Android history and basics of Kotlin

Android Lecture 1

2024

Lukáš Prokop
Simona Kurňavová

About this course

Course page: https://d3s.mff.cuni.cz/teaching/nprg056/

Garant: Jan Kofroň

Lecturers: Lukáš Prokop, Simona Kurňavová

Schedule for semestral projects:

- *October 1 – December 1*: Forming project groups (1-3 students) and creating project specifications
- *December 1*: The project specification has to be accepted by a lecturer
- *February 28:* Final version of the project
- *April 15*: Issues identified by lecturers fixed

How to submit project specification:

Via email to Jan Kofroň (jan.kofron@d3s.mff.cuni.cz) with Lukáš Prokop (Lukas.Prokop@gendigital.com) and Simona Kurňavová (Simona.Kurnavova@gendigital.com) as cc. *Email should contain:*

- What is the purpose of the application
- Description of features and functionalities of the application
- Optionally: technical stack and other clarifying information.

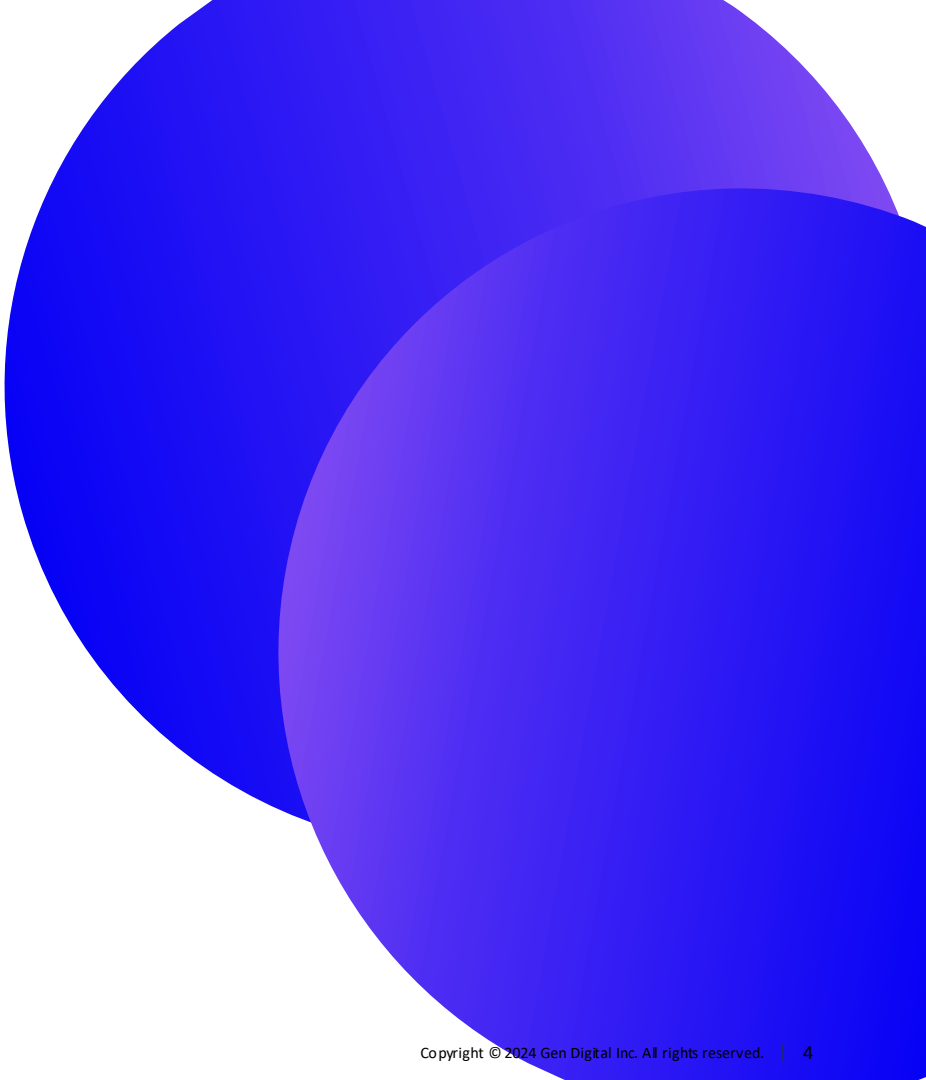
How to hand over project:

Ideally using Github/Gitlab/Bitbucket repository link (please make sure it would be accessible to us).

