

# Flutter

**tl;dr : Flutter : 1 React Native : 0\*\***

Many of us have heard about React Native(RN) and what it **does** and **doesn't**. RN is rather a large investment by Facebook to build a modularised, cross-platform native framework for mobile apps development under the mantra *learn once write anywhere*'. There are many production apps in the market powered by RN notably the cure.fit app. What seemed very attractive from a developer standpoint is the cross-platform nature and everything being components rather than monolithic classes or routines. The very idea of having a one way data flow and everything being composed of small components was very refreshing for a native developer like me dealing with Cocoa Touch. The JS part was a little weird coming from Swift, but fellow web developers say TypeScript solves that. I didn't get a chance to build anything significant with RN.

This brings us to the topic at hand, Flutter. Flutter is Google's answer to the *write once run anywhere* problem statement. Flutter, available as a beta since 2015, is a cross-platform native mobile app development framework for building iOS, Android and **Fuchsia** apps. The first thing that struck me when trying flutter was how easy it is to set it up for development, unlike RN. Flutter is composed of widgets, everything in flutter is a widget. Widgets compose very well to form complex widgets like Legos. Another mind-blowing feature of flutter(and RN) is live/hot reload. It's my favourite feature coming from native development since

the feedback loop of running the code to viewing the output is nearly instantaneous. Composition is refreshing.

Dart is the programming language used for writing flutter apps. Its pretty simple to write once you get a hang of the semantics and conventions which is still evolving.

UI components in flutter are written in code using dart eliminating the need for another domain specific language to be added alongside the SDK for UI creation.

Flutter renders UI components on its own using a rendering engine called [Skia](#), unlike RN where the JS code is bridged to native components.

This technique was many advantages as Google doesn't need to rely on other platform creators if something goes wrong while interacting with the respective platform components. Everything on the screen, be it a navigation bar or an action bar is drawn by Skia while running a flutter app on the platform. This results in a very performant, almost 60 fps rendering as the code is translated from dart to machine code.

Now the native developer in you might ask, what about the shiny new frameworks that Apple / Google introduces every year and the wealth of other amazing frameworks available already, how do we use them in a flutter app?

For that, you need to head to this [link](#). I haven't tried\*\* this technique

but if it can scale to all the different platform specific-frameworks, Flutter could shape up to the next big thing in the mobile development landscape.

Hope that these developments push Apple to build features like Hot reload and frameworks around composability, which makes a world of difference in developer productivity.

Found this article and flutter interesting, get started [here](#). Hated it, please send your “why this article sucks” mail to [paulsumit1993@gmail.com](mailto:paulsumit1993@gmail.com).

\*\* Just kidding, technology is a tool and it’s up to us as developers to decide which is most appropriate for a given task/problem.

\*\*\* I’m skeptical as RN tried to do something similar and the results were varying.

References:

[Flutter - Beautiful native apps in record time](#)

[Flutter \(software\) - Wikipedia](#)

<https://hackernoon.com/whats-revolutionary-about-flutter-946915b09514>

[Out of Depth with Flutter – Flutter – Medium](#)

## Sunsetting React Native – Airbnb Engineering & Data Science – Medium