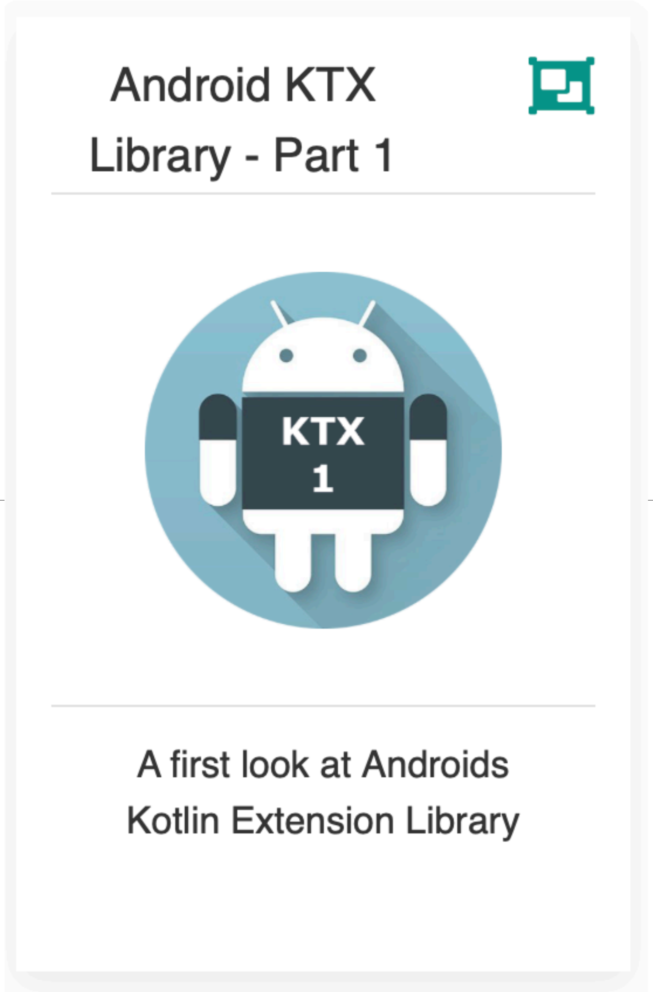
Mobile Application Development

Produced by David Drohan (ddrohan@wit.ie)

Department of Computing & Mathematics
Waterford Institute of Technology
http://www.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Introducing Android KTX Library



Android KTX Library - Part 1

A first look at Androids Kotlin Extension Library

Agenda

- ❑Background
- ❑Extension Functions
- ❑The KTX Library Packages
  (Core, Fragment, Palette, SQLite, Collections)
- ❑KTX in our Case Study (Donation)

Agenda

❑ Background

❑ Extension Functions

❑ The KTX Library Packages
(Core, Fragment, Palette, SQLite, Collections)

❑ KTX in our Case Study (Donation)

What is it?

□ **Android KTX** is one of the new additions to the Android Jetpack family.

□ It’s a collection of Kotlin **extension functions** that enhances and give a Kotlin-ish touch to existing Android APIs.

□ KTX takes advantage of many of the new features in the Kotlin language to make all the old Java Android APIs feel like brand new.