Agenda

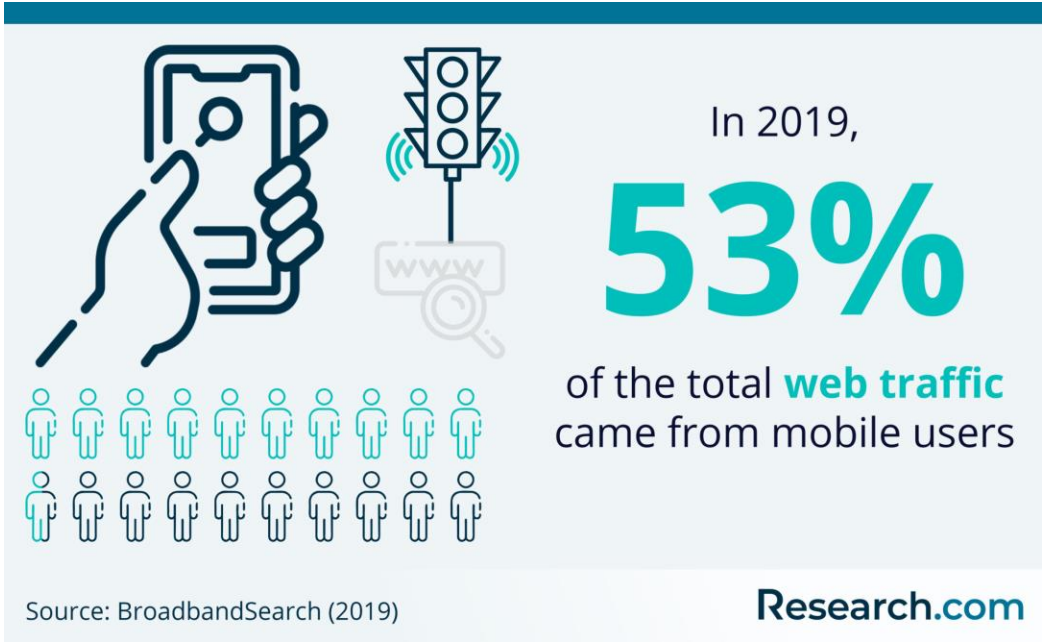
- Mobile and Android history
- Android development
- Kotlin
- Q&A

Why mobile development



Why mobile development

- **Ubiquitous**
- **Quick**
- **Convenient**



Source

Mobile device specifics

- **Not enough computation power**
- **Changing state:**
  - Screen orientation
  - Foldables
- **Unstable network connection**
- **Small battery**
- **Small and variable display size**
- **Never ending interruptions**
  - Engaging notifications
  - Alarms

History

Timeline

2000

APRIL 2000

Pocket PC 2000

Windows Mobile

OCT 2003

Android Inc.

SW for digital cameras

JUL 2005

Android Inc. acquired by Google

JUN 2007

iPhone

JUN 2007

OHA & AOSP

OCT 2008

T-Mobile G1 (HTC)

MAY 2009

Windows mobile 6.5

2010

Timeline

OCT 2017

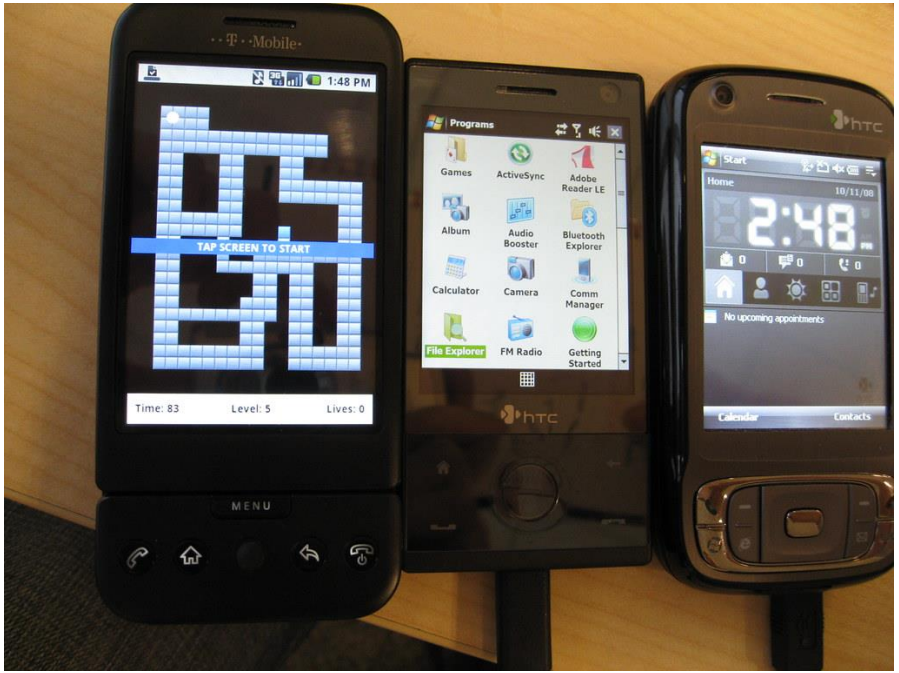
Microsoft discontinued Windows 10 Mobile

