Full-Stack Service Programming

Lecture 3
Flutter 기반 모바일 앱 개발

2023. 09. 01

Sungwon Lee

Department of Software Convergence

경희대학교
KYUNG HEE UNIVERSITY

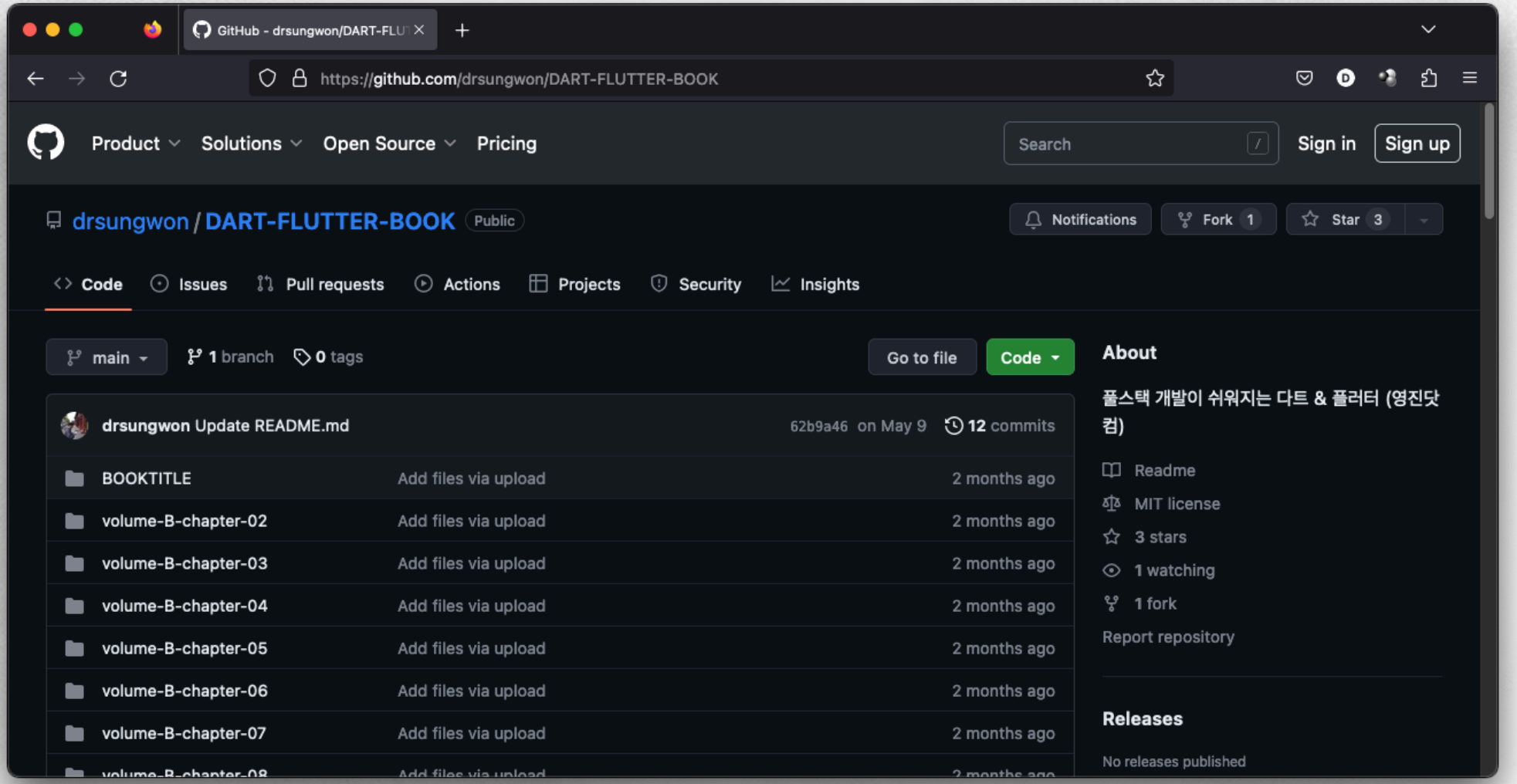
교재 목차

- Volume.E Flutter로 Mobile App 개발

TEXT BOOK & SOURCE CODE

소스 코드

- https://github.com/drsungwon/DART-FLUTTER-BOOK

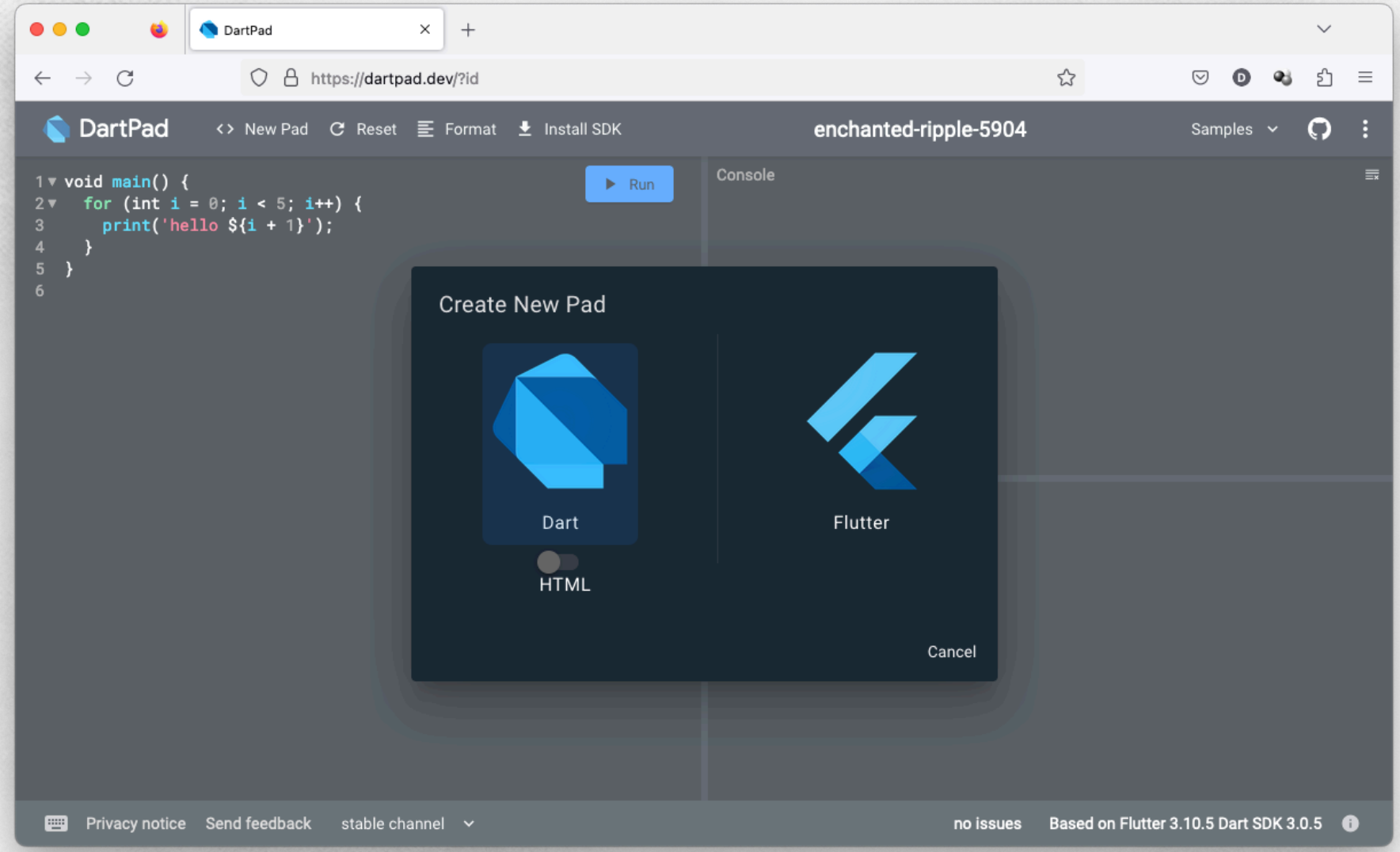


Flutter 실행 환경

- DartPad : Volume.B Chapter.1 참조
  - https://dartpad.dev/?
- Smart Phone & Tablet : Flutter SDK Install 참조
  - https://docs.flutter.dev/get-started/install
  - https://docs.flutter.dev/get-started/editor
  - https://docs.flutter.dev/get-started/test-drive
- Desktop (macOS, Linux, Windows) : Volume.F 예정
  - https://docs.flutter.dev/platform-integration/desktop
- Web Browser : Volume.G 예정
  - https://docs.flutter.dev/platform-integration/web/building

