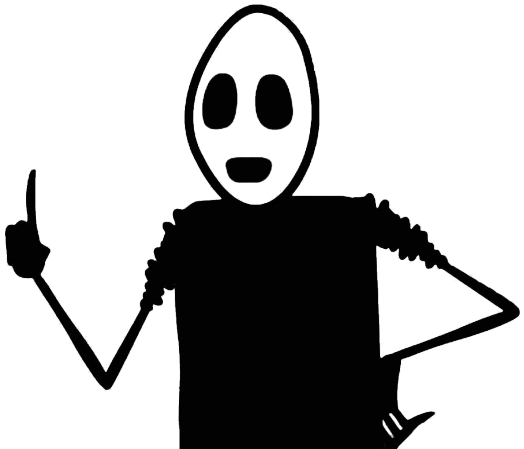
State of Kotlin Ecosystem

Ruslan Ibragimov / ibragimov.by

План

1. Проекты
    a. JVM
    b. Android
    c. JS
    d. Native
    e. Tools
2. Популярность Kotlin
3. Ресурсы для изучения



Краткий
Безопасный
Совместимый
Поддержка IDE

~~~~ Kotlin/JVM ~~~~


spring
by Pivotal™

Spring

- написали экстеншенов
- разметили API с помощью @NotNull аннотаций
- учитывают Nullable/NotNull котлин типы в MVC
- ...

https://docs.spring.io/spring/docs/current/spring-framework-reference/languages.html#kotlin

Bad Java + Kotlin

```kotlin
// Kotlin
@Component
class ЭтоБудетКомпонентСпринга
```

```java
// Java
@Component
public final class ЭтоБудетКомпонентСпринга
```

Bad Java + Kotlin

```kotlin
// Kotlin
@Component
open class ЭтоБудетКомпонентСпринга
```

```java
// Java
@Component
public class ЭтоБудетКомпонентСпринга
```

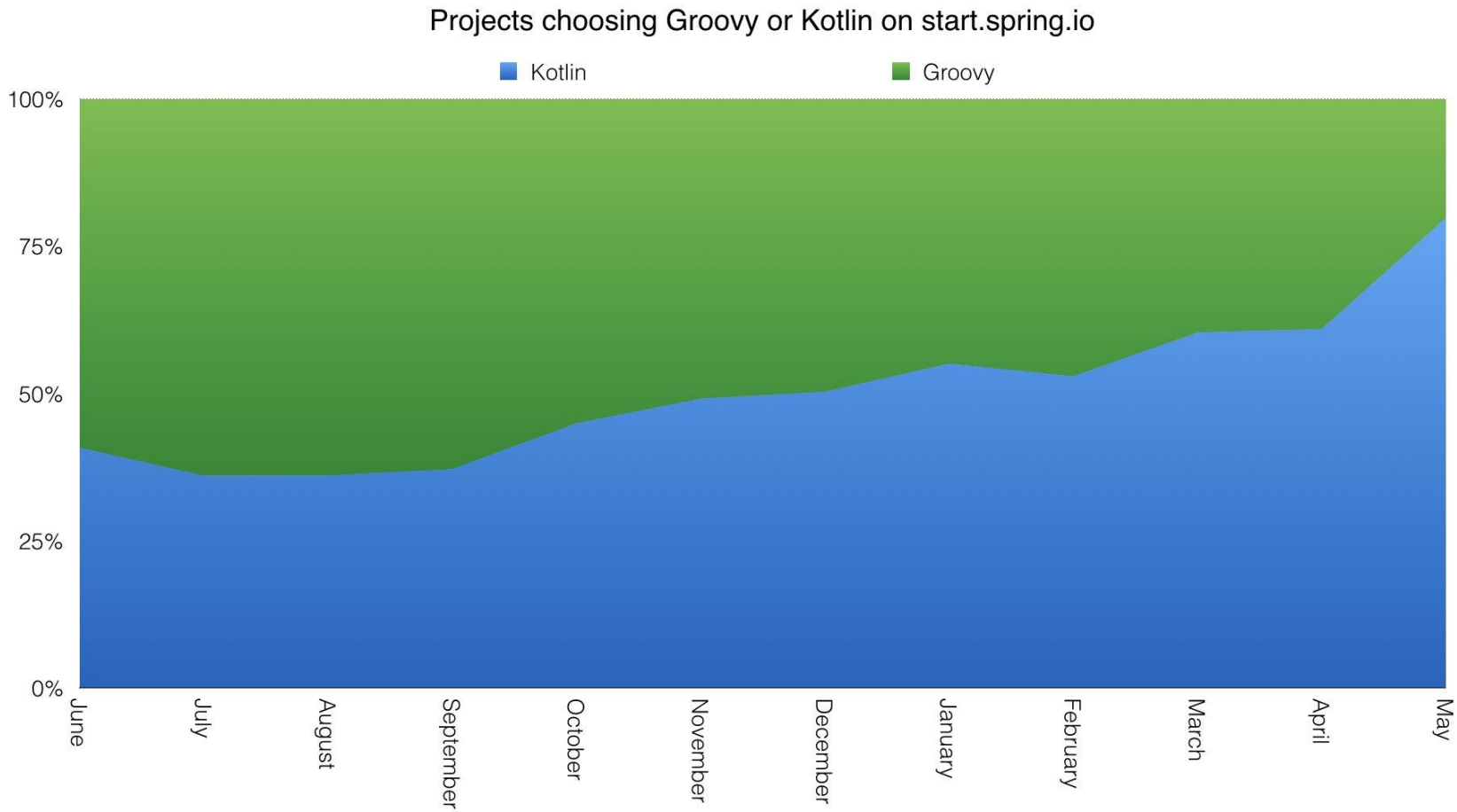
Bad Java + Kotlin

```
// Compiler Plugins:

kotlin-allopen (kotlin-spring)

koltin-noarg (kotlin-jpa)
```

Bad Java + Kotlin

```
// Gralde
apply plugin: "kotlin-allopen"

// Kotlin
@Component
class ЭтоБудетКомпонентСпринга

// Decompiled
@Component
public class ЭтоБудетКомпонентСпринга
```


spring
by Pivotal



https://twitter.com/ankinson/status/874656449665880065

Andy Wilkinson
@ankinson

Follow

Replying to @heapyhop

No new graphs, but the trend continues. Java dominates. Kotlin is gaining in popularity.

4:54 PM - 13 Mar 2018

1 1

 https://twitter.com/ankinson/status/874656449665880065

https://github.com/vert-x3/vertx-lang-kotlin
http://vertx.io/docs/vertx-lang-kotlin-coroutines/kotlin/

Vert.x

```
val select = "SELECT TITLE FROM MOVIE WHERE ID=?"
connection.queryWithParams(select, json { array(movie) }) {
   if (it.succeeded()) {
       if (it.result().rows.size == 1) {
           val insert = "INSERT INTO RATING (VALUE, MOVIE_ID) VALUES ?, ?"
           connection.updateWithParams(insert, json { array(rating, movie) }) {
               if (it.succeeded()) {
                   ctx.response().setStatusCode(200).end()
               } else {
                   ctx.response().setStatusCode(500).end()
               }
           }
       } else {
           ctx.response().setStatusCode(404).end()
       }
   } else {
       ctx.response().setStatusCode(500).end()
   }
}
```

Vert.x

