

# Flutter 프레임워크 기초

---

## Flutter 개발 환경 구성과 위젯 이해

## 학습목표

- Flutter 프레임워크의 기본 개념과 구조 이해
- Flutter 애플리케이션 개발을 위한 기초 환경 설정
- Flutter 위젯의 개념과 위젯 트리 구조 파악

## Flutter 개발 준비

- 애플리케이션 인터페이스 목업 작성을 통한 레이아웃 설계
- 개발 도구 선택 (Excalidraw, Figma, FlutterFlow 등)
- Flutter SDK와 개발환경 설정 완료
- 보일러플레이트 프로젝트 생성 및 기본 구조 이해

## Flutter 프로젝트 구성

- Flutter create 명령어를 통한 프로젝트 템플릿 생성
- 프로젝트 유형별 템플릿 선택 (app, module, package, plugin)
- 디버그 배너 설정 및 개발 모드 구성
- 위젯 기반의 사용자 인터페이스 설계 방법론 적용

# main.dart

기본 Flutter 애플리케이션의 진입점

# main.dart 초기 설정

```
import 'package:flutter/material.dart'; // Flutter UI 라이브러리 import

void main() {
  runApp(const MyApp()); // 앱 시작, MyApp 위젯을 루트 위젯으로 설정
}

class MyApp extends StatelessWidget {
  // MyApp은 StatelessWidget으로, 앱의 기본 설정을 담당
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    // build 메서드는 위젯의 UI를 생성
    return MaterialApp(
      // MaterialApp은 Flutter 앱의 기본 구조를 제공
      title: 'Flutter Demo', // 앱의 제목
      theme: ThemeData(
        // 앱의 테마 설정
        primarySwatch: Colors.blue, // 기본 색상 설정
      ),
      home: const MyHomePage(title: 'Flutter Demo Home Page'), // 앱의 홈 화면을 MyHomePage로 설정
      debugShowCheckedModeBanner: false, // 디버그 배너 제거
    );
  }
}
```

# MyHomePage 위젯 구현

```
class MyHomePage extends StatefulWidget {  
  // MyHomePage는 StatefulWidget으로, 화면의 상태를 관리  
  const MyHomePage({Key? key, required this.title}) : super(key: key);  
  // title은 홈 화면의 제목으로 사용될 문자열  
  final String title;  
  
  @override  
  State<MyHomePage> createState() => _MyHomePageState();  
  // createState 메서드는 MyHomePage의 상태를 생성  
}  
  
class _MyHomePageState extends State<MyHomePage> {  
  // _MyHomePageState는 MyHomePage의 실제 상태를 관리  
  int _counter = 0; // _counter는 버튼 클릭 횟수를 저장하는 변수  
  
  void _incrementCounter() {  
    // _incrementCounter는 버튼 클릭 시 호출되는 메서드  
    setState(() {  
      // setState를 호출하여 화면을 다시 그리도록 지시  
      _counter++; // _counter 값을 1 증가  
    });  
  }  
}
```

# 위젯 빌드 메서드

```
@override
Widget build(BuildContext context) {
  // build 메서드는 위젯의 UI를 생성
  return Scaffold(
    // Scaffold는 앱의 기본 레이아웃을 제공 (AppBar, Body, FloatingActionButton 등)
    appBar: AppBar(
      // AppBar는 앱 상단에 위치하는 톨바
      title: Text(widget.title), // AppBar의 제목을 설정
    ),
    body: Center(
      // Center는 자식 위젯을 화면 중앙에 배치
      child: Column(
        // Column은 자식 위젯을 세로로 배치
        mainAxisAlignment: MainAxisAlignment.center, // 자식 위젯들을 중앙 정렬
        children: <Widget>[
          // children은 Column에 포함될 위젯 목록
          const Text('You have pushed the button this many times:'), // 텍스트 위젯
          Text(
            // 텍스트 위젯
            '$_counter', // $_counter 값을 화면에 표시
            style: Theme.of(context).textTheme.headlineMedium, // 텍스트 스타일 설정
          ),
        ],
      ),
    ),
    floatingActionButton: FloatingActionButton(
      // FloatingActionButton은 화면 하단에 떠 있는 버튼
      onPressed: _incrementCounter, // 버튼 클릭 시 _incrementCounter 메서드 호출
      tooltip: 'Increment', // 톨팁 텍스트 설정
      child: const Icon(Icons.add), // 버튼 내부에 표시될 아이콘
    ),
  );
}
```



## 디버그 배너 제거

```
class MyApp extends StatelessWidget {  
  // MyApp은 StatelessWidget으로, 앱의 기본 설정을 담당  
  const MyApp({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    // build 메서드는 위젯의 UI를 생성  
    return MaterialApp(  
      debugShowCheckedModeBanner: false, // 디버그 배너를 제거하는 설정  
      title: 'Flutter Demo', // 앱의 제목  
      theme: ThemeData(  
        // 앱의 테마 설정  
        primarySwatch: Colors.blue, // 기본 색상 설정  
      ),  
      home: const MyHomePage(title: 'Flutter Demo Home Page'), // 앱의 홈 화면을 MyHomePage로 설정  
    );  
  }  
}
```

## 애플리케이션 인터페이스 만들기

제품	링크	가격	설명
Excalidraw	<a href="https://excalidraw.com">https://excalidraw.com</a>	무료	일반 웹 기반 그래픽 디자인 도구
Figma	<a href="https://www.figma.com">https://www.figma.com</a>	무료/유료	코드 없이 UI 템플릿을 생성하기 위한 공유 디자인 및 빌드 솔루션
FlutterFlow	<a href="https://flutterflow.io">https://flutterflow.io</a>	무료/유료	Flutter 코드를 생성하는 대화형 UI 템플릿 및 구성 요소

## 보일러플레이트 Flutter 프로젝트 생성

타입	설명
app	기본 Flutter 애플리케이션을 생성하는 기본 템플릿
module	다른 애플리케이션과 통합할 수 있는 모듈을 생성하는 템플릿
package	공유 가능한 Flutter 프로젝트 생성 템플릿
plugin	Android 및 iOS에서 사용할 API 기반 템플릿
skeleton	Detail View에 기반한 모범 사례 애플리케이션 템플릿

기본 애플리케이션 생성:

```
flutter create my_awesome_app
```

모듈 생성:

```
flutter create -t module my_awesome_module
```

패키지 생성:

```
flutter create -t package my_awesome_package
```

## 플러그인 생성:

```
flutter create -t plugin my_awesome_plugin --platforms web --platform android
```

### 참고

플러그인을 생성할 때는 지원할 플랫폼을 지정해야 합니다. 추가할 플랫폼마다 `--platform` 접두사를 추가해야 합니다.

## 스켈레톤 생성:

```
flutter create -t skeleton my_awesome_skeleton
```

## 요약

- Flutter 프레임워크의 기본 개념과 구조 이해
- Flutter 애플리케이션 개발을 위한 기초 환경 설정
- Flutter 위젯의 개념과 위젯 트리 구조 파악

# END

---