



Android lecture 1

Android history, Kotlin

Agenda

- History of Android OS
- Development options
- Kotlin

Why mobile development

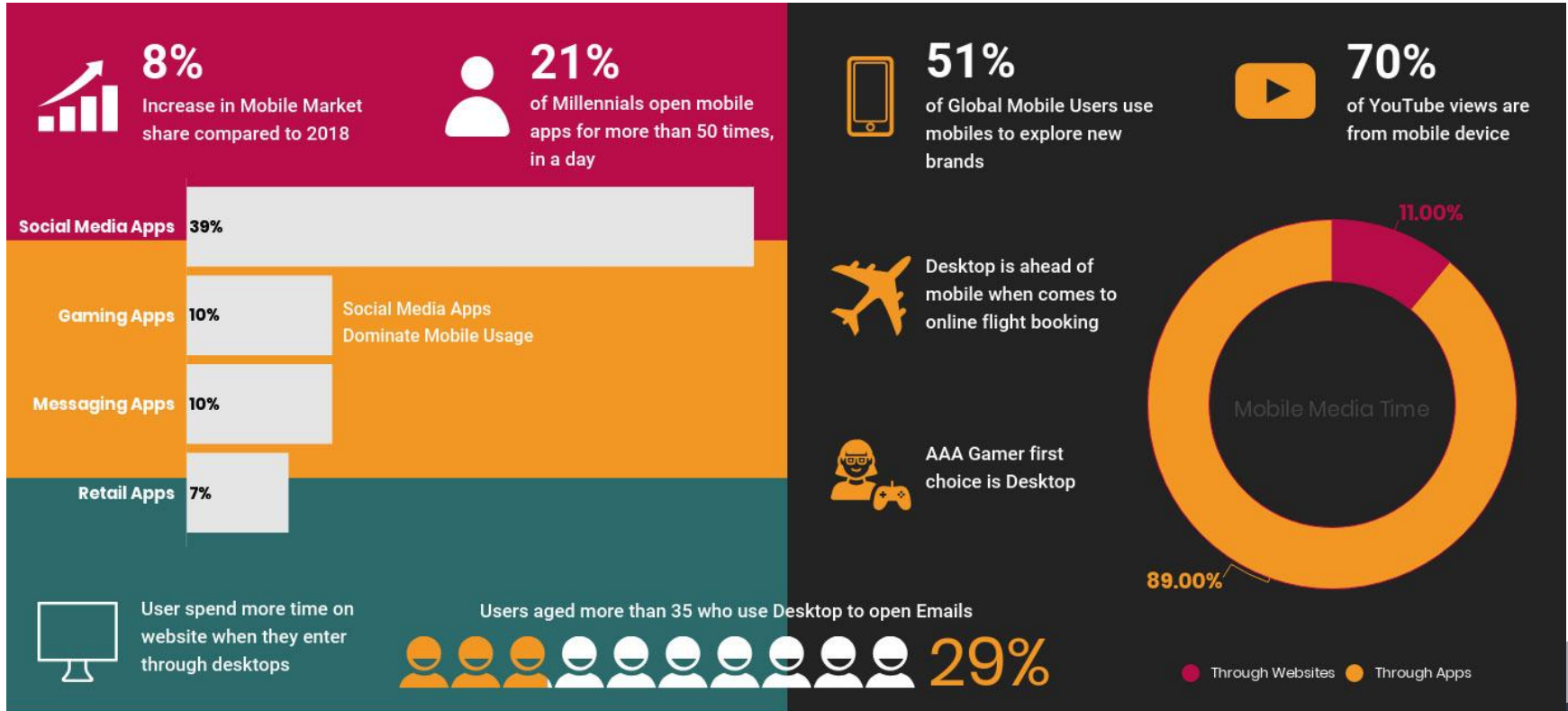
Why mobile matters

Ubiquitous

Quick

Convenient

Why mobile matters



Created By: LaptopDiscovery.com

<https://laptopdiscovery.com/mobile-vs-desktop-usage-statistics/>

Mobile devices specifics

- Not enough computation power
- Changing state - screen orientation
- Unstable network connection
- Small battery

History

Mobile devices history

- Psion, handhelds
- April 2000 - Pocket PC 2000 (Windows Mobile)
- October 2003 - Android Inc.
 - Intended as software for digital cameras
- July 2005 - Android Inc. acquired by Google
- November 5, 2007 - Open Handset Alliance
 - Consortium of technology companies (Google), manufacturers (HTC, Sony,...), carriers (Sprint, T-mobile,..) and chipset makers (Qualcomm, ...)
- June 29, 2007 - iPhone
- October 22, 2008 - T-mobile G1 (HTC)
- May 2009 - Windows Mobile 6.5
- November 8 2010 - Windows phone 7 release
- 2017 - Microsoft discontinued Windows 10 Mobile development