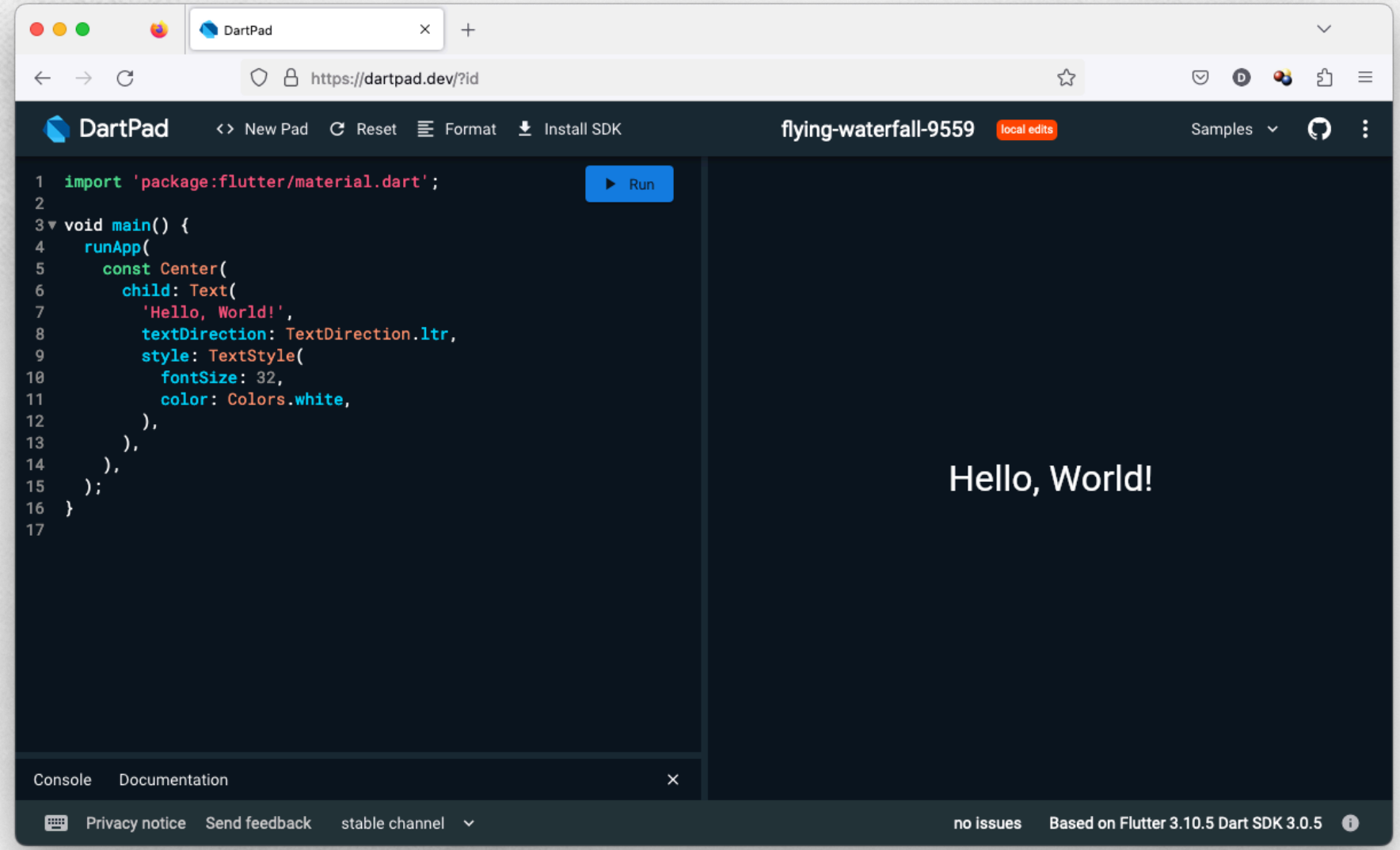
DartPad를 이용한 Hello World 개발하기

- DartPad에서 Flutter를 위한 New Pad 생성



DartPad를 이용한 Hello World 개발하기

- volume-E-chapter-01.dart 코드의 copy & paste 후, 실행



volume-E-chapter-01.dart 이해하기

- 패키지
  - Dart/Flutter에서 사용하는 표준 혹은 3rd-party 라이브러리
  - flutter/material.dart는 Android의 표준 GUI 체계인 Material Design을 지원하는 Flutter의 표준 라이브러리임

- 코드 구조

```
runApp( 표현 1, );
표현 1 = const Center( child : 표현 2, )
표현 2 = Text( 'Hello, World!', textDirection: TextDirection.ltr, style: 표현 3, )
표현 3 = TextStyle( fontSize: 32, color: Colors.white, )
```

volume-E-chapter-01.dart 이해하기

- TextStyle 클래스
  - Text의 형태 및 화면에 보여지는 (immutable) 스타일을 표현함
  - https://api.flutter.dev/flutter/painting/TextStyle-class.html
- Text 클래스
  - 한 가지 스타일에 기반한 Text 위젯을 정의함
  - https://api.flutter.dev/flutter/widgets/Text-class.html
- Center 클래스
  - 포함하는 Child를 화면 중간에 위치시키는 위젯을 정의함
  - https://api.flutter.dev/flutter/widgets/Center-class.html

volume-E-chapter-01.dart 이해하기

- runApp()
  - runApp 함수 호출은 위젯들로 구성된 Flutter 애플리케이션을 시작하는 지점으로, Flutter 프로그램 의 main 함수
  - https://api.flutter.dev/flutter/widgets/runApp.html

"Inflate the given widget and attach it to the screen. The widget is given constraints during layout that force it to fill the entire screen. If you wish to align your widget to one side of the screen (e.g., the top), consider using the Align widget. If you wish to center your widget, you can also use the Center widget."

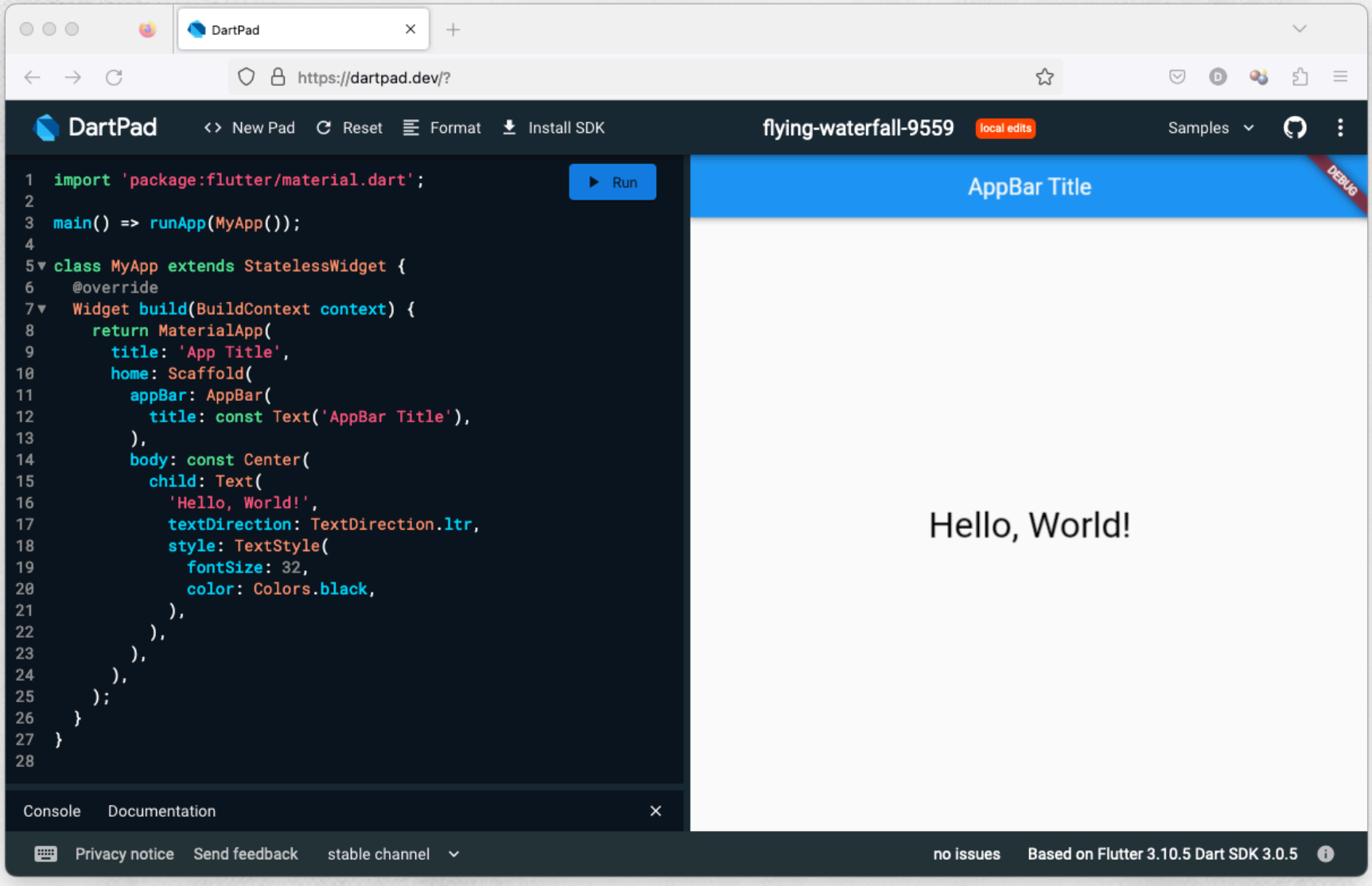
"주어진 위젯(widget)을 부풀게 하여 화면에 붙입니다. 위젯은 레이아웃 중에 전체 화면을 채우도록 강제하는 제약 조건이 부여됩니다. 위젯을 화면의 원하는 위치에 정렬하려면 Align 위젯을 사용합니다. 만약 화면 중앙에 위치하도록 하려면 Center 위젯을 사용합니다."

