ROME - APRIL 13/14 2018

{codemotion}

Kotlin for Android developers

Victor Kropp, JetBrains

@kropp

Kotlin on

JVM + Android

JS

In development: **Kotlin/Native**

iOS/macOS/Windows/Linux

Kotlin

Links

Kotlin
https://kotlinlang.org

Kotlin Koans

https://try.kotl.in

Kotlin in Action book

by Dmitry Jemerov & Svetlana Isakova

Kotlin

Sample Java App

```
package kropp.name.myapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);
   }
}
```

Kotlin

Converting to Kotlin

```
package kropp.name.myapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);
   }
}
```

Converting to Kotlin

```
package kropp.name.myapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);
   }
}
```

Kotlin

Converting to Kotlin

```
package kropp.name.myapp

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState)
       setContentView(R.layout.activity_main)
   }
}
```

Converting to Kotlin

```
package kropp.name.myapp

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState)
       setContentView(R.layout.activity_main)
   }
}
```

Kotlin

Converting to Kotlin

```
package kropp.name.myapp

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState)
       setContentView(R.layout.activity_main)
   }
}
```

Converting to Kotlin

```
package kropp.name.myapp

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState)
       setContentView(R.layout.activity_main)
   }
}
```

Converting to Kotlin

```
package kropp.name.myapp

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState)
       setContentView(R.layout.activity_main)
   }
}
```

Converting to Kotlin

```
package kropp.name.myapp

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity {

   override fun onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState)
       setContentView(R.layout.activity_main)
   }
}
```

Converting to Kotlin

```
package kropp.name.myapp

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity {
   override fun onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
   }
}
```

Converting to Kotlin

```
package kropp.name.myapp

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity {
   override fun onCreate(savedInstanceState: Bundle) {
       super.onCreate(savedInstanceState)
       setContentView(R.layout.activity_main)
   }
}
```

Converting to Kotlin

```
package kropp.name.myapp

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity() {
   override fun onCreate(savedInstanceState: Bundle) {
       super.onCreate(savedInstanceState)
       setContentView(R.layout.activity_main)
   }
}
```

Sample Kotlin App

```
package kropp.name.myapp

import android.support.v7.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity() {
   override fun onCreate(savedInstanceState: Bundle) {
       super.onCreate(savedInstanceState)
       setContentView(R.layout.activity_main)
   }
}
```

Data classes

```kotlin
data class Person(
  var firstName: String,
  var lastName: String,
  var age: Int
)
```

Java equivalent

```java
package kotlindemo;

import java.util.Objects;

public class Person {
 private String firstName;
 private String lastName;
 private int age;

 public Person(String firstName, String lastName, int age) {
   this.firstName = firstName;
   this.lastName = lastName;
   this.age = age;
 }

 public String getFirstName() {
   return firstName;
 }

 public void setFirstName(String firstName) {
   this.firstName = firstName;
 }

 public String getLastName() {
   return lastName;
 }

 public void setLastName(String lastName) {
   this.lastName = lastName;
 }

 public int getAge() {
   return age;
 }

 public void setAge(int age) {
   this.age = age;
 }

 @Override
 public boolean equals(Object o) {
   if (this == o) return true;
   if (o == null || getClass() != o.getClass()) return false;
   Person person = (Person) o;
   return age == person.age &&
       Objects.equals(firstName, person.firstName) &&
       Objects.equals(lastName, person.lastName);
 }

 @Override
 public int hashCode() {
   return Objects.hash(firstName, lastName, age);
 }
}
```

Data classes

```
data class Person(
  var firstName: String,
  var lastName: String,
  var age: Int
)
```

Properties

```
class Person {
  var firstName: String = ""
}
```

Kotlin

Properties

```
public class Person {
  private String firstName;
  public String getFirstName() {
    return firstName;
  }
  public void setFirstName(String firstName) {
    this.firstName = firstName;
  }
  public Person() { this.firstName = ""; }
}
```

Make val not var

```
var   mutable: String
val immutable: String
```

Properties

```kotlin
class Person(
  var firstName: String
)
```

Properties

```
public class Person {
  private String firstName;
  public String getFirstName() {
    return firstName;
  }
  public void setFirstName(String firstName) {
    this.firstName = firstName;
  }
  public Person(String firstName) {
    this.firstName = firstName;
  }
}
```

Null safety

```
val canBeNull: String?
val notNull: String
```

Kotlin

Null safety

```
fun nullability(str: String?) {
  val dot = str.indexOf(".")
}
```

Null safety

```kotlin
fun nullability(str: String?) {
  val dot = str.indexOf(".")
}
```

Only safe (?.) or non-null asserted (!!.) calls are allowed on a nullable receiver of type String?