```
val select = "SELECT TITLE FROM MOVIE WHERE ID=?"
connection.queryWithParams(select, json { array(movie) }) {
   if (it.succeeded()) {
       if (it.result().rows.size == 1) {
           val insert = "INSERT INTO RATING (VALUE, MOVIE_ID) VALUES ?, ?"
           connection.updateWithParams(insert, json { array(rating, movie) }) {
               if (it.succeeded()) {
                   ctx.response().setStatusCode(200).end()
               } else {
                   ctx.response().setStatusCode(500).end()
               }
           }
       } else {
           ctx.response().setStatusCode(404).end()
       }
   } else {
       ctx.response().setStatusCode(500).end()
   }
}
```

Vert.x

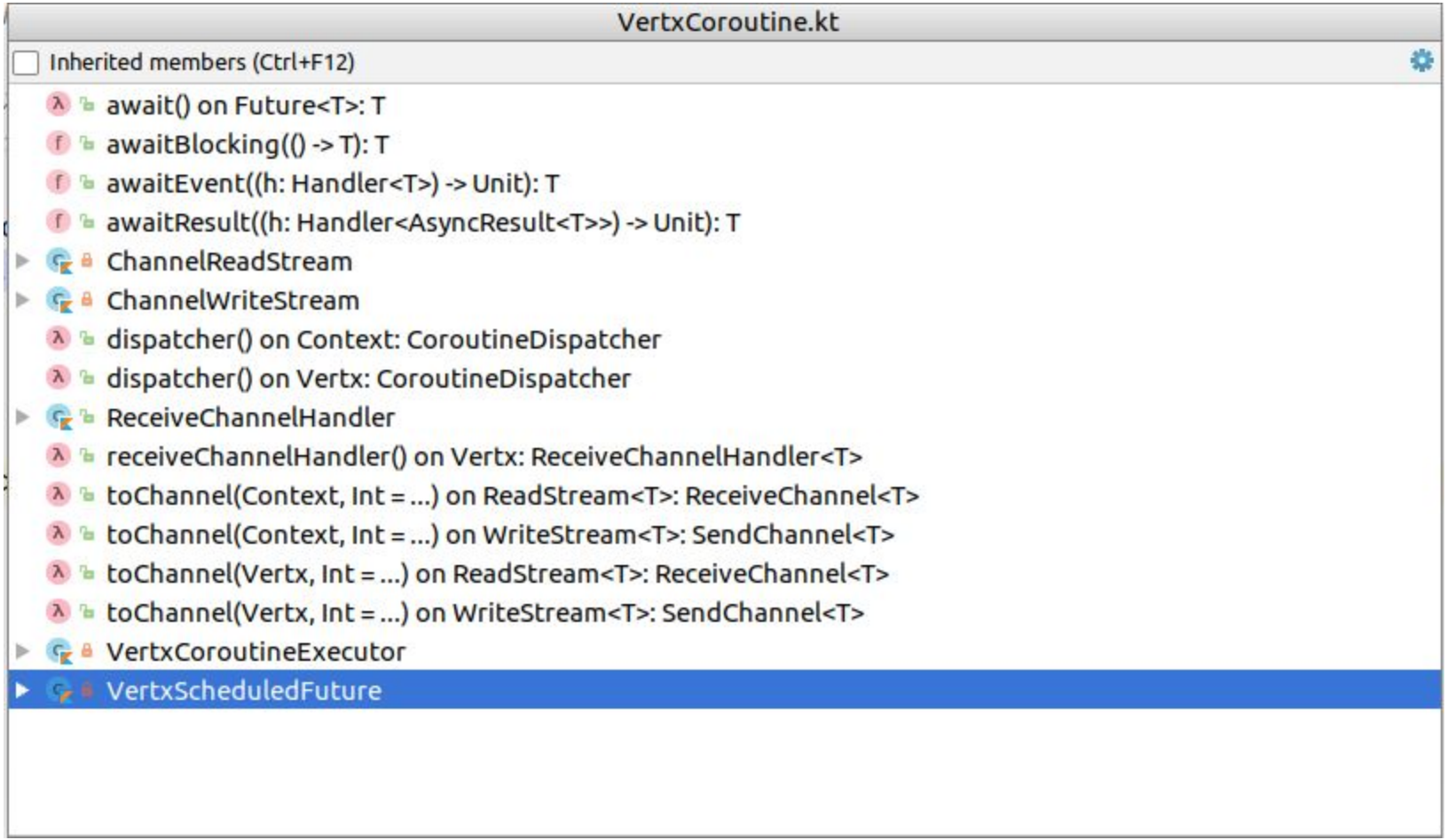
```
val select = "SELECT TITLE FROM MOVIE WHERE ID=?"
connection.queryWithParams(select, json { array(movie) }) {
   if (it.succeeded()) {
       if (it.result().rows.size == 1) {
           val insert = "INSERT INTO RATING (VALUE, MOVIE_ID) VALUES ?, ?"
           connection.updateWithParams(insert, json { array(rating, movie) }) {
               if (it.succeeded()) {
                   ctx.response().setStatusCode(200).end()
               } else {
                   ctx.response().setStatusCode(500).end()
               }
           }
       } else {
           ctx.response().setStatusCode(404).end()
       }
   } else {
       ctx.response().setStatusCode(500).end()
   }
}
```

Vert.x

```
val select = "SELECT TITLE FROM MOVIE WHERE ID=?"
connection.queryWithParams(select, json { array(movie) }) {
   if (it.succeeded()) {
       if (it.result().rows.size == 1) {
           val insert = "INSERT INTO RATING (VALUE, MOVIE_ID) VALUES ?, ?"
           connection.updateWithParams(insert, json { array(rating, movie) }) {
               if (it.succeeded()) {
                   ctx.response().setStatusCode(200).end()
               } else {
                   ctx.response().setStatusCode(500).end()
               }
           }
       } else {
           ctx.response().setStatusCode(404).end()
       }
   } else {
       ctx.response().setStatusCode(500).end()
   }
}
```

Vert.x

```
val select = "SELECT TITLE FROM MOVIE WHERE ID=?"
connection.queryWithParams(select, json { array(movie) }) {
   if (it.succeeded()) {
       if (it.result().rows.size == 1) {
           val insert = "INSERT INTO RATING (VALUE, MOVIE_ID) VALUES ?, ?"
           connection.updateWithParams(insert, json { array(rating, movie) }) {
               if (it.succeeded()) {
                   ctx.response().setStatusCode(200).end()
               } else {
                   ctx.response().setStatusCode(500).end()
               }
           }
       } else {
           ctx.response().setStatusCode(404).end()
       }
   } else {
       ctx.response().setStatusCode(500).end()
   }
}
```


VertxCoroutine.kt
Inherited members (Ctrl+F12)
await() on Future<T>: T
awaitBlocking(() -> T): T
awaitEvent((h: Handler<T>) -> Unit): T
awaitResult((h: Handler<AsyncResult<T>>) -> Unit): T
ChannelReadStream
ChannelWriteStream
dispatcher() on Context: CoroutineDispatcher
dispatcher() on Vertx: CoroutineDispatcher
ReceiveChannelHandler
receiveChannelHandler() on Vertx: ReceiveChannelHandler<T>
toChannel(Context, Int = ...) on ReadStream<T>: ReceiveChannel<T>
toChannel(Context, Int = ...) on WriteStream<T>: SendChannel<T>
toChannel(Vertx, Int = ...) on ReadStream<T>: ReceiveChannel<T>
toChannel(Vertx, Int = ...) on WriteStream<T>: SendChannel<T>
VertxCoroutineExecutor
VertxScheduledFuture

Vert.x

```
fun main(args: Array<String>) {
    val vertx = Vertx.vertx()

    launch(vertx.dispatcher()) {
        // use coroutines here
    }
}
```

Vert.x

```
fun main(args: Array<String>) {
    val vertx = Vertx.vertx()

    launch(vertx.dispatcher()) {
        // use coroutines here
    }
}
```

Vert.x

```
try {
    val select = "SELECT TITLE FROM MOVIE WHERE ID=?"
    val result = awaitResult<ResultSet> {
        connection.queryWithParams(select, json { array(movie) }, it)
    }

    if (result.rows.size == 1) {
        awaitResult<UpdateResult> {
            val insert = "INSERT INTO RATING (VALUE, MOVIE_ID) VALUES ?, ?"
            connection.updateWithParams(insert, json { array(rating, movie) }, it)
        }
        ctx.response().setStatusCode(200).end()
    } else {
        ctx.response().setStatusCode(404).end()
    }
} catch (e: Exception) {
    ctx.response().setStatusCode(500).end()
}
```

Vert.x

```
try {
    val select = "SELECT TITLE FROM MOVIE WHERE ID=?"
    val result = awaitResult<ResultSet> {
        connection.queryWithParams(select, json { array(movie) }, it)
    }

    if (result.rows.size == 1) {
        awaitResult<UpdateResult> {
            val insert = "INSERT INTO RATING (VALUE, MOVIE_ID) VALUES ?, ?"
            connection.updateWithParams(insert, json { array(rating, movie) }, it)
        }
        ctx.response().setStatusCode(200).end()
    } else {
        ctx.response().setStatusCode(404).end()
    }
} catch (e: Exception) {
    ctx.response().setStatusCode(500).end()
}
```

Vert.x