T-Mobile G1 (HTC Dream)



Nokia N95 (2007), iPhone 2G(2007), T-mobile G1(2008), HTC HD2(2009)

Android

- Linux based OS
 - No bash, no access to root (by default)
 - GNU C library (glibc) replaced by bionic
- Open source
 - <https://source.android.com/>
 - Just the OS, not the Play store or Google play services

Android versions

- 1.0 - 23 September 2008 (API 1)
- 1.1 - 9 February 2009 (API 2)
- 1.5 - Cupcake 27 April 2009 (API 3)
- 1.6 - Donut 26 September 2009 (API 4)
- 2.0 -2.1 - Eclair October 2009 (API 5 - 7)
- 2.2 - Froyo May 2010 (API 8)
- 2.3.x - Gingerbread December 2010 (API 9 - 10)

Android versions

- 3.x - Honeycomb February 2011 (API 11 - 13) - Tablet only
- 4.0.x - Ice cream sandwich October 2011 (API 14-15)
- 4.1 - 4.3 - Jelly Bean July 2012 (API 16 - 18)
- 4.4 - Kitkat October 2013 (API 19)
- 4.4W - Kitkat for wearables (API 20)
- 5.0 - 5.1 - Lollipop November 2014 (API 21 - 22)
 - Material design
 - ART runtime - Just In Time vs Ahead Of Time
- 6.0 - Marshmallow October 2015 (API 23)
 - Runtime permissions
 - Doze mode

Android versions

- 7.0 - Nougat (API 24) - August 2016
 - Multi window
 - Doze on the go
 - Vulkan API
 - Quick settings tiles
- 7.1 Nougat (API 25) - October, 2016
 - App shortcuts

Android versions

- 8.0 Oreo (API 26) August, 2017
 - Picture in picture
 - Notification channels
 - Custom fonts and downloadable fonts
 - Autosizing textview
 - Multi display support
 - Permissions
 - Fix granting permission to whole group
 - Some packages from java 8
 - java.time
 - java.nio.file
 - java.lang.invoke
 - Project treble
 - Sony, Nokia, OnePlus

Android versions

- 8.1 Oreo (API 27) December 5, 2017
 - Neural Network API
 - Programmatic safe browsing actions
 - Video thumbnail extractor
 - Wallpapers color API
 - Fingerprint updates
 - New error codes
 - Cryptography updates
 - Prefer Conscrypt over Bouncy castle

Android versions

- 9.0 Pie (API 28) August, 2018
 - Display cutout
 - Notifications - messaging
 - Multi camera support
 - ImageDecoder
 - Animation
 - GIF, WebP animated images
 - HDR VP9, HEIF and Media APIs

Android versions

- 10 (API 29) September, 2019
 - Gesture navigation
 - Smart replies
 - Dark theme
 - Foldable devices
 - More privacy control

Android versions

- 11 (API 30) September, 2020
 - Chat bubbles
 - Notification history
 - One time permission
 - Permission auto-reset
 - 5G detection API

Android today - versions

<https://androidx.tech/artifacts/ui/ui-saved-instance-state/0.1.0-dev14>

ANDROID PLATFORM VERSION		API LEVEL	CUMULATIVE DISTRIBUTION
4.0	Ice Cream Sandwich	15	
4.1	Jelly Bean	16	99.8%
4.2	Jelly Bean	17	99.2%
4.3	Jelly Bean	18	98.4%
4.4	KitKat	19	98.1%
5.0	Lollipop	21	94.1%
5.1	Lollipop	22	92.3%
6.0	Marshmallow	23	84.9%
7.0	Nougat	24	73.7%
7.1	Nougat	25	66.2%
8.0	Oreo	26	60.8%
8.1	Oreo	27	53.5%
			39.5%
9.0	Pie	28	
10.	Android 10	29	8.2%

<https://developer.android.com/about/dashboards/>

<https://www.bidouille.org/misc/androidcharts>

<https://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide>

Android

- Android - phones and tablets
- Wear OS - smartwatch
 - Extended notification center
- Android TV - televisions, consoles
 - Native TV app experience for IPTV
- Android auto - cars
 - Connecting phone with the car
- Android things
 - Platform for IoT
 - Development board Raspberry Pi 3B
- Google glass
 - Glass explorer program - retired
 - Now glass for enterprise - to be used in logistic, manufactures...

