Introduction
Development of mobile devices

Anders Kalhauge

Spring 2017

Outline

Who

Anders Kalhauge

`aka@cphbusiness.dk` (21 72 44 11)

- 21 years experience as IT consultant in the private sector
- 16 years teaching computer science for students and private companies
- Main interests
  - Programming and programming languages
  - Development of large scale systems
  - Software architecture

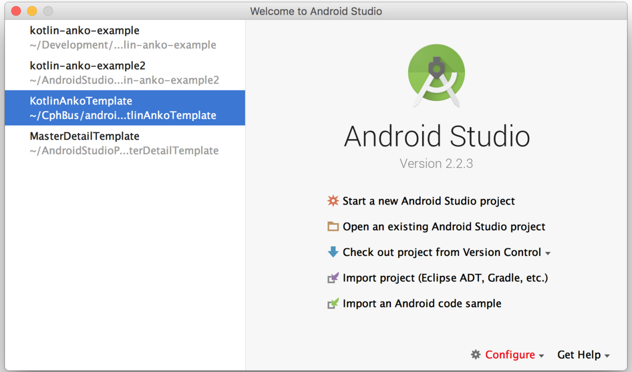
Tools - Android Studio

Made by IntelliJ - fathers of Kotlin

`https://developer.android.com/studio/index.html`

Tools - Kotlin Plugin I

Android Studio


Welcome to Android Studio

kotlin-anko-example
~/Development/...lin-anko-example

kotlin-anko-example2
~/AndroidStudio...in-anko-example2

KotlinAnkoTemplate
~/CphBus/androi...tlinAnkoTemplate

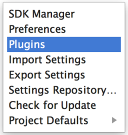
MasterDetailTemplate
~/AndroidStudioP...terDetailTemplate

Android Studio
Version 2.2.3

Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control
Import project (Eclipse ADT, Gradle, etc.)
Import an Android code sample
Configure
Get Help

Tools - Kotlin Plugin II

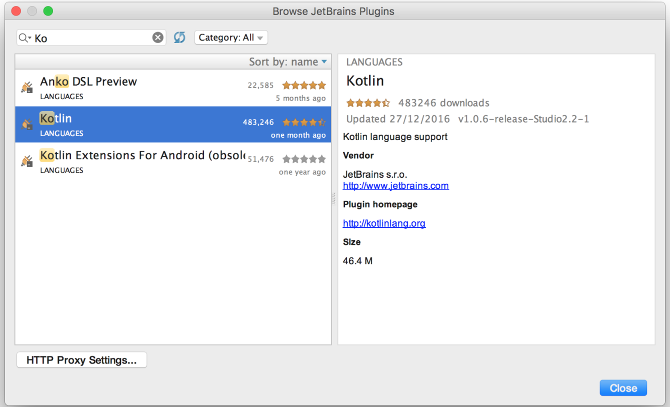
Android Studio



Tools - Kotlin Plugin III

cphbusiness
COPENHAGEN BUSINESS ACADEMY

Android Studio



Plugin	Category	Downloads	Updated
Anko DSL Preview	LANGUAGES	22,585	5 months ago
Kotlin	LANGUAGES	483,246	one month ago
Kotlin Extensions For Android (obsol	LANGUAGES	51,476	one year ago

LANGUAGES
Kotlin
483246 downloads
Updated 27/12/2016 v1.0.6-release-Studio2.2-1
Kotlin language support
Vendor
JetBrains s.r.o.
http://www.jetbrains.com
Plugin homepage
http://kotlinlang.org
Size
46.4 M

Tools - Template project

Android Studio

Use Git to clone:

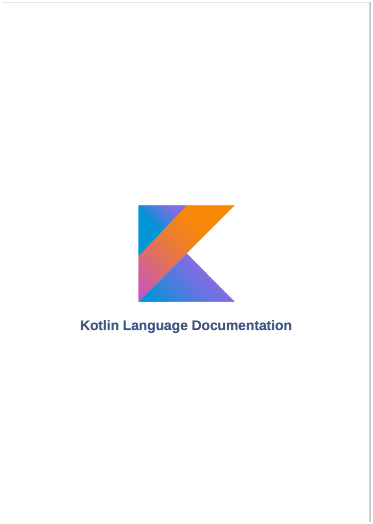
`https://github.com/cphbus-mobile/android-anko-template.git`

Kotlin for Android Developers

Covers most aspects of the curriculum.

Step by step introduction to Kotlin and Android. Large friendly letters and nice examples.

Kotlin Language Documentation


Kotlin Language Documentation

Readable documentation of all Kotlin features with examples. Good as reference.

Kotlin oddities

There are some differences between Kotlin and Java, that can seem a little bit unnescesary.

- The most general class in Kotlin is `Any`, where it is `Object` in Java. Kotlin has its own hiarchy so `Any` is not just another name for `Object`
- The same this goes for `Unit`, which is used where Java would use `void`.
- Types are specified in the end of the signatue, after a colon. Makes it easier to parse, making room for more constructs. Also it follows UML standards.
- More than one class, property, of function can be defined in each file. Directory structure and package names does not have to match, but it is a help if they does.

Kotlin functions

Kotlin functions are first class members as in C. Meaning that they can be defined at package level, outside classes.

- Funtion definition is started with the `fun` keyword.
- Functions can't be overridden unless they are marked `open`.
- If the type is `Unit` don't specify it.
- If the body is an expression and the type can be inferred from the expression, a `=` can be used instead of Type, and return.
- parameters can have default values.

```
fun calculate(a: Int, b: Int) = 7*a + 8*b

fun print(tree: Tree, ind: String = "") {
  if (tree.left != null) print(tree.left, ind+"␣␣")
  println("$ind${tree.data}")
  if (tree.right != null) print(tree.right, ind+"␣␣")
  }
```

Kotlin classes

Kotlin classes have the following properties

- the default constructor in first line.
- the body of the default constructor, if any, is declared in the `init{...}` block
- the default constructor must be called from any other constructor
- additional constructors must be marked with the `constructor` keyword
- only classes marked `open` can be extended
- Kotlin doesn't use the `new` keyword to instantiate objects, constructors are just invoked.

