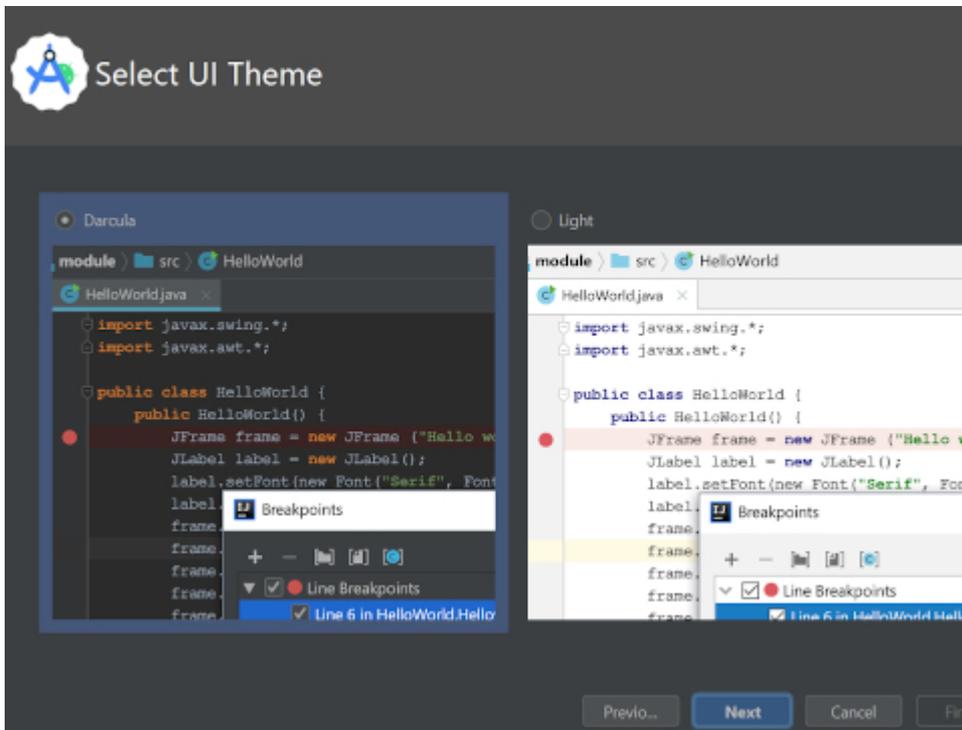


# Empty Activity oder: Unter IT-Evangelisten



Ein indischer Arbeitskollege, der *Software Engineer* und Mathematiker ist, schickte mir ein [Instagram-Video](#) eingedenk dessen, dass hier (wo ich arbeite) die meisten wissen, dass die Programmierer hier alle meine Kinder, wenn nicht sogar meine Enkelkinder sein könnten, nähme man nur das Alter. Das inspirierte mich. Ich sollte das, was die Dame im Filmchen kann, auch können sollte oder es zumindest versuchen. Ich bin ja noch viel jünger. Warum eigentlich nicht?

Frage also an das Publikum: Welche App braucht man als alter weißer Mann, die es noch nicht gibt?



programming an app for android, kotlin, java, -no people -chaos 100 -s 750

Da ich bei Apps absolut ahnungslos bin, fragte ich natürlich zuerst ~~meinen Dienstreboter~~ die künstliche Intelligenz: Wie programmiert man eine App für Android?

**ChatGPT:** Das Entwickeln einer Android-App erfordert grundlegende Programmierkenntnisse, insbesondere in [Java](#) oder [Kotlin](#) (Ach was! So etwas sollte die KI auch können!), da dies die Hauptprogrammiersprachen für die Android-Entwicklung sind. Hier ist eine grundlegende Schritt-für-Schritt-Anleitung, wie man eine Android-App entwickelt:

### **Vorbereitung:**

Installiere [Android Studio](#): Dies ist die offizielle integrierte Entwicklungsumgebung (IDE) von Google für die Android-Entwicklung.

Lade die erforderlichen [SDKs](#) und Tools über den Android Studio SDK Manager herunter.

### **Projekt erstellen:**

Starte Android Studio und erstelle ein neues Projekt. Wähle den Projekttyp (z. B. „Phone and Tablet“, „Empty Activity“) und gib den Projektnamen sowie andere Einstellungen an.

### **Design der Benutzeroberfläche:**

Verwende den integrierten Layout-Editor, um das UI deiner App zu entwerfen.

Definiere XML-Layouts für verschiedene Bildschirmgrößen und -orientierungen.

### **Programmierung:**

Verwende Java oder Kotlin, um die Logik deiner App zu programmieren.

Definiere Aktivitäten, die verschiedene Bildschirme und Interaktionen repräsentieren.

Implementiere [UI-Elemente](#) und verknüpfe sie mit Code.



