Kotlin

android

Basics

Kotlin

Overview

What is Kotlin?

Kotlin is...

Cross-platform
Statically typed
General-purpose

Kotlin compiles to...

Java ByteCode
JavaScript
Native Code

What is Kotlin?

Of course, runs on every **Java Virtual Machine (JVM)**

Libraries written in Kotlin are compatible to be used and run in a Java project...

... and vice versa

What is Kotlin?

Inspired by Java, but...

cleaner

simpler

faster to compile

a mix of OOP and functional programming

Warum überhaupt?

- Wir wollen Nachrichten loggen.
- Nicht alle Nachrichtigen sind gleich wichtig.
- Zu viele Nachrichten füllen mir mein Log an.
- Manchmal möchte ich manche Nachrichten gar nicht sehen.

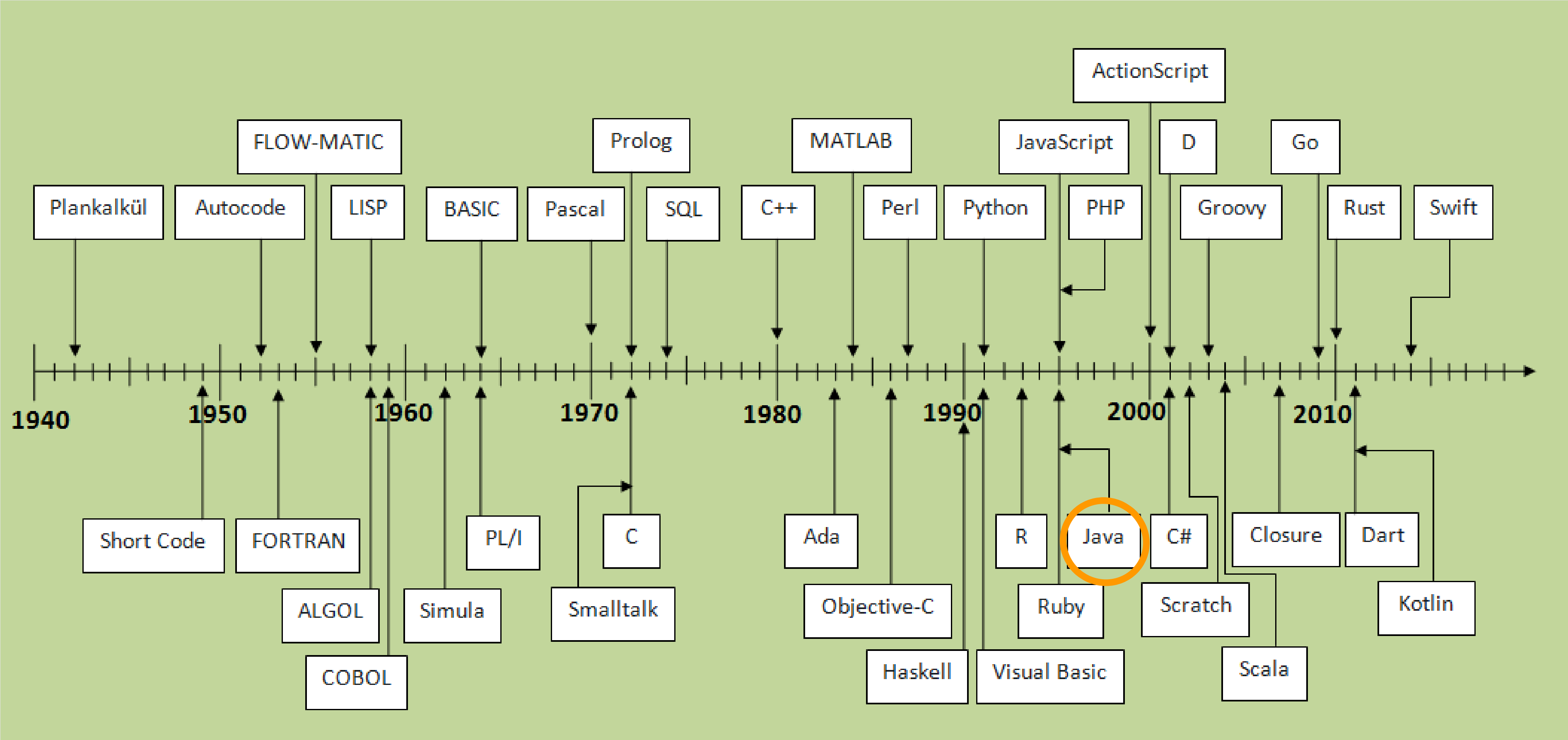
WENN KEINER ZUHÖRT WILL ICH AUCH NICHTS LOGGEN

Schneller Exit wenn kein Appender aktiv.

Kotlin

Motivation

Timeline



https://javaconceptoftheday.com/history-of-programming-languages/

Java is pretty old... arguably...

But why?

Since 2019 it is the favored programming language for Android applications at Google

Underwent an incredible growth since then

Kotlin

Examples

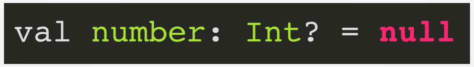
Null Safety

Java

Every object defaults to null

Kotlin

