Android Hello World - Complete Setup Guide

Prerequisites Installation

Step 1: Install Android Studio

1. Download Android Studio from https://developer.android.com/studio
2. Run the installer and follow the setup wizard
3. When prompted, choose "Standard" installation
4. This will install Android SDK, Android SDK Platform, and Android Virtual Device

Step 2: Configure Android Studio (First Launch)

1. Launch Android Studio
2. Complete the initial setup wizard
3. Wait for all SDK components to download (this can take 15-30 minutes)

Project Setup

Step 3: Create New Project

1. Open Android Studio
2. Click "New Project"
3. Select "Empty Activity" (with Compose)
4. Configure your project:
    - **Name**: HelloWorld
    - **Package name**: com.example.helloworld
    - **Save location**: Choose a folder
    - **Language**: Kotlin
    - **Minimum SDK**: API 24 (Android 7.0)
    - **Build configuration language**: Kotlin DSL
5. Click "Finish"

Step 4: Project Structure

Android Studio will create a project structure. You need to replace/create these files:

```
HelloWorld/
├── app/
│   ├── src/
│   │   └── main/
│   │       ├── java/com/example/helloworld/
│   │       │   ├── MainActivity.kt
│   │       │   └── ui/theme/
│   │       │       ├── Color.kt
│   │       │       ├── Theme.kt
│   │       │       └── Type.kt
│   │       ├── res/
│   │       │   ├── values/
│   │       │   │   ├── strings.xml
│   │       │   │   └── themes.xml
│   │       │   └── (drawable, mipmap folders auto-generated)
│   │       └── AndroidManifest.xml
│   └── build.gradle.kts
├── build.gradle.kts (Project level)
└── settings.gradle.kts
```

Step 5: Copy the Code Files

Copy the code from the artifacts I provided into these exact locations:

1. **MainActivity.kt** → `app/src/main/java/com/example/helloworld/MainActivity.kt`
2. **Color.kt** → `app/src/main/java/com/example/helloworld/ui/theme/Color.kt`
3. **Theme.kt** → `app/src/main/java/com/example/helloworld/ui/theme/Theme.kt`
4. **Type.kt** → `app/src/main/java/com/example/helloworld/ui/theme/Type.kt`
5. **AndroidManifest.xml** → `app/src/main/AndroidManifest.xml`
6. **build.gradle.kts (app)** → `app/build.gradle.kts`
7. **build.gradle.kts (project)** → `build.gradle.kts` (root level)
8. **settings.gradle.kts** → `settings.gradle.kts` (root level)
9. **strings.xml** → `app/src/main/res/values/strings.xml`
10. **themes.xml** → `app/src/main/res/values/themes.xml`

Step 6: Sync Gradle

1. After adding all files, click "Sync Now" in the banner that appears at the top

2. Or click: File → Sync Project with Gradle Files
3. Wait for sync to complete (may take a few minutes the first time)

Testing the App

Step 7: Run on Emulator

1. Click "Device Manager" in the right sidebar
2. Click "Create Virtual Device"
3. Select a phone (e.g., "Pixel 6")
4. Select a system image (e.g., "Tiramisu" - API 33), click "Download" if needed
5. Click "Next" then "Finish"
6. Click the green "Play" button (▶) in the toolbar
7. Select your virtual device
8. Wait for the emulator to start and the app to install
9. You should see "Hello Android!" displayed in the center

Step 8: Run on Physical Device (Optional)

1. Enable Developer Options on your phone:
    - Go to Settings → About Phone
    - Tap "Build Number" 7 times
2. Enable USB Debugging:
    - Settings → Developer Options → USB Debugging
3. Connect phone via USB
4. Click the green "Play" button
5. Select your physical device
6. If prompted on phone, allow USB debugging

Building for Release

Step 9: Generate Signed APK

1. Go to: Build → Generate Signed Bundle / APK
2. Select "APK" → Click "Next"

3. Click "Create new..." to create a keystore:
    - **Key store path**: Choose a secure location (SAVE THIS!)
    - **Password**: Create a strong password (REMEMBER THIS!)
    - **Alias**: "helloworld"
    - **Alias password**: Create a password
    - **Validity**: 25 years minimum
    - **Certificate info**: Fill in your details
4. Click "OK" then "Next"
5. Select "release" build variant
6. Check both signature versions (V1 and V2)
7. Click "Finish"
8. APK will be created in: `app/release/app-release.apk`

Step 10: Install APK on Device

1. Transfer the APK to your Android device
2. Open it on the device
3. If prompted, allow installation from unknown sources
4. Install and run

Deploying to Google Play Store

Step 11: Create Google Play Developer Account

1. Go to https://play.google.com/console
2. Sign up ($25 one-time fee)
3. Complete account verification

Step 12: Prepare App for Store

1. Create high-quality app screenshots (at least 2)
2. Design an app icon (512x512 PNG)
3. Write app description
4. Create a feature graphic (1024x500)

5. Set content rating via questionnaire

Step 13: Upload to Play Console

1. Click "Create app" in Play Console
2. Fill in app details
3. Go to "Release" → "Production"
4. Click "Create new release"
5. Upload your signed APK (or better: AAB bundle)
6. Complete store listing with descriptions and graphics
7. Set pricing (free or paid)
8. Submit for review

Step 14: Generate AAB (Recommended for Play Store)

Instead of APK, Google prefers Android App Bundle (AAB):

1. Build → Generate Signed Bundle / APK
2. Select "Android App Bundle"
3. Use same keystore as before
4. Upload the `.aab` file to Play Console

Important Notes

- **Keep your keystore file safe!** You cannot update your app without it
- **Remember your passwords!** Store them securely
- First app review can take 1-3 days
- Updates typically review faster
- Minimum SDK 24 covers ~95% of devices

Troubleshooting

Gradle sync fails: Check your internet connection, try "File → Invalidate Caches"

Emulator won't start: Enable virtualization in BIOS, or try a different API level

App crashes: Check Logcat window (bottom of Android Studio) for error messages

Build errors: Make sure all files are in correct locations and package names match

Next Steps

Once your Hello World is working, you can:

- Modify the greeting text in MainActivity.kt
- Change colors in Color.kt
- Add buttons and interactive elements
- Learn about Jetpack Compose components
- Add navigation between screens