

Entwicklung mobiler Applikationen

WS2022/2023

Jan Brodhaecker | Oktober 2022

Agenda

- Vorstellungsrunde
- Organisatorisches
 - Email-Verteiler / Moodle
 - Prüfungsleistung
- Fahrplan
- Einleitung

Vorstellungsrunde

- **Jan Brodhaecker, 31 Jahre alt, Ilvesheim**

- Berufserfahrung

- 10/2011 – 10/2014 Duales Studium @ IBM, Mainz/Mannheim
 - 11/2014 – 10/2019 (Senior) Development Engineer @ sovanta AG, Heidelberg
 - 11/2019 – 08/2022 Senior Cloud Engineer @ SAP SE, St. Leon-Rot
 - 09/2022 – heute Development Expert @ SAP SE, Walldorf

- Ausbildung

- 2011 – 2014 Bachelor of Science @ DHBW Mannheim
 - 2016 – 2019 Master of Science @ DHBW CAS Heilbronn

Vorstellungsrunde

- **Kontakt** (bevorzugt auch für Rückfragen bzgl. der Vorlesung)
 - jan.brodhaecker@googlemail.com
- **LinkedIn**
 - www.linkedin.com/in/jan-brodhaecker-1709
- **Xing**
 - www.xing.com/profile/Jan_Brodhaecker/cv
- **Studienarbeiten oder Bachelor-Arbeiten?** Schlagen Sie gerne Themen vor!

Vorstellungsrunde

- **Wer sind Sie?**
 - Name, Firma, Tätigkeit ...
 - Optional: besondere Anforderungen an die Vorlesung

Organisatorisches

- Email-Verteiler
 - tinf21ai1@lists.posteo.de
- Folien per Moodle? E-Mail?

Organisatorisches

Option A

- **60% Klausur (60 Minuten)**
- **40% Präsentation ***
 - Gruppe von 2,3 Studierenden
 - freie Themenwahl (relevante für Themen der Vorlesung, bspw. Wearables, Automotive, Firebase, ...)
 - Implementierungsbeispiel(e)
 - Handout (max. 1 DIN A4 Seite)
 - 20 - 25 Minuten

Option B

- **60% Programmierentwurf - Entwicklung einer App ***
 - Gruppe von 3,4 Studierenden
 - freie Themen- und Technologiewahl
 - gemeinsam erarbeiteter Kriterienkatalog**
 - Einzelbewertung, keine Gruppenbewertung (Verteilung der Themen innerhalb der Gruppe)
 - Zeit während der Vorlesung für Vorbereitung etc.
- **40% Präsentation**
 - in den o.g. Gruppen
 - Vorstellung der App-Idee
 - Vorstellung eines prägnanten Problems während der Entwicklung (bspw. Anbindung an API, OAuth, ...)
 - 20 - 25 Minuten

* Themen sollten im Vorfeld abgestimmt sein

** für beide Kurse identisch

Fahrplan

Einleitung / Motivation



Einleitung

**Hat schon Jemand Erfahrung mit der Entwicklung mobiler Applikationen?
Hat schon jemand eine App entwickelt (privat/beruflich)?**