```
try {
    val select = "SELECT TITLE FROM MOVIE WHERE ID=?"
    val result = awaitResult<ResultSet> {
        connection.queryWithParams(select, json { array(movie) }, it)
    }

    if (result.rows.size == 1) {
        awaitResult<UpdateResult> {
            val insert = "INSERT INTO RATING (VALUE, MOVIE_ID) VALUES ?, ?"
            connection.updateWithParams(insert, json { array(rating, movie) }, it)
        }
        ctx.response().setStatusCode(200).end()
    } else {
        ctx.response().setStatusCode(404).end()
    }
} catch (e: Exception) {
    ctx.response().setStatusCode(500).end()
}
```

Vert.x

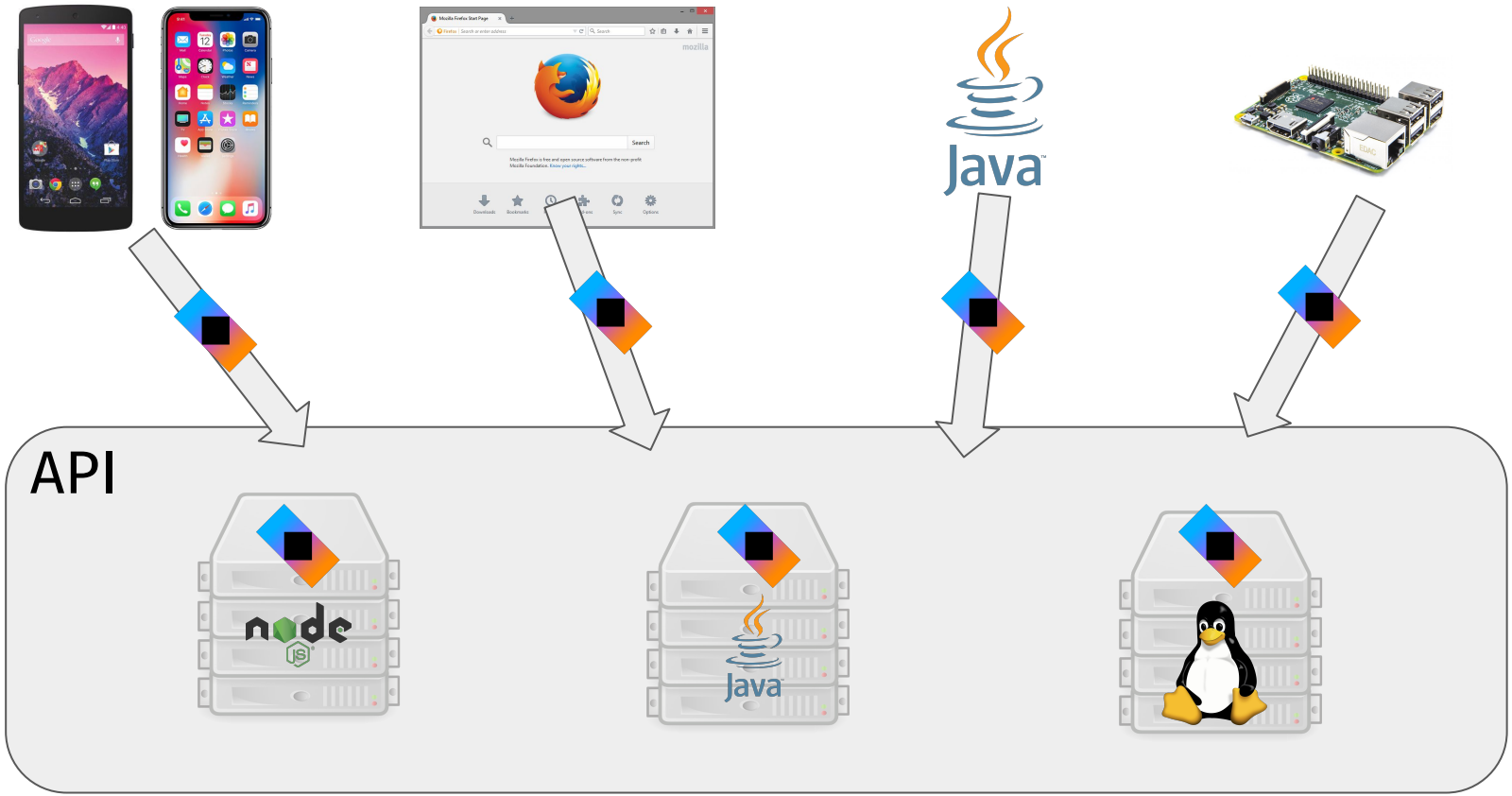
```
try {
    val select = "SELECT TITLE FROM MOVIE WHERE ID=?"
    val result = awaitResult<ResultSet> {
        connection.queryWithParams(select, json { array(movie) }, it)
    }

    if (result.rows.size == 1) {
        awaitResult<UpdateResult> {
            val insert = "INSERT INTO RATING (VALUE, MOVIE_ID) VALUES ?, ?"
            connection.updateWithParams(insert, json { array(rating, movie) }, it)
        }
        ctx.response().setStatusCode(200).end()
    } else {
        ctx.response().setStatusCode(404).end()
    }
} catch (e: Exception) {
    ctx.response().setStatusCode(500).end()
}
```


VERY NICE!!!

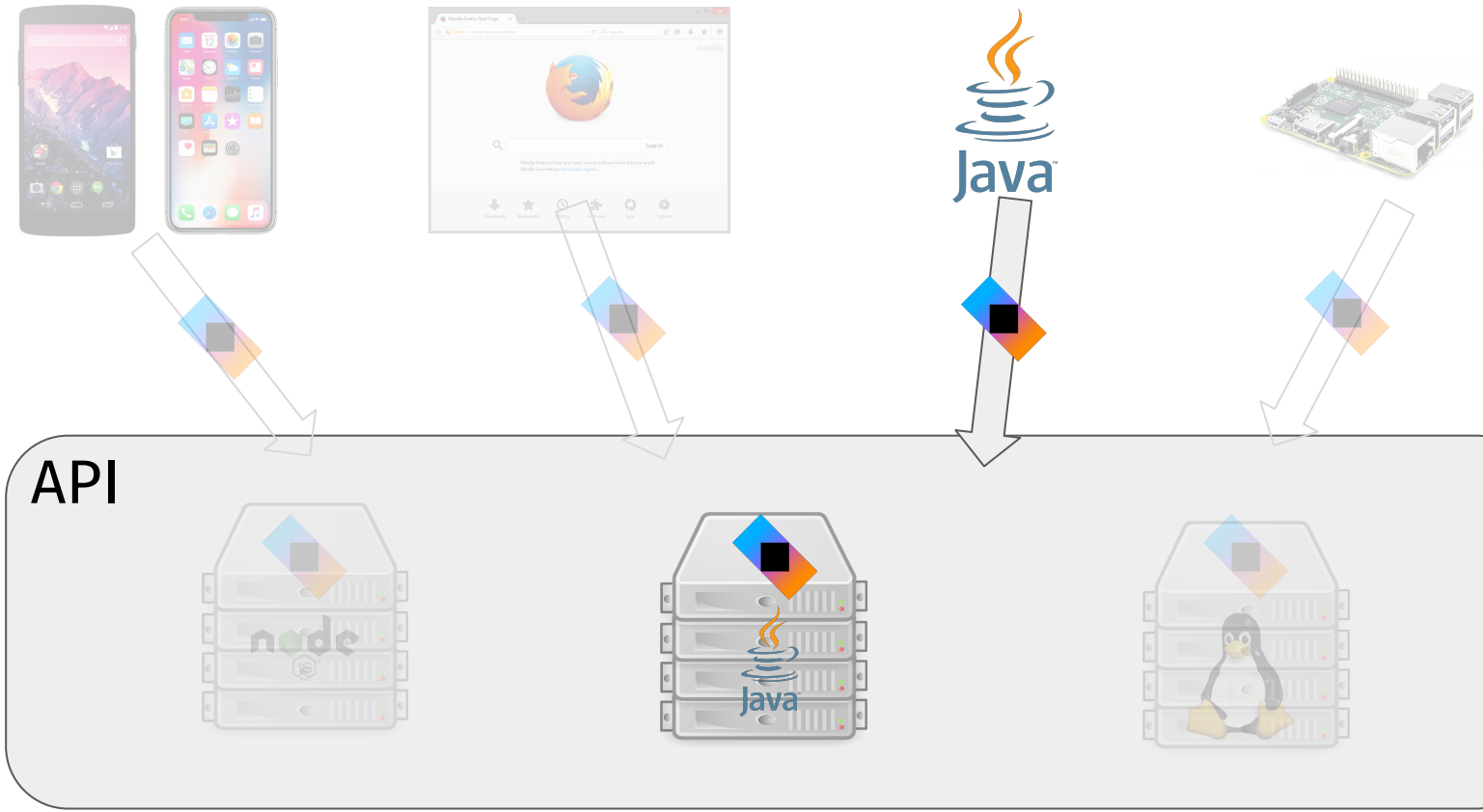
Ktor

★ 2400

https://ktor.io/



Мультиплатформа



Java, Java, Java-Java Jing-Jing-Jing

https://bkug.by/2018/02/23/otchet-o-bkug-8/

jackson-module-kotlin

400

jackson-module-kotlin

```
data class KotlinNight(
    val venue: String,
    val organizers: List<Organizer>,
    val time: ZonedDateTime
)
```

jackson-module-kotlin

