

01

Flutter

01. Flutter란

- 구글에서 개발한 오픈소스 UI 킷
- 하나의 코드베이스를 사용하여 iOS, Andriod, 웹 데스크탑 (Windows, macOS, Linux) 애플리케이션 개발
- 특징
 - 다양한 플랫폼 지원
 - Dart 언어 사용
 - Widget 기반 구조
 - Hot Reload
 - 고성능 렌더링
 - 풍부한 블러그인

02. Widget

- 독립적으로 실행되는 작은 프로그램
- 주로 바탕화면 등에서 날씨나 뉴스 생활정보를 보여줌
- 그래픽이나 데이터 요소를 처리하는 함수를 가짐

03. Widget in Flutter

- UI를 만들고 구성하는 모든 기본 단위 요소
 - Text, Icon, Image, Text field...
- 눈에 보이지 않는 구조를 표현하는 요소들
 - Center, Colmun, Row, Padding...
- Widget의 종류
 - Stateless Widget
 - Stateful Widget
 - Inherited Widget

04. Widget Tree

- Widget들은 Tree구조로 설계되어 있음
- 하나의 Widget안에 다른 Widget들이 포함될 수 있음
- Widget은 부모 Widget과 자식 Widget으로 구성
- Parent Widget을 Widget Container라고도 함

04. Widget Tree

MyApp: 논리적 앱 이름

MaterialApp: 앱의 전반적인 구조와 설정을 정의

MyHomePage: 논리적 페이지 이름

Scaffold: 앱의 하나의 화면에 해당하는
Widget들을 포함

05. Stateless Widget

- 변하지 않는 상태를 가지는 위젯
- 상태가 고정되어 있고 빌드 시점 이후에는 변경되지 않음
- 화면에 데이터를 렌더링 하지만 사용자 상호작용이나 기타 이벤트에 의해 자체적으로 상태를 변경할 수 없음
- Flutter가 제공하는 대부분의 위젯

05. Stateless Widget

■ 특징

■ 불변성

- 위젯이 한번 생성되면 상태나 데이터가 변하지 않음
- 모든 데이터는 위젯의 생성자에게 전달받아 화면에 표시

■ 단순한 UI 구성

- 정적 콘텐츠(Text, Icon, Button등)을 표시

■ 효율적인 리렌더링

- 상태가 없기 때문에 Flutter의 효율적 트리 리렌더링 메커니즘을 활용

06. 기초 Widget

■ MaterialApp

- 앱의 초기화면, 테마 등 기본 설정 정의
- 주 속성 : title, theme, home, routes...

■ Scaffold

- 한 페이지에 해당하는 위젯들의 모음
- 주 속성 : appBar, body..

■ Text

- 문자열 출력
- 주 속성 : style, textAlign ..

■ Column

- 자식 Widget들을 세로로 배치
- 주 속성 : children, mainAxisAlignment..

06. 기초 Widget

■ Row

- 자식 Widget들을 가로로 배치
- 주 속성 : children, mainAxisAlignment..

■ Center

- 자식 Widget을 가운데로 배치
- 주 속성 : child

■ Padding

- 자식 Widget들의 여백을 설정
- 주 속성 : padding , child

■ Divider

- 구분선 추가
- 주 속성 : color, thickness, endIndent, height...

■ SizedBox

- 빈공간 설정
- 주 속성 : height, width

06. Button Widget

■ ElevatedButton

- 음영 효과가 있어 떠 있는 듯한 입체감을 주는 버튼
- 주 속성 : `child`, `onPress`, `style..`

■ TextButton

- 배경이 없는 간단한 버튼, 주로 텍스트만 표시
- 주 속성 : `child`, `onPress`, `style..`

■ OutlinedButton

- 버튼 테두리가 강조된 형태의 버튼
- 주 속성 : `child`, `onPress`, `style..`

■ IconButton

- 아이콘만 포함된 버튼으로 짧은 작업에 사용
- 주 속성 : `icon`, `iconSize`, `onPress`, `style..`

06. Button Widget

■ **TextButton.icon**

- icon과 텍스트의 조합을 표현하는 Widget
- 주 속성 : icon, onPressed, label, style..

■ **ElevatedButton.icon**

- icon과 ElevatedButton의 조합을 표현하는 Widget
- 주 속성 : icon, onPressed, label, style..

07. SnackBar

- 앱 화면 하단에 잠깐 표시되는 알람 Widget
- 주 속성 : content, action, backgroundColor, duration..
- 생성 예

```
ScaffoldMessenger.of(context).showSnackBar(  
  SnackBar(  
    content: Text('This is a SnackBar!'),  
    duration: Duration(seconds: 2),  
  ));
```

08. Fluttermtoast

- 앱에 간단한 알림 메시지를 잠시 나타나게 하는 Toast 메시지 라이브러리
- Fluttermtoast 패키지 설치
 - fluttertoast: ^8.2.1
 - flutter pub get
- 생성 예

```
Fluttertoast.showToast(  
  msg: "This is a Toast message",  
  toastLength: Toast.LENGTH_SHORT,  
  gravity: ToastGravity.BOTTOM,  
  backgroundColor: Colors.black,  
  textColor: Colors.white,  
  fontSize: 16.0,  
);
```

09. Navigator

- 화면 간 Navigation을 관리하는 Widget
- 페이지 전환, 스택 관리, 화면 뒤로가기 등을 처리
- 스택(Stack) 구조를 기반으로 동작
 - 화면을 스택에 추가하거나 제거하여 Navigation 흐름을 관리
- Route 관리
 - `MaterialPageRoute`를 사용해 새로운 페이지를 추가하거나 제거
- Named Routes 지원
 - 이름 있는 Route를 정의하여 Navigation을 간소화
- 메서드 : `push`, `pop`...

09. Navigator

■ 생성 예

```
ElevatedButton(  
  onPressed: () {  
    Navigator.push( context, MaterialPageRoute(  
      builder: (context) => SecondPage()), );  
  },
```

```
MaterialApp(  
  initialRoute: '/',  
  routes: {  
    '/': (context)=>ScreenA(),  
    '/b': (context)=>ScreenB(),  
    '/c': (context)=>ScreenC(),  
  },  
);
```