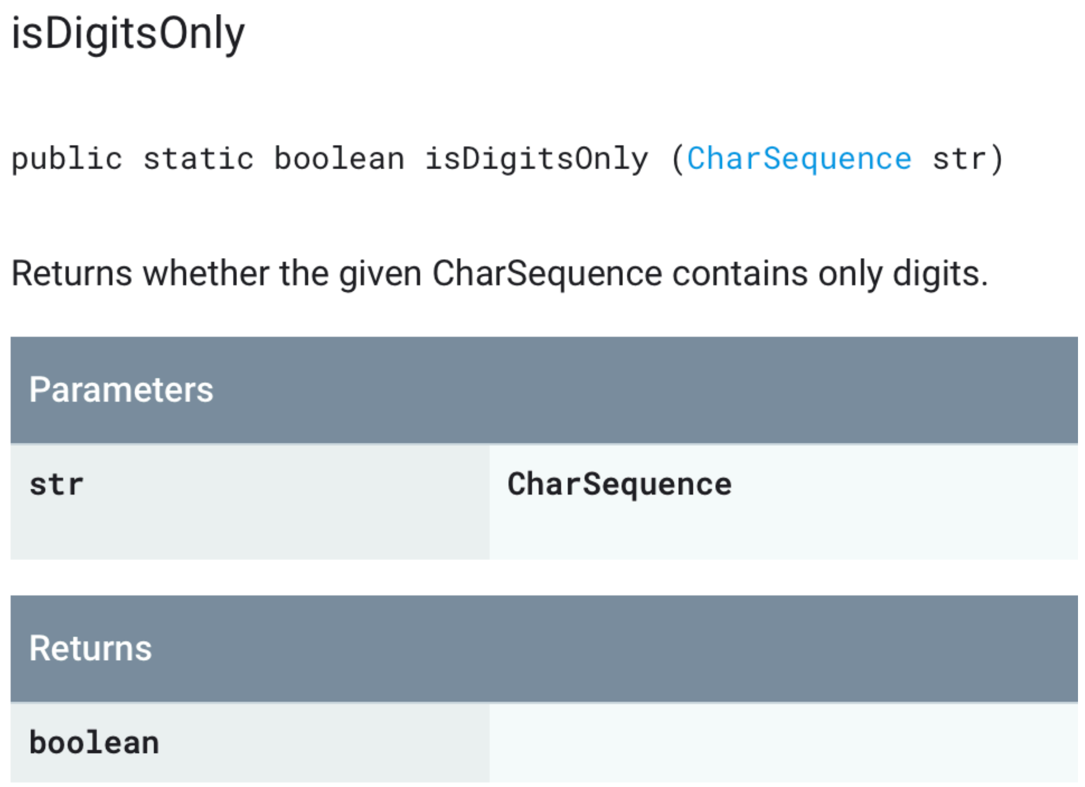
Why is it Useful?

- ❑ It *simplifies* the developers interaction with Android APIs by using those new features of the Kotlin language (*extension functions, lambdas, and named and default parameters*)

- ❑ For example… did you know that the class **TextUtils** has the methods **isDigitsOnly** and **getTrimmedLength**

- ❑ It helps you discover many features of the Android APIs like these that you did not previously know existed because they were buried under tons of utility classes or static methods.

Java



isDigitsOnly

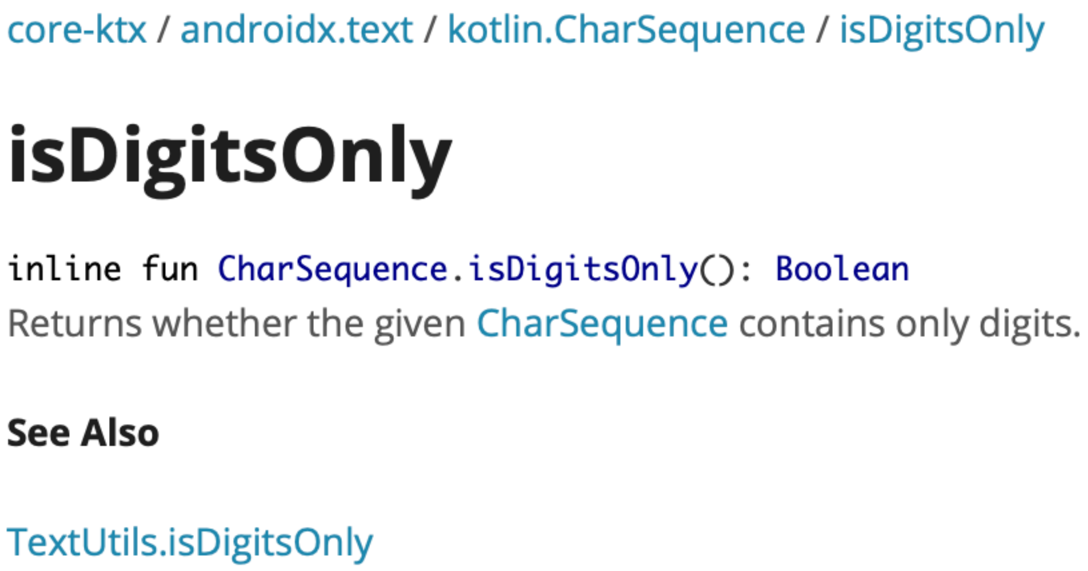
```
public static boolean isDigitsOnly (CharSequence str)
```

Returns whether the given CharSequence contains only digits.

