Kotlin

Introduction to DSLs in Kotlin

Victor Kropp

@kropp

Kotlin

What is DSL?

- A **domain-specific language (DSL)** is
a computer language specialized to a particular application domain

- Simpler DSLs, particularly ones used by a single application, are sometimes informally called **mini-languages**

https://en.wikipedia.org/wiki/Domain-specific_language

Kotlin

External DSLs

SQL

```
INSERT INTO Users (FirstName, LastName, Company)
         VALUES ("Victor", "Kropp", "JetBrains")
```

RegExp

```
\s*([^:]+)\s*[^a-z]:(.*?)\s*$
```

Kotlin

Internal DSLs

Examples

Kotlin

kotlinx.html

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

kotlinx.html

```
<html>
 <body>
   <div><a href="http://kotlinlang.org"
 target="_blank">Main site</a></div>
 </body>
</html>
```

Kotlin

Anko

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

Kotlin

Why DSL?

```
verticalLayout {
  for (i in 1..10) {
    button("Say Hello $i")
  }
}
```

Why DSL?

```
verticalLayout {
  generateButtons(10)
}

fun VerticalLayout.generateButtons(count: Int) {
  for (i in 1..count) {
    button("Say Hello $i")
  }
}
```

Kotlin

Building blocks

Extension functions

```
fun Int.days(): Period = ...

fun Period.ago(): Date = ...



3.days().ago()
2.months().later()
```

Extension properties

```
val Int.days: Period
  get() = ...
val Period.ago: Date
  get() = ...

3.days.ago
2.months.later
```

Lambda as last parameter

```
fun f(lambda: () -> Unit) {
 …
}


f() { doSomething() }
```

Lambda with receiver

```
fun f(lambda: StringBuilder.() -> Unit) {
 …
}
f { append("Kotlin") }) }
```

Our own mini DSL

```kotlin
open class Tag(val name: String) {
   val children = mutableListOf<Tag>()
   override fun toString() =
              "<$name>${children.joinToString("")}</$name>"
}

class Html : Tag("html")
class Body : Tag("body")
class P : Tag("p")
```

Kotlin

Our own mini DSL

```
fun html(builder: Html.() -> Unit) =
      Html().apply(builder).toString()


fun Html.body(builder: Body.() -> Unit) =
      children.add(Body().apply(builder))


fun Body.p(builder: P.() -> Unit = {}) =
      children.add(P().apply(builder))
```

Our own mini DSL

```
html {
  body {
    p()
  }
}
```

```
<html><body><p></p></body></html>
```

A problem

```
html {
  body {
    p()
    body {} // shouldn't be allowed here
  }
}
```

@DslMarker annotation

```
@DslMarker
annotation class HtmlTag

@HtmlTag
open class Tag(val name: String) {
   ...
}
```

Context checking

```
html {
  body {
    p()
    body {}
    fun Html.body() can’t be called here by implicit receiver
  }
}
```

Kotlin

Operator overloading

```
html {
  body {
    p {
      +"Hello Kotlin!"
    }
  }
}
```

```
<html><body><p>Hello Kotlin!</p></body></html>
```

Operator overloading

```
class P : Tag("p") {
   private var text = ""
   override fun toString() = "<p>$text</p>"
   operator fun String.unaryPlus() { text = this }
}
```

Infix functions

```kotlin
object Users : Table() {
  val name = varchar("name")
  val company = varchar("company", length = 50)
}

Users.select { Users.company eq "JetBrains" }
```

Kotlin

Infix functions

```
inline fun FieldSet.select(where:
 SqlExpressionBuilder.() -> Op<Boolean>) : Query = …

infix fun<T> ExpressionWithColumnType<T>.eq(t: T) :
 Op<Boolean> = …

Users.select { Users.company.eq("JetBrains") }
```

Kotlin

invoke() convention

```
dependencies.compile("org.jetbrains.kotlinx:kotlinx-html-jvm:0.6.4")

dependencies {
  compile("org.jetbrains.kotlinx:kotlinx-html-jvm:0.6.4")
}
```

Kotlin

invoke() convention

```
fun DependencyObj.invoke(builder: DependencyObj.() -> Unit)
        = this.apply(builder)
```

Kotlin

Links

`kotlinx.html`

https://github.com/Kotlin/kotlinx.html

Anko

https://github.com/Kotlin/anko

Exposed

https://github.com/JetBrains/Exposed

Kotlin Community

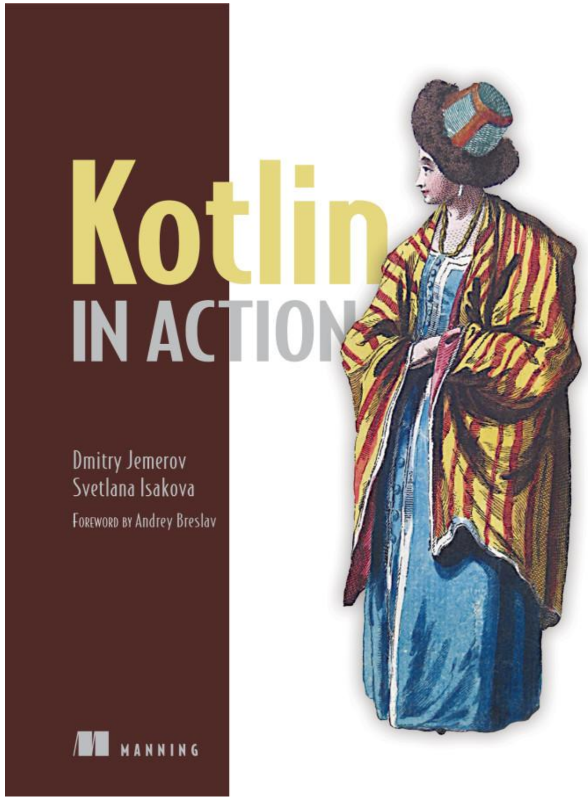
slack

https://kotlinlang.slack.com/

Get invite at

http://slack.kotlinlang.org/

Kotlin in Action


Kotlin
IN ACTION
Dmitry Jemerov
Svetlana Isakova
Foreword by Andrey Breslav
MANNING

Kotlin Conf

kotlinconf.com

2-3 NOV. 2017 SAN FRANCISCO, PIER 27

TWO DAYS OF CONTENT
FROM KOTLIN CREATORS
AND COMMUNITY ENTHUSIASTS

JET BRAINS

AN OFFICIAL EVENT BY JETBRAINS

Kotlin

Thank you!

Victor Kropp

@kropp

victor.kropp.name

Kotlin

Questions?

Victor Kropp

@kropp

victor.kropp.name