2010 — 2020

NOV 2010

Windows Phone 7

HTC G1



android

- **Linux-based OS**
  - No shell, no access to root (by default)
  - GNU C library (glibc) replaced by Bionic
- **Open source**
  - https://source.android.com
  - Just the OS (Google play store and services are proprietary)

android

2008

1.0 SEPTEMBER 2008

API 1

1.1 FEBRUARY 2009

1.5 CUPCAKE

APRIL 2009

API 3

1.6 DONUT

SEPTEMBER 2009

API 4

2.0 – 2.1 ÉCLAIR

OCTOBER 2009

API 5-7

2.2 FROYO

MAY 2010

API 8

Android 2.3

2.3 GINGERBREAD

DECEMBER 2010

API 9 - 10

2010

android

2011

2010

3.X HONEYCOMB

FEBRUARY 2011

API 11 – 13

4.0.X ICE CREAM SANDWICH

OCTOBER 2009

API 14 – 15

4.1 – 4.3 JELLY BEAN

JULY 2012

API 16 – 18

4.4 KITKAT

OCTOBER 2013

API 19

4.4W KITKAT

OCTOBER 2009

API 20

5.0 – 5.1 LOLLIPOP

NOVEMBER 2014

API 21 - 22

- Material design

6.0 MARSHMALLOW

OCTOBER 2015

API 23

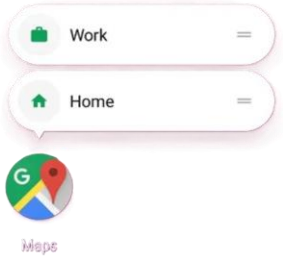
Android Nougat (API 24, 25)

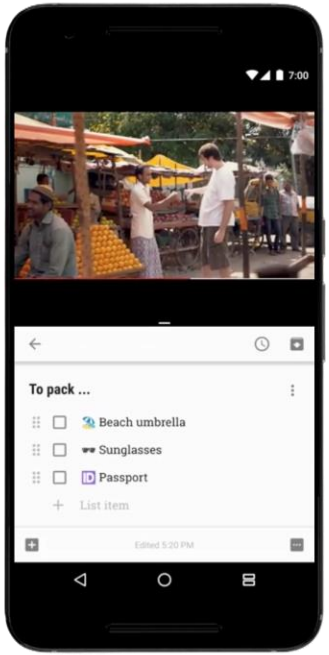
August 2016 (Android 7.0, API 24)

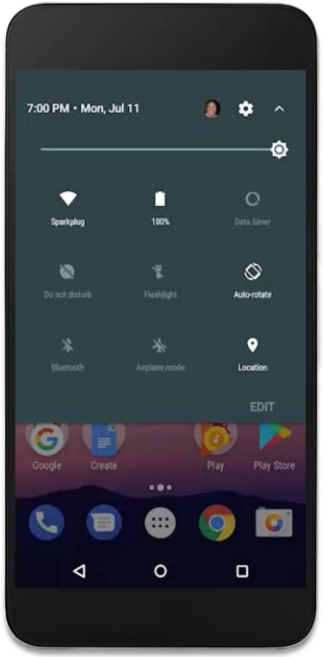
- Multi window
- Quick setting tiles
- Vulkan API

October 2016 (Android 7.1, API 25)

- App shortcuts







Source

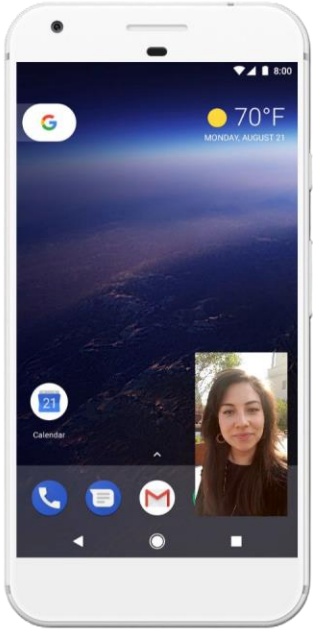
Android Oreo (API 26, 27)

August 2017 (Android 8.0, API 26)

- Picture in Picture
- Notification channels
- Custom fonts
- Autosize text view
- Multi display
- Project treble
  - Sony, Nokia, OnePlus

October 2017 (Android 8.1, API 27)

- Neural Network API
- Video thumbnail extractor
- Wallpapers color API
- Cryptography update
  - Conscrypt over Bouncy castle



Source

Android Pie (API 28)

August 2018 (Android 9.0)