### **Funktionalität hinzufügen:**

Füge Funktionen wie Buttons, Menüs, Texteingaben, Bilder usw. hinzu.

Implementiere die gewünschte App-Logik, wie Datenverarbeitung, Berechnungen usw.

### **Ressourcen verwalten:**

Füge Ressourcen wie Bilder, Zeichenketten und Farben hinzu.

Organisiere diese Ressourcen in den entsprechenden Verzeichnissen im Ressourcenordner.

### **Testing:**

Verwende den Android Emulator oder teste die App auf einem physischen Gerät.

Führe umfassende Tests durch, um sicherzustellen, dass deine App einwandfrei funktioniert.

### **Optimierung:**

Achte auf die Leistung deiner App und optimiere bei Bedarf den Code und die Ressourcennutzung.

### **Veröffentlichung:**

Registriere dich für ein Google Play Developer-Konto.

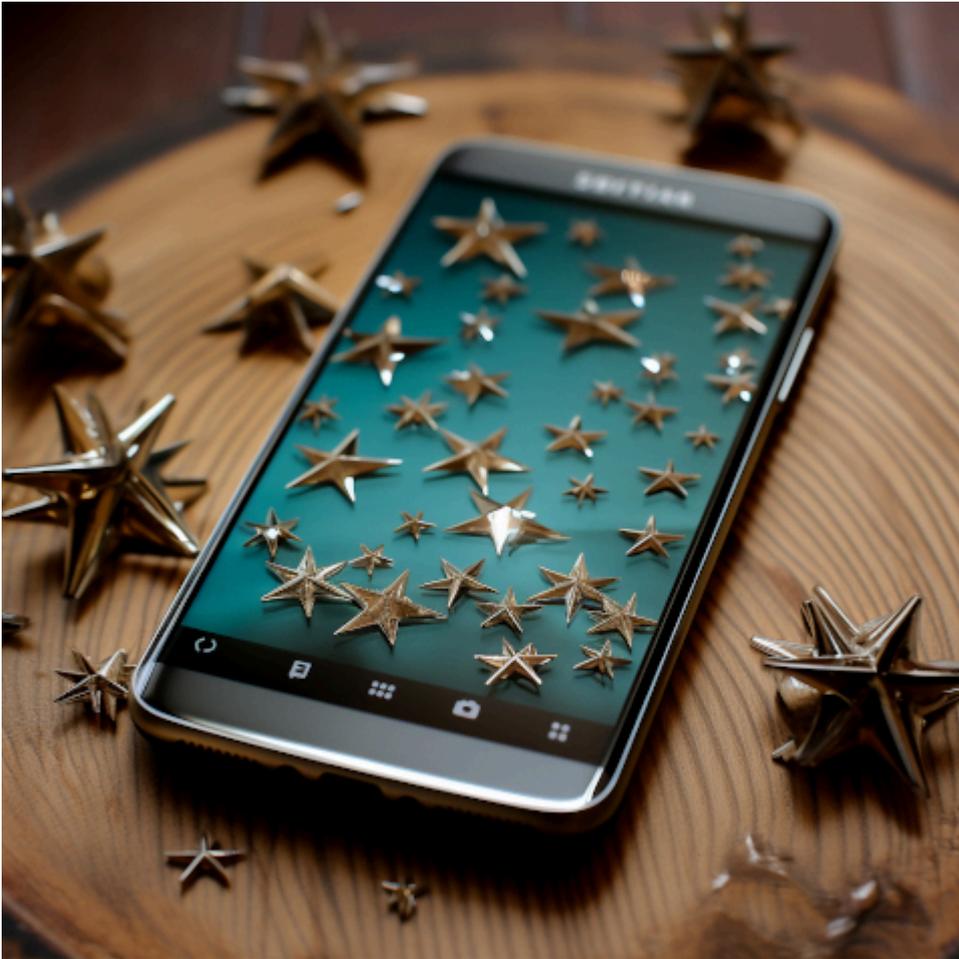
Generiere ein signiertes APK (Android Package) deiner App.

Veröffentliche deine App auf dem Google Play Store.

### **Aktualisierung und Wartung:**

Behalte deine App im Auge, sammle Feedback von Nutzern und füge neue Funktionen hinzu.

Halte die Android-Plattform-Updates im Blick und aktualisiere deine App entsprechend.



Ich konnte mich nicht zurückhalten und fragte nach: Gib ein Beispiel für einen Code in Kotlin, um eine App für Android zu programmieren! Auch da bekam ich eine Antwort ([Textdatei](#)). Hehe! Soll ich jetzt damit anfangen? Was ist mit einer Marktanalyse? Sollte ich schon Gelder für mein Startup einsammeln?

Und ab sofort nennt ihr mich [Herr IT-Evangelist!](#)