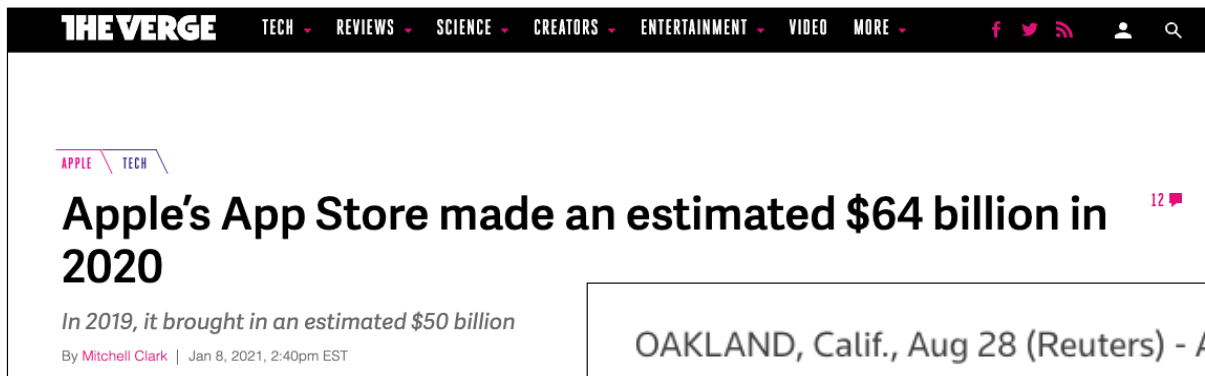
Einleitung

Programmiersprachen ... ?
privat/beruflich/DHBW

Einleitung

Warum beschäftigen wir uns eigentlich mit dem Thema?

<https://www.theverge.com/2021/1/8/22220873/apple-2020-app-store-revenue-60-billion-dollars>



THE VERGE TECH REVIEWS SCIENCE CREATORS ENTERTAINMENT VIDEO MORE f t r u q

APPLE TECH

Apple's App Store made an estimated \$64 billion in 2020 12

In 2019, it brought in an estimated \$50 billion

By Mitchell Clark | Jan 8, 2021, 2:40pm EST

OAKLAND, Calif., Aug 28 (Reuters) - Alphabet Inc's (GOOGL.O) Google generated \$11.2 billion in revenue from its mobile app store in 2019, according to a court filing unsealed on Saturday, offering a clear view into the service's financial results for the first time.

<https://www.reuters.com/technology/google-play-app-store-revenue-reached-112-bln-2019-lawsuit-says-2021-08-28/>

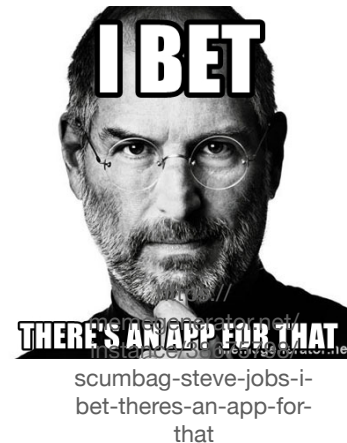
Einleitung

Warum beschäftigen wir uns eigentlich mit dem Thema?

Key Mobile App Statistics

- Mobile apps are expected to generate over **\$935 billion** in revenue by 2023.
- The Apple App Store has **1.96 million apps** available for download.
- There are 2.87 million apps available for download on the Google Play Store.
- **21% of Millennials** open an app 50+ times per day.
- 49% of people open an app **11+ times each day**.
- 69% of all **US digital media time** comes from mobile apps.
- The **average smartphone owner** uses 10 apps per day and 30 apps each month.