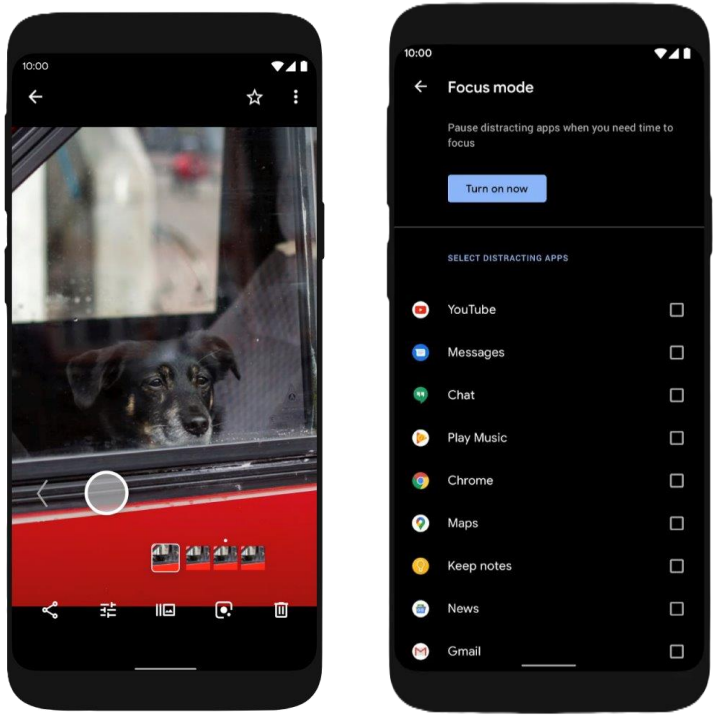
- Display cutout supports
- Notification – messaging
- Multicamera support
- Image decoder (HEIF, HDR)
- Gesture navigation
- Animation
  - GIF, WebP animated images
  - HDR VP9, HEIF and Media APIs

Source

Android 10 (API 29)

September 2019

- Gesture navigation (new version)
- Smart replies
- Dark theme
- Foldables

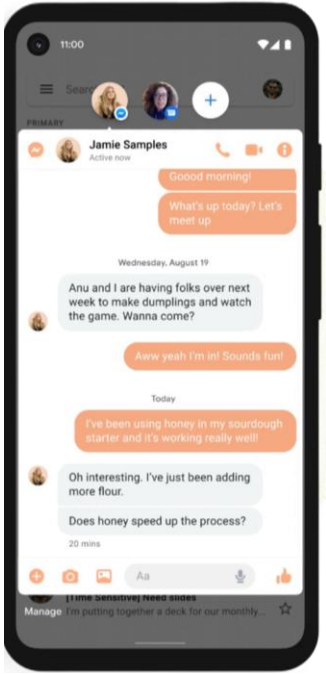


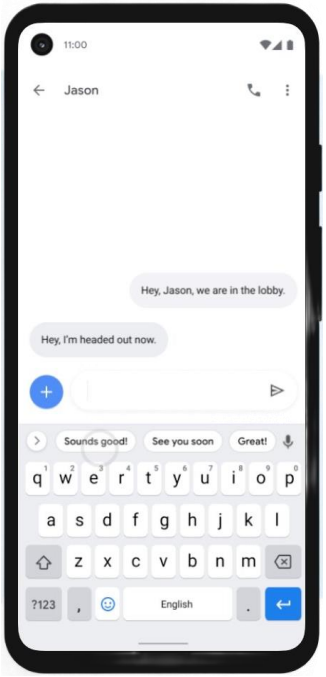
Source

Android 11 (API 30)

September 2020

- Chat bubbles
- Smart replies
- Notification history
- One time permission
- Permission auto-reset
- 5G detection API





Source

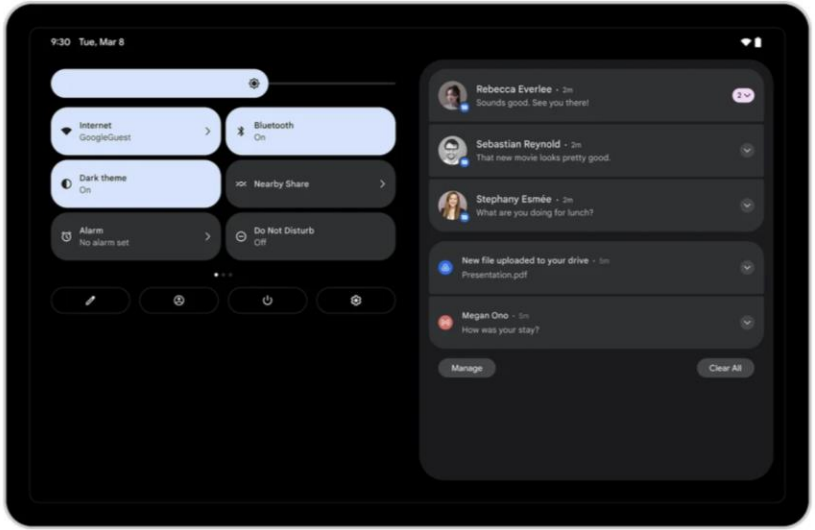
Android 12 (API 31, 32)

October 2021 (Android 12.0, API 31)

- Material You – design refresh
- Splash screen API
- Extending screenshot beyond screen
- Bluetooth permissions
  - Scan for nearby device don’t need location

March 2022 (Android 12L, API 32)