DartPad를 이용한 Hello World 개발 (리뷰 & 실습)

- volume-E-chapter-01.dart

Hello World 진화시키기 Part.1

- volume-E-chapter-02.dart 코드의 copy & paste 후, 실행



Hello World 진화시키기 Part.1

```
main() → runApp() → StatelessWidget.build() [ = MyApp().build() ]
```

- StatelessWidget 클래스
  - https://api.flutter.dev/flutter/widgets/StatelessWidget-class.html
- MaterialApp 클래스
  - https://api.flutter.dev/flutter/material/MaterialApp-class.html
- Scaffold 클래스
  - https://api.flutter.dev/flutter/material/Scaffold-class.html
- AppBar 클래스
  - https://api.flutter.dev/flutter/material/AppBar-class.html

StatelessWidget 클래스

- 고정된 정보를 다루는 경우에 Base 클래스로 사용하는 위젯임
- 공식 문서에서는 다음과 같이 설명함

  A stateless widget is a widget that describes part of the user interface by building a constellation of other widgets that describe the user interface more concretely. The building process continues recursively until the description of the user interface is fully concrete.

- Derived 클래스에서 반드시 override 해야 하는 메소드는 build() 로서, OS가 앱을 실행하는 경우, 화면에 출력해야 하는 내용들을 준비하고, 화면에 나타나도록 하는 핵심 기능을 수행함

MaterialApp 클래스

- 구글의 Material Design을 준수하는 Application임
- 공식 문서에서는 다음과 같이 설명함

  A convenience widget that wraps a number of widgets that are commonly required for Material Design applications.

- 주요 데이터
  - title : OS가 앱을 인식하기 위한 프로그램의 이름
  - home : 앱의 홈 화면을 채울 내용

Scaffold 클래스

- Material Design 앱이 화면에 어떻게 나타날지를 표현함
- 공식 문서에서는 다음과 같이 설명함

  Implements the basic Material Design visual layout structure.

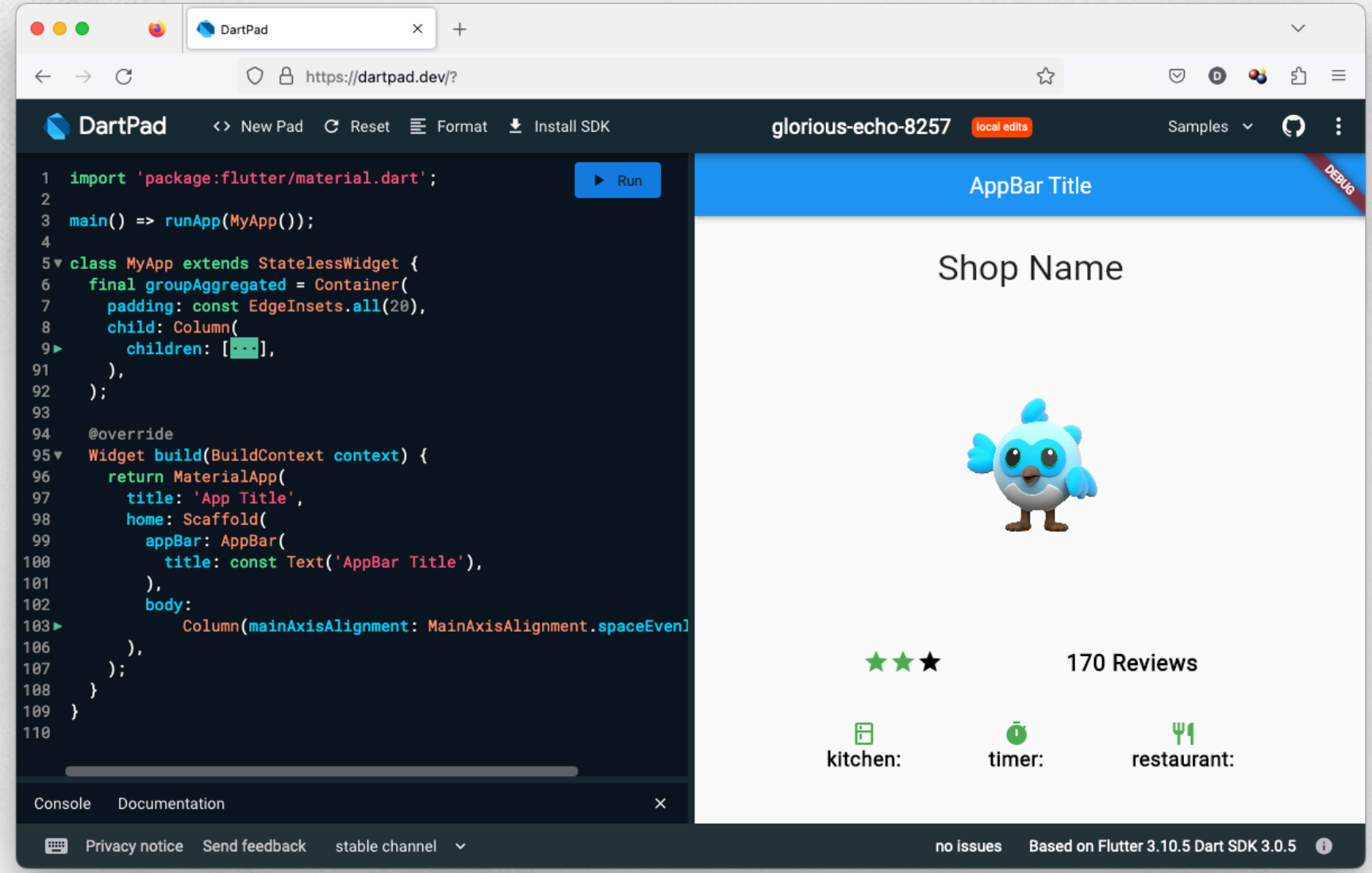
- 주요 데이터
  - appBar : Scaffold 화면 상단에 보여줄 애플리케이션 바 내용 (혹은 타이틀과 메뉴)
  - body : Scaffold 화면의 메인 컨텐츠 출력 내용

Hello World 진화시키기 Part.1 (리뷰 및 실습)

- volume-E-chapter-02.dart

Hello World 진화시키기 Part.2

- volume-E-chapter-03-code-22-final.dart 실행



새롭게 등장한 위젯들

- Icon : https://api.flutter.dev/flutter/widgets/Icon-class.html
- Image : https://api.flutter.dev/flutter/widgets/Image-class.html
- Row : https://api.flutter.dev/flutter/widgets/Row-class.html
- Column : https://api.flutter.dev/flutter/widgets/Column-class.html
- Expanded : https://api.flutter.dev/flutter/widgets/Expanded-class.html
- DefaultTextStyle : https://api.flutter.dev/flutter/widgets/DefaultTextStyle-class.html
- Container : https://api.flutter.dev/flutter/widgets/Container-class.html

Hello World 진화시키기 Part.2 (리뷰 및 실습)