```
data class KotlinNight(
    val venue: String,
    val organizers: List<Organizer>,
    val time: ZonedDateTime
)

fun main(args: Array<String>) {
    ObjectMapper().readValue(json, KotlinNight::class.java)
}
```

jackson-module-kotlin

```
Exception in thread "main"
com.fasterxml.jackson.databind.exc.InvalidDefinitionException:
Cannot construct instance of `ua.knight.KotlinNight` (no
Creators, like default construct, exist): cannot deserialize
from Object value (no delegate- or property-based Creator)
```

jackson-module-kotlin

```
data class KotlinNight(
    val venue: String,
    val organizers: List<Organizer>,
    val time: ZonedDateTime
)

fun main(args: Array<String>) {
    ObjectMapper()
        .registerKotlinModule()
        .readValue(json, KotlinNight::class.java)
}
```

kotlinx.serialization

480

klaxon

660

Kodein/Injekt

 1000

Kodein/Injekt

```
val kodein = Kodein {
    bind<Dice>() with provider { RandomDice(0, 5) }
    bind<DataSource>() with singleton { SqliteDS.open("file") }
}

class Controller(
    private val kodein: Kodein
) {
    private val ds: DataSource = kodein.instance()
}
```

kategory/funKtionale

kategory + funKtionale = Λrrow

 1100

http://arrow-kt.io/

Λrrow

- тайпклассы (как бы)
- фп типы: Option, Try, Either
- интеграции: Rx2, Coroutines

ReactiveX/RxKotlin

3900

Exposed

1200

Exposed

```
object Users : Table() {
    val id = varchar("id", 10).primaryKey()
    val name = varchar("name", length = 50)
    val cityId = (integer("city_id") references Cities.id).nullable()
}
```

Exposed

```
Database.connect("jdbc:h2:mem:test", driver =
"org.h2.Driver")

transaction {
    create(Users)

    Users.insert {
        it[id] = "sergey"
        it[name] = "Sergey"
        it[cityId] = munichId
    }
}
```

Spek

 1100

Spec

```
class SimpleTest : Spek({
  describe("a calculator") {
    val calculator = SampleCalculator()

    it("should return the result of adding the first number to the second number") {
      val sum = calculator.sum(2, 4)
      assertEquals(6, sum)
    }

    it("should return the result of subtracting the second number from the first number") {
      val subtract = calculator.subtract(4, 2)
      assertEquals(2, subtract)
    }
  }
})
```

KotlinTest

700

JUnit 5.1

JUnit 5.1

```
@Test fun test() {
    assertAll(
        Executable { assertEquals(20, testFun(10)) },
        Executable { assertEquals(0, testFun(0)) }
    )
}
```

JUnit 5.1

```
@Test fun test() {
    assertAll(
        Executable { assertEquals(20, testFun(10)) },
        Executable { assertEquals(0, testFun(0)) }
    )
}
```

```
@Test fun test() {
    assertAll(
        { assertEquals(20, testFun(10)) },
        { assertEquals(0, testFun(0)) }
    )
}
```

mockito-kotlin

1000

mockito-kotlin

```
open class Foo {
    fun bar(s: String) {}
}

@Test
fun myTest() {
    val foo = Foo()
    whenever(foo.bar(any()))
}
```

mockito-kotlin

```
open class Foo {
    fun bar(s: String) {}
}

@Test
fun myTest() {
    val foo = Foo()
    whenever(foo.bar(any()))
}
```

MockK

400

TornadoFX

1400

TornadoFx

```
class FormView : View("My View") {
    override val root = vbox {
        label("Enter your login")
        form {
            fieldset {
                field("Username") {
                    textfield()
                }
                field("Password") {
                    passwordfield()
                }
            }
        }
        button("Go!") {
            addClass(Styles.redStyle)
        }
    }
}
```

TornadoFx

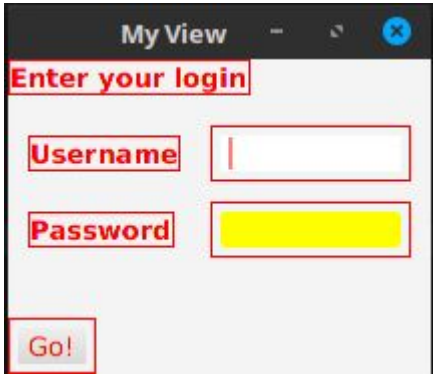
```
class FormView : View("My View") {
    override val root = vbox {
        label("Enter your login")
        form {
            fieldset {
                field("Username") {
                    textfield()
                }
                field("Password") {
                    passwordfield()
                }
            }
        }
        button("Go!") {
            addClass(Styles.redStyle)
        }
    }
}
```

```
class MyApp : App(FormView::class, Styles::class)

fun main(args: Array<String>) {
    Application.launch(MyApp::class.java, *args)
}
```

TornadoFx

```
class FormView : View("My View") {
    override val root = vbox {
        label("Enter your login")
        form {
            fieldset {
                field("Username") {
                    textfield()
                }
                field("Password") {
                    passwordfield()
                }
            }
        }
        button("Go!") {
            addClass(Styles.redStyle)
        }
    }
}
```

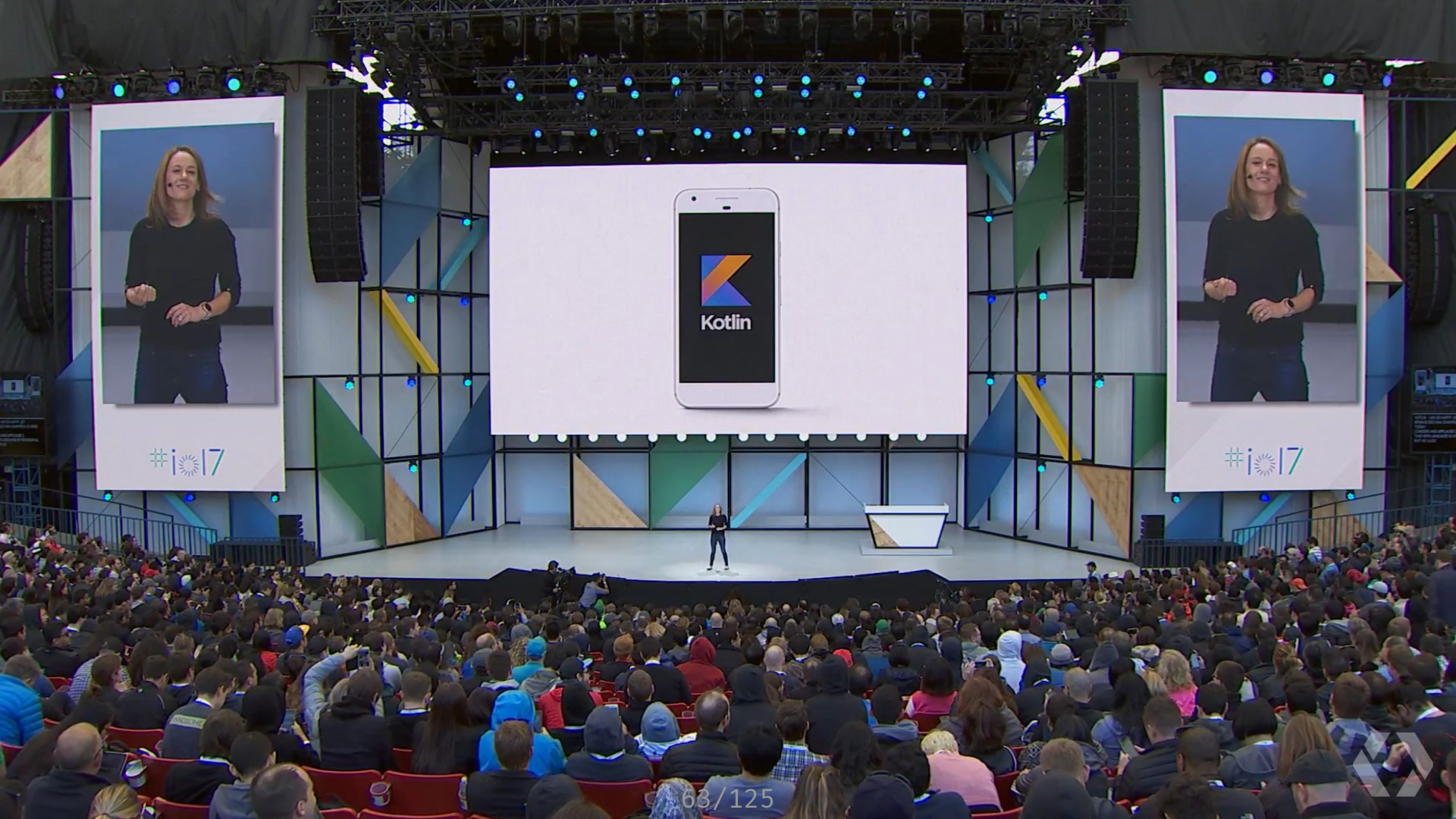
```
class MyApp : App(FormView::class, Styles::class)

fun main(args: Array<String>) {
    Application.launch(MyApp::class.java, *args)
}
```



Kotlin/JVM

- новые библиотеки
- DSL для всего что можно
- адаптеры для Java библиотек

