  
  is String -> s.startsWith("prefix")  
  else -> false  
}
```

```
when {  
  x.isOdd() -> print("x is odd")  
  x.isEven() -> print("x is even")  
  else -> print("x is funny")  
}
```

ADT

```
sealed class Tree {  
  object Empty: Tree()  
  data class Leaf(val x: Int): Tree()  
  data class Node(val left: Tree, val right: Tree): Tree()  
  
  fun max(): Int = when (this) {  
    Empty -> Int.MIN_VALUE  
    is Leaf -> this.x  
    is Node -> Math.max(this.left.max(), this.right.max())  
  }  
}
```

Visibility

internal

Delegation

```
interface Base {  
    fun print()  
}
```

```
class BaseImpl(val x: Int) : Base {  
    override fun print() {  
        print(x)  
    }  
}
```

```
class Derived(b: Base) : Base by b
```

Delegation Properties

```
val lazyVal: String by lazy {  
    // ...  
    "Hello"  
}
```

Inline Functions

```
val a = Any()
```

```
synchronized(a) {
```

```
}
```

Inline Functions

```
fun getNumber(): Int {  
    var count = 0
```

```
    synchronized(count) {  
        count++  
        return count  
    }
```

```
}
```

Inline Functions

```
inline fun <R> synchronized(  
    lock: kotlin.Any,  
    block: () -> R  
): R { ... }
```

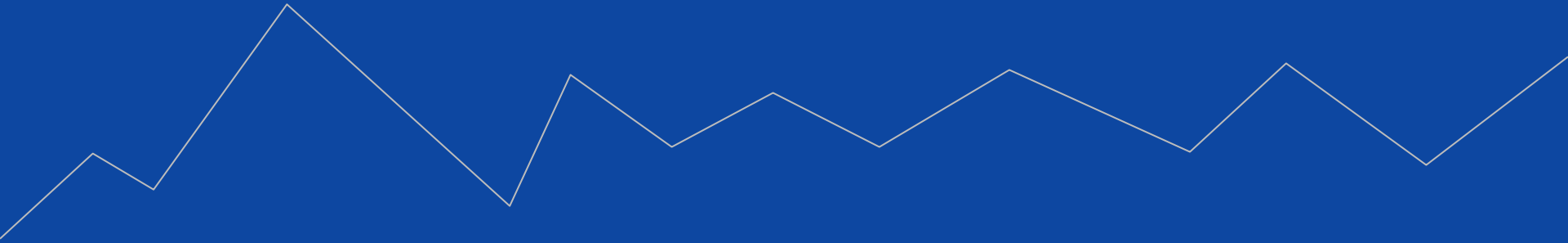
```
fun getNumber(): Int {  
    var count = 0
```

```
    synchronized(count) {  
        count++  
        return count  
    }  
}
```

Inline Functions + Generics = Reified Generics