- Optimized for large screens and foldables

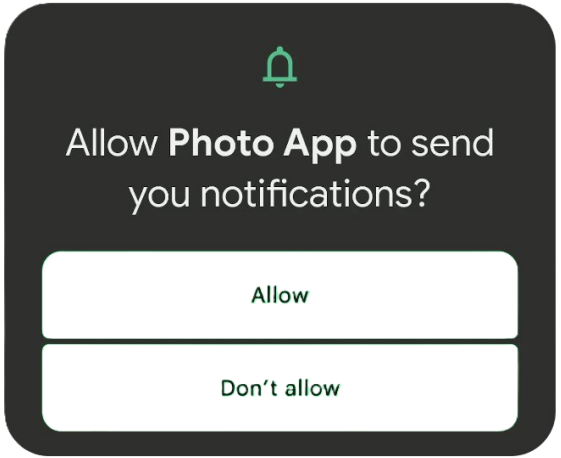


Source

Android 13 (API 33)

August 2022

- Notification permission
- Nearby wifi devices permission

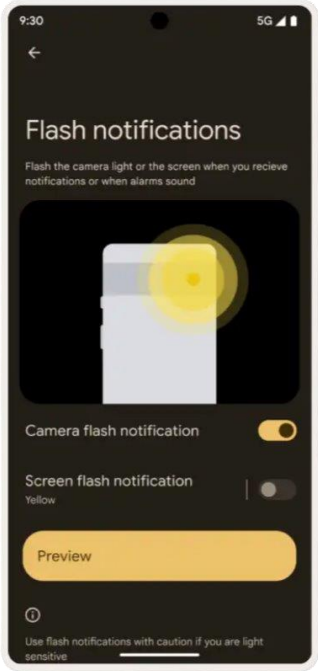


Source

Android 14 (API 34)

October 2023

- Exact alarm permission denied by default
- Partial access to photos and videos
- Change to non-dismissable notifications
- Security changes
- Accessibility improvements
  - Flash notifications
  - Better readability with font scaling



Source

Android 15 (API 35)

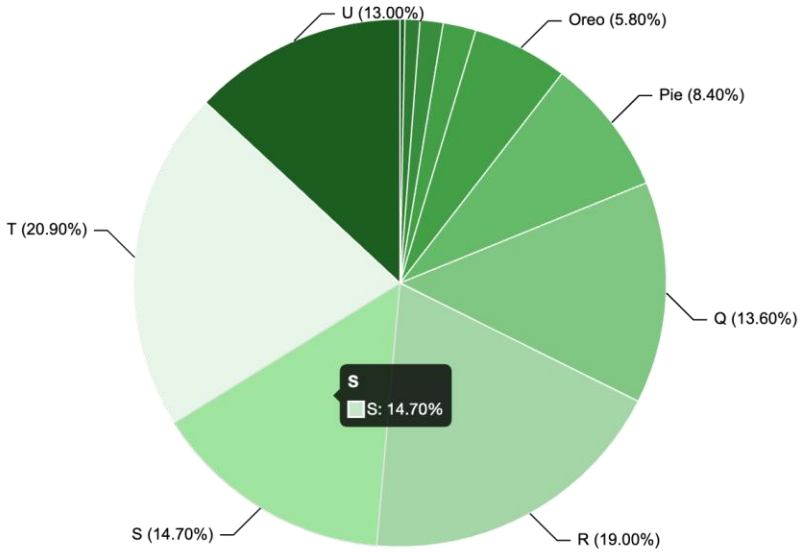
October 2024 (in beta) docs

- Private space
- Screen recording detection
- Improvements to Picture-in-picture

Source

Android version distribution

May 1, 2024:



API Distribution

Platform Version	API	Distribution
Android 4.4 (KitKat)	19	0.3%
Android 5 (Lollipop)	21	0.1%
Android 5.1 (Lollipop)	22	0.8%
Android 6 (Marshmallow)	23	1.4%
Android 7 (Nougat)	24	1.0%
Android 7.1 (Nougat)	25	1.0%
Android 8 (Oreo)	26	1.5%
Android 8.1 (Oreo)	27	4.3%
Android 9 (Pie)	28	8.4%
Android 10 (Q)	29	13.6%
Android 11 (R)	30	19.0%
Android 12 (S)	31	14.7%
Android 13 (T)	33	20.9%
Android 14 (U)	34	13.0%

