Full Stack Web with Kotlin

Ruslan Ibragimov

Disclaimer

- I’m not related to Kotlin Team or JetBrains
- I might be wrong

My languages background

Java 6+

Kotlin M8+

EcmaScript 5+

TypeScript 1.6+

Haskell

Clojure

Plan

- Kotlin
- Problems of current mainstream solutions
- Kotlin in Browser
- Future...

Is Kotlin Silver Bullet?

Trade-offs, Trade-offs Everywhere!

Kotlin Targets

Java Bytecode (Compile Target)
Android Platform (Performance, Method Count, Size)
Java (Interoperability)
JavaScript (Interoperability, Compile Target)
Native (LLVM* Compile Target -> iOS, IoT?)

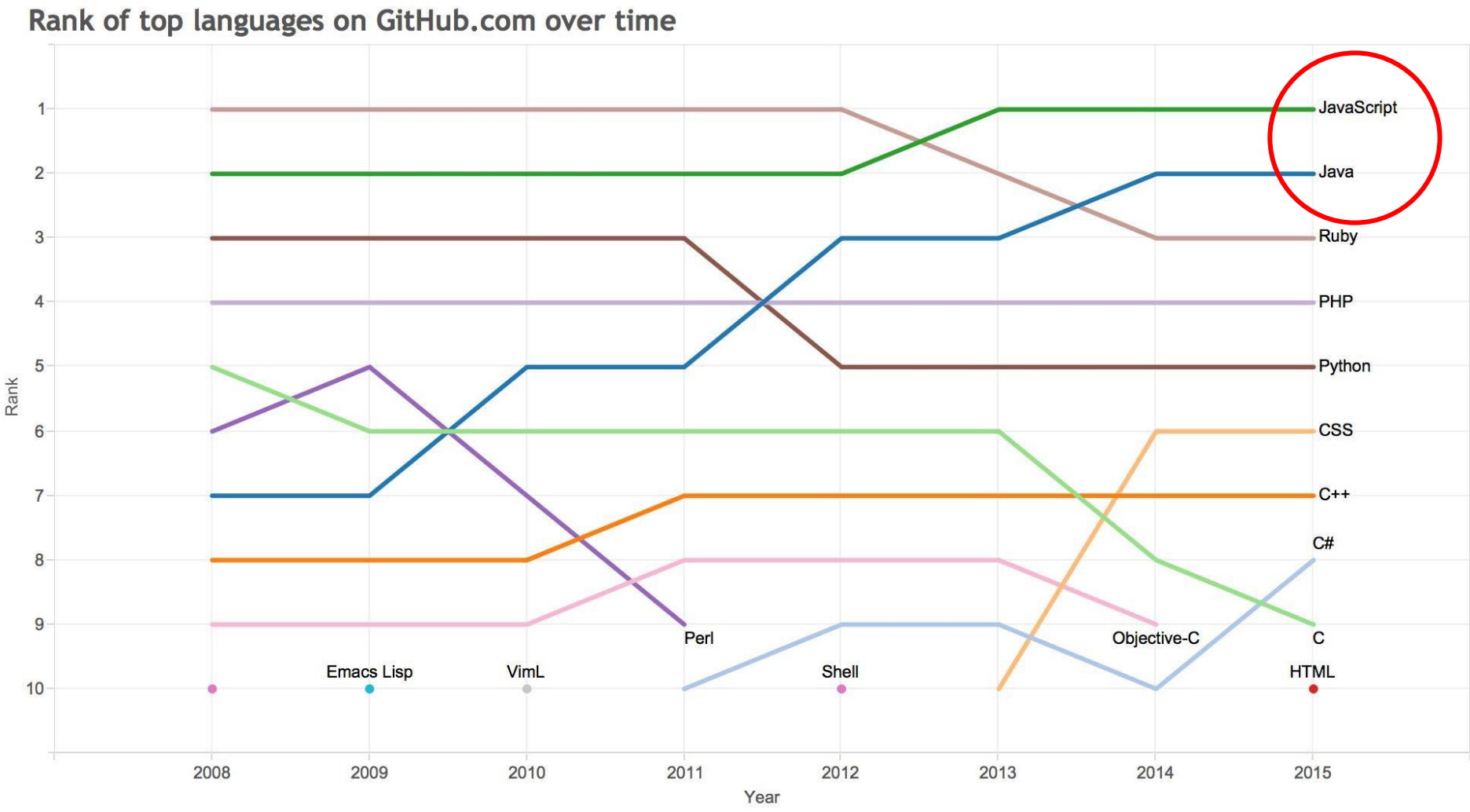
Language Design

What do you prefer?
- Simple or Fast
- Clever or Readable
- Shiny New or Good Old
- Ground-Breaking or Compatible

Java Bytecode

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

So why not “X”?