By default there is no null value until you specify it like this



```
val number: Int? = null
```

Extension Functions

Java

Not available

(available by using Project Lombok)

(maybe by inheritance, but you don't own every object you'd like to extend)

Kotlin

A standard language construct by simply prefixing the function name with the class name it should be added to

Code

Java

Verbose (e.g. getter, setters)

Been here before some of the more modern language constructs were invented (e.g. async)

Kotlin

Very concise language (to the point)

Fewer lines of code

Better to code / read / maintain

Coroutines

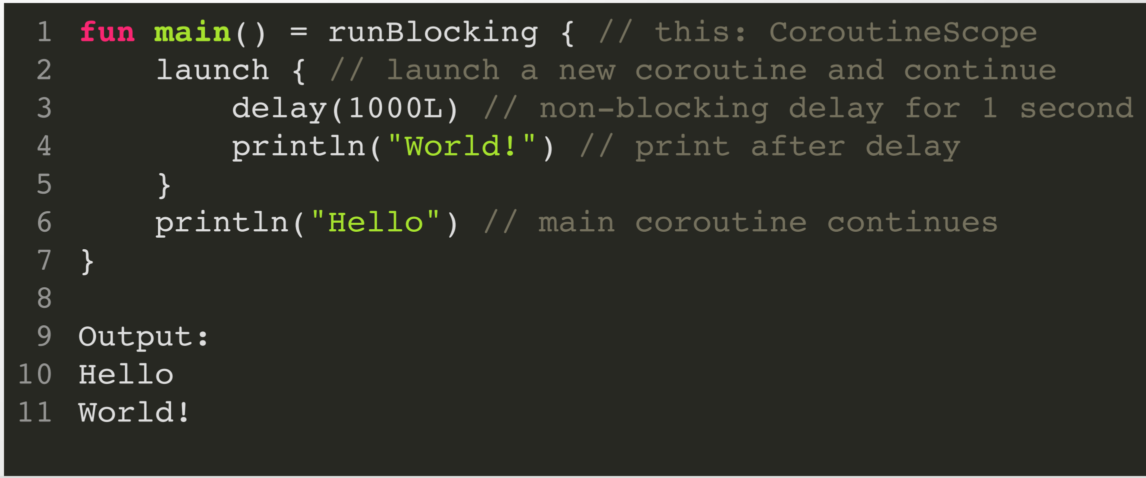
Java

Background threads (ExecutorService, etc.)

Kotlin

Own threadpool

Part of the language



```
fun main() = runBlocking { // this: CoroutineScope
    launch { // launch a new coroutine and continue
        delay(1000L) // non-blocking delay for 1 second
        println("World!") // print after delay
    }
    println("Hello") // main coroutine continues
}

Output:
Hello
World!
```

Data Classes

Java

Manually (verbose)

Solved by using Project Lombok

Kotlin

A language construct

Automatically implements

getters, setters,

hashCode(),

equals(),

toString()...