- volume-E-chapter-03-code-22-final.dart

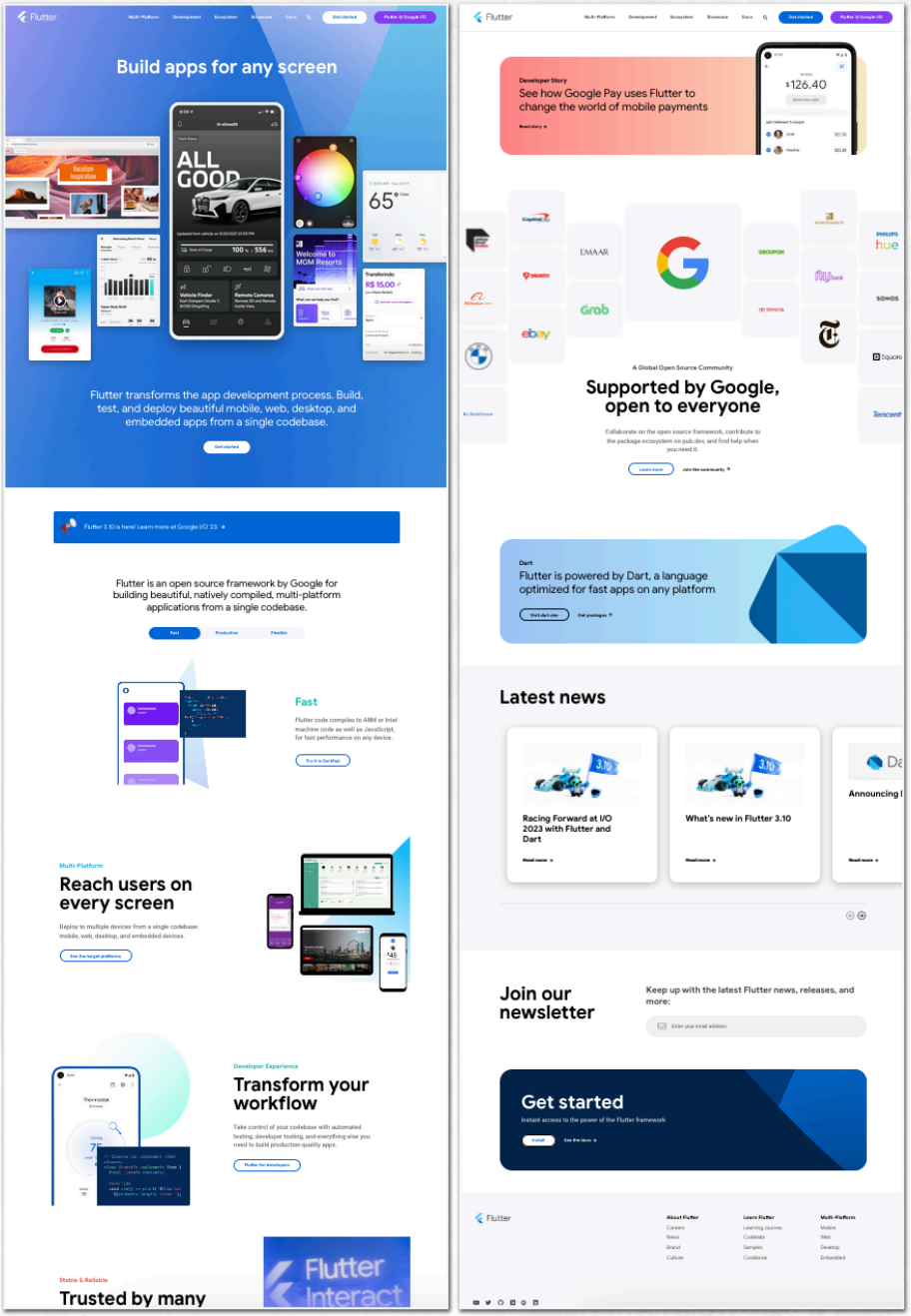
ONE MORE : Step by Step ?

- Hello World 부터 시작해서 최종적인 코드로 발전하는 과정을 보고 싶다면?
- volume-E-chapter-03-code-01.dart 부터 시작해서, volume-E-chapter-03-code-22-final.dart 까지의 코드가 어떻게 변해가는지 과정을 확인함

Flutter 이해하기

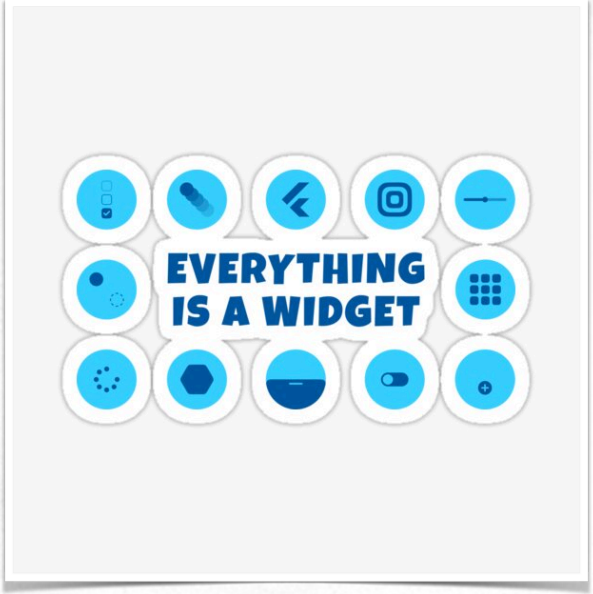
- Flutter 공식 사이트 방문
  (https://flutter.dev/)

- 키워드 확인
  - Open source
  - By Google
  - Multi-platform
  - Natively complied
  - Single codebase



Flutter 이해하기

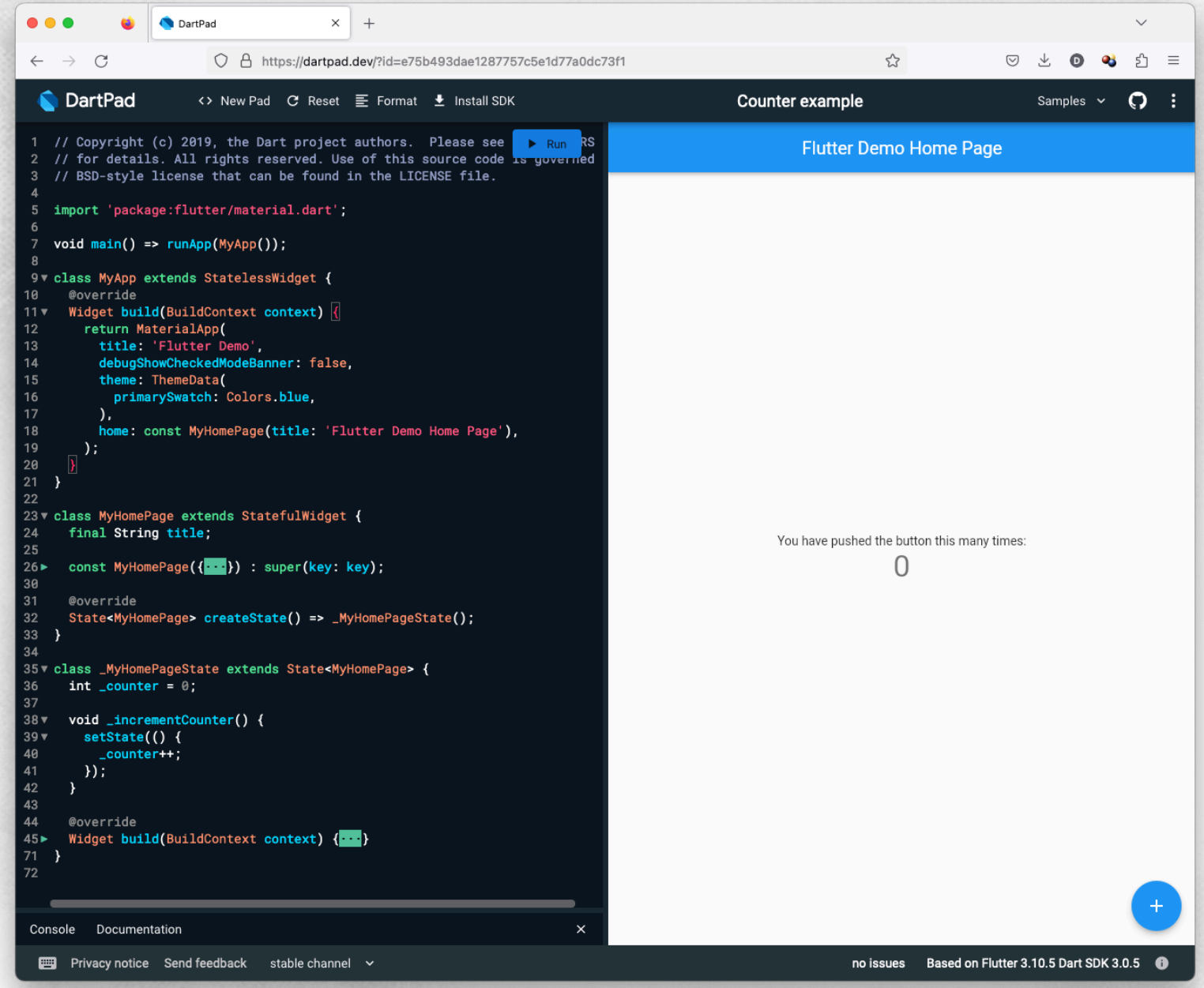
- "Everything is a Widget."
  - Widget catalog : https://docs.flutter.dev/ui/widgets
  - Cookbook : https://docs.flutter.dev/cookbook
  - Sample : https://flutter.github.io/samples/#
  - Flutter widget index : https://docs.flutter.dev/reference/widgets
  - Flutter documentation : https://docs.flutter.dev/



