June 7th 2017

Native Android applications in Kotlin

Gilbert Adrian Guevara Sabogal
@gilbertguevara @MedellinAndroid
gilbert.guevara@gmail.com

Agenda

- What is Kotlin?
- Why Kotlin?
- Syntax.
- Extension Functions.
- Default values in Parameters.
- Android extensions.
- Delegated properties.
- Init constructor.
- Object expressions.
- Single expressions.
- Data classes.
- Lists Functions and lambdas.

What is Kotlin?

Kotlin is a statically-typed programming language that runs on the Java Virtual Machine and also can be compiled to JavaScript source code or uses the LLVM compiler infrastructure.

Why Kotlin

- Concise.
- Safe.
- Statically typed.
- Interoperable with Java.

Syntax

```kotlin
package com.example.kotlinexample

import android.os.Bundle
import android.support.design.widget.FloatingActionButton
import android.support.design.widget.Snackbar
import android.support.v7.app.AppCompatActivity
import android.support.v7.widget.Toolbar

class KotlinMainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_java_main)
        val toolbar = findViewById(R.id.toolbar) as Toolbar
        setSupportActionBar(toolbar)

        val fab = findViewById(R.id.fab) as FloatingActionButton
        fab.setOnClickListener { view ->
            Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show()
        }
    }
}
```

Syntax

extends/implements

```
class KotlinMainActivity : AppCompatActivity()
{
```

Syntax

Fun – ctions, parameters

```
override fun onCreate(savedInstanceState:
Bundle?) {
```

Syntax

```
override fun onCreate(savedInstanceState:
Bundle?) : Unit {
```

Syntax

Values/variables

```
val price = 100        // Int
price = 30             // don't compile! it's a
constant
var total = price * 3  // Int
val name = "John"      // String
val pi = 3.1416        // Double
val piFloat = 3.1416f   // Float
```

Syntax

Type inference

```
val lastName : String = "Connor" // explicit type definition
var size : Double = 30.0
var time : Float = 15f
val isValid : Boolean = true
```

Syntax

Properties/fields

```
resources.getString(R.string.title_home)
```

Syntax

Safe ?

```
val dontCompile : String = null
var mustBeInitialized : Int
val correct : String? = null
```

Syntax

Safe call

```
val context : Context? = null
val res = context?.resources
```

Syntax

Smart cast

```
val context : Context? = null
context?.let {
    val res = context.resources
    val appName = res.getString(R.string.app_name)
    val shortName = appName.substring(0, 2)
}
```

Syntax

Elvis operator

```
try {
    // code...
} catch (e: Throwable) {
    Log.e("TAG", e.message ?: "Error message")
}
```

Syntax

Statically typed

```
val toolbar = findViewById(R.id.toolbar) as Toolbar
```

Extension Functions

Extension functions allow us to extend the functionality of a class by adding new functions.

```kotlin
fun ViewGroup.inflate(@LayoutRes layoutRes: Int, attachToRoot: Boolean = false): View {
  return LayoutInflater.from(context).inflate(layoutRes, this, attachToRoot)
}
```

```kotlin
/**
 * converts to px
 */
val Float.px: Float get() = (this * Resources.getSystem().displayMetrics.density)
```

Default values in parameters

```
fun build(url: String = BuildConfig.BASE_URL): Retrofit {

  val interceptor = HttpLoggingInterceptor()
  interceptor.level = HttpLoggingInterceptor.Level.BODY

  val okHttpClient = OkHttpClient
      .Builder()
      .addNetworkInterceptor(StethoInterceptor())
      .addInterceptor(interceptor)
      .build()

  return Retrofit.Builder().client(okHttpClient).baseUrl(url)
      .addConverterFactory(MoshiConverterFactory.create())
      .addCallAdapterFactory(RxJava2CallAdapterFactory.create())
      .build()
}
```

Android extensions

```
apply plugin: 'kotlin-android-extensions'
```

```
import kotlinx.android.synthetic.main.home_activity.*
```

```
<android.support.v4.view.ViewPager
    android:id="@+id/pager"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    />
```

```
pager.adapter = HomeAdapter(supportFragmentManager)
```

Delegated properties

Lazy properties

```
private val guideViewLazy by lazy {
  guideView.layoutManager = LinearLayoutManager(activity)
  guideView
}
```

Lateinit modifier

```
lateinit var component: AppComponent

override fun onCreate() {
  super.onCreate()
  this.component = createAppComponent()
  Stetho.initializeWithDefaults(this)
  this.initializeFonts()
}
```

Init constructor

```
class HugersAdapter : EpoxyAdapter() {
  init {
    addModels(HugersMenuModel())
  }
}
```

Object expressions

```
private val immediate = object : Scheduler() {
  override fun createWorker(): Scheduler.Worker {
    return ExecutorScheduler.ExecutorWorker(Executor { it.run() })
  }
}
```

Single Expressions

```
fun String.isEmail() = !this.isNullOrEmpty() &&
EMAIL_ADDRESS.matcher(this).matches()
```

Data classes

```
data class Food(val price: Int,
                val name: String,
                val address: String,
                val description: String,
                val tip: String)
```

Lists Functions and lambdas

```
val foodModels = food.items.map { GuideFoodAndDrinkModel(it).hide()
}
```

References and Links

- https://developer.android.com/kotlin/index.html
- https://kotlinlang.org/docs/tutorials/kotlin-android.html
- https://antonioleiva.com/kotlin/
- https://android.jlelse.eu/learn-kotlin-while-developing-an-android-app-part-1-e0f51fc1a8b3
- https://blog.mindorks.com/a-complete-guide-to-learn-kotlin-for-android-development-b1e5d23cc2d8
- https://github.com/Kotlin/anko

Questions?

Next talks

- Architecture components
- Data binding
- Realm. (ORM)
- Image Processing
- Physics animations
- Custom views
- Accessibility
- Speech to Text. Text to Speech
- Android Things
- What’s new on Android O
- Play Services
- Firebase
- Augmented Reality