

# Native vs Non-Native Mobile Apps

Ernesto Garcia

February 4, 2024

## 1 Native vs Non-Native mobile apps

Mobile apps can be broken into 2 top level categories (native and non-native apps). Then within the category of non-native apps, there are various sub-categories (hybrid-apps, cross-platform apps, and progressive web apps (PWAs)).

**NATIVE** Android apps written in either Java or Kotlin. The libraries used in these apps can access all of the bespoke functionality built into the Android operating system, as well as the various hardware/firmware variants and chip architectures of the Android ecosystem.

**Non-native** mobile apps are written in higher-level programming languages / frameworks (e.g. JavaScript, CSS, HTML5, Python, .net, etc) and cannot directly access all of the proprietary operating system functions and hardware components — they require an additional layer/framework in between in order to make use of all the bespoke OS controls specifically built and designed by Google and Apple for the Android and iOS ecosystems respectively.

## 2 Comparative

- **Native Apps:**

- **Performance:** Native apps typically offer better performance since they are developed specifically for a particular platform (iOS or Android) and can directly access the device's hardware and software features.
- **User Experience:** Native apps provide a more seamless and intuitive user experience as they adhere to platform-specific design guidelines and behaviors.
- **Development Time:** Developing native apps can take longer since separate codebases need to be maintained for each platform (iOS and Android).

- **Access to Platform Features:** Native apps have full access to platform-specific features and APIs, allowing developers to leverage the latest functionalities offered by the operating system.
- **Non-Native Apps:**
  - **Development Time:** Non-native apps can be developed more quickly as they often use cross-platform frameworks like React Native, Flutter, or Xamarin, allowing developers to write code once and deploy it across multiple platforms.
  - **Cost-Effectiveness:** Developing non-native apps can be more cost-effective, especially for small businesses or startups, as it requires less development time and resources compared to maintaining separate native codebases.
  - **Portability:** Non-native apps are inherently more portable since they can run on multiple platforms with minimal modifications to the codebase.
  - **Performance:** Non-native apps may suffer from slightly degraded performance compared to native apps, especially when handling complex animations or interactions.

### 3 References )

- <https://medium.com/swlh/native-vs-non-native-mobile-apps-whats-the-difference-b3a641e06f52>Medium Article: Native vs Non-Native Mobile Apps: What's the Difference?
- <https://www.heady.io/blog/the-great-debate-native-vs.-non-native-mobile-apps-:text=that>