~~~~ Kotlin/Android ~~~~


#io17
Kotlin
#io17


IN ALL
NOTHER
ICK

Anko

★ 10000

Anko Layouts

```
verticalLayout {
    val name = editText()
    button("Say Hello") {
        onClick { toast("Hello, ${name.text}!") }
    }
}
```

android/ktx

 5200

kotterknife

★ 1900

Koin

900

Kotlin/Android

- новые библиотеки
- DSL для всего что можно
- адаптеры для Android SDK

~~~~ Kotlin/JS ~~~~

create-react-kotlin-app

 1300

kotlinx.html

345

kotlinx.html

```
html {
    body {
        div {
            a("http://kotlinlang.org") {
                target = ATarget.blank
                +"Main site"
            }
        }
    }
}
```

kotlinx.html

Плюсы:

- это код - статическая типизация
- это код - не нужно учить язык шаблонизатора
- это код - можно писать код, легче переиспользовать

kotlinx.html

Плюсы:

- это код - статическая типизация
- это код - не нужно учить язык шаблонизатора
- это код - можно писать код, легче переиспользовать

Минусы:

- это код - сложнее верстальщикам
- это код - сложнее сделать внешним ресурсом

Aza-Kotlin-CSS

50

Aza-Kotlin-CSS

```
val css = Stylesheet {
    a {
        width = 10.px
        color = 0xffffff
        opacity = .8

        hover {
            color = 0xf2cacf
        }
    }
}

css.render()
```

Aza-Kotlin-CSS

Плюсы:

- это код - статическая типизация
- это код - не нужно учить язык шаблонизатора (sass, less)
- это код - можно писать код, легче переиспользовать
- это код - инспекции : неиспользуемая переменная

Минусы:

- это код - сложнее верстальщикам
- это код - сложнее сделать внешним ресурсом

kotlinx.html - **JSX**
Aza-Kotlin-CSS - **CSS in JS**

ts2kt

200

kotlin-wrappers

140

kotlin-fullstack-sample

700

Kotlin/JS

- DSL для всего что можно
- адаптеры для JavaScript

~~~~ Kotlin/Native ~~~~
~~~~

JetBrains/kotlinconf-app

1300

Kotlin/Native

https://bkug.by/2018/02/23/otchet-o-bkug-8
https://soundcloud.com/podlodka/podlodka-50-kotlin-i-swift

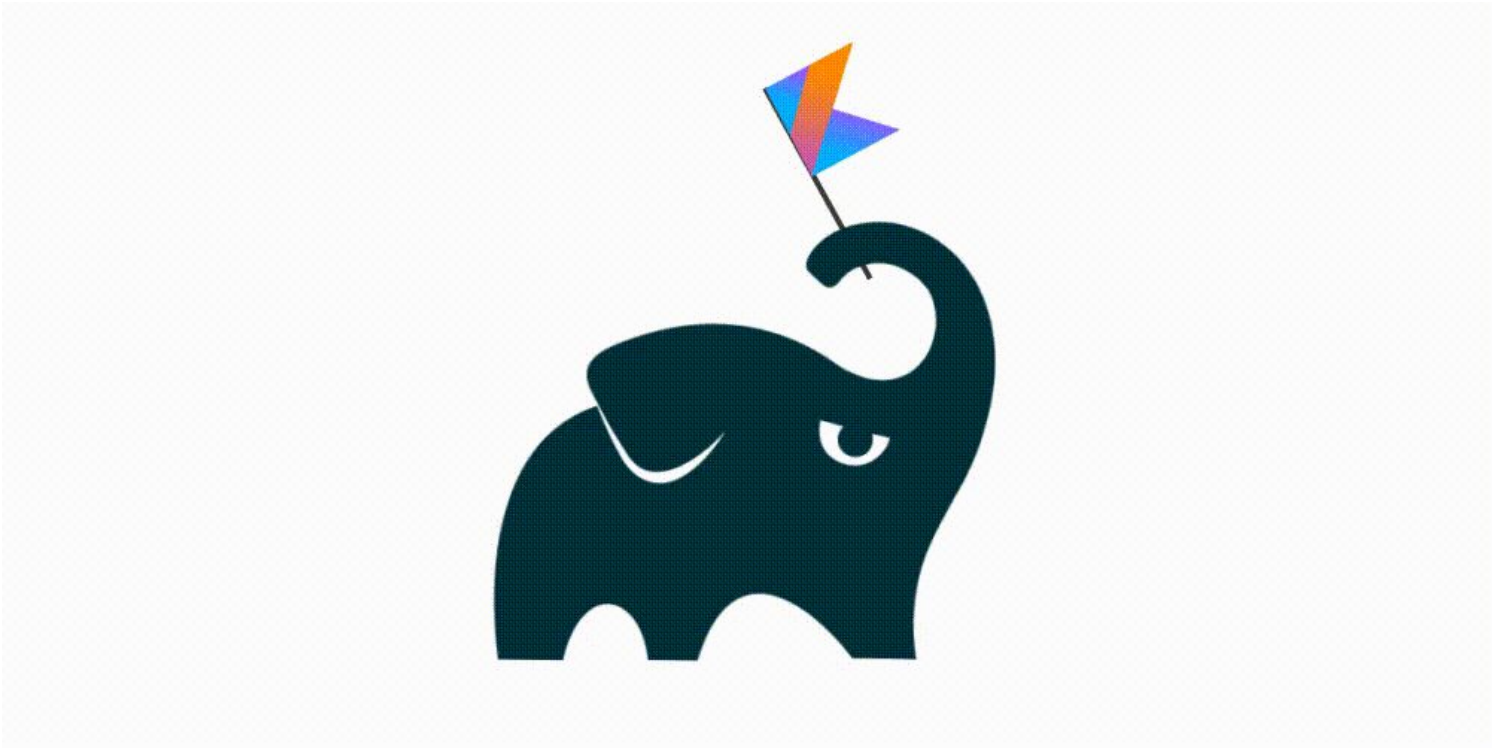
~~~~ Tools ~~~~

kscript

360

Kobalt

340



Gradle Kotlin DSL

1350

ktlint

800

detekt

 900

Fabrice Bellingard
@bellingard

Follow

Replying to @benjohnde

On our side, built-in support for Kotlin in @SonarQube & @sonarcloud is planned for this year.

