How does JIT and AOT work in Dart?

Just-in-time for another program compilation and probably ahead of time in a good way.

> *“ the problem isn’t what is to be done, the problem lies in deciding when it should be done and how to get the stage ready for it to be done” — Anonymous*

Dart is the main programming language to develop cross-platform mobile applications using flutter framework.

Dart uses two major ways for program compilation, the **Just-In-Time** and the **Ahead-of-Time** method.

In this article you will learn what each of the compilation method means,and a comparison between the two methods.

What is Just-In-Time (JIT)?

A JIT compiler converts program source code into native machine code just before program execution. Compilers are usually one of the deciding factors in the speed of an application both in development and when pushed to production. The JIT compilers can be used to improve performance speed and improve application runtime. It is common in JAVA programming language and is also one of the ways program source codes are executed in DART. Just-in-time compilers try to predict which instructions will be executed next so that it can compile the code in advance.

A common way to say this is that a JIT compiler compiles code on the fly, or in other words, just in time.

The Dart VM has a just-in-time compiler (JIT) that supports both pure interpretation (as required on iOS devices, for example) and runtime optimization. This is necessary for a fast development cycle, readily switching between writing the codes and testing on devices on the fly.

Photo by Jiyeon Park on Unsplash

> "Just-in-Time" means making "only what is needed, when it is needed, and in the amount needed."

What is Ahead-of-Time (AOT)?

The AOT compiler works by compiling your code before it is "delivered" to whatever runtime environment runs the code. The AOT compiler is typically used when the app is ready to be deployed to either an appstore or an in-house production backend.

The AOT-compiled code runs inside an efficient Dart runtime that enforces the sound Dart type system and manages memory using fast object allocation and a generational garbage collector as specified on the dart documentation website.

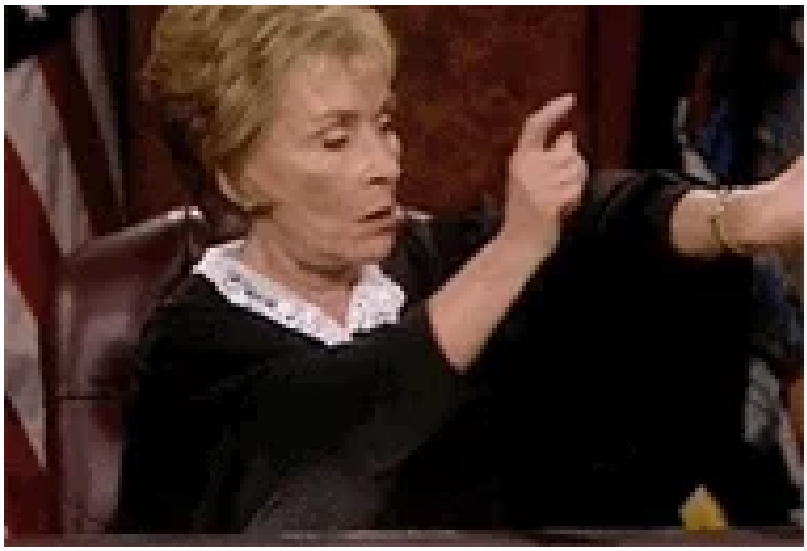
An AOT snapshot for a dart app can be generated by adding *-k aot*.

```
dart2native bin/main.dart -k aot
```

Photo by Lukas Blazek on Unsplash

> *Here is what the Dart official team has to say about this:*
>
> *Code that's compiled ahead-of-time (AOT) with a compiler such as dart2native has different performance characteristics from code that's compiled just-in-time (JIT) in the Dart VM. AOT-compiled code is guaranteed to have fast startup and consistent runtime performance, with no latency during early runs. JIT-compiled code is slower at startup, but it can have better peak performance after it runs long enough for runtime optimizations to be applied*

The major difference that exists between Ahead-Of-Time and Just-In-Time compilation is the time in which the compilation happens.



There’s no time... Go and compile it.

sources:
https://dart.dev/platforms#dart-native-vm-jit-and-aot
https://www.quora.com/What-is-a-JIT-compiler