Parameters	
`str`	`CharSequence`

Returns	
`boolean`	

KTX



core-ktx / androidx.text / kotlin.CharSequence / isDigitsOnly

isDigitsOnly

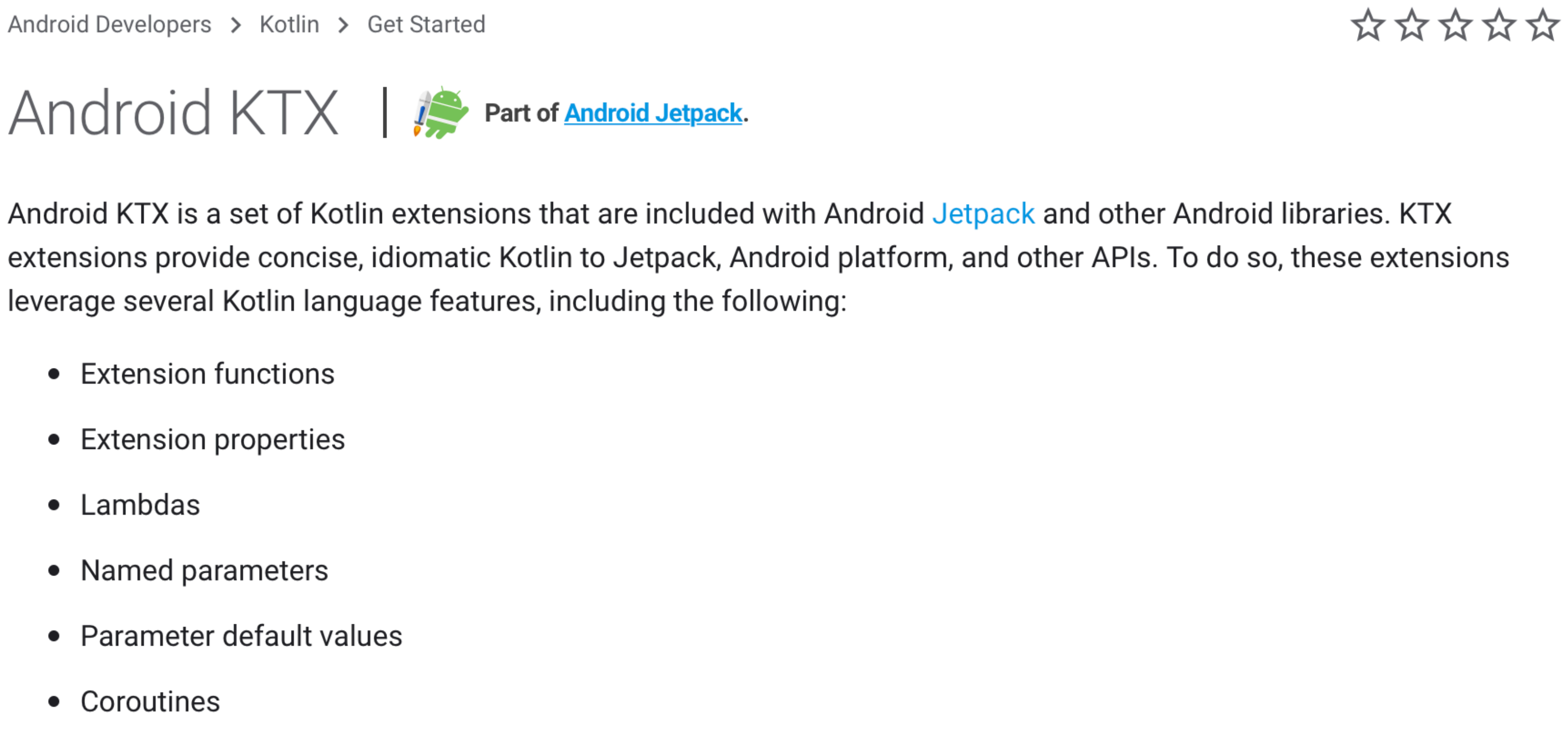
```
inline fun CharSequence.isDigitsOnly(): Boolean
```

Returns whether the given CharSequence contains only digits.

See Also

TextUtils.isDigitsOnly

Official Docs (https://developer.android.com/kotlin/ktx)