<https://buildfire.com/app-statistics/>



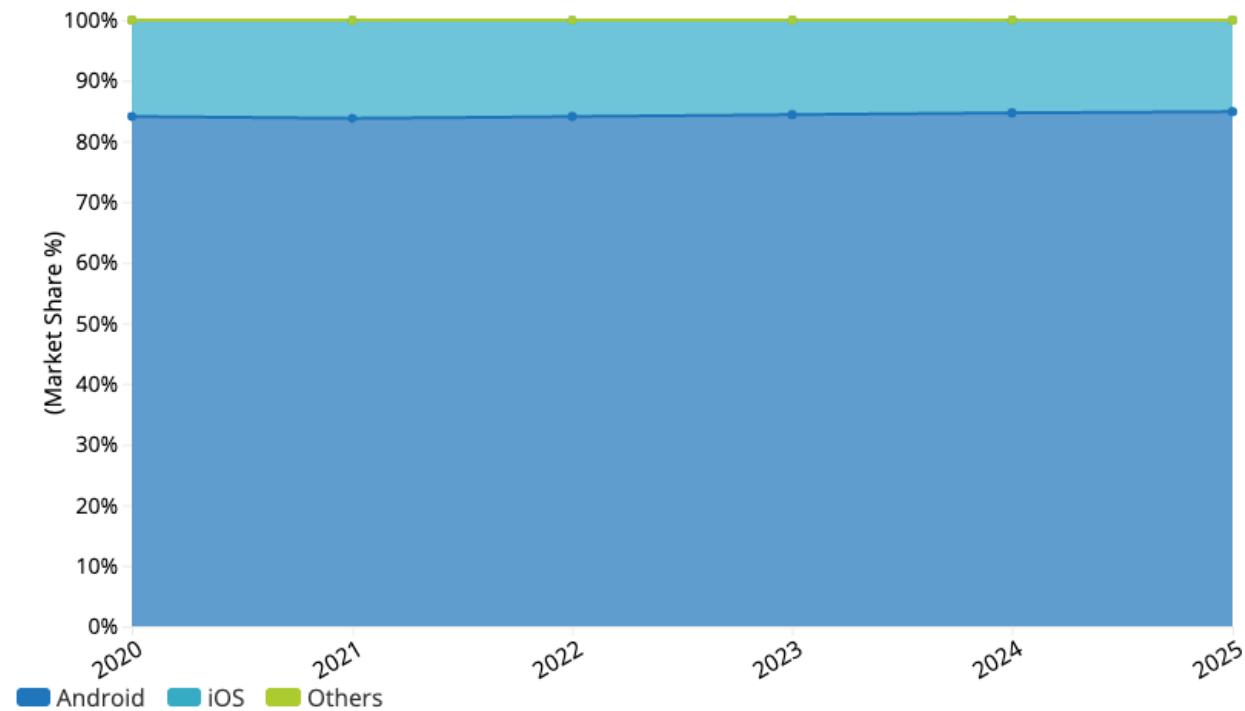
Einleitung

- Was sind eigentlich mobile Applikationen?
 - Software Applikationen, die auf einem mobilen Gerät (Smartphone, Tablet, etc.) ausgeführt werden
 - Dabei wird häufig eine Netzwerkverbindung verwendet um Rechenlast auf entfernte Systeme auszulagern (bspw. Verwendung von Webservices)
 - Aktuell zwei „Big Player“
 - iOS, Apple
 - Android, Google

<https://aws.amazon.com/mobile/mobile-application-development/>

Einleitung

Worldwide Smartphone Shipment OS Market Share Forecast



<https://www.idc.com/promo/smartphone-market-share>

Einleitung

Native Applications

Native mobile applications are written in the programming language and frameworks provided by the platform owner and running directly on the operating system of the device such as **iOS** and **Android**.

Cross-Platform Applications

Cross-platform native mobile applications can be written in variety of different programming languages and frameworks, but they are compiled into a native application running directly on the operating system of the device.

Hybrid-Web Applications

Hybrid mobile applications are built with standard web technologies - such as JavaScript, CSS, and HTML5 - and they are bundled as app installation packages. Contrary to the native apps, hybrid apps work on a 'web container' which provides a browser runtime and a bridge for native device APIs via Apache Cordova.

Progressive Web Applications

PWAs offer an alternative approach to traditional mobile app development by skipping app store delivery and app installations. PWAs are web applications that utilize a set of browser capabilities - such as working offline, running a background process, and adding a link to the device home screen - to provide an 'app like' user experience.

<https://aws.amazon.com/mobile/mobile-application-development/>

Einleitung



android

<https://bit.ly/2FcMB6l>

Einleitung

- Android
 - mobiles Betriebssystem für ca. 2,5 Milliarden Geräte
 - verfügbar seit 21. Oktober 2008
 - entwickelt von der Open Handset Alliance Konsortium
 - ... welche 2007 von Google ins Leben gerufen wurde
 - basiert auf einem Linux-Kernel
 - verfügbar in Open-Source
 - ohne Google-Play Services



Einleitung



- Android – einfache und günstige Möglichkeit (IoT) Projekte zu realisieren (Security-Cam, GPS-Tracker, Bitcoin-Miner, ...)