source1, source2, source3

Gen

android

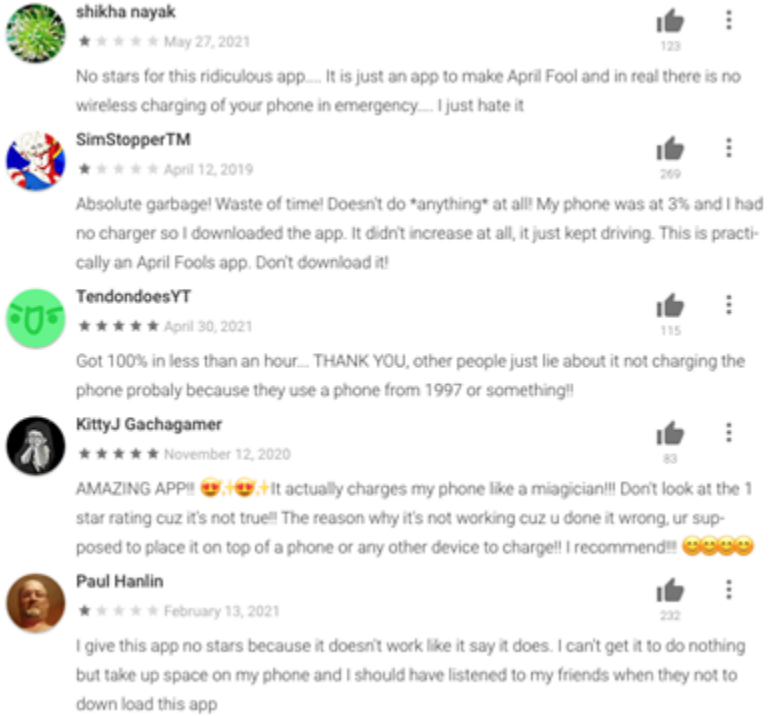
- **Android – Phones and tablets**
- **WearOS - Smartwatch**
  - Extended notification center
  - Sporttester
- **Chrome OS**
- **Android Auto**
  - Mirror optimized UI to built-in infotaiment
- **Android automotive**
  - Standalone OS in cars
- **Android Things** (deprecated January 2022)
  - IoT
- **Google glass**
  - Glass explorer program (retired 2015)
  - Glass for enterprise (retired 2023)

Android Ecosystem

- **Distribution:**
  - Primary Google play store
  - Alternative stores: Samsung Galaxy store, Amazon store, …
  - Direct downloads
- **Different monetization models**
  - Free
  - Paid
  - In-App products (subscriptions and one-time)
- Android users not willing to pay for apps -> Ads
- Available in most countries

Android: Cons

- **Fragmentation**
  - OS adoption
  - Sizes
  - Performance differences
  - CPUs
- Low end devices
- Low quality apps
  - Wireless charger simulator



shikha nayak
★ ☆ ☆ ☆ ☆ May 27, 2021
123
No stars for this ridiculous app.... It is just an app to make April Fool and in real there is no wireless charging of your phone in emergency.... I just hate it

SimStopperTM
★ ☆ ☆ ☆ ☆ April 12, 2019
269
Absolute garbage! Waste of time! Doesn't do *anything* at all! My phone was at 3% and I had no charger so I downloaded the app. It didn't increase at all, it just kept driving. This is practically an April Fools app. Don't download it!

TendondoesYT
★ ★ ★ ★ ★ April 30, 2021
115
Got 100% in less than an hour... THANK YOU, other people just lie about it not charging the phone probaly because they use a phone from 1997 or something!!

KittyJ Gachagamer
★ ★ ★ ★ ★ November 12, 2020
83
AMAZING APP!! 😍✨😍✨It actually charges my phone like a miagician!!! Don't look at the 1 star rating cuz it's not true!! The reason why it's not working cuz u done it wrong, ur supposed to place it on top of a phone or any other device to charge!! I recommend!!! 😊😊😊😊

Paul Hanlin
★ ☆ ☆ ☆ ☆ February 13, 2021
232
I give this app no stars because it doesn't work like it say it does. I can't get it to do nothing but take up space on my phone and I should have listened to my friends when they not to down load this app

Android: Pros

- **Open source**
- **Many users**
- **Many apps**
- **Customisation (Keyboard, Launcher)**
- **Developer one-time fee $25 (Apple $99 Annual)**
  - Alpha, beta channels, stage rollout
  - Basic crash reporting
  - Pre-launch reports
  - Android Vitals
- Dev tools available on all major platforms (Linux, Windows, MacOS, ChromeOS)
- Huge open source community
  - OkHttp, Retrofit, Ktor, Dagger, Hilt, Koin, Flipper, RxJava
- Reference Pixel devices

Android: Security

- **Root user not available by default**
  - Rooting to get extra functionality, often breaks warranty
- **Permissions**
  - < API-23 – All permissions granted during installation
  - >= API-23 – Runtime permissions, user have to approve “dangerous” permissions
  - TargetSDK changes the behaviour
- **Bouncer**
  - Service which scans Google play store for mallicious apps

Mobile Development

Options (iOS, Android)