Flutter 공식 Counter 프로그램 이해하기

- Counter 프로그램 이란?
  - Flutter SDK의 공식 template (자동 생성) 프로그램
  - DartPad 사이트의 'Counter example' 예제
  - https://docs.flutter.dev/get-started/test-drive
  - https://api.flutter.dev/flutter/material/Scaffold-class.html

Flutter 공식 Counter 프로그램 이해하기



Flutter 공식 Counter 프로그램 이해하기

MyHomePage.createState() → _MyHomePageState.build() → return Scaffold()

```dart
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
    );
  }
}
```

```dart
class MyHomePage extends StatefulWidget {
  final String title;

  const MyHomePage({...}) : super(key: key);

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}
```

```dart
class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [...],
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment',
        child: const Icon(Icons.add),
      ),
    );
```

새롭게 등장한 위젯들

- StatefulWidget : https://api.flutter.dev/flutter/widgets/StatefulWidget-class.html
- State : https://api.flutter.dev/flutter/widgets/State-class.html

StatefulWidget 클래스

- 앱의 변경 가능한 상태(state)를 저장하는 State 위젯을 관리함
  - 프로그램 실행 동안 바뀌는 값은 StatefulWidget이 아닌, State 위젯에 저장함
  - StatefulWidget은 State 위젯을 생성하고 관리함
- StatelessWidget과 달리, 앱의 화면에 대한 build()는 상태 정보를 가지고 있는 State 위젯을 통해서 수행됨
  - Count 프로그램 예제처럼, State 위젯 안의 build()를 통해서 Scaffold와 같은 화면 제어 기능이 수행됨

State 클래스

- 프로그램 내부에서 변경되는 정보를 저장할 변수를 정의함
- 변경되는 정보가 화면에 표시될 때의 모습을 구성하는 build() 메소드를 정의함
  - Count 프로그램 예제는 Scaffold 위젯을 정의하고 있음
- 변경되는 정보를 저장한 변수의 변경을 Flutter에게 알려서, 화면의 업데이트를 요청하는, setState() 메소드를 정의함
  - Count 프로그램 예제는 카운터 값을 1만큼 증가시키는 기능으로 작성되어 있음
  - setState() 메소드는 Flutter에게 정보의 업데이트를 알리는 용도임으로, 복잡한 작업은 setState() 밖에서 이루어져야 함

StatefulWidget와 State의 관계

StatefulWidget를 extends한 MyStatefulWidget 클래스가 있고, State를 extends한 _MyState 클래스가 있습니다. 지금까지의 설명을 잘 읽었다면 다음의 동작을 유추할 수 있습니다.

① MyStatefulWidget이 _MyState을 생성하겠지

② _MyState는 변하는 정보를 저장하겠지

③ _MyState는 변하는 정보를 화면에 출력하는 방법을 가지고 있겠지

④ _MyState는 변하는 정보가 언제 출력될지를 결정하는 방법을 가지고 있겠지

결론부터 이야기하면, 맞습니다. _MyState를 생성하기 위해서 MyStatefulWidget 안에는 createState() 메서드가 오버라이드되어 있고 실제로 _MyState 객체를 생성합니다. ②를 위해서, _MyState 클래스 안에는 변수 _y가 있고, changeMyState() 메서드에서 이 값을 변경하는 작업을 합니다. ③을 위해서 _MyState 안에는 build() 메서드가 오버라이드 되어 있고, 메서드 안을 보면 _y 값을 출력 용도로 사용하고 있습니다. 그리고 ④를 위해서 _MyState 클래스의 메서드인 changeMyState() 메서드가 setState()를 호출하고 있습니다.

```
class MyStatefulWidget extends StatefulWidget {
  const MyStatefulWidget({
    Key? key,
    this.x = value_of_x,
  }) : super(key : key);

  final X x;

  @override
  State<MyStatefulWidget> createState() => _MyState();
}

class _MyState extends state<MyStatefulWidget> {
  Y _y;

  void changeMyState() {
    setState(() { _y = value_of_y; });
  }

  @override
  Widget build(BuildContext context) {
    return CredentialsContainer(
      Z: widget.x,
      V: SomeFunction(_y),
    )
  }
}
```

Flutter 공식 Counter 프로그램 (리뷰 & 실습)

- volume-E-chapter-05.dart

ONE MORE : FloatingActionButton class ?

- Material Design 기반의 floating action button 임
- https://api.flutter.dev/flutter/material/FloatingActionButton-class.html

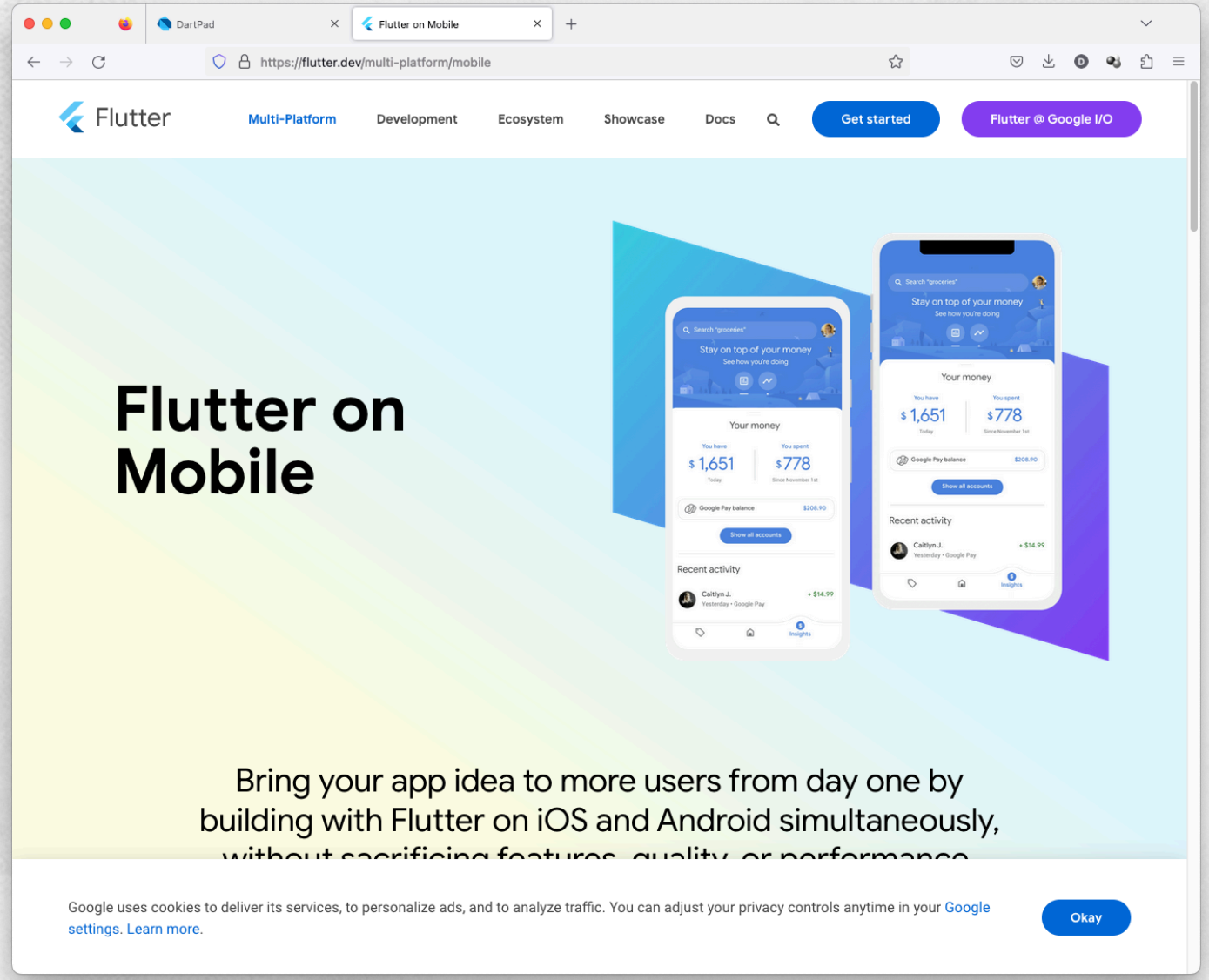
Stateless Widget 활용하기

- Stateless 위젯에 기반하여, (고정된 정보를 제공하는) 웹 화면을 앱으로 모사 개발함

- 새롭게 등장하는 기능과 위젯
  - Theme.of(context). : context는 상위 위젯의 정보로서, 이 문법은 상위 위젯의 테마 정보에 접근함. 예를 들어, Theme.of(context).primaryColor는 현재 위젯이 속한 상위 위젯에서 Default 색상 정보를 리턴함
  - ListView : https://api.flutter.dev/flutter/widgets/ListView-class.html