Smart Casts

Java

Developer has to check the types

Kotlin

Casting checks are handled by the smart casts feature

Redundant checks are removed

No Checked Exceptions

Java

Checked exceptions are available

(IMHO this is a good thing)

Kotlin

No checked exceptions

(IMHO this is a bad thing, because, well ... exception handling)

Higher-Order Funcs & Lambdas

```
max(strings, { a, b -> a.length < b.length })

/**
The function max is a higher-order function,
as it takes a function value as its second argument.
This second argument is an expression that is itself a function,
called a function literal,
which is equivalent to the following named function:
**/

fun compare(a: String, b: String): Boolean = a.length < b.length
```

available in Java as well, to some extent

Primitive Types

Java

Variables of primitive types are not an object

Kotlin

All variables are objects

Public Fields

Java

Available, but should not be used

Kotlin

Not available at all

Wildcard Types

(Generics)

Java

? can be used to specify a type of <any>

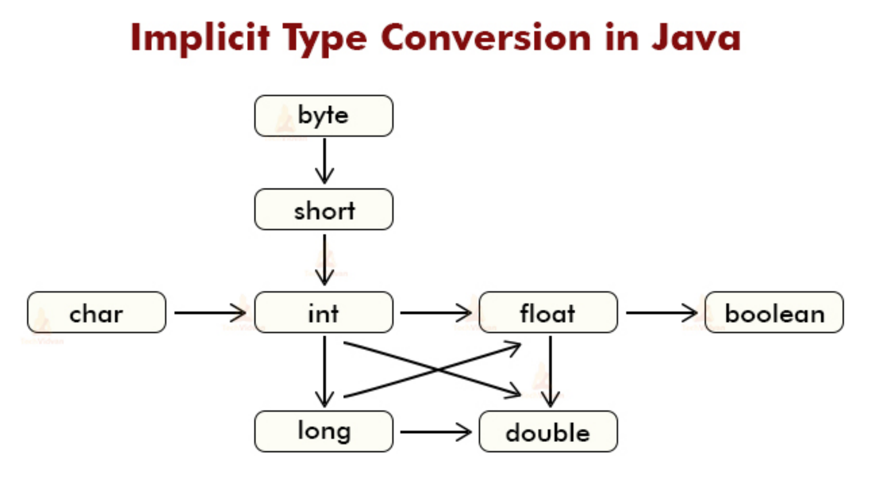
Kotlin

Not available. declaration-site variance and type projections as alternative

Explicit Conversions

Java

Supports implicit conversions (called 'widening')



Kotlin

No implicit conversions. You have to convert explicitly.

Continue here...

https://kotlinlang.org

(Try Kotlin, then Why Kotlin?)

then...

https://kotlinlang.org/docs/home.html

android

History

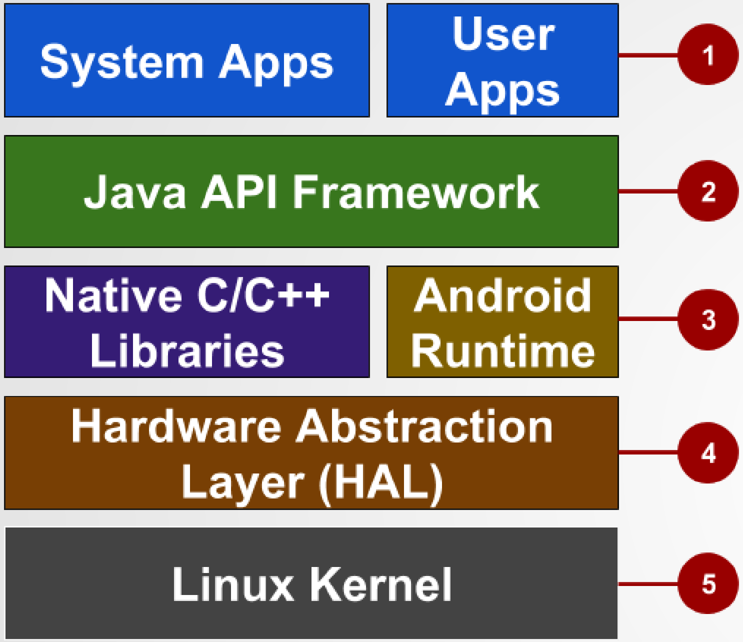
What is Android?