Rank of top languages on GitHub.com over time
Rank
1
2
3
4
5
6
7
8
9
10
JavaScript
Java
Ruby
PHP
Python
CSS
C++
C#
C
HTML
Objective-C
Shell
Perl
VimL
Emacs Lisp
2008
2009
2010
2011
2012
2013
2014
2015
Year
Source: GitHub.com

Java, Java, Java, JavaScript

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!

Dan Lew
@danlew42

Follow

Kotlin is a Trojan horse: You think you're getting Java++ then SURPRISE here comes an army of functional programming constructs.

RETWEETS	LIKES
43	106

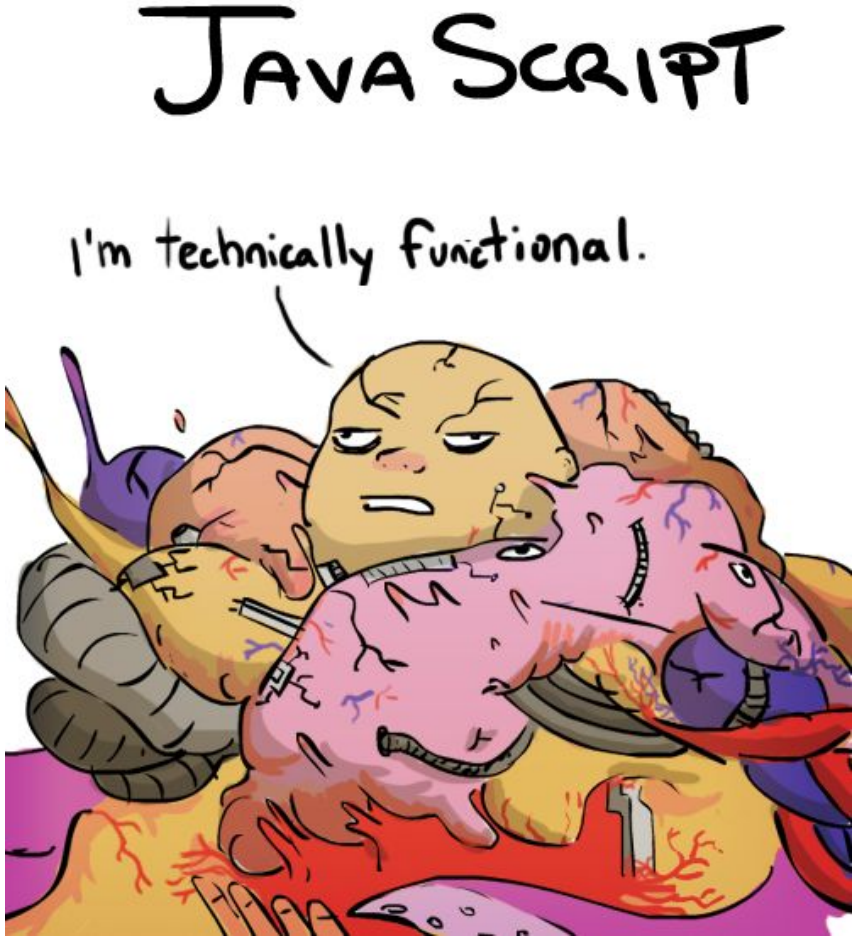
12:03 PM - 8 Dec 2016

4 43 106 •••

So… What Wrong \w Java(Script)?

Why Not?

- Doesn’t Scale (big teams)
- Refactoring?
- Error-prone
- IDE Support
- Technically Functional



© Leftover Salad [source]

Java?

Programming in Kotlin

Backend

Use existing libraries:

- Spring
- Bootique
- Vert.x
- Ratpack
- etc

Use Kotlin libraries:

- Wasabi
- kara
- Ktor
- etc

Example: Wasabi

```
var server = AppServer()

server.get("/", { response.send("Hello World!") })

server.start()
```

Frontend

KSX (Kotlin JSX)

```
const TextBlock = ({text}) => (
 <div className="text">
   {text}
 </div>
);
```

```
const Page = () => (
 <div className="page">
   <TextBlock text="Hello, World!"/>
 </div>
);
```

KSX (Kotlin JSX)

```
const TextBlock = ({text}) => (
 <div className="text">
   {text}
 </div>
);
```

```
var TextBlock = function TextBlock(_ref) {
 var text = _ref.text;
 return React.createElement(
   "div",
   { className: "text" },
   text
 );
};
```