11:06 PM - 20 Feb 2018

22 Retweets **41** Likes

1 22 41

dokka

★ 550

Резюме

- Kotlin/JVM
- Kotlin/Android
- Kotlin/JS
- Kotlin/Native
- Tools

Рейтинги

Февраль 2016 - 168

Сентябрь 2016 - 99

Июнь 2017 - 43

Март 2018 - 38

https://www.tiobe.com/tiobe-index/

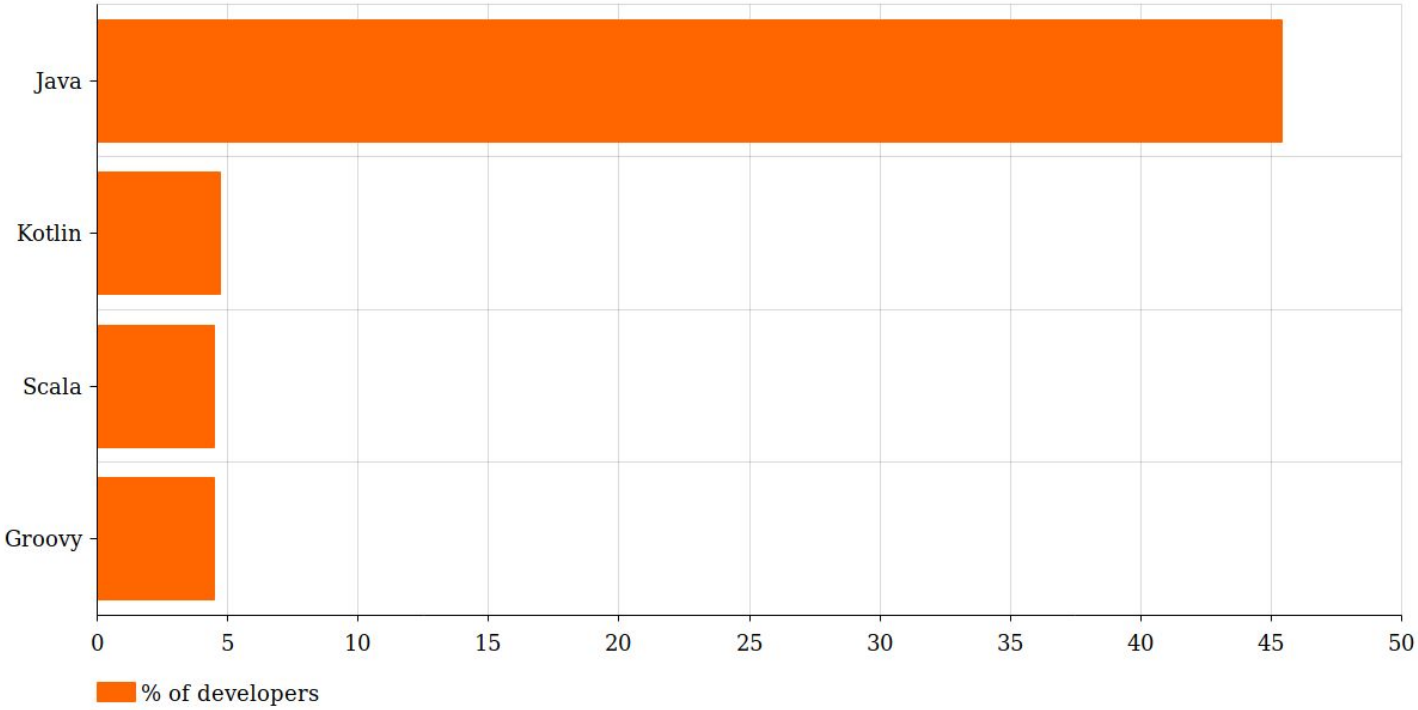
PYPL

Rank	Change	Language	Share	Trend
1		Java	22.7 %	-0.8 %
2		Python	21.69 %	+5.4 %
3	↑↑	Javascript	8.53 %	+0.3 %
4	↓	PHP	8.33 %	-1.7 %
5	↓	C#	7.99 %	-0.7 %
6		C	6.42 %	-1.3 %
7	↑	R	4.23 %	+0.4 %
8	↓	Objective-C	3.81 %	-1.1 %
9		Swift	3.0 %	-0.6 %
10		Matlab	2.39 %	-0.4 %
11		Ruby	1.72 %	-0.4 %
12	↑↑↑	TypeScript	1.52 %	+0.4 %
13		VBA	1.42 %	+0.0 %
14	↓↓	Visual Basic	1.26 %	-0.3 %
15	↓	Scala	1.21 %	-0.0 %
16	↑↑↑↑↑↑	Kotlin	0.9 %	+0.8 %