€ 27,00 bis € 69,99 (06.10.2022)

[Info beim Hersteller](#)

Aktueller Preisbereich

€ 23,00 bis € 59,00

Preisentwicklung 1W 1M 3M 6M 1J



[Preisentwicklung öffnen](#)

[Preisalarm setzen](#)

[Zur Wunschliste hinzufügen](#)

[Zur Vergleichsliste hinzufügen](#)

[Feedback senden](#)

via Amazon Partnerprogramm



Motorola Fire XT316

★★★★★ 4.0 / 1 Bewertung

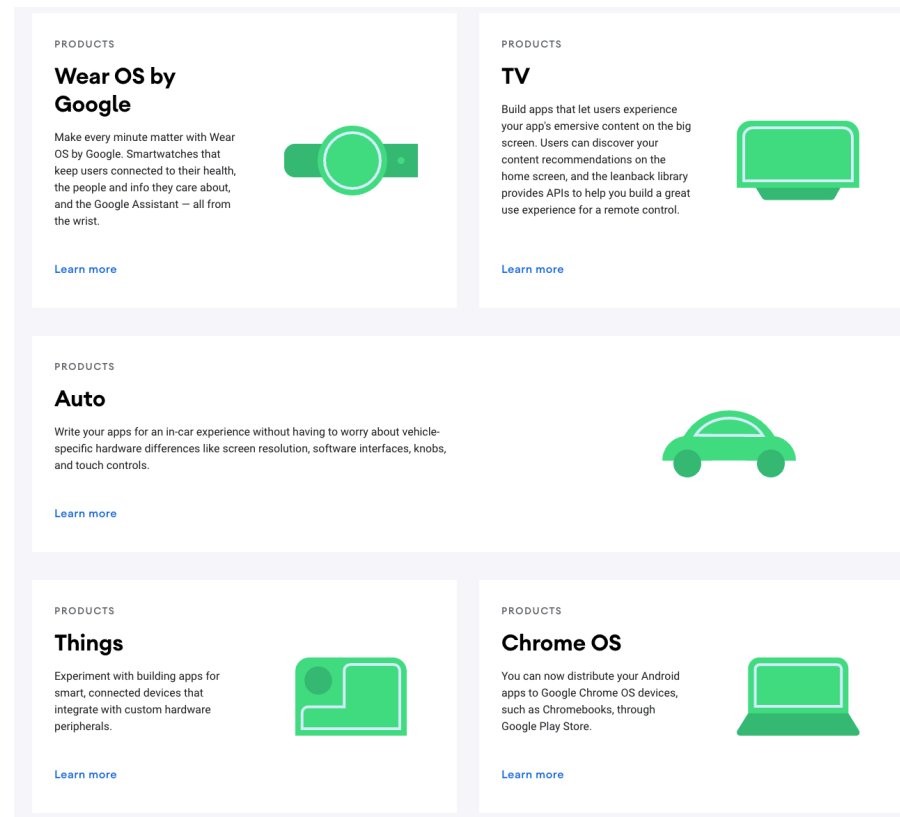
Betriebssystem	Android 2.3
Display	2.8", 320x240 Pixel, 262.144 Farben, kapazitiver Touchscreen
Kamera hinten	3.0MP
Kamera vorne	nein
Schnittstellen	Micro-USB-B 2.0, 3.5mm-Klinke, WLAN 802.11b/g/n, Bluetooth 2.1
Sensoren	Beschleunigungssensor, Annäherungssensor, Lichtsensor, Kompass
CPU	600MHz
RAM	256MB
Speicher	150MB, microSD-Slot (dediziert, bis 32GB)
Navigation	A-GPS
Modem	GSM (0.2Mbps/0.1Mbps), UMTS
Frequenzbänder	2G (850/900/1800/1900)

