

# Different Types of Providers

Riverpod 프로바이더의 종류

# Provider의 종류

- Provider
- StateProvider
- StateNotifierProvider
- FutureProvider
- StreamProvider
- ChangeNotifierProvider (사용 안함 - Provider 마이그레이션 용도)
- 각각 다른 타입을 반환해주고 사용 목적이 다르다
- 모든 Provider는 글로벌하게 선언된다

# Provider

- 가장 기본 베이스가되는 Provider
- 아무 타입이나 반환 가능
- Service, 계산한 값등을 반환할때 사용
- 반환값을 캐싱할때 유용하게 사용된다
  - 빌드 횟수 최소화 가능
- 여러 Provider의 값들을 묶어서 한번에 반환값을 만들어낼 수 있다

# StateProvider

- UI에서 “직접적으로” 데이터를 변경할 수 있도록 하고싶을때 사용
- 단순한 형태의 데이터만 관리 (int, double, String등)
- Map, List등 복잡한 형태의 데이터는 다루지 않음
- 복잡한 로직이 필요한경우 사용하지 않음
  - number++ 정도의 간단한 로직으로만 한정

# StateNotifierProvider

- StateProvider와 마찬가지로 UI에서 “직접적으로” 데이터를 변경할 수 있도록 하고싶을 때 사용
- 복잡한 형태의 데이터 관리가능 (클래스의 메소드를 이용한 상태관리)
- StateNotifier를 상속한 클래스를 반환

# FutureProvider

- Future 타입만 반환가능
- API 요청의 결과를 반환할때 자주 사용
- 복잡한 로직 또는 사용자의 특정 행동뒤에 Future를 재실행하는 기능이 없음
  - 필요할경우 StateNotifierProvider 사용

# StreamProvider

- Stream 타입만 반환가능
- API 요청의 결과를 Stream으로 반환할때 자주 사용
  - Socket등

# 비교차트

Provider 종류	반환값	사용 예제
Provider	아무 타입	데이터 캐싱
StateProvider	아무 타입	간단한 상태값 관리
StateNotifierProvider	StateNotifier를 상속한 값 반환	복잡한 상태값 관리
FutureProvider	Future 타입	API 요청의 Future 결과값
StreamProvider	Stream 타입	API 요청의 Stream 결과값



# ref.read vs ref.watch

- ref.watch는 반환값의 업데이트가 있을때 지속적으로 build 함수를 다시 실행해준다.
- ref.watch는 필수적으로 UI관련 코드에만 사용한다.
- ref.read는 실행되는순간 단 한번만 provider 값을 가져온다.
- ref.read는 onPressed 콜백처럼 특정 액션 뒤에 실행되는 함수 내부에서 사용된다.