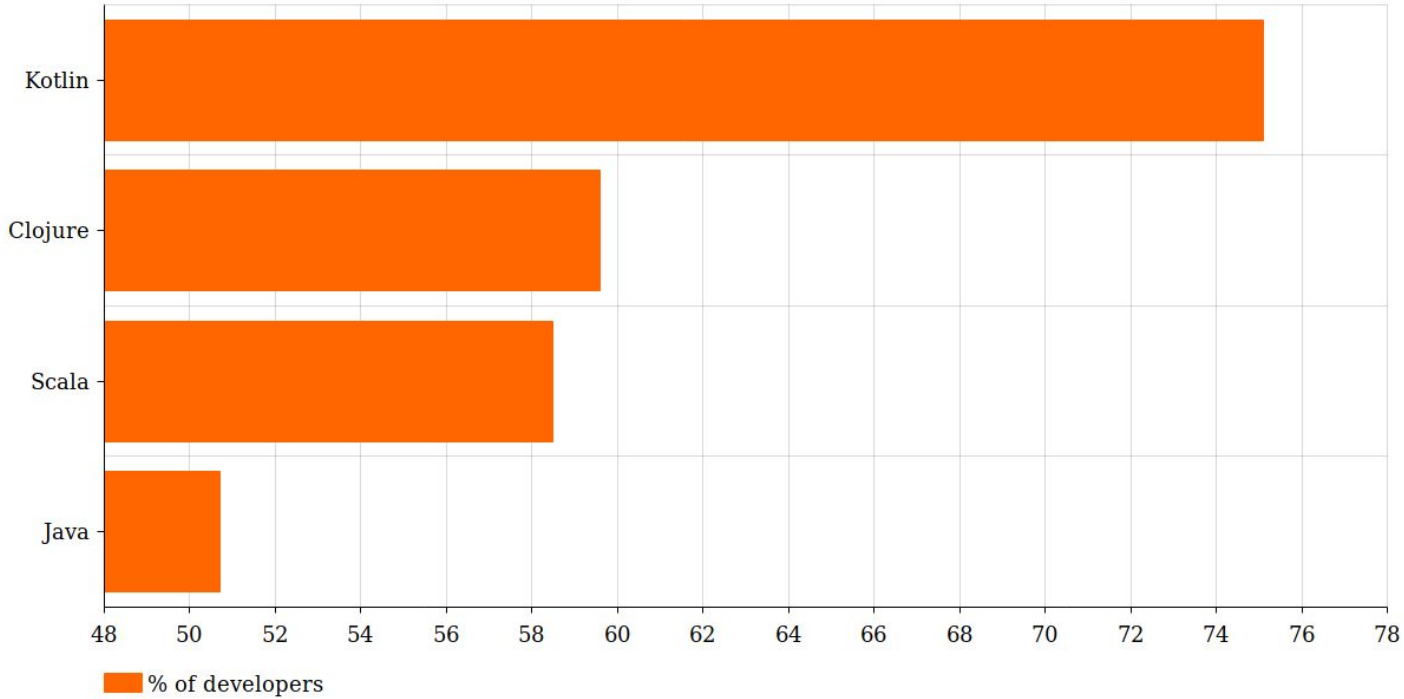
https://pypl.github.io/PYPL.html

https://insights.stackoverflow.com/survey/2018

stackoverflow



stackoverflow

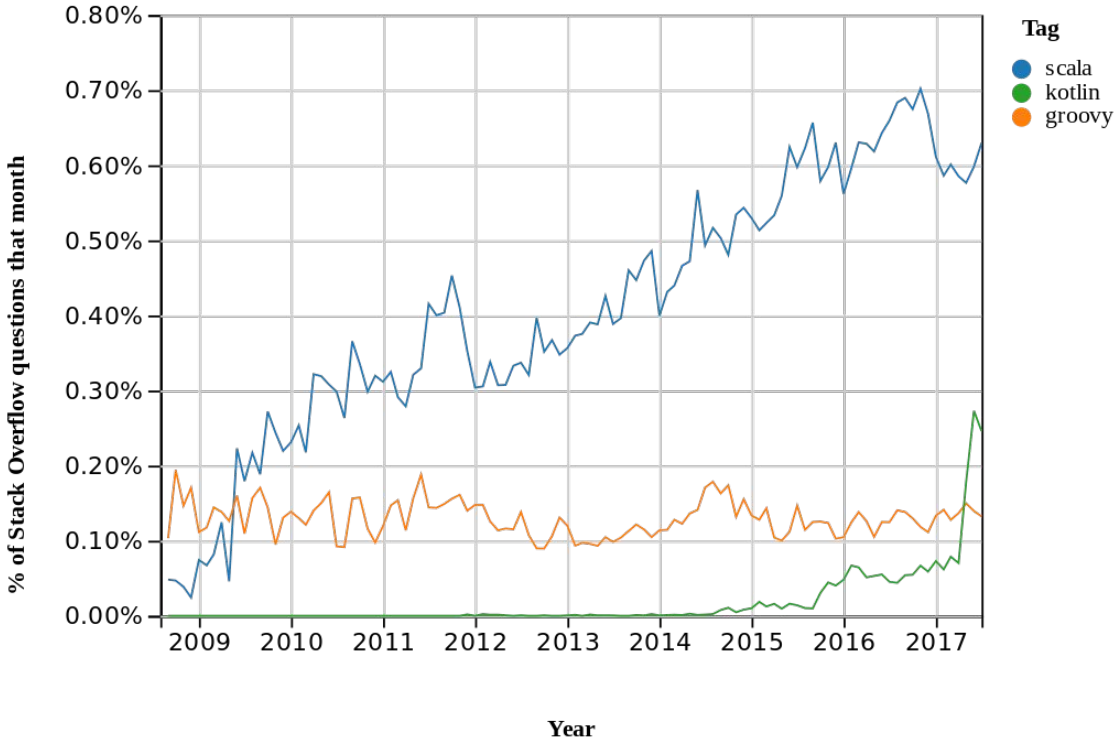


stackoverflow

½ - Java

⅔ - Groovy

stackoverflow



https://insights.stackoverflow.com/trends?tags=kotlin%2Cgroovy%2Cscala

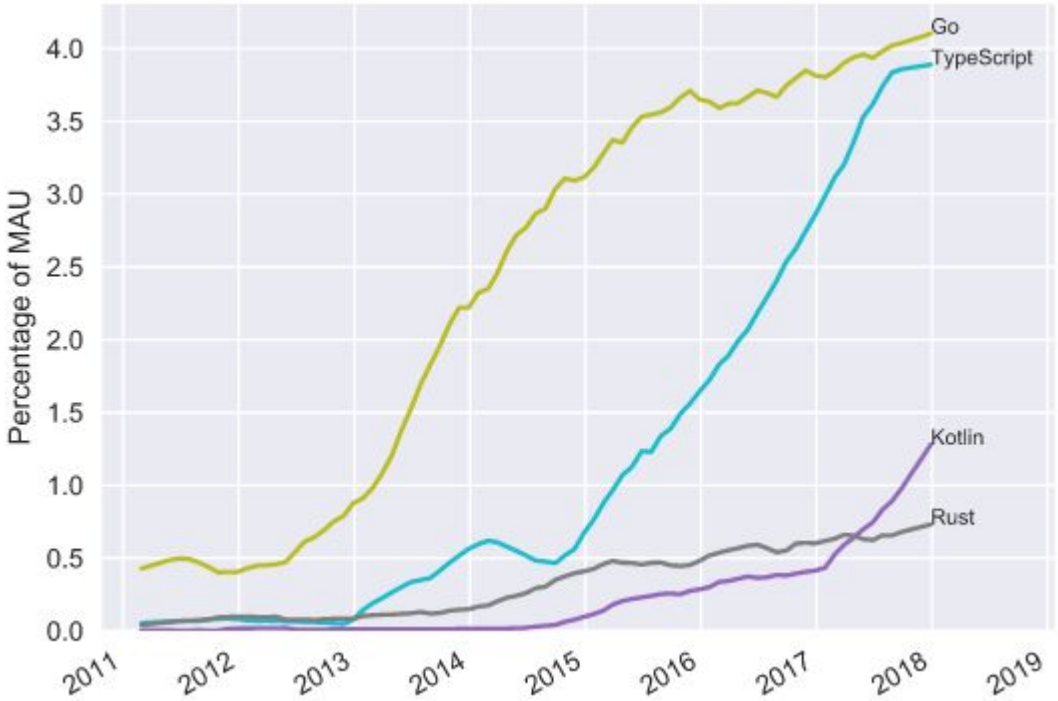
stackoverflow

stackoverflow

GitHub

http://www.benfrederickson.com/ranking-programming-languages-by-github-users/

GitHub


Percentage of MAU
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
2011
2012
2013
2014
2015
2016
2017
2018
2019
Go
TypeScript
Kotlin
Rust

GitHub

Java - 14%

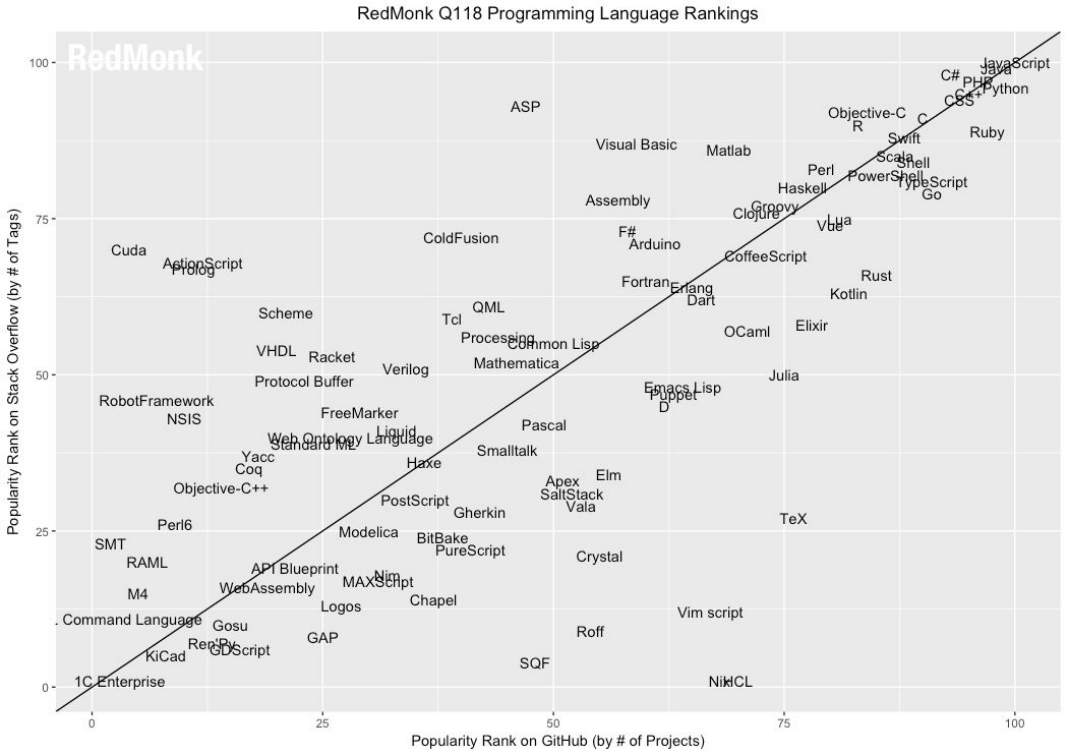
Kotlin - 1.28%

Scala - 0.78%

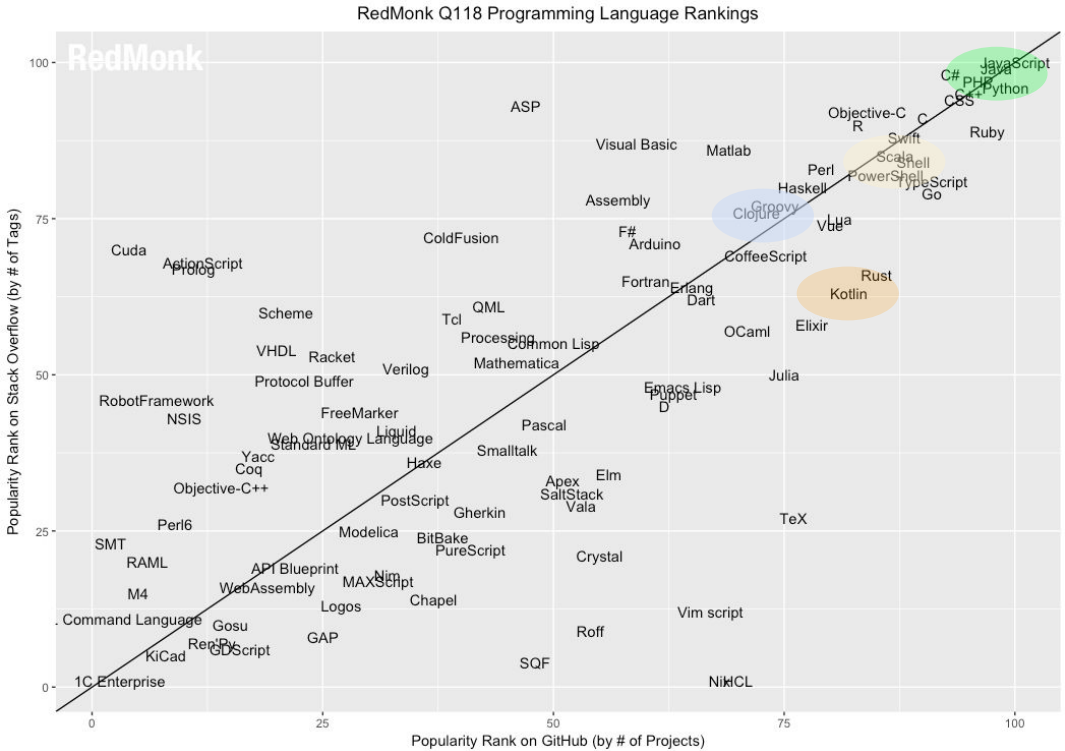
Groovy - 0.4%

https://redmonk.com/sogrady/2018/03/07/language-rankings-1-18/

RedMonk


RedMonk Q118 Programming Language Rankings
RedMonk
Popularity Rank on Stack Overflow (by # of Tags)
Popularity Rank on GitHub (by # of Projects)
0
25
50
75
100
JavaScript
Java
C#
PHP
Python
C++
CSS
Objective-C
R
C
Ruby
Swift
Scala
Shell
Perl
PowerShell
TypeScript
Go
Haskell
Groovy
Clojure
Lua
Vue
ASP
Visual Basic
Matlab
Assembly
F#
Arduino
CoffeeScript
Rust
Fortran
Erlang
Dart
Kotlin
ColdFusion
Cuda
ActionScript
Prolog
Scheme
QML
Tcl
Processing
Common Lisp
OCaml
Elixir
VHDL
Racket
Verilog
Mathematica
Julia
Protocol Buffer
Emacs Lisp
Puppet
D
RobotFramework
NSIS
FreeMarker
Liquid
Pascal
Web Ontology Language
Standard ML
Smalltalk
Yacc
Coq
Haxe
Apex
Elm
SaltStack
Vala
Objective-C++
PostScript
Gherkin
TeX
Perl6
Modelica
BitBake
SMT
PureScript
Crystal
RAML
API Blueprint
Nim
MAXScript
WebAssembly
M4
Chapel
Logos
Vim script
Command Language
Gosu
Roff
GAP
KiCad
GDScript
SQF
1C Enterprise
NikiCL