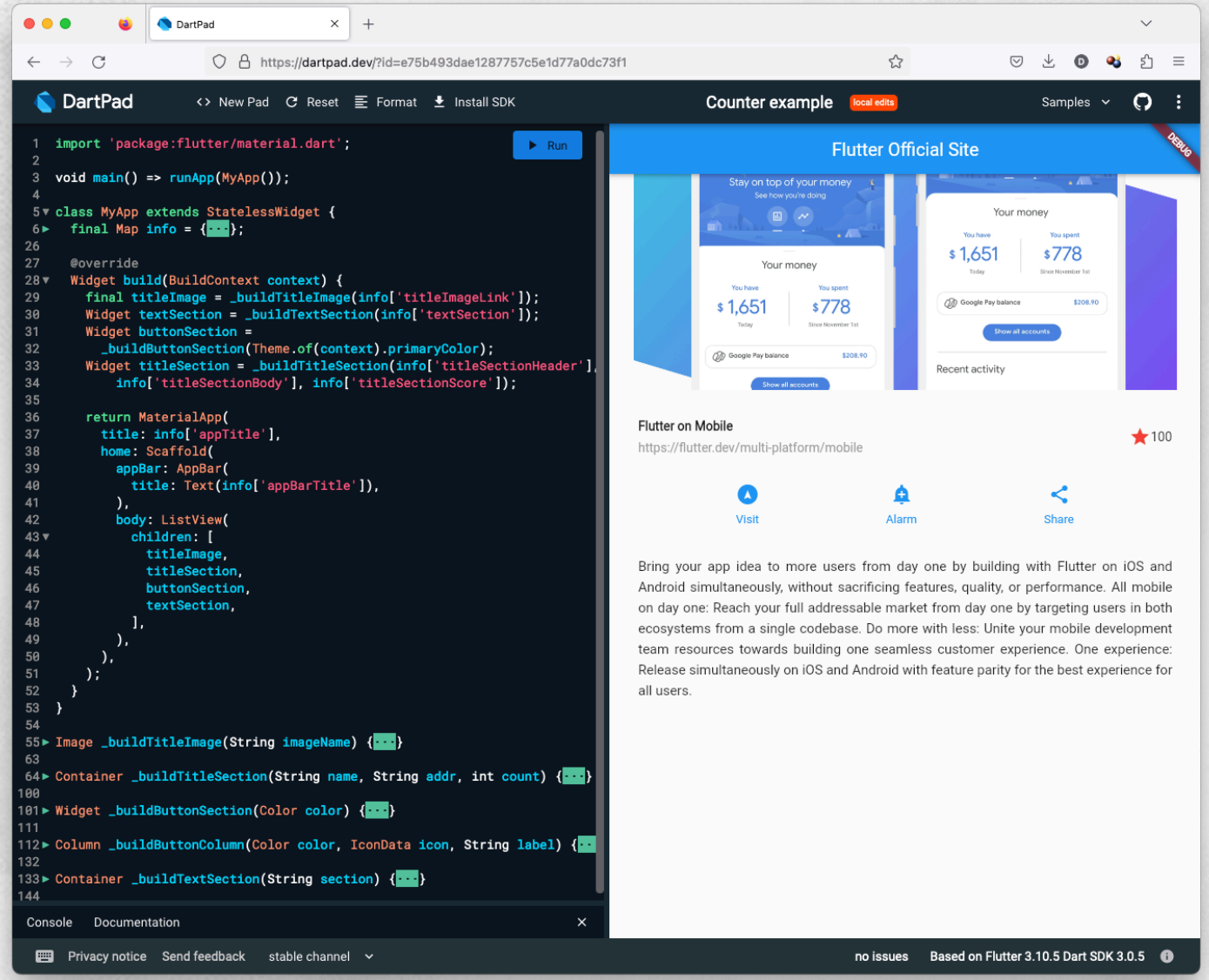
Stateless Widget 활용하기

- 원본 : https://flutter.dev/multi-platform/mobile



Stateless Widget 활용하기

- 앱 버전 : volume-E-chapter-06.dart

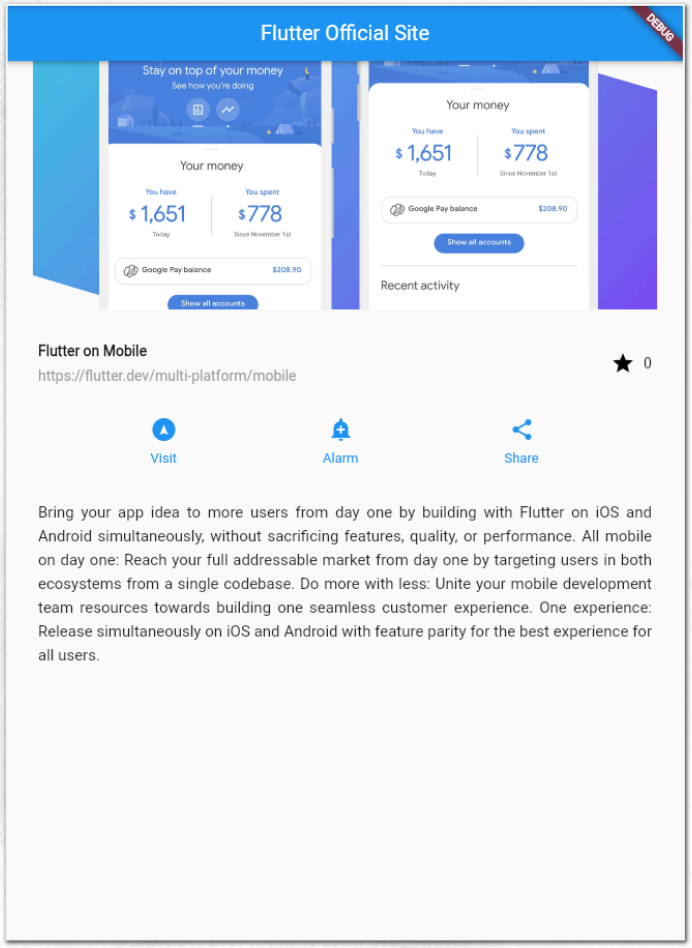


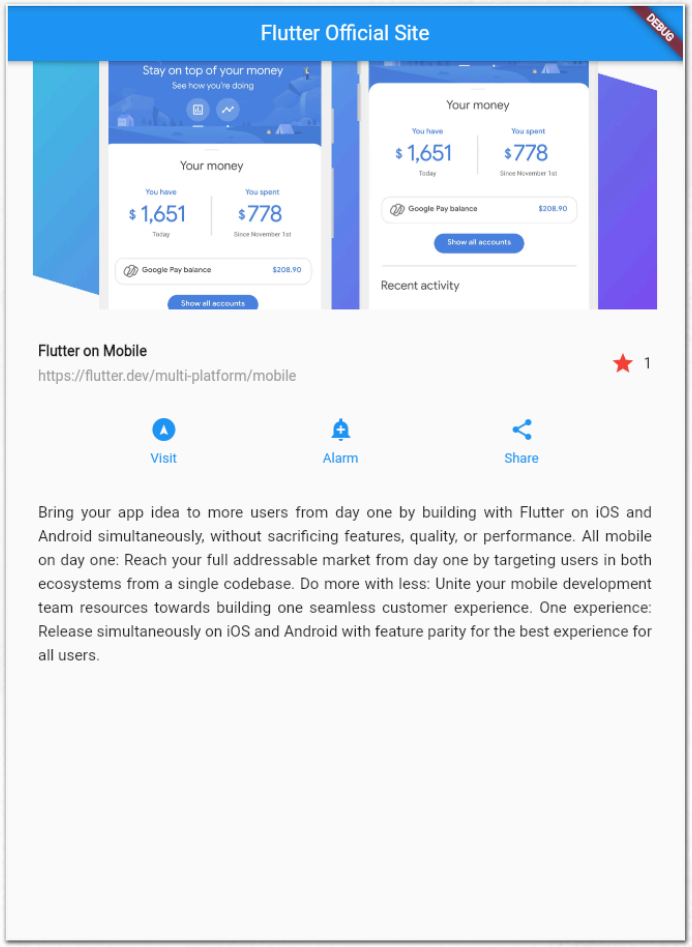
Stateless Widget 활용하기 (리뷰 및 실습)