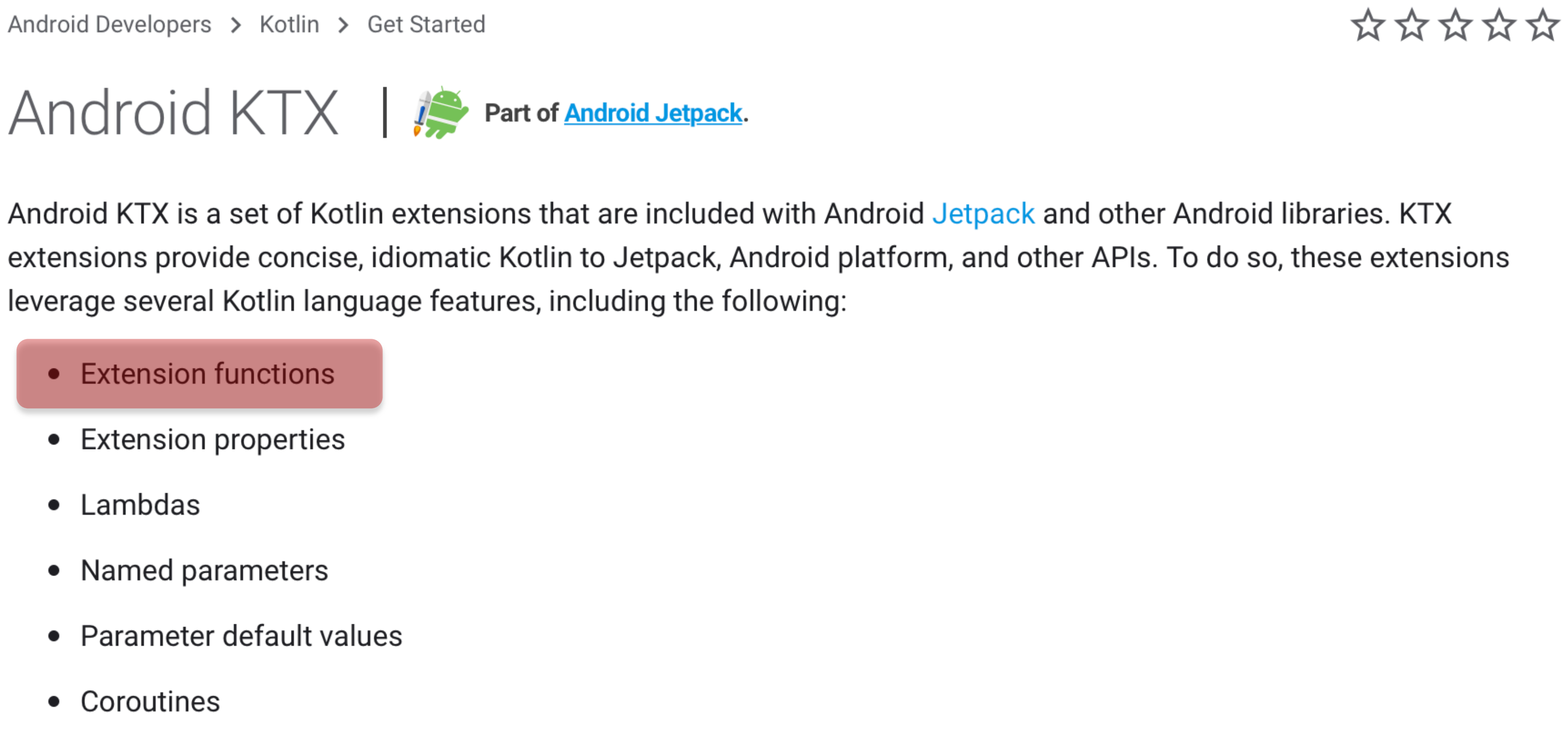
Android Developers > Kotlin > Get Started ☆☆☆☆☆

Android KTX | Part of Android Jetpack.

Android KTX is a set of Kotlin extensions that are included with Android Jetpack and other Android libraries. KTX extensions provide concise, idiomatic Kotlin to Jetpack, Android platform, and other APIs. To do so, these extensions leverage several Kotlin language features, including the following:

- Extension functions
- Extension properties
- Lambdas
- Named parameters
- Parameter default values
- Coroutines

Official Docs (https://developer.android.com/kotlin/ktx)



Android Developers > Kotlin > Get Started ☆☆☆☆☆

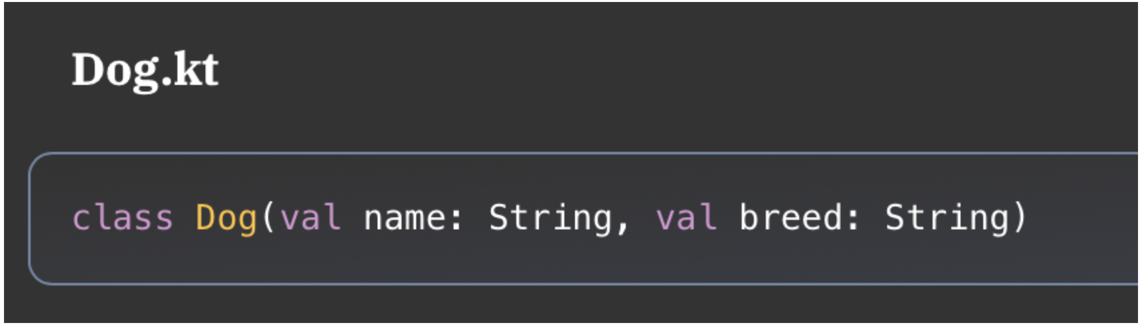
Android KTX | Part of Android Jetpack.