RedMonk


RedMonk Q118 Programming Language Rankings
RedMonk
Popularity Rank on Stack Overflow (by # of Tags)
Popularity Rank on GitHub (by # of Projects)
0
25
50
75
100
JavaScript
Java
C#
PHP
Python
CSS
C++
Objective-C
R
C
Ruby
Swift
Scala
Shell
PowerShell
Perl
TypeScript
Haskell
Go
Assembly
Groovy
Clojure
Lua
Vue
F#
Arduino
ColdFusion
Cuda
ActionScript
Prolog
CoffeeScript
Fortran
Erlang
Dart
Rust
Kotlin
Scheme
Tcl
QML
Processing
Common Lisp
OCaml
Elixir
VHDL
Racket
Verilog
Mathematica
Julia
Protocol Buffer
Emacs Lisp
Puppet
D
RobotFramework
NSIS
FreeMarker
Liquid
Pascal
Web Ontology Language
Standard ML
Smalltalk
Yacc
Coq
Haxe
Apex
Elm
SaltStack
Vala
Objective-C++
PostScript
Gherkin
TeX
Perl6
Modelica
BitBake
SMT
PureScript
Crystal
RAML
API Blueprint
Nim
MAXScript
WebAssembly
M4
Chapel
Logos
Vim script
Command Language
Gosu
Roff
GAP
KiCad
GDScript
SQF
1C Enterprise
NikiCL

Сообщество


Greenland
Finland
Iceland
Russia
Canada
Kazakhstan
Mongolia
North Atlantic Ocean
Algeria
Libya
Saudi Arabia
Iraq
Iran
Afghanistan
Pakistan
Mali
Niger
Chad
Sudan
Ethiopia
Venezuela
Colombia
DRC
Tanzania
Brazil
Angola
Namibia
Madagascar
Indian Ocean
South Pacific Ocean
South Atlantic Ocean
Chile
South Africa
Australia
Papua New Guinea
Argentina

145 Kotlin User Groups

https://kotlinconf.com/

https://kotlinslack.herokuapp.com/

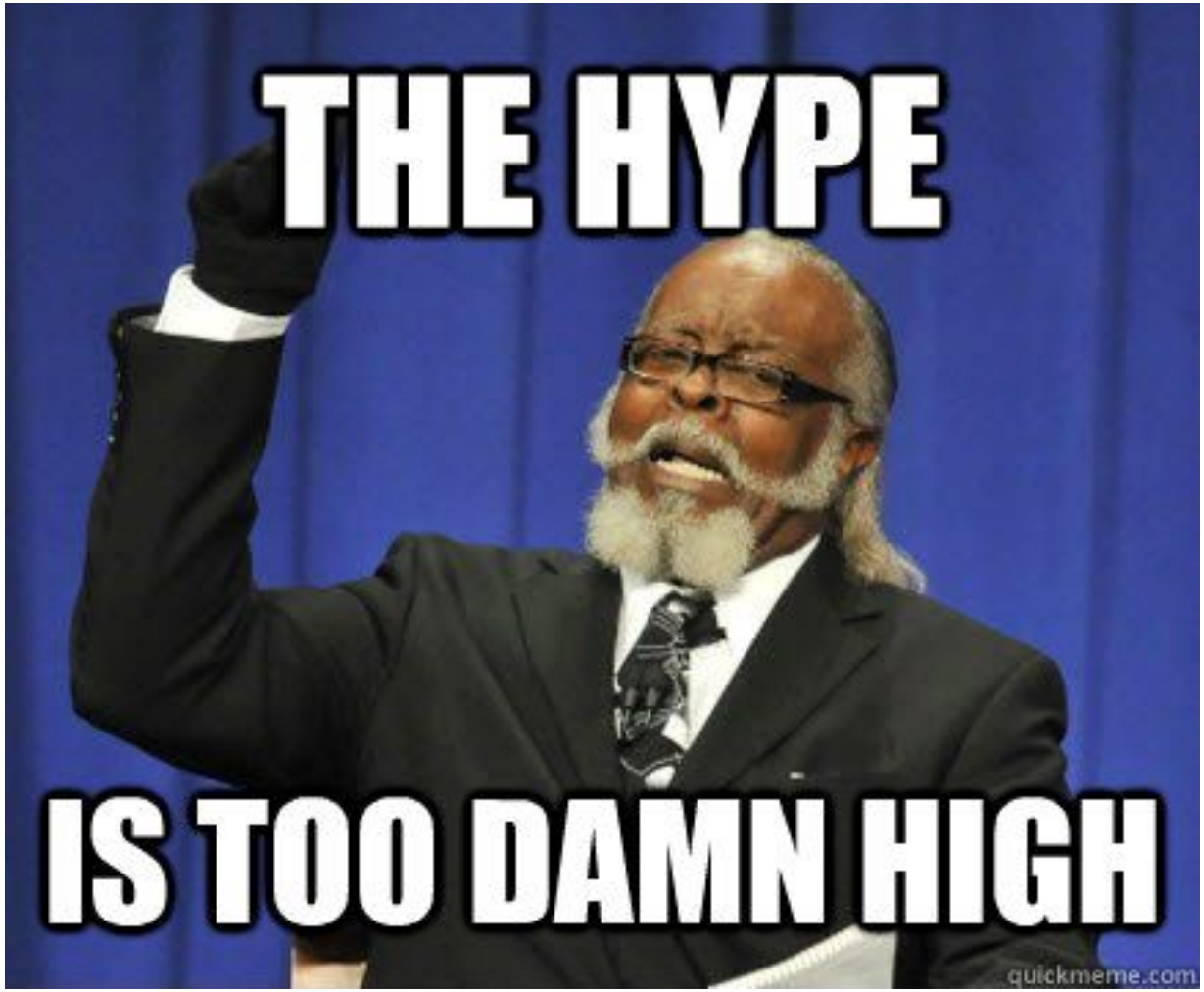
@kotlin_lang

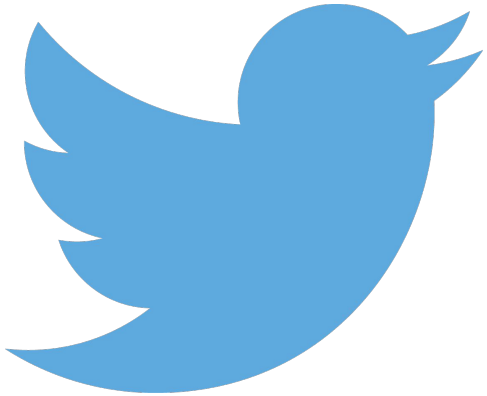
Компании

Google
Amazon
Netflix
Pinterest
Evernote
Uber
Corda

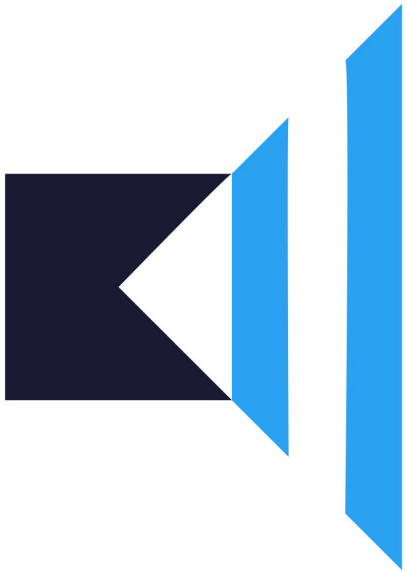
Coursera
Pivotal
Foursquare
Trello
Atlassian
Zalando
Juno

Instamotor
Keepsafe
Coinbase
Hired
Disney
Qiwi
Udemy


THE HYPE
IS TOO DAMN HIGH
quickmeme.com



@Kotlin



@talkingkotlin

https://kotlin.link/

Вопросы?

- Awesome Kotlin: **kotlin.link**
- Twitter/Telegram: **@HeapyHop**