<https://www.google.com/glass/start/>

Android ecosystem

- Apps are available on Google Play store
- Paid and free apps
- Subscription
- In-app purchases
- Android users not like to pay for apps => a lot of advertisements
- Available in most countries

Android - cons

- Fragmentation, slow upgrades,
- Low-end devices
- Low-quality apps in Google play
 - Solar Charger Prank

User reviews



Felicia Nany

★★★★★ May 6, 2019



how do i charge my battery with this solar???????????? it didn't work for me when i put it in sun. my battery was 51% but it has now reduced to 49%. why???



Suzie Iyals

★★★★★ May 10, 2019



This app is so stupid... All it does is to charge itself and waste your battery. I regret ever installing it... ☹️❌❌



Philip De Mutisoh

★★★★★ October 3, 2018



Fake app.....pretending to charge but not charging anything



A Google user

★★★★★ January 12, 2019



Please the panel has no impact on the battery



A Google user

★★★★★ October 2, 2018



Very very bbed app Very very bad don't download it



A Google user

★★★★★ December 15, 2018



It sucks even in darkness it shows that it's charging wen it's not

Android - pros

- Open source
- Lot of users
- Lot of apps
- Freedom (keyboard, launcher, ...)
- Developers one time fee 25\$ (Apple 99USD annual)
 - Alpha, beta channels, staged rollout
 - Basic crash reporting
 - Pre-launch reports
 - Android vitals
- Dev tools available for Linux, Windows and Mac
- A lot of OSS libraries
 - [Retrofit](#), [OkHttp](#), [Dagger](#), [Flipper](#), [RxJava](#)
- Nexus/Pixel devices

Android - security

- Root user is not available by default
 - Rooting devices to get more features, often void warranty
- Permissions
 - Until API-23 before installation “all or nothing”
 - Since API-23 permissions are requested at runtime
 - Empty data or crash, depends on TargetSDK
- Bouncer
 - Service that scan google play store for malicious apps

Development

Development options

- App-like mobile web
- Other language frameworks ([Xamarin](#), [flutter.io](#))
- WebView based frameworks ([PhoneGap](#))
- Native
 - Kotlin - officially announced support for android development on Google IO 2017
 - Java
 - C/C++ (mainly games and libraries)

Development tools

- Android studio
- SDK
 - ADB
 - Lint, UiAutomatorViewer
 - Emulator
 - Aapt, aidl, dx, (jack & jill - java 8, now deprecated)
 - D8
 - Proguard/R8
- NDK (C/C++ development)
 - Cross compile
 - Native libraries
- Gradle
 - Build system
 - Use Groovy (kotlin) as DSL
 - Extensible

Kotlin

Ketchup



Island



(c) VK.com/Piloterrr

Kotlin



<https://kotlinlang.org/>

History

- Developed by JetBrains
 - No other language except scala did not have features, they needed
- July 2011 unveiled Project Kotlin
- February 2012 open sourced
- February 2016 Kotlin v1.0
- Google IO 2017 first-class support on Android
- November 2017 Kotlin v1.2
- September 2018 1.3-RC
- April 2020 1.3.72
- September 2020 1.4.0

Why Kotlin

- Concise
 - Reduce boilerplate code
- Safe
 - Avoid NullPointerExceptions
- Interoperable
 - Works well with legacy java code
- Tool-friendly
 - Support in IDEs (IntelliJ IDEA, Android Studio, Eclipse, Netbeans, standalone compiler)

Kotlin - concise

- Data classes
- Named parameters
- Operator overloading
- Smart casting
- Default parameter values
- Extension functions
- Lambda expressions
- String interpolation

Data classes

- Methods generated by compiler
 - equals()/hashCode()
 - toString()
 - copy()
 - Only for fields defined in primary constructor

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

```
val boris = Person(  
    name = "Boris",  
    surname = "Johnson",  
    street = "Downing Street",  
    buildingNumber = 10  
)
```

```
val borris = boris.copy(name = "borris")
```

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)  
    }  
}
```

Smartcast

```
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")  
    }  
}
```

Kotlin - safe

- Null safety
- Immutable by default
 - List vs. MutableList, Map vs. MutableMap

Kotlin - interoperability

- Java
- Swift
- Kotlin multiplatform



Thank you Q&A

Feedback is appreciated

prokop@avast.com

Please use [mff-android] in subject