▼ Alle Produkteigenschaften anzeigen

<https://geizhals.de/motorola-fire-xt316-a660929.html?hloc=at&hloc=de>

Einleitung

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



Einleitung

13. November 2012

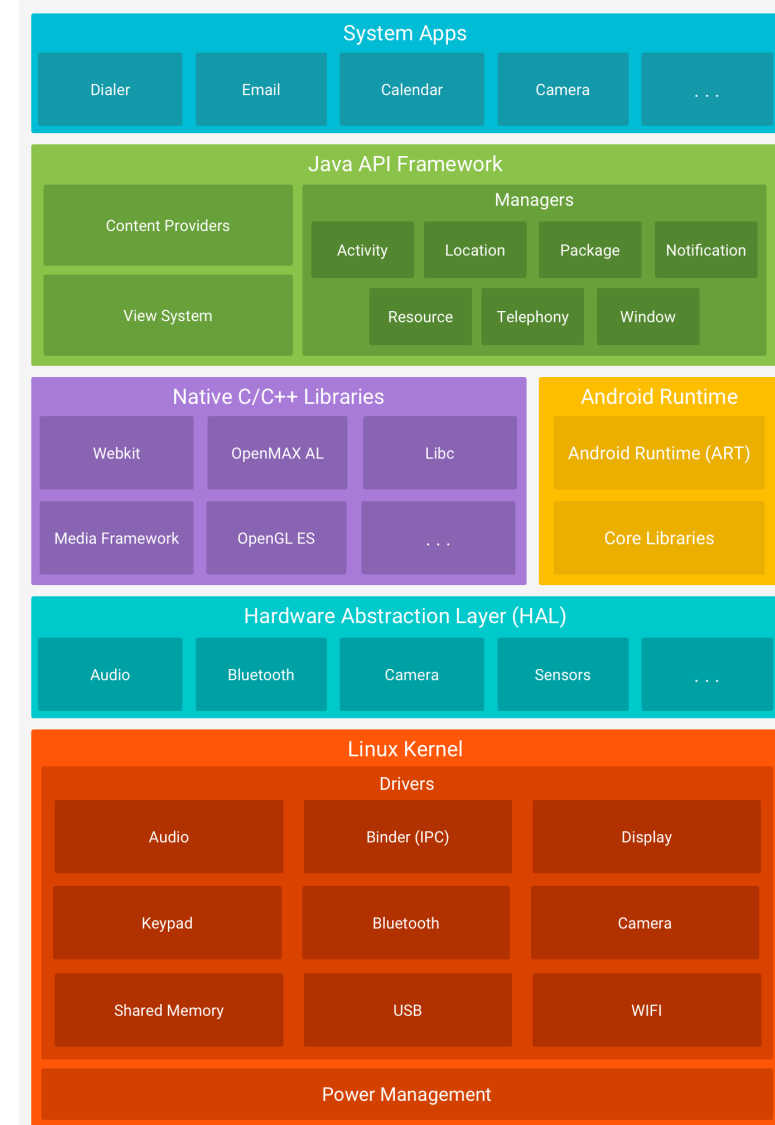
ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.1 Jelly Bean	16	
4.2 Jelly Bean	17	99,9%
4.3 Jelly Bean	18	99,7%
4.4 KitKat	19	99,7%
5.0 Lollipop	21	98,8%
5.1 Lollipop	22	98,4%
6.0 Marshmallow	23	96,2%
7.0 Nougat	24	92,7%
7.1 Nougat	25	90,4%
8.0 Oreo	26	88,2%
8.1 Oreo	27	85,2%
9.0 Pie	28	77,3%
10. Q	29	62,8%
11. R	30	40,5%
12. S	31	13,5%