- volume-E-chapter-06.dart

Stateful Widget 활용하기

- CHAPTER.6 프로그램에 변경 가능한 정보 처리 기능을 추가함
  - 화면 중간의 별표(좋아요) 클릭시 숫자가 0/1로 toggle 됨





Stateful & State Widget 적용하기

- 단계1 : 고정된 콘텐츠를 변경 가능한 StatefulWidget로 변경

```
Container _buildTitleSection(String name, String addr, int count) {
  return Container(
    padding: const EdgeInsets.all(32),
    child: Row(
      children: [
        Expanded(...),
/* old
        Icon(
          Icons.star,
          color: Colors.red[500],
        ),
        Text('$count');
*/
// New #1 : Start
        const Counter(),
// New #1 : End
      ],
    ),
  );
}
```

Stateful & State Widget 적용하기

단계2 : StatefulWidget 도입

```
150 // New #2 : Start
151
152 class Counter extends StatefulWidget {
153   counst Counter({
154     Key? key,
155   }) : super(key: key);(
156
157   @override
158   State<Counter> createState() => CounterState();
159 }
160
```

Stateful & State Widget 적용하기

단계3 : State 도입

```
161 class CounterState extends State<Counter> {
162   int _counter = 0;
163   bool _boolStatus = false;
164   Color _statusColor = Colors.black;
165
166   void _buttonPressed() {
167     setState(() {
168       if (_boolStatus == true) {,
169         _boolStatus = false;
170         _counter--;
171         _statusColor = Colors.black;
172       } else {
173         _boolStatus = true;
174         _counter++;
175         _statusColor = Colors.red;
176       }
177     });
178   }
179
180   @override
181   Widget build(BuildContext context) {
182     return Row(
183       children: [
184         IconButton(
185           icon: const Icon(Icons.star),
186           color: statusColor,
187           onPressed: _buttonPressed,
188         ),
189         Text('$_counter'),
190       ],
191     );
192   }
193 }
194
195 // New #2 : End
```


```
161 class CounterState extends State<Counter> {
162   int _counter = 0;
163   bool _boolStatus = false;
164   Color _statusColor = Colors.black;
165
166   void _buttonPressed() {
167     setState(() {
168       if (_boolStatus == true) {,
169         _boolStatus = false;
170         _counter--;
171         _statusColor = Colors.black;
172       } else {
173         _boolStatus = true;
174         _counter++;
175         _statusColor = Colors.red;
176       }
177     });
178   }
179
```

Stateful & State Widget 적용하기

단계3 : State 도입

```
161 class CounterState extends State<Counter> {
162   int _counter = 0;
163   bool _boolStatus = false;
164   Color _statusColor = Colors.black;
165
166   void _buttonPressed() {
167     setState(() {
168       if (_boolStatus == true) {,
169         _boolStatus = false;
170         _counter--;
171         _statusColor = Colors.black;
172       } else {
173         _boolStatus = true;
174         _counter++;
175         _statusColor = Colors.red;
176       }
177     });
178   }
179
180   @override
181   Widget build(BuildContext context) {
182     return Row(
183       children: [
184         IconButton(
185           icon: const Icon(Icons.star),
186           color: statusColor,
187           onPressed: _buttonPressed,
188         ),
189         Text('$_counter'),
190       ],
191     );
192   }
193 }
194
195 // New #2 : End
```

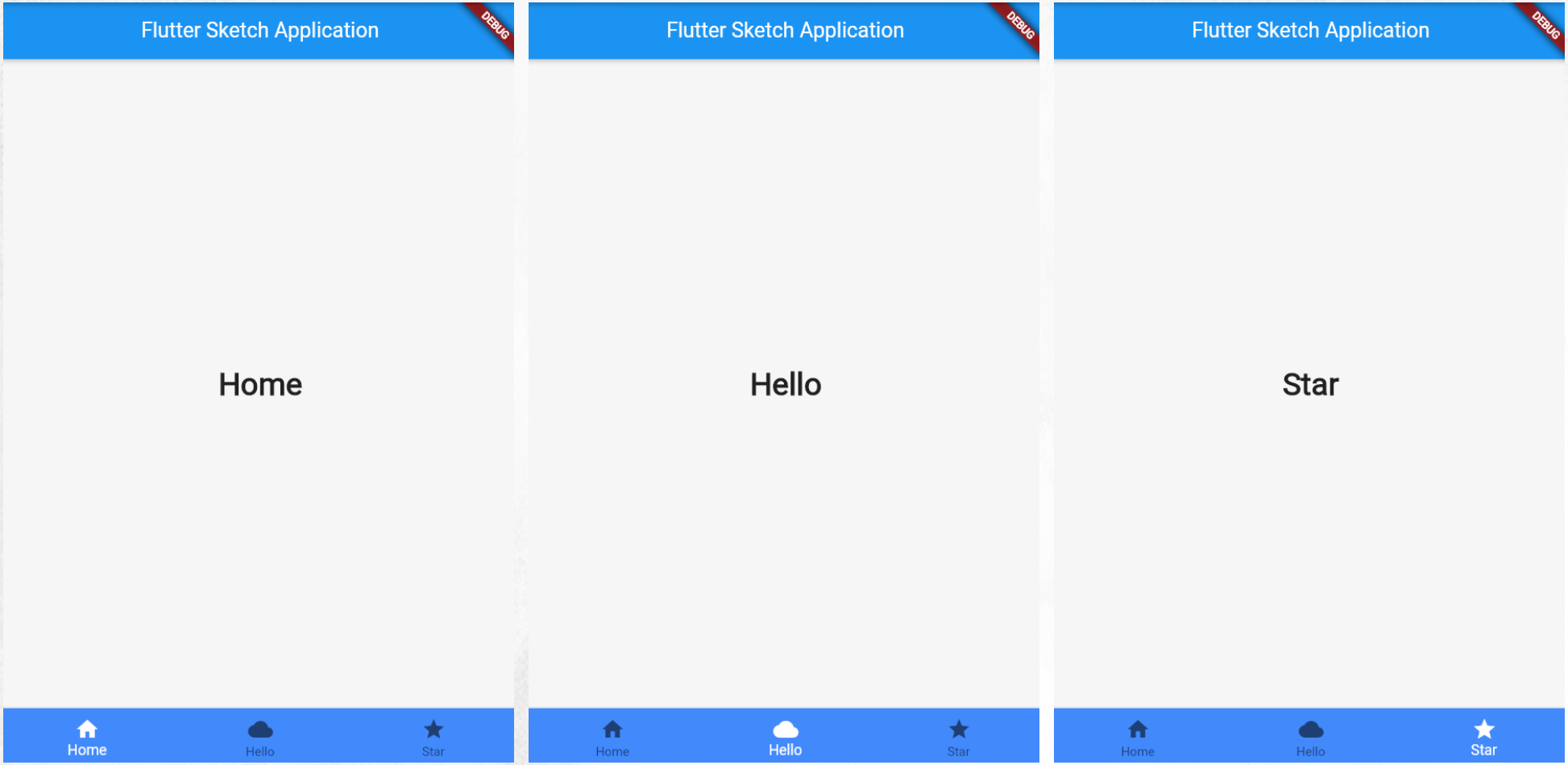
```
180   @override
181   Widget build(BuildContext context) {
182     return Row(
183       children: [
184         IconButton(
185           icon: const Icon(Icons.star),
186           color: statusColor,
187           onPressed: _buttonPressed,
188         ),
189         Text('$_counter'),
190       ],
191     );
192   }
193 }
194
195 // New #2 : End
```

Stateful Widget 활용하기 (리뷰 및 실습)