KSX (Kotlin JSX)

```
val Text = ksx { props:TextProps ->
  div("text") {
      + props.text
  }
}
```

CSS in JS

```
const styles = {
 button: {
   fontSize: 12,
   '&:hover': {
     background: 'blue'
   }
 },
 '@media (min-width: 1024px)': {
   button: {
     width: 200
   }
 }
};
```

CSS in Kotlin (Aza-Kotlin-CSS)

```
Stylesheet {
    div {
        width = AUTO
        a {
            color = 0xffffff
            hover {
                color = 0xff0000
            }
        }
    }
}
```

Building DSL \w Kotlin

Ext.ension functions

```
fun String.hello() = "Hello, $this!"

println("Baruch".hello())
// Hello, Baruch!
```

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

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

class Div {
   var classes = listOf<Cl>()
   var text = ""

   operator fun Cl.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 greetings = StringBuilder().apply {
   append("Hello")
   append("f(by)")
}.toString()
```

with() {}

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


val greetings = with(StringBuilder()) {
   append("Hello")
   append("f(by)")
   toString()
}
```

DSL in Kotlin

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

JavaScript Interop

dynamic
@native
@JsName
noImpl

dynamic

```
val res: dynamic = eval("console.log('abc'); {a:true};")

res.a = 42
res.func {
   x -> x + 1
}
```

@native

```
@native("$")
class jquery
```

@JsName

```
class Component {
    @JsName("render")
    fun foo() {

    }
}
```

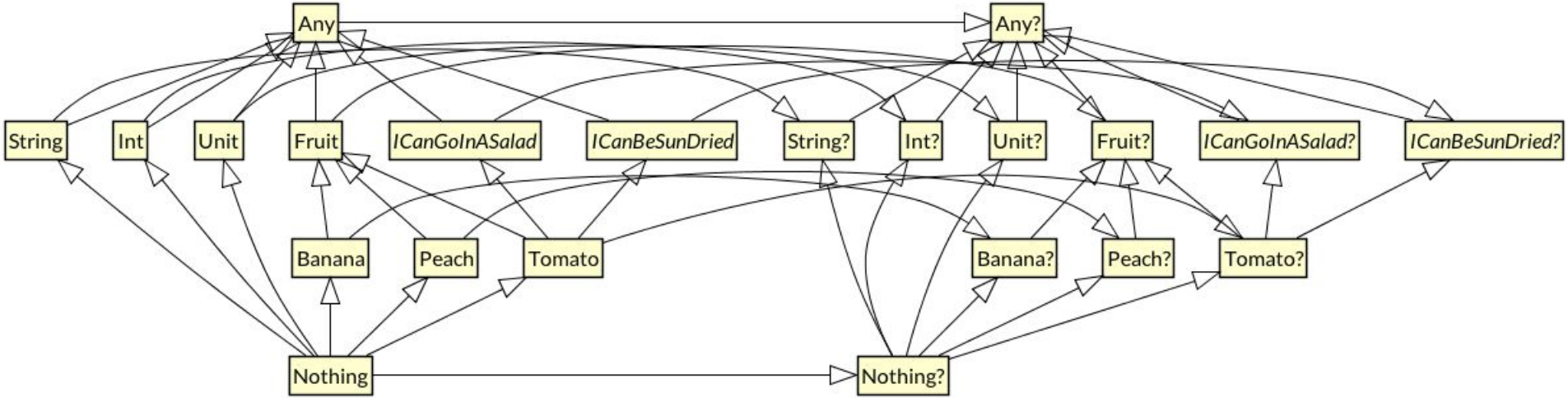
```
function Component() {
}

Component.prototype.render = function () {
};
```

noImpl

```
interface LoDash {
  fun forEach(): Unit = noImpl
}
```

```
@native
public val noImpl: Nothing
  get() = throw Exception()
```

KotΛin Type Hierarchy



Inlining Js

```
fun main(args: Array<String>) {
   js("console.log('abc')")
}
```

```
function main(args) {
 console.log('abc');
}
```

ts2kt

Converter of TypeScript definition files to Kotlin declarations

Universal Application

Universal Apps

- Possible!
- Rewrite existing framework (port preact to Kotlin)
- Use Nashorn

Universal Apps

- Define API
- JVM/JS Implementation
- Multitarget build

Problems

- Tooling (Module Bundler, Assets, Hot Reload, etc.)
- Size of the result JavaScript Bundle
- WebAssembly?
- Documentation

Questions?

Ruslan Ibragimov
Twitter: @HeapyHop
Belarus Kotlin User Group: https://bkug.by/
Awesome Kotlin: https://kotlin.link/