Kotlin class example

```
open class Pet(val name: String)

class Cat(name: String) : Pet(name) {
  var lifeCount: Int

  init { lifeCount = 9 }

  constructor(lifeCount: Int, name: String)
    : this(name) {
    this.lifeCount =
        if (lifeCount > 9) 9
        else if (lifeCount < 0) 0
        else lifeCount
        }
    }
```

Kotlin data classes

In Kotlin classes marked with `data` are special classes for entities or DTOs. They are automatically equipped with:

- `equals(...)`/`hashCode()` pair
- `toString()` of the form `"ClassName(propertyName=value,...)"`
- `component1()`, `component2()`, . . . properties
- `copy(...)` function

Kotin data class example

```kotlin
data class Person(val name: String, var age: Int)

fun main(arguments: Array<String>) {
  val kurt = Person("Kurt", 34)
  println(kurt)
  val (n, a) = kurt
  println("The␣person␣named␣$n␣is␣$a␣years␣of␣age")
  val twin = kurt.copy(name = "Hermann")
  println(twin)
  }
```

```
Person(name=Kurt, age=34)
The person named Kurt is 34 years of age
Person(name=Hermann, age=34)
```

Kotlin properties

Kotlin properties has (as in Java):

- a getter, a setter, or both.
- a state
    - in a backing field
    - defined by the getter/setter
    - delegated to another object.

Kotlin properties example I

```
class User1(
    val id: Int,
    var firstName: String,
    var lastName: String,
    age: Int = 0
    ) {
  var name: String
    get() = "$firstName␣$lastName"
    set(value) {
      firstName = value.substringBeforeLast("␣")
      lastName = value.substringAfterLast("␣")
      }
  var age: Int = age
    get() = field
    private set(value) { field = value }
  }
```

Kotlin properties example Ia

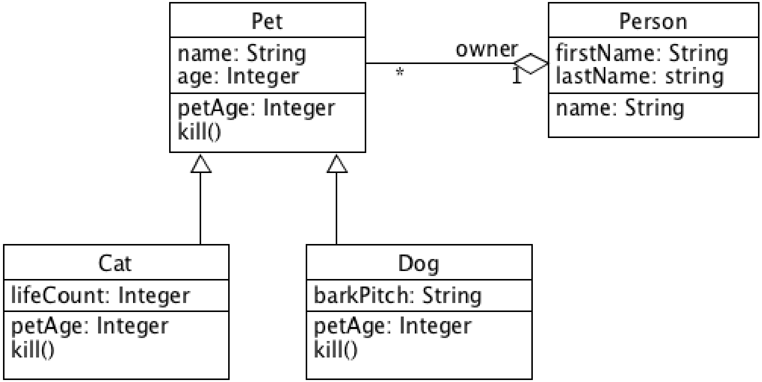
```
import com.google.gson.Gson

fun main(args: Array<String>) {
  val json = """
␣␣␣␣␣␣␣{␣"id":␣7
␣␣␣␣␣␣␣,␣"firstName":␣"Kurt"
␣␣␣␣␣␣␣,␣"lastName":␣"Hansen"
␣␣␣␣␣␣␣,␣"age":␣34
␣␣␣␣␣␣␣}
␣␣␣␣␣␣␣"""
  val gson = Gson()
  val user1 = gson.fromJson(json, User1::class.java)
  println("${user1.firstName}␣is␣${user1.age}␣years")
  }
```

Kotlin properties example II

```
class User2(val data: MutableMap<String, Any?>) {
  val id: Int by data
  var firstName: String by data
  var lastName: String by data
  var age: Int by data
  }
```

Exercise 1 - Kotlin code

Create Kotlin classes for the following:

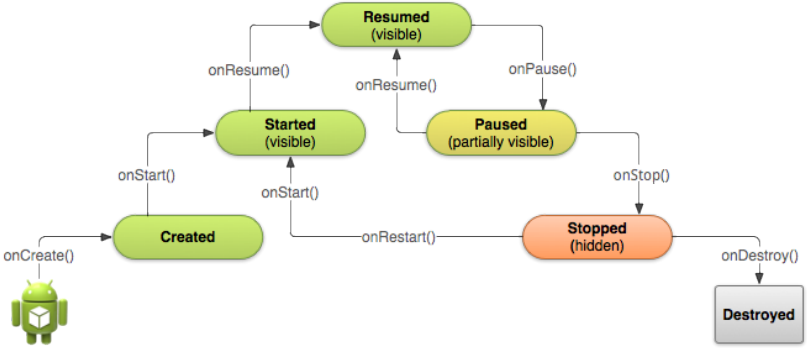


The pet age factor for a cat is 5, for a dog it is 7.

More exercises and reading

Android Activities

Activities are the central building blocks of an Android app.



Please clone:

`https://github.com/cphbus-android/kotlin-anko-template.git`

Android Services

Services are activities without display.
They are not paused when in the background as activities are.
Services run as long as they are not in use any more, or until they are stopped.

Android Broadcast Receivers

Listens for broadcasted events, like incoming calls, sms'es, phone turned on, etc.
Broadcast receivers normally starts the appropriate services or activities.

Android Data Providers

Data providers, provide data between services and activities.
Data providers are often build up around a SQLite database.
The contact book is implemented using a data provider.