Android KTX is a set of Kotlin extensions that are included with Android Jetpack and other Android libraries. KTX extensions provide concise, idiomatic Kotlin to Jetpack, Android platform, and other APIs. To do so, these extensions leverage several Kotlin language features, including the following:

- Extension functions
- Extension properties
- Lambdas
- Named parameters
- Parameter default values
- Coroutines

Extension Functions

- To understand how **KTX** works, you need to understand Kotlin **Extension Functions.** They allow you to add a function to an existing class *without* modifying the class.
- For example, say you had a **`Dog`** class:

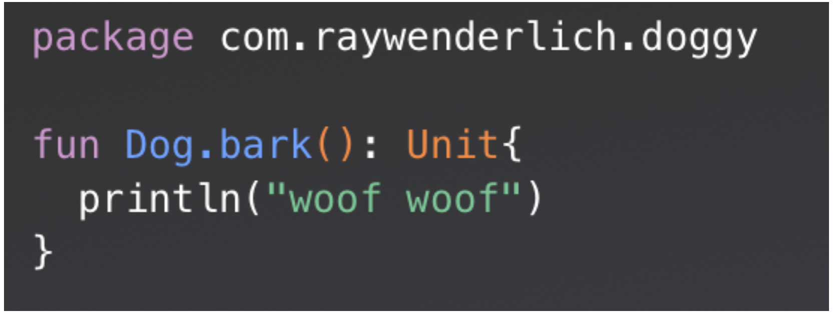


Dog.kt

```kotlin
class Dog(val name: String, val breed: String)
```

Extension Functions

- In another Kotlin file you could add a function to **`Dog`** without modifying the original file:



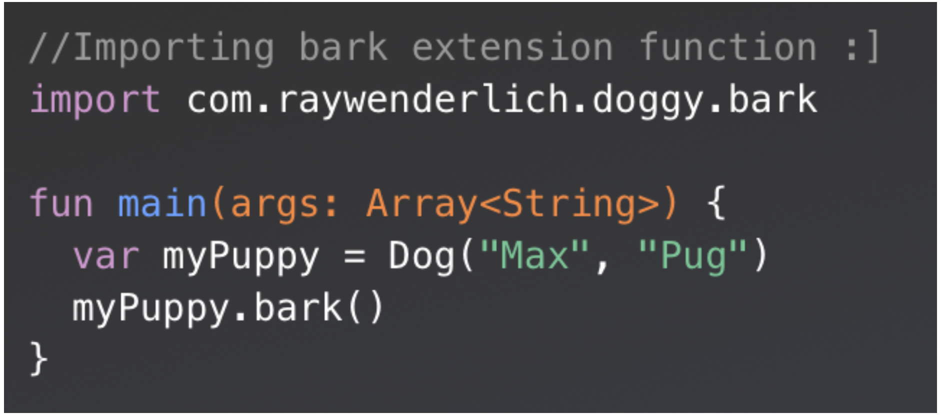
```
package com.raywenderlich.doggy

fun Dog.bark(): Unit{
  println("woof woof")
}
```

- To create the extension function,
  after **`fun`** type the class, then a dot,
  then the name of the extension function.

Extension Functions

- You could test your extension function in another file as follows:



```
//Importing bark extension function :]
import com.raywenderlich.doggy.bark

fun main(args: Array<String>) {
  var myPuppy = Dog("Max", "Pug")
  myPuppy.bark()
}
```

- To use the extension function, you only import bark, and then every Dog object will be able to use the bark() function.

