

Forum:[Hack Forums](#)

Thread: [Need recommendations on which language should I code my mobile app in? \[Android/iOS\]](#)

LiF3H4CK3R - January 7, 2019, 8:15 pm

Hello!

I'm about to hire someone to develop a Dictionary app.

I actually have a project for coding a Dictionary app. I'm about to hire Developers and Programmers for this project.

This is the first mobile app development project for our company, so I don't know much about the requirements.

Can someone please give me information about what language should the app be coded in? We are developing for both Android and iOS. It is based on your answers that I'll be recruiting my team. I want to code this app in a stable and not outdated programming language.

Please do let me know. Thank you :)

HFDone - January 8, 2019, 1:39 am

Android ide AndroidStudio use java iOS ide xCode use swift

sdk - January 8, 2019, 8:14 am

A simple and nice median to not need two different projects is [React Native](#) and similar frameworks, that allow for one project and design etc for multiple platforms. Currently use this at my place of work.

Disease - January 8, 2019, 3:57 pm

(Jan 7, 2019 09:15 PM)LiF3H4CK3R Wrote: [\[CITE\]](#) Hello!

I'm about to hire someone to develop a Dictionary app.

I actually have a project for coding a Dictionary app. I'm about to hire Developers and Programmers for this project.

This is the first mobile app development project for our company, so I don't know much about the requirements.

Can someone please give me information about what language should the app be coded in? We are developing for both Android and iOS. It is based on your answers that I'll be recruiting my team. I want to code this app in a stable and not outdated programming language.

Please do let me know. Thank you :)

There's a number of facets to this question. And nobody has provided you with incorrect information thusfar, but fairly incomplete. My hope here is to sum up the most popular, maybe even viable, options available to you.

First thing's first- by default Android and iOS development are entirely different beasts. When writing native applications, and not a Progressive Web Application (PWA... discussed later), you are typically confined to writing, compiling, and distributing the application via the platform's unique workflow. There are options available to you to streamline this, and allow you to write once and deploy everywhere. I'll also mention my favorite solution in that regard.

With Android, you're looking at Kotlin and Dart for a native application. Java had been the primary

language for Android development for a long time. But Kotlin, based on the JVM, effectively pushed Java out of favor. Google's language, Dart, has also been picking up steam. It had a relatively uncertain future up until recently, though. Flutter, the up and coming technology that is absolutely worth looking into, is new to the Android scene, is going to be the primary tech used to write apps with Google Fuschia, and it is powered by Dart. This simple act has solidified Dart's position, in my opinion. Making it equally safe to choose as Kotlin.

With iOS, the reigning tech was Objective-C for a very long time. But Swift has taken over, and Obj-C is now largely legacy. Newcomers to iOS should go with Dart, in my opinion. If you have any basic understanding of Objective-C then it'll help you tremendously in picking up Swift. But is not required at all.

So to recap the native options:Android: Kotlin or Dart. Dart is now backing Google's newest initiative, making it a solid choice.

iOS: Swift... Technically Objective-C is still somewhere in the game. But go Swift.

Now with that out of the way, there are a couple technologies available to you to write a single codebase and deploy to both iOS and Android. sdk has already mentioned one of those options: React Native. Right now this would be my recommended option. It's backed by a very solid framework (React), and as such is JavaScript. A relatively easy language to adopt if you don't know it already. Another option is actually Flutter... it just had its 1.0 release. It will be the primary tech behind apps written for Fuschia, which could potentially be replacing Android down the road. It may not be a bad idea to get ahead of the curve and adopt Flutter.

Recapping two options to write a single codebase and deploy for both Android and iOS:React Native

Flutter

Lastly, you have another option: Progressive Web Apps. This is effectively a website that's optimized for mobile. It can be setup to be installable to the phone. It effectively runs in the browser, though, at all times. Some hardware limitations do exist, but you'll find most of what you need to run an effective app is exposed to you. It's an excellent option if your app is totally data driven, and thus is tethered to requiring an Internet connection anyways. A number of major apps, like Twitter, have moved to PWAs.

Owl - January 9, 2019, 6:49 am

(Jan 8, 2019 04:57 PM)Disease Wrote: [\[CITE\]](#) With iOS, the reigning tech was Objective-C for a very long time. But Swift has taken over, and Obj-C is now largely legacy. Newcomers to iOS should go with Dart, in my opinion. If you have any basic understanding of Objective-C then it'll help you tremendously in picking up Swift. But is not required at all.

So to recap the native options:Android: Kotlin or Dart. Dart is now backing Google's newest initiative, making it a solid choice.

iOS: Swift... Technically Objective-C is still somewhere in the game. But go Dart.

ObjC is not legacy and Dart is not a native iOS language.

Disease - January 9, 2019, 3:28 pm

(Jan 9, 2019 07:49 AM)Soev Wrote: [\[CITE\]](#) ObjC is not legacy and Dart is not a native iOS language. Objective-C has almost entirely fallen from favor in iOS development. It is effectively the industry standard now. And it outperforms Objective-C in essentially every manner. Thus it is effectively legacy, as it's primary use case now is in the maintenance of existing codebases.

And you're right, Dart isn't. That was a typo, I was referring to Swift. Thank you for pointing that out.

Owl - January 10, 2019, 8:33 am

(Jan 9, 2019 04:28 PM)Disease Wrote: [\[CITE\]](#) Objective-C has almost entirely fallen from favor in iOS development. It is effectively the industry standard now. And it outperforms Objective-C in essentially every manner. Thus it is effectively legacy, as it's primary use case now is in the maintenance of existing codebases.

And you're right, Dart isn't. That was a typo, I was referring to Swift. Thank you for pointing that out.

The bulk of iOS internals are in ObjC and if you're serious about developing for the platform then you must be able to at least comprehend it. You also mentioned React Native which relies on the ObjC's dynamic runtime and, sooner or later, you will find a need to write natively and bridge back.

I'm not disputing Swift (I write in it daily) but I disagree with your dismissal of ObjC as legacy and suggesting it's used only in maintenance. Apple have presented Swift as an alternative, not a replacement (yet).