



플러터 : 7월 17일 (웹툰)

☀ 상태	시작 전
🎯 프로젝트	<u>플러터</u>
🏷 태그	

<https://nomadcoders.co/flutter-for-beginners/lectures/4127>

#1.3 How Flutter Works (12:12)

Flutter 어플리케이션의 동작과정

1. 다른 Framework들은 OS를 통하여 버튼이나 근사한 애니메이션 등의 모든 요소를 만들 어달라고 요청하는 직접적인 소통을 하는 반면, Flutter는 C나 C++로 짜인 엔진을 통하여 Framework를 동작시키고 UI를 그려낸다. (직접적 소통이 아니라서 OS에 대한 어떠한 제약 사항이 없다.)
 2. 안드로이드, 윈도우 등의 플랫폼이 엔진을 동작하면 엔진은 "runner"프로젝트를 열어 엔진을 실행시키고 우리가 짠 Dart Flutter 코드를 동작시켜 화면에 UI를 보여준다.
 3. Flutter : 엔진을 어플리케이션 내부에 집어넣기 -> 우리가 짠 Dart 코드 컴파일하기
 4. 유저가 어플리케이션을 실행시킬 때 엔진을 가동시키는 "runner"프로젝트를 실행시킨 후, 엔진이 모든 UI를 Framework와 함께 그려준다.
-
5. Embedder(= "runner"프로젝트) : 특정 플랫폼에 특화된 것
 6. iOS, 안드로이드, 맥OS, 윈도우, 리눅스를 위한 다양한 Embedder이 존재한다.
 7. Embedder의 역할: 엔진 가동시키기



code → engine(runner) → 구현(그려주기)

#1.4 Flutter vs React Native (07:19)

위젯(widget)은 컴퓨터나 스마트폰 화면에서 사용자가 자주 사용하는 기능이나 정보를 빠르게 접근할 수 있도록 만든 작은 프로그램

예를 들어,

날씨 위젯은 현재 날씨를 실시간으로 보여주고,

달력 위젯은 오늘의 일정이나 날짜를 쉽게 확인할 수 있게 해줌

위젯은 보통 홈 화면에 배치되며, 클릭만으로 간단히 정보를 확인하거나 특정 기능을 실행

React Native👍

네이티브 앱(ios, android) 운영체제 상에서 가능한 위젯을 사용해야 하는 경우

디자인이 iOS 혹은 Android 앱처럼 보이게끔 만들고 싶은 경우

Flutter 🍌

세밀한 디자인 요구사항이 있거나 100% 커스터마이징하고 싶은 경우

외부 패키지에 의존하지 않고 고수준의 애니메이션을 구현하고 싶은 경우

구체적으로 설명하면:

- Flutter 앱은 여전히 **Android, iOS, Windows, macOS, Linux** 등의 운영체제에서 실행됨.
- Flutter는 Dart 언어로 작성된 코드로부터 네이티브 코드(각 운영체제에서 실행 가능한 코드)를 생성
- 운영체제에 따라 다르게 동작해야 하는 기능(예: 파일 시스템 접근, 카메라, 센서 등)을 사용하기 위해 Flutter는 플랫폼 채널(platform channel)을 제공
 - 이를 통해 Flutter 코드가 네이티브 코드와 상호작용할 수 있습니다.
- Flutter의 위젯은 기본적으로 Flutter 엔진 위에서 작동하지만, 네이티브 위젯이 필요한 경우 해당 운영체제의 네이티브 API를 직접 호출하여 사용 가능.

#2.0 Installation (08:17)

-android studio

#2.1 Dart Pad (03:04)

-android studio 설치해서 application 만들 때

최대한 호환성 좋은 걸로 깔아야 오류가 적음

#2.2 Running Flutter (11:15)