- OS and programming platform developed by Google for mobile devices.

Why Android?

- Most popular platform for mobile apps
- Easy to develop apps
- Many distribution options

How to Android?



1. Your apps live here along with cores-system apps (email, SMS, ...).
2. UI components, resource management, and lifecycle management. Makes native libraries available to the app.
3. Every app runs in its own instance of the Android Runtime.
4. Expose device hardware capabilities to the higher-level Java API framework
5. Foundation. Manages threading, low-level memory management, etc...

android

Overview

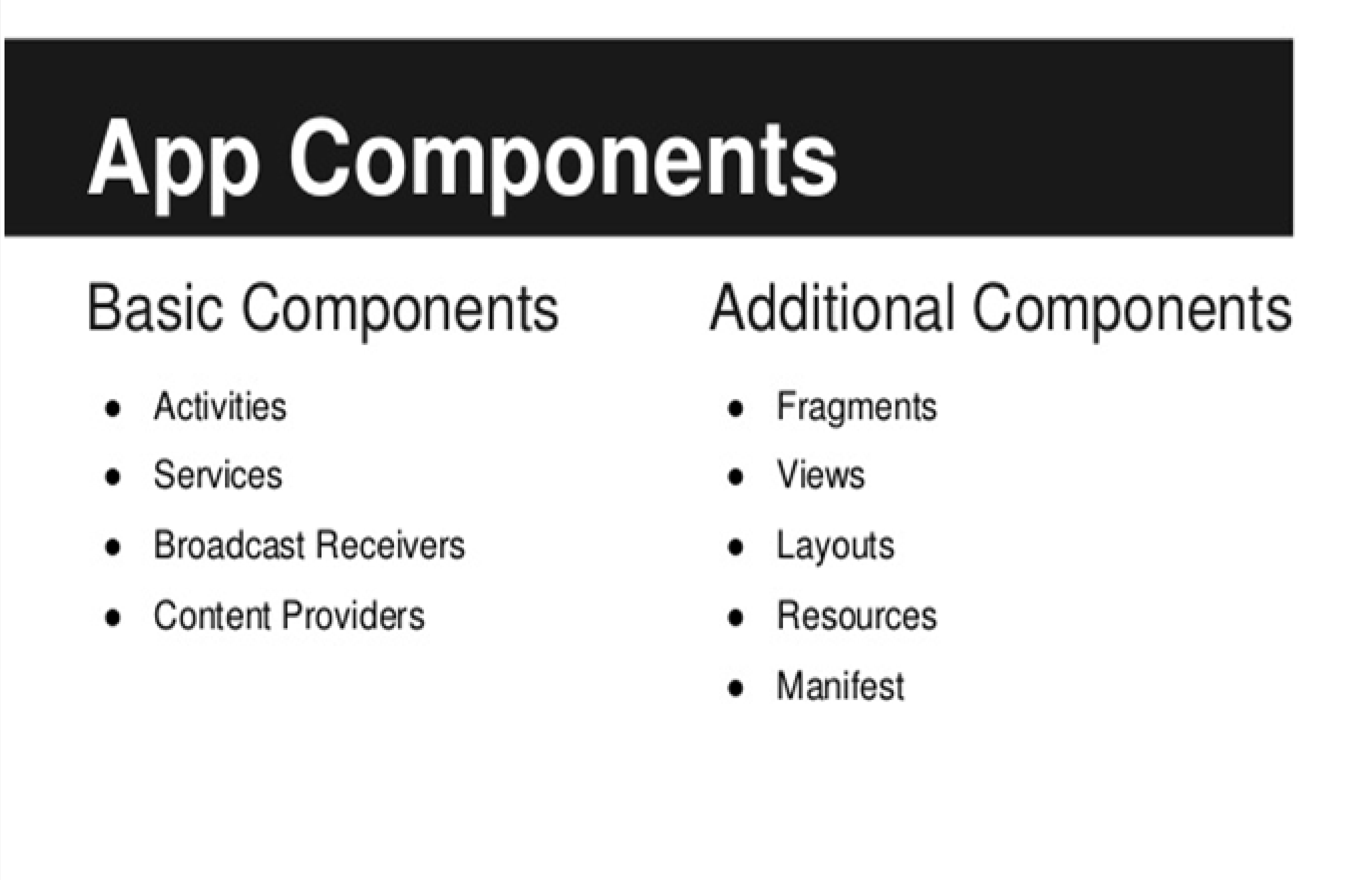
AndroidManifest.xml

- Every Android app project must contain one in the root of project source set.
- This file holds important information about your app for:
  - Build Tools
  - The Android Operating System
  - Google Play

- Is XML because it's easy to read for developers.
- Is compiled to a binary format upon compilation.

AndroidManifest – Contents

- All of the apps...
  - functions
  - services
  - broadcast receivers
  - content producers
  - components
- The apps permissions
- The apps rights (permissions other apps need to have in order to get data from it)



