

<Flutter>

- Dart 언어 하나만 알면 됨
- Dart: 자바스크립트 대용품이라 비슷하게 생김.
- 만들어진 UI를 가져다쓰고 그걸 커스터마이징하는 식으로 개발.

Flutter -> UI Framework

Dart -> 프로그래밍 언어

<위젯>

- StatelessWidget : 변경 X , 정적인 텍스트
=> 하위 클래스에는 Text, Image, Container, Row, Column, ListView 등이 있다.
- StatefulWidget : 변경 O
=> 하위 클래스에는
TextButton, IconButton, Checkbox, Radio, TextFormField, DropdownButton 등이 있다.
* MaterialButton(상, 디자인 스타일을 가지는 버튼) - TextButton(하, 텍스트를 보여주고 입력에 반응하는 버튼)
- DefaultTabController: 탭을 관리하는 컨트롤러를 제공. 탭 간 전환 쉽게 관리. TabBar 와 TabBarView 위젯과 함께 사용하여 탭 전환. // 사용법: length 속성으로 탭의 수를 지정, 하위 위젯으로 TabBar 과 TabBarView 포함.
- TabBarView: 탭 간의 스와이프를 통해 전환할 수 있는 뷰 제공. 각 탭에 해당하는 콘텐츠 표시. // children 속성에 가 ㄴ 탭에 해당하는 위젯을 리스트 형태로 전달.
- Container : 다른 위젯을 포함하고, 그 위젯을 스타일링하거나 배치하는데 사용. / 자식 위젯을 포함하고 그것의 레이아웃 설정. (CircleAvatar 위젯이 이 컨테이너의 자식) / 자식 위젯과 컨테이너의 경계 사이에 여백 만들기. 패딩 설정 / 배경 설정(색상, 그라디언트, 이미지) / 자식 위젯의 크기 조절(width, height) /

<클래스>

- State
- Scaffold: AppBar, Body, BottomNavigationBar 지원해주는 클래스
- MyAppState: State<MyApp>을 상속받아 MyApp 의 상태 관리. 이 클래스 내에서 build 메서드를 구현하여 UI 빌드.

<State 클래스>

- * StatefulWidget을 사용할 때는 항상 createState 메서드를 오버라이드하여 상태를 관리하는 State 객체를 반환해야 합니다.
- * StatefulWidget의 상태를 저장하고 관리하는 클래스. 상태가 변경되면 setState 메서드를 호출하여 Flutter에 UI를 다시 그리도록 요청.(변경되지 않는 부분이니까?)

<메서드(Method)>

- createState: StatefulWidget 을 구현할 때 반드시 오버라이드(상속 받은 메서드를 재정의, ex) 바퀴의 회전 횟수 5회 -> 상속받은 더 빠른 바퀴의 회전 횟수 10회)해야하는 메서드.
StatefulWidget 생성 시 처음 한 번 호출됨.
- build

<프로필 화면을 구현해보기>

- Stateless: 프로필 화면 전체적인 틀 디자인

- Stateful: 이름, 직업 입력하고 변경하는 칸, 관심있는 영상, 강사 좋아요 목록