3. September 2019

* Android Studio

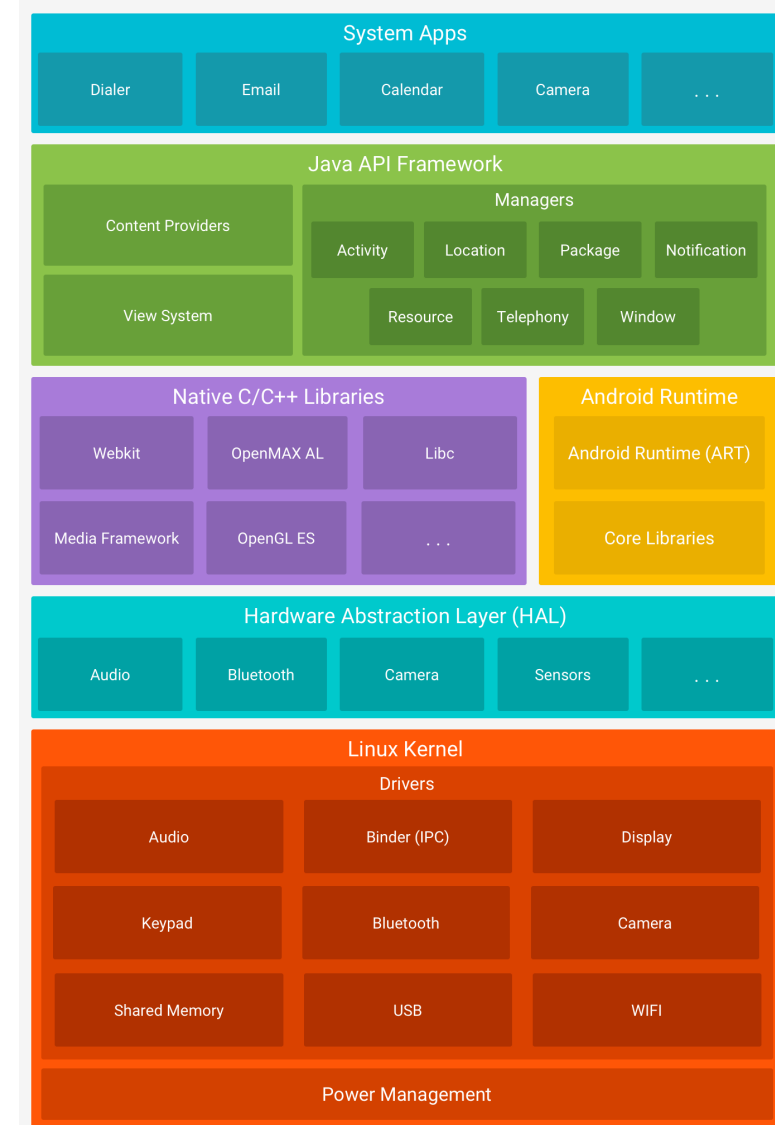
Einleitung

- Linux Kernel
 - Grundlage, Threading, Low-Level Memory Management, ...
- HAL (Hardware Abstraction Layer)
 - ermöglicht Hardware Zugriffe (Bluetooth, Kamera, ...)
- Android Runtime
 - jede App läuft in einem eigenen Prozess, in eigener Instanz der ART, führt mit wenig Speicher DEX Dateien aus (spezielles Android Bytecode Format, d8: Java Bytecode -> DEX Bytecode)



Einleitung

- Native C/C++ Libraries
 - ART, HAL benötigen native Bibliotheken in C/C++, teilw. können APIs auch über Java APIs angesprochen werden (bspw.: OpenGL API)
 - Android NDK erlaubt Teile einer App in nativem Code (C/C++) zu entwickeln
- Java API Framework
 - Activity Manager, View System, Content-Providers, ...
- System Apps
 - SMS, Calendar, Contacts, ...