App Components

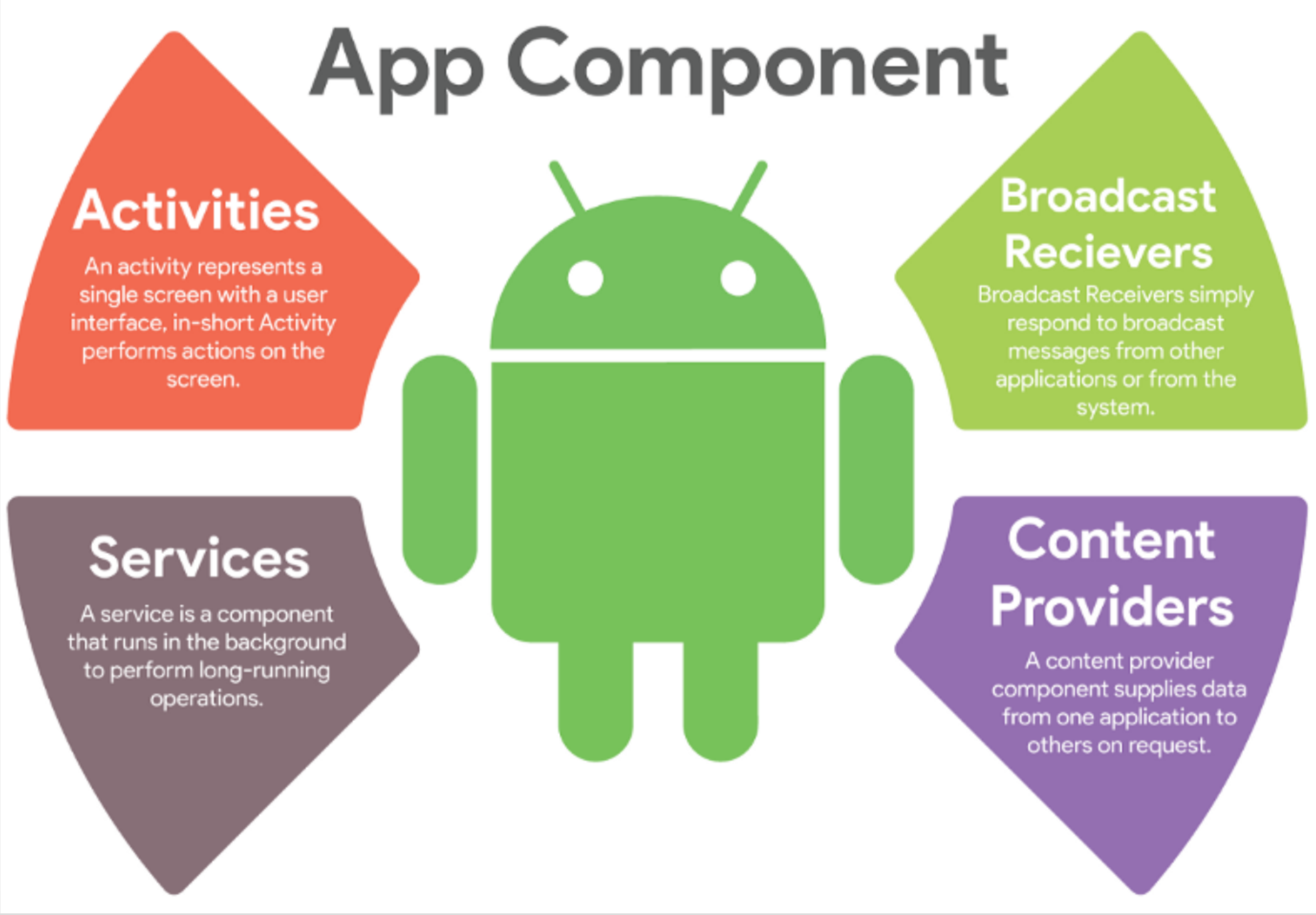
Basic Components

- Activities
- Services
- Broadcast Receivers
- Content Providers

Additional Components

- Fragments
- Views
- Layouts
- Resources
- Manifest

Basic Components



App Component

Activities

An activity represents a single screen with a user interface, in-short Activity performs actions on the screen.

Broadcast Recievers

Broadcast Receivers simply respond to broadcast messages from other applications or from the system.

Services

A service is a component that runs in the background to perform long-running operations.

Content Providers

A content provider component supplies data from one application to others on request.

Additional Components

Fragments

Parts of a screen used inside of activities.

Views and ViewGroups

Views are individual GUI Elements.
ViewGroups are containers for views.

Layouts

XML files containing layout data for activities, fragments or ViewGroups.

Resources

Images, Strings, UI Layouts...

Manifest

Explained above.

android

Jetpack Compose

Overview

History

- Android development follows many programming paradigms => gets complicated
- Library to abstract that complexity
- Announced 2019 (by Google)

Jetpack Compose

- Modern toolkit for Android development
- Reactive programming model
- Uses Kotlin language constructs
- Fully declarative like flutter, Swift UI or React Native (tell it WHAT you want, not HOW you want to do it)

