# Android Development with Kotlin

## Types of mobile apps

## Mobile platforms:

Much like the Linux or Windows operating system controls your desktop or laptop computer, a mobile operating system is the software platform on top of which other programs can run on mobile devices. The operating system is responsible for determining the functions and features available on your device, such as thumb wheel, keyboards, WAP, synchronization with applications, email, text messaging and more. The mobile OS will also determine which third-party applications (mobile apps) can be used on your device. The two popular mobile platforms are:

**iOS:**

iOS, an operating system from Apple, was originally developed for the iPhone. Later it was extended to support iPod Touch, iPad and Apple TV. Apple's App Store contains more than 500,000 applications and boasts more than 25 billion downloads collectively. It holds the reputation of intelligent UI creator which is based on the concept of direct manipulation, using multi-touch gestures.

**Android:**

Android is a Linux based mobile operating system developed by the Open Handset Alliance led by Google. Android boasts large community of developers writing applications extending the functionality of the devices. It has 450,000 apps in its Android Market and download exceeds 10 billion count.

## Native Apps:

Native apps live on the device and are accessed through icons on the device home screen. Native apps are installed through an application store (such as Google Play or Apple's App Store). They are developed specifically for one platform, and can take full advantage of all the device features — they can use the camera, the GPS, the accelerometer, the compass, the list of contacts, and so on. They can also incorporate gestures (either standard operating-system gestures or new, app-defined gestures). And native apps can use the device's notification system and can work offline. **E.g.: WhatsApp**

**Pros:**

- Native Apps Offer Speed
- Native Apps Work Offline
- Native Apps Are More Interactive And Intuitive
- Native Apps Allow Developers To Access The Full Feature Set Of Devices

**Cons:**

- Expensive Development
- Time-Consuming Development
- Native Apps Require Frequent Upgrades

## Hybrid Apps:

Hybrid applications are web applications, such as UIWebView in iOS and WebView in Android (not Safari or Chrome). Hybrid apps are developed using HTML, CSS and Javascript, and then wrapped in a native application using platforms like **Cordova**. This allows you to use any web-native framework you want, and there are plenty of these. **E.g**.: **Instagram, Evernote**

**Pros:**

- Low cost with ease of development
- Faster development
- Single codebase

**Cons:**

- Performance
- Feature accessibility

## Cross Platform Apps:

Cross-platform tools allow you to write your code once and then have that code translated into the native code of multiple operating systems, allowing you to publish your mobile app on different platforms with minimal effort. Native cross-platform apps are the perfect combination of hybrid apps and native apps, giving you the code reuse features of hybrid apps with an improved performance similar to that of native apps. **E.g.: Alibaba, Google Ads**

**Pros:**

- Single codebase
- Lower development cost
- Native UI Look and Feel
- Choice of development language

**Cons:**

- Performance
- Feature accessibility

## Progressive Web Apps:

Web apps are apps that run on the web and are stored on remote servers. These apps are then loaded and shown on devices through a browser interface.

Even though web apps can be great in some cases, they do have their own setbacks. The first problem with web apps is that they require internet connectivity to be able to run on your device as they're loaded from the web and aren't stored on devices. Another problem is that the apps aren't available in any mobile app store, which can make it harder for users to find and use them. **E.g.: AliExpress, Twitter Lite, Flipkart**

**Pros:**

- Can work offline
- Single codebase
- Lower development cost
- Faster development
- No installation from app/play store

**Cons:**

- Performance
- Feature accessibility