Kotlin

Null safety

```
fun nullability(str: String?) {
   val dot = str!!.indexOf(".")
}
```

Non-null asserted call

May throw `NullPointerException`

Usually a bad style,
use only when you know what you are doing

Null safety

```kotlin
fun nullability(str: String?) {
   val dot = str?.indexOf(".")
}
```

Safe call

The result will be `null` if `str` is `null`

Kotlin

Null safety

```
fun nullability(str: String?) {
  val dot = str?.indexOf(".") ?: 0
}
```

Elvis operator

The result will be `0` if `str?.indexOf()` returns `null`

Kotlin

Null safety

```
fun nullability(str: String?) {
    val dot = str?.indexOf(".") ?: throw Exception()
}
```

Kotlin

Type casts

```kotlin
fun cast(obj: Any) {
  if (obj is String) {
    val dot = obj.indexOf(".")
  }
}
```

Smart cast

obj is String inside ‘then’ branch

Type casts

```kotlin
fun cast(obj: Any) {
  val str = obj as String
  val dot = str.indexOf(".")
}
```

Kotlin

Type casts

```kotlin
fun cast(obj: Any) {
   val str = obj as? String

   val dot = str.indexOf(".")
}
```

Safe cast

`str` is `null` if `obj` is not a `String`

Kotlin

Extension functions

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

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



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

Kotlin

Extension properties

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

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

Lambda expressions

```
val list = listOf<Int>()

list.filter({ it > 0 })
```

Lambda expressions

```
val list = listOf<Int>()

list.filter { it > 0 }
```

Lambda expressions

```kotlin
val list = listOf<Int>()

list.filter { it > 0 }.map { it*2 }
```

Kotlin

inline functions

```
inline fun <T> Iterable<T>.filter(predicate: (T) -> Boolean):
                                                      List<T> {
  val result = mutableListOf<T>()
  for (it in this) {
    if (predicate(it)) {
      result.add(it)
    }
  }
  return result
}
```

inline functions

```
inline fun <T> Iterable<T>.filter(predicate: (T) -> Boolean):
                                                        List<T> {
  val result = mutableListOf<T>()
  for (it in this) {
    if (predicate(it)) {
      result.add(it)
    }
  }
  return result
}
```

Anko

Anko is a Kotlin library which makes Android application development faster and easier.

https://github.com/Kotlin/anko/

Anko

Anko Commons: a lightweight library with helpers for:
Intents, Dialogs and toasts, Logging, Resources and dimensions

Anko Layouts: a fast and type-safe way to write dynamic Android layouts

Anko SQLite: a query DSL and parser collection for Android SQLite

Anko Coroutines: utilities based on the kotlinx.coroutines library

Kotlin

Anko

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

Android KTX

A set of Kotlin extensions for Android app development

https://github.com/android/android-ktx

Android KTX

```
sharedPreferences.edit()
        .putBoolean("key", value)
        .apply()
```

Kotlin

Android KTX

```
sharedPreferences.edit {
    putBoolean("key", value)
}
```

Android KTX

```kotlin
val spannedString = buildSpannedString {
    bold { "Hello" }
    italic { "KTX!" }
}
```

Kotlin Android Extensions

Kotlin Android Extensions

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

View binding

```
findViewById<TextView>(R.id.label)
```

View binding

```
// Using R.layout.activity_main from the 'main' source set
import kotlinx.android.synthetic.main.activity_main.*


findViewById<TextView>(R.id.label)
```

View binding

```kotlin
// Using R.layout.activity_main from the 'main' source set
import kotlinx.android.synthetic.main.activity_main.*


findViewById<TextView>(R.id.label)
```

View binding

```
// Using R.layout.activity_main from the 'main' source set
import kotlinx.android.synthetic.main.activity_main.*

findViewById<TextView>(R.id.label)
label.text = "Hello!"
```

View binding

```kotlin
// Using R.layout.activity_main from the 'main' source set
import kotlinx.android.synthetic.main.activity_main.*



label.text = "Hello!"
```

Parcelable

```kotlin
import android.os.Parcelable
import kotlinx.android.parcel.Parcelize

@Parcelize
data class Person(
  var firstName: String,
  var lastName: String,
  var age: Int
) : Parcelable
```

Kotlin

Coroutines (Kotlin 1.1)

Asynchronous programming made easy

Write asynchronous code in synchronous style

Coroutines (Kotlin 1.1)

Asynchronous programming made easy

Write asynchronous code in synchronous style

```
val team = api.team().await()
val lead = api.profile(team.lead.id).await()
```

Kotlin

Multi-platform projects

Share code between different platforms

JVM + Android

JS

In development:

iOS/macOS/Windows/Linux **Kotlin/Native**

Thank you!

Victor Kropp
@kropp
victor.kropp.name

Kotlin

Questions?

Victor Kropp
@kropp
victor.kropp.name