Jetpack Compose

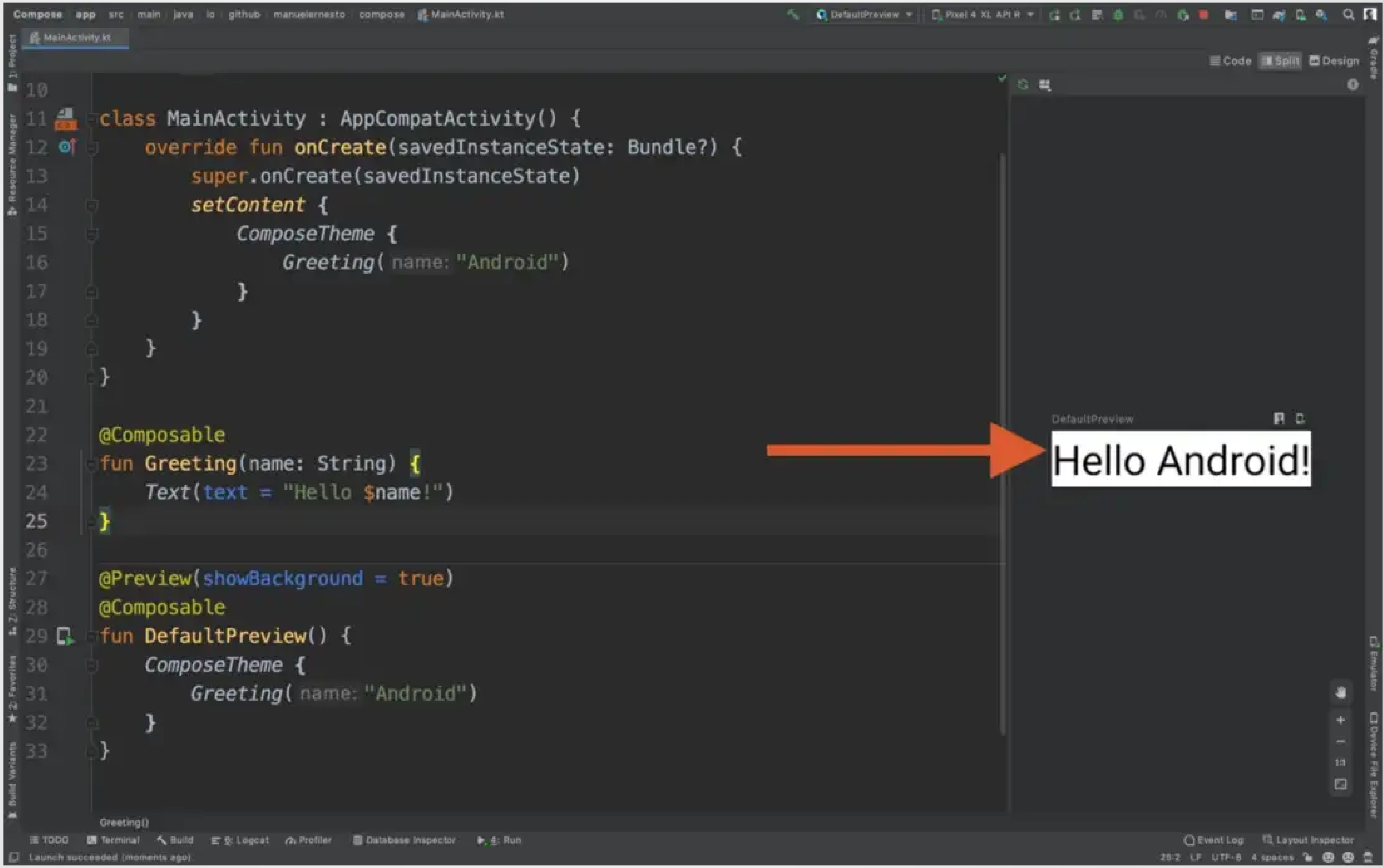
- …describes the UI by calling functions that transform data into a UI hierarchy
- …when data changes, the framework automatically calls these functions, thus updating the UI