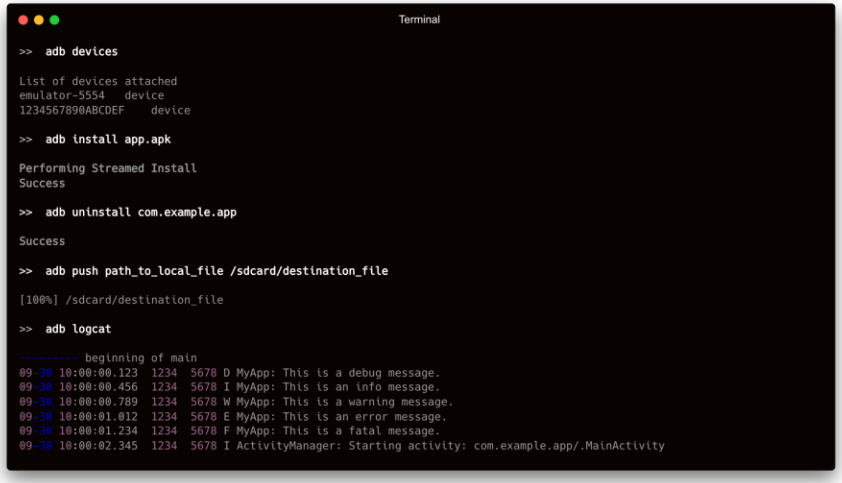


- **App-like mobile web**
- **Multiplatform frameworks: Flutter (Google), Xamarin (Microsoft), React native (Facebook)**
- **WebView based frameworks**
- **Kotlin Multiplatform**
- **Native:**
  - **IOS**:
    - *Swift*: The primary language for iOS development, announced by Apple in 2014.
    - *Objective-C*: Supported, but gradually being replaced by Swift.
    - *C/C++*: Mainly games and libraries.
  - **Android**:
    - *Kotlin*: The primary language for Android development, support announced on GoogleIO in 2017.
    - *Java:* Supported, not recommended anymore (i.e. Compose not working with Java)
    - *C/C++:* Mainly games and libraries.

Android Development

Development tools

- **Android Studio** (editor, emulators, logcat debugging and support tools)
- **SDK**
  - ADB
  - Lint
  - Emulator
  - aapt, aidl, dx, (Jack & Jill – Java 8, now deprecated)
  - D8
  - Proguard/R8
- **NDK** (C/C++ development)
  - Cross compile
  - Native libraries



- Robust build system
- Dependency management
- Compilation, packaging, testing
- Build scripts languages:
  - Kotlin DSL
  - Groovy

app/build.gradle.kts

```kotlin
plugins {
    id("com.android.application") // Android application plugin
}

android {
    compileSdk = 34 // Compile SDK version

    defaultConfig {
        applicationId = "com.example.myapp" // App package name
        minSdk = 21 // Minimum SDK version
        targetSdk = 34 // Target SDK version
        versionCode = 1 // App version code
        versionName = "1.0" // App version name
        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner" // Test runner
    }

    buildTypes {
        getByName("release") {
            isMinifyEnabled = false // Disable code shrinking
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro" // ProGuard rules
            )
        }
    }
}

dependencies {
    implementation("androidx.core:core-ktx:1.10.1") // AndroidX core with Kotlin extensions
    implementation("androidx.appcompat:appcompat:1.6.1") // AppCompat for backward compatibility
    implementation("com.google.android.material:material:1.9.0") // Material Design components
    implementation("androidx.constraintlayout:constraintlayout:2.1.4") // ConstraintLayout for UI

    testImplementation("junit:junit:4.13.2") // JUnit for unit testing
    androidTestImplementation("androidx.test.ext:junit:1.1.5") // Android JUnit extension
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1") // Espresso for UI testing
}
```

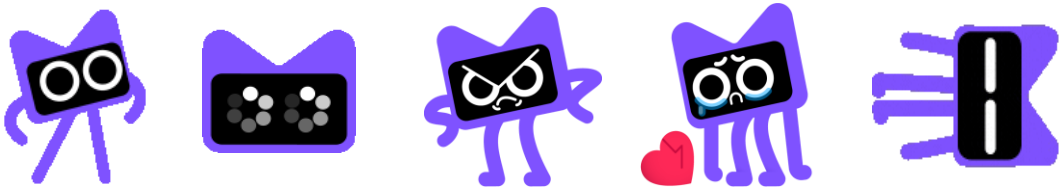
Kotlin

Ketchup

Island

Kotlin

- **Documentation**
- Open source (Apache 2.0 license)
- First released in 2016
- Developed by JetBrains
- Tools support: IntelliJ, Android Studio, standalone compiler, ...
- Interoperable with Java (legacy code)
- Coroutines for asynchronous programming - lightweight concurrency framework
- Garbage collection
- Concise, null-safety, typesafe, type inference, immutability (List vs. MutableList)
- Kotlin playground



Kotlin assets

Kotlin current version

Source

Kotlin Basics

- **Values and variables**
  - Statically typed
  - If no type is specified, Kotlin infers the type from the initial assignment.
- **Classes**
  - Primary constructors are directly declared in the class header (optional init block)
- **Objects** (Singleton class)
- **Functions:**
  - *Single-expression functions:* functions that omit the return keyword and braces. Body is specified after "=" symbol.
  - Function return type must be specified (unless returns Unit or it is a single-expression function)