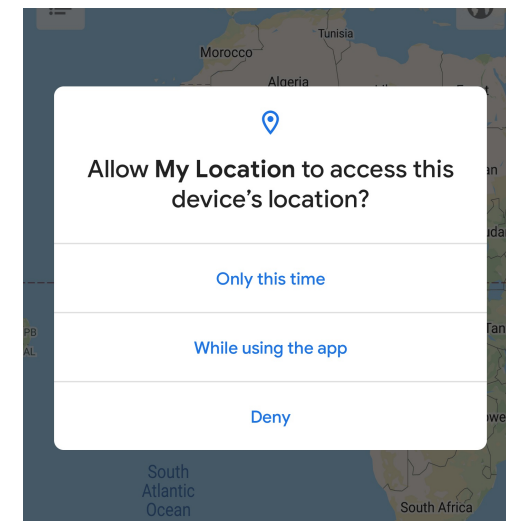
Einleitung

- Android App Fundamentals
 - <https://developer.android.com/guide/components/fundamentals>
- Android Apps können Kotlin, Java und C++ entwickelt werden und mit Hilfe des Android SDK in eine APK kompiliert werden
- Jede Android App lebt in ihrer eigenen Sandbox, die durch die folgenden Sicherheitseigenschaften geschützt wird:
 - Android ist ein multi-user Linux system, wobei App gleich unterschiedlicher User
 - Jede App hat eine ein-eindeutige Linux User ID (verhindert Zugriff durch andere User)
 - Jeder Prozess läuft in einer eigenen VM (isoliert)
 - Jede App läuft in eigenem Linux Prozess

<https://developer.android.com/guide/platform>

Einleitung

- Android App Fundamentals
 - <https://developer.android.com/guide/components/fundamentals>
 - Zugriffe auf Daten oder Funktionen (Kamera, GPS, etc.) werden über sogenannte *Permissions* geregelt
 - Manifest deklariert Komponenten und benötigte Gerätefunktionalität



<https://www.androidauthority.com/android-11-location-access-1085357/>

Einleitung

- Android App Fundamentals
 - <https://developer.android.com/guide/components/fundamentals>
- App components
 - Activities Was befindet sich aktuell auf dem Bildschirm (main()-Methode)
 - Services läuft (länger) im Hintergrund (bspw. Musik, Downloads, ...)
 - Broadcast Receivers empfängt system-weite Events (bspw. Batterie leer, WLAN gefunden)
 - Content Providers geteilte (System-)App-Daten (bspw. Kontakte, Kalender, ...)

Einleitung

- Android App Fundamentals
 - <https://developer.android.com/guide/components/fundamentals>
- Intents = asynchrone Nachrichten, um Activities, Services oder Broadcast Receiver zu aktivieren
 - können **implizit** (Ziel ist unbekannt) oder **explizit** (Ziel ist bekannt) sein

Einleitung

- App components
 - Activities
 - Services
 - Broadcast Receivers
 - Content Providers

Setup

- Bitte installieren Sie auf ihrer Maschine ...
 - JDK & Android Studio



AVD
Manager

SDK
Manager

- Installieren Sie ein SDK (.. 10, 11, 12 ...)
- Richten Sie sich einen Emulator ein
- Erzeugen Sie ein Hello-World-Projekt

SDK Platforms		SDK Tools	SDK Update Sites
Each Android SDK Platform package includes the Android platform API level by default. Once installed, the IDE will automatically check for package details" to display individual SDK components.			
Name	API Level		
<input type="checkbox"/> Android 12 Preview (S)	31		
<input checked="" type="checkbox"/> Android 11.0 (R)	30		
<input type="checkbox"/> Android 10.0 (Q)	29		
<input type="checkbox"/> Android 9.0 (Pie)	28		
<input type="checkbox"/> Android 8.1 (Oreo)	27		

Setup

- Neues Projekt erzeugen
 - Android Studio öffnen
 - Datei
 - Neues Projekt
 - Telefon & Tablet
 - Leere Activity
- Starten
 - Emulator auswählen -> Run / Debug