android

Jetpack Compose

Example

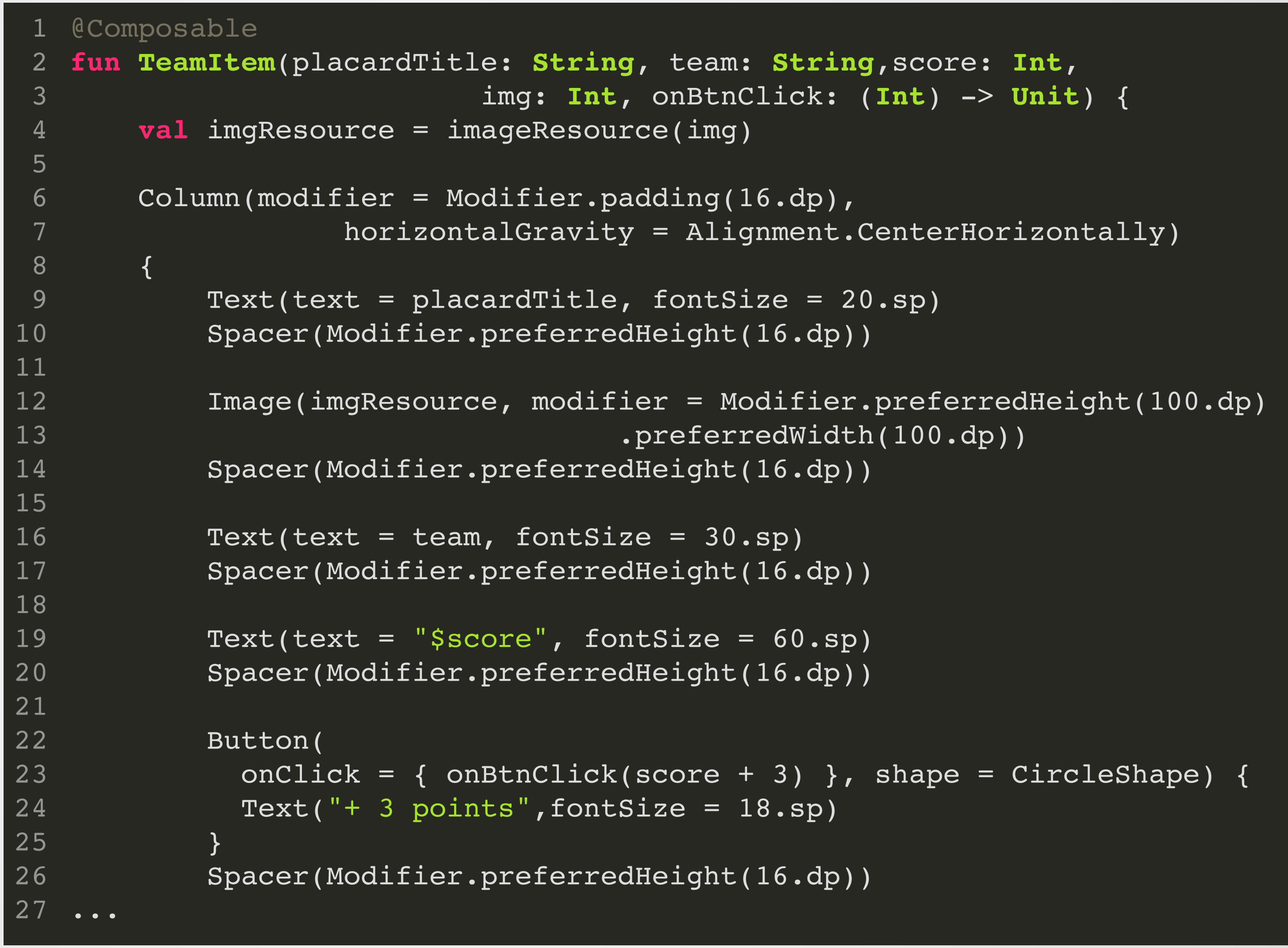
Jetpack Compose



```kotlin

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            ComposeTheme {
                Greeting( name: "Android")
            }
        }
    }
}

@Composable
fun Greeting(name: String) {
    Text(text = "Hello $name!")
}

@Preview(showBackground = true)
@Composable
fun DefaultPreview() {
    ComposeTheme {
        Greeting( name: "Android")
    }
}
```

Launch succeeded (moments ago)

Jetpack Compose



```
@Composable
fun TeamItem(placardTitle: String, team: String,score: Int,
                        img: Int, onBtnClick: (Int) -> Unit) {
    val imgResource = imageResource(img)

    Column(modifier = Modifier.padding(16.dp),
                horizontalGravity = Alignment.CenterHorizontally)
    {
        Text(text = placardTitle, fontSize = 20.sp)
        Spacer(Modifier.preferredHeight(16.dp))

        Image(imgResource, modifier = Modifier.preferredHeight(100.dp)
                                .preferredWidth(100.dp))
        Spacer(Modifier.preferredHeight(16.dp))

        Text(text = team, fontSize = 30.sp)
        Spacer(Modifier.preferredHeight(16.dp))

        Text(text = "$score", fontSize = 60.sp)
        Spacer(Modifier.preferredHeight(16.dp))

        Button(
          onClick = { onBtnClick(score + 3) }, shape = CircleShape) {
          Text("+ 3 points",fontSize = 18.sp)
        }
        Spacer(Modifier.preferredHeight(16.dp))
...
```

Jetpack Compose

- No more 'drawing' in the editor
- Declarative language
- Data binding as it should be
- Like JS single-page languages (Angular, React JS, etc...)

References

- https://www.imaginarycloud.com/blog/kotlin-vs-java/
- https://google-developer-training.github.io/android-developer-fundamentals-course-concepts-v2/unit-1-get-started/lesson-1-build-your-first-app/1-0-c-introduction-to-android/1-0-c-introduction-to-android.html
- https://www.techplayon.com/applications-component/
- https://proandroiddev.com/the-journey-of-jetpack-compose-i-eef660bc546b
-