

Flutter vs Kotlin(Android)

항목	Flutter	Kotlin(Android)
핵심 철학	모든 것이 위젯(Widget)	모든 것은 컴포넌트(Component)/View
주요 언어	Dart	Kotlin
화면 구성	위젯 트리(Widget Tree)로 구성	뷰 계층(View Hierarchy)로 구성
동작 방식	Flutter 엔진이 직접 그림(스스로 렌더링)	Android OS가 View를 관리(플랫폼에 의존)

- Flutter: 화면을 직접 그린다. (Canvas에 그림 그리듯)
- Kotlin(Android): 운영체제(Android 시스템)가 View를 띄운다.

2. 기본 빌딩 블록 비교

Flutter	Kotlin(Android)	설명
Widget	View	UI를 구성하는 기본 단위
StatefulWidget	Activity or Fragment + ViewModel	상태를 가질 수 있는 화면
StatelessWidget	단순 View	상태 없는 고정된 화면 요소
setState()	LiveData, StateFlow + observe()	화면 갱신 방법
Navigator.push()	startActivity(Intent)	화면 이동 방법
BuildContext	Context	리소스 접근, 화면 제어용 객체

- Flutter는 **Widget**이 모든 것의 시작이야.
- Kotlin은 **Activity/Fragment + View** 조합이 기본 구조야.

코틀린(Android) 주요 컴포넌트 4개 (Flutter랑 매칭)

코틀린(Android)은 '4대 컴포넌트'라고 부르는 게 있어.

Kotlin 컴포넌트	Flutter 대응 관계	역할
-------------	---------------	----

Activity	Flutter 전체 앱 or 특정 Route	하나의 화면(페이지)
Service	백그라운드 처리 (Flutter는 isolate나 플랫폼 채널로)	알람, 음악 재생 같은 백그라운드 작업
Broadcast Receiver	Flutter에 직접 대응 없음(Platform Channel)	시스템 이벤트 수신(예: 배터리 부족 알림)
Content Provider	직접 대응 없음(Platform Channel)	앱 간 데이터 공유(DB, 파일 등)

4. UI 작성 방식 차이

Flutter	Kotlin(Android)
코드로 UI를 직접 작성	보통 XML 파일로 레이아웃을 작성 후 코드와 연결
Declarative 방식 (선언형)	Imperative + Declarative 혼합 방식
예시: <code>Container(width: 100)</code>	예시: <code><LinearLayout width="100dp"></code>

✅ 요약

- Flutter는 **UI = 코드** (모든 게 Dart 파일 안에 있음)
- Kotlin은 **UI = XML + Kotlin 코드 분리**

5. State 관리

Flutter	Kotlin(Android)
<code>setState()</code> , <code>Provider</code> , <code>Riverpod</code> , <code>Bloc</code> 등 사용	<code>ViewModel</code> , <code>LiveData</code> , <code>StateFlow</code> 사용

6. 생명주기 (Lifecycle)

Flutter	Kotlin(Android)
<code>initState()</code> , <code>dispose()</code> 등으로 관리	<code>onCreate()</code> , <code>onStart()</code> , <code>onResume()</code> , <code>onPause()</code> , <code>onDestroy()</code> 등 세세히 쪼개져 있음