```
inline fun <reified T : Any> Gson.fromJson(json): T =  
    this.fromJson(json, T::class.java)
```

Yet Another Language?



Kotlin Targets

Java Bytecode (Compile Target)

Android Platform (Performance, Method Count, Size)

Java (Interoperability)

JavaScript (Interoperability, Compile Target)

Native (LLVM* Compile Target -> iOS, IoT?)

Language Design Trade-offs

What do you prefer?

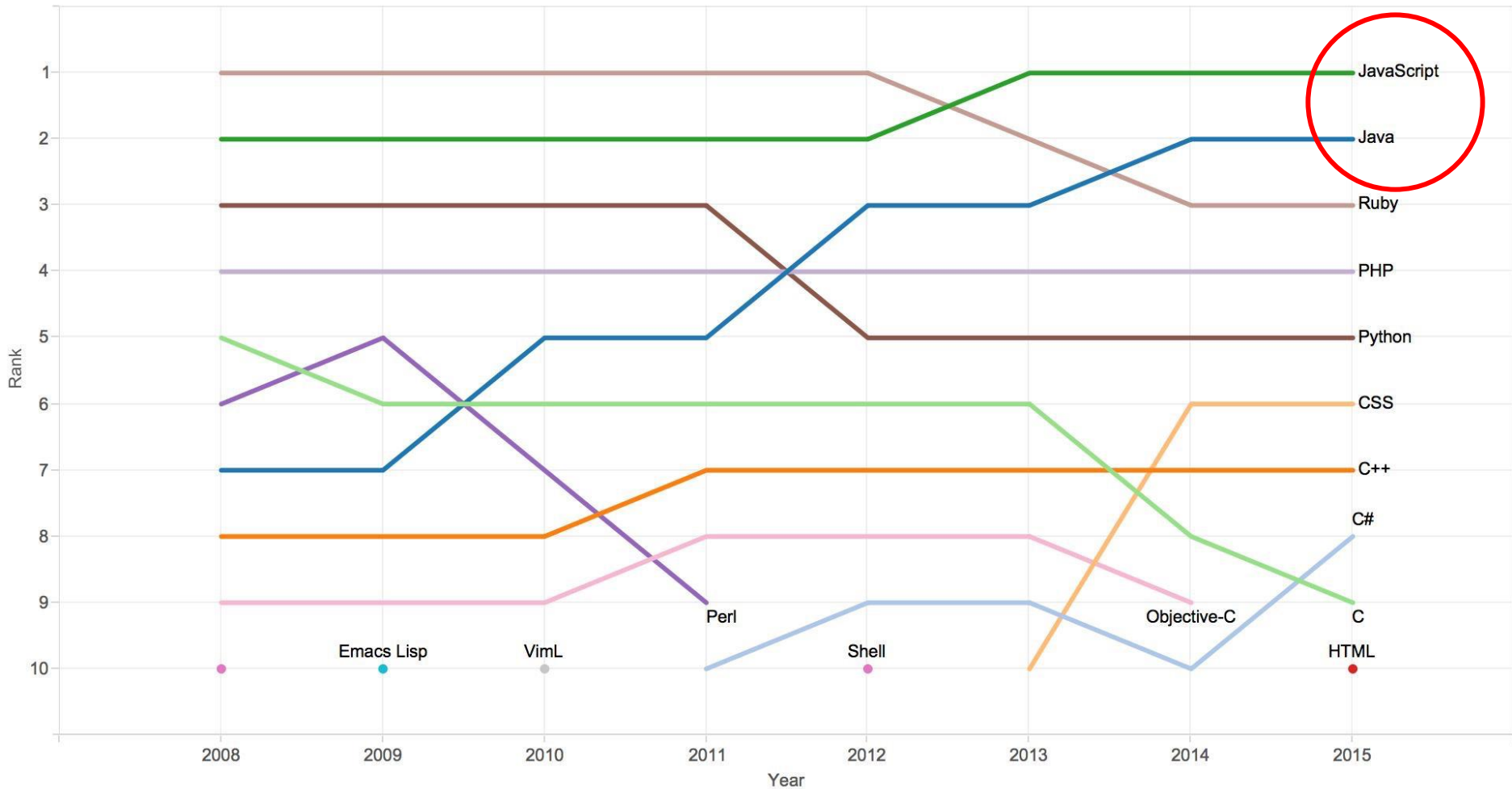
- Simple or Fast
- Clever or Readable
- Shiny New or Good Old
- Ground-Breaking or Compatible

© Andrey Breslav 2013 [[source](#)]

Java Bytecode

- Performance: patterns that understand JVM
- Target Bytecode version - 6 (Android, IntelliJ Idea)
- ABI Compatibility (Hi, Scala_2.11_1.2.3)
- ...

Rank of top languages on GitHub.com over time



Why Kotlin?

- First-class interop \w Java
- Intuitive, Easy to learn \w Java Background
- Tooling (Ide, Build Tools, Converter Java -> Kotlin)
- JavaScript Target Coming Soon!

First-class interop \w Java

- Call Java From Kotlin and vice versa
- Put Kotlin class in folder with Java class
- Seamlessly integrate Kotlin in Java code base



Evolutionary, rather than
Revolutionary change

Interop FTW!

Building DSL \w Kotlin



Function type with receiver

```
class Div {  
    var classes = ""  
    var text = ""  
}
```

```
div {  
    classes = "text pull left"  
    text = "Hello!"  
}
```

Function type with receiver

```
div {  
  classes = "text pull left"  
  text = "Hello!"  
}
```

```
fun div(body: Div.() -> Unit) {  
  val div = Div()  
  body(div)  
  // ...  
}
```

Operator overloading

```
div {  
  + CI("text")  
  + CI("pull left")  
  + "Hello!"  
}
```

```
data class CI(val name: String)
```

```
class Div {  
  var classes = listOf<CI>()  
  var text = ""  
  
  operator fun CI.unaryPlus() {  
    classes += this  
  }  
  
  operator fun String.unaryPlus() {  
    text += this  
  }  
}
```

Extension Properties

```
div {  
  + "text".cl  
  + "pull left".cl  
  + "Hello!"  
}
```

```
data class Cl(val name: String)
```

```
class Div {  
  var classes = listOf<Cl>()  
  var text = ""  
  
  val String.cl: Cl  
    get() = Cl(this)  
  // ...  
}
```

.apply {}

```
public inline fun <T> T.apply(block: T.() -> Unit): T { block(); return this }
```

```
val p = Person().apply {  
    firstName = "Baruch"  
    lastName = "Sadogursky"  
}
```

with() {}

```
public inline fun <T, R> with(receiver: T, block: T.() -> R): R = receiver.block()
```

```
val greetings = with(StringBuilder()) {  
    append("Hello")  
    append("JUG Latvia")  
    toString()  
}
```

DSL in Kotlin

- + IDE support out of the box
- + Simple
- + Limited set of operators
- Simple
- Limited set of operators

Q&A

Ruslan Ibragimov

Twitter: @HeapyHop

Belarus Kotlin User Group: <https://bkug.by/>

Awesome Kotlin: <https://kotlin.link/>