Android KTX Packages

- The KTX toolset is broken up into a number of different packages, so that you can import only the ones you need into your app project:
  - Core KTX: for use with framework APIs such as Toasts, Spans, and Menus
  - Fragment KTX: simplifies Fragment transactions
  - Palette KTX: for working with color palette APIs
  - SQLite KTX: simplifies SQLite database transactions
  - Collections KTX: for use with collection APIs
  - Navigation KTX: for use with Jetpack Navigation

Android KTX Packages

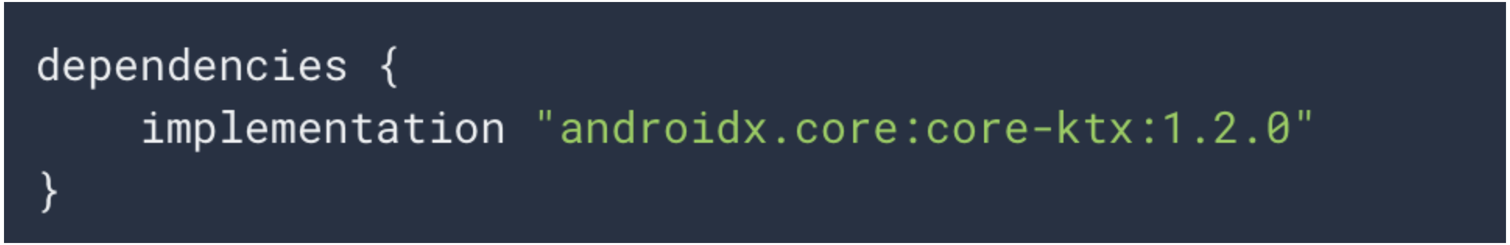
- The KTX toolset is broken up into a number of different packages, so that you can import only the ones you need into your app project:
  - Core KTX: for use with framework APIs such as Toasts, Spans, and Menus
  - Fragment KTX: simplifies Fragment transactions
  - Palette KTX: for working with color palette APIs
  - SQLite KTX: simplifies SQLite database transactions
  - Collections KTX: for use with collection APIs
  - Navigation KTX: for use with Jetpack Navigation

Core KTX

helpers for util, os, view, net & more…

Core KTX

- The **Core KTX** module provides extensions for common libraries that are part of the Android framework.
- These libraries do not have Java-based dependencies that you need to add to **`build.gradle`**.
- To include this module, add the following to your app's **`build.gradle`** file:



```
dependencies {
    implementation "androidx.core:core-ktx:1.2.0"
}
```

Core KTX packages/libraries

- ❑ androidx.core.animation
- ❑ androidx.core.content
- ❑ androidx.core.graphics
- ❑ androidx.core.graphics.drawable
- ❑ androidx.core.net
- ❑ androidx.core.os
- ❑ androidx.core.preference
- ❑ androidx.core.text
- ❑ androidx.core.transition
- ❑ androidx.core.util
- ❑ androidx.core.view
- ❑ androidx.core.widget

Core KTX packages/libraries

- ❑ androidx.core.animation
- ❑ androidx.core.content
- ❑ androidx.core.graphics
- ❑ androidx.core.graphics.drawable
- ❑ androidx.core.net
- ❑ androidx.core.os
- ❑ androidx.core.preference
- ❑ androidx.core.text
- ❑ androidx.core.transition
- ❑ androidx.core.util
- ❑ androidx.core.view
- ❑ androidx.core.widget

Core KTX – **bundleOf** (androidx.core.os)

- **bundleOf** is a Core KTX helper function that lets you create a **`Bundle`** object from anywhere.

```
var bundle = bundleOf("ARG_1" to "Bob", "ARG_2" to false)
```

- Instead of

```
var bundle = bundle()
bundle.putString("ARG_1”, "Bob”
bundle.putBoolean("ARG_2”, false)
```

Core KTX – **drawToBitmap** (androidx.core.view)

- This helper function converts a **`View`** to a **`Bitmap`** object

```
private fun viewToBitmap(view: View) = view.drawToBitmap()
```

- Instead of

```
private fun viewToBitmap(view: View): Bitmap {
// 1 Create a new bitmap with the view info
   val bitmap = Bitmap.createBitmap(view.width, view.height,
                                     Bitmap.Config.ARGB_8888)
// 2 Draw the view pixes into the bitmap
   val canvas = Canvas(bitmap)
   view.draw(canvas)
return bitmap
}
```

Agenda

❑ Background

❑ Extension Functions

❑ The KTX Library Packages
(Core, Fragment, Palette, SQLite, Collections)

❑ KTX in our Case Study (Donation)

References

Sources:

https://www.raywenderlich.com/5576-android-ktx-tutorial-getting-started
https://blog.mindorks.com/android-ktx-android-development-with-kotlin
https://developer.android.com/kotlin/ktx
https://codinginfinite.com/android-ktx-example-android-ktx-core-ktx/

Thanks.