Kotlin basics

```kotlin
var age = 25  // Mutable variable
age = 26      // Can be reasigned

val name = "Alice"  // Immutable variable
// name = "Bob"     // Error: Val cannot be reassigned

class Person(val name: String, var age: Int) {
    fun introduce() {
        println("My name is $name and I am $age years old.")
    }
}

val person = Person("Bob", 30)
person.introduce() // Output: My name is Bob and I am 30 years old.

object Singleton {
    fun showMessage(): String = "This is a singleton object."
}

println(Singleton.showMessage())  // Output: This is a singleton object.
```


Data classes

- Used mainly for data modeling
- Methods generated by compiler:
  - Equals() / hashCode()
  - toString()
  - Copy()
- Data objects (since Kotlin 1.9)

Data Classes

```
data class Person(
    val name: String,
    val surname: String,
    val street: String,
    val buldingNumber: Int
)

val bruceWayne = Person(
    name = "Bruce",
    surname = "Wayne",
    street = "Wayne Manor",
    buldingNumber = 1
)

val robin = bruceWayne.copy(name = "Robin")
```

Named, default parameters, String interpolation

- **Default parameters:** replace method overloading
- **String interpolation**:
  - For variables: $variableName
  - For expressions: ${expression}

Parameters

```
fun greetUser(name: String, age: Int = 30) {
    println("Name: $name, Age: $age") // String interpolation
}

fun main() {
    greetUser(name = "Bob")                // Output: Name: Bob, Age: 30
    greetUser(name = "Charlie", age = 25)  // Output: Name: Charlie, Age: 25
}
```

When expression

When expression

```
val x = 15
val y = 10

when { // As statement
    x > y -> println("x is greater than y")
    x == y -> println("x is equal to y")
    else -> println("x is less than y")
}

val day = 3

val dayName = when (day) { // As expression
    1 -> "Monday"
    2 -> "Tuesday"
    3 -> "Wednesday"
    4 -> "Thursday"
    5 -> "Friday"
    else -> "Weekend"
}
```

Operator overloading

- Arithmetic operators (+, -, *, /, ...)
- "in" operator (contains)
- Indexed access operator
- Equality/Inequality operator (equals)
- Comparison operators (compareTo)

Operator overloading

```
data class ComplexNumber(
   val real: Double,
   val imaginary: Double
) {
   operator fun plus(increment: ComplexNumber): ComplexNumber {
       return ComplexNumber(real + increment.real, imaginary + increment.imaginary)
   }
}
```

Gen

Extension functions

- Adds methods or variables to any class
- Doesn't have access to internal state of the class
- Powerful, but easy to misuse

Extension Functions

```
fun Int.isEven() = this % 2 == 0
fun Int.isOdd() = this % 2 == 1

val Int.absoluteValue: Int
    get() = if (this < 0) -this else this
```

Smartcast

- Automatically casts a variable to a specific type after checking its type with conditions

Smart Cast

```
fun smartCastDemo(x: Any) {
    when (x) {
        is Int -> println("Integer value abs(x): ${x.absoluteValue}")
        is String -> println("String x.toLowerCase: ${x.toLowerCase(Locale.getDefault())}")
        else -> println("x is not Integer or String")
    }
}
```

Lambda expressions

- Function can be passed as argument or returned from other function

Lambda expressions

```
fun <T, R> transform(item: T, transformation: (T) -> R): R {
    return transformation(item)
}

fun <T, R> T.transform(transformation: (T) -> R): R {
    return transformation(this)
}

val number = 1
number.transform { it.toString() }
transform(number) { it.toString() }
```

Companion object

- Object associated with class
- Accessed via class name
- Methods/properties do not belong to instance
- Similar to static in java
- Can implement/extend another interface/class
- Top level declaration vs. Companion object: article

Companion object

```
class User(val name: String) {
    companion object {
        fun createGuest() = User("Guest")
    }
}

fun main() {
    val guest = User.createGuest()
    println(guest.name)  // Output: Guest
}
```

Null safety

- **Null safety enforced at compile time**
  - It’s part of type system

Null safety

```
// Safe access operator
val length: Int? = nullableString?.length

// Elvis operator
val result: String = nullableString ?: "Default Value"

// Doesn't print anything if nullableString is null.
nullableString?.let { print("Hey, $it") }

// Throws NullPointerException if nullableString is null. Do not use!
nullableString!!.let { print ("Hey, $it") }
```

Kotlin Multiplatform (KMP)

- **Documentation:** https://kotlinlang.org/docs/multiplatform.html
- JetBrains
- Announced in 2017

- Allows to write shared business logic in Kotlin and use it across multiple platforms

- Built on Kotlin/Native
- Primary used for iOS and Android mobile development
- Kotlin Multiplatform Gradle plugin
- Multiplatform libraries

- More information in 2023 lesson about KMP: slides
- Example of KMP project from mdevcamp workshop (contains both version: 2 separate apps, and KMP variant): repository

Thank you

Lukáš Prokop
Simona Kurňavová

Lukas.Prokop@gendigital.com
Simona.Kurnavova@gendigital.com

Gen™