- volume-E-chapter-07.dart

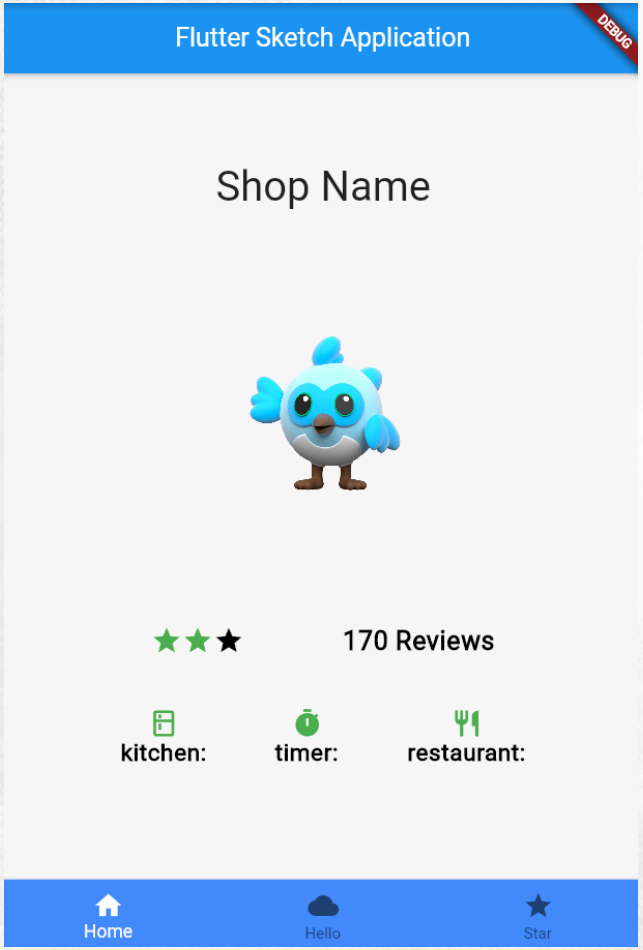
두고 두고 활용할 레퍼런스 프로그램 개발하기

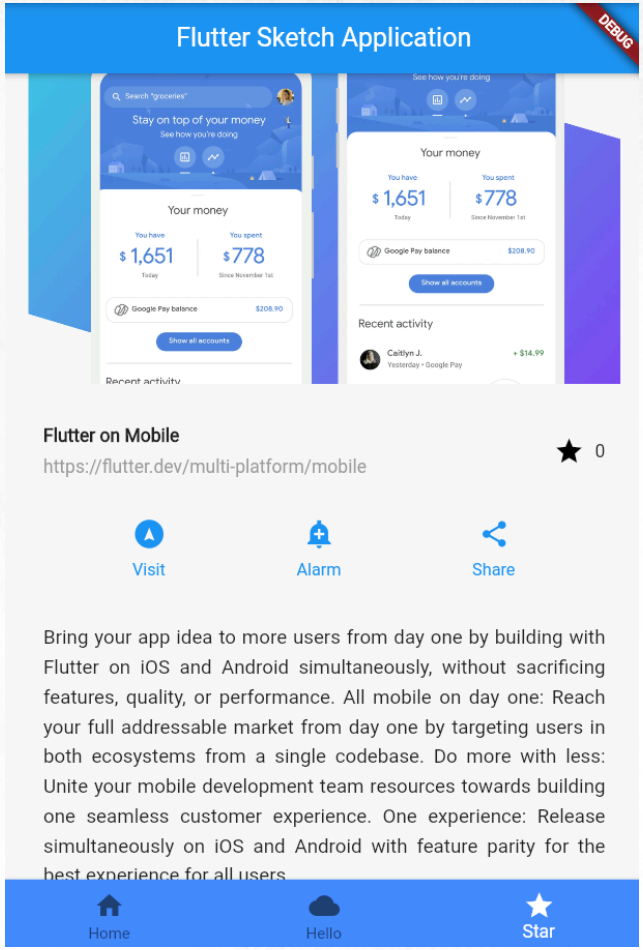
- volume-E-chapter-08-code-01.dart



두고 두고 활용할 레퍼런스 프로그램 개발하기

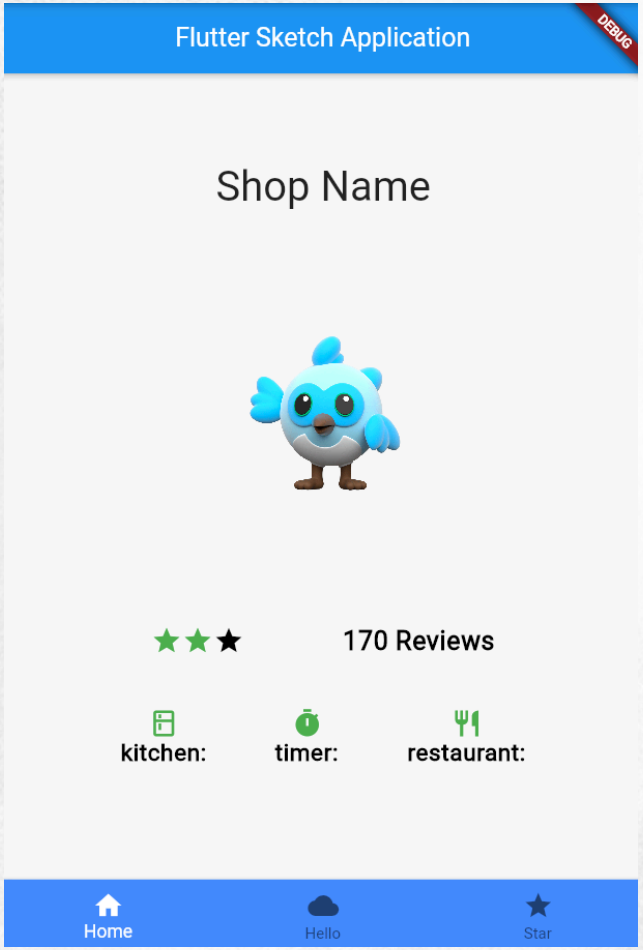
- volume-E-chapter-08-code-02.dart

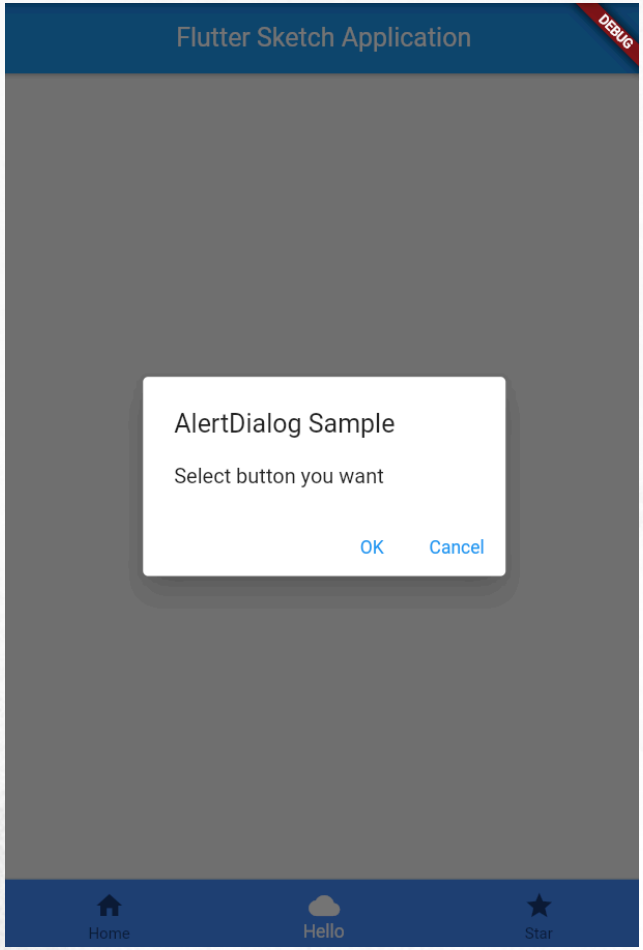


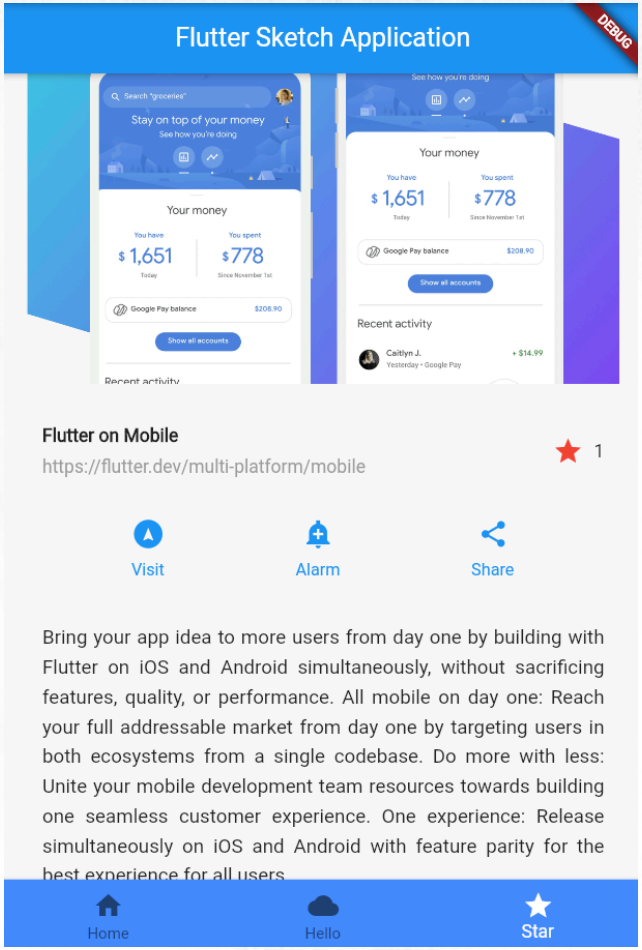


두고 두고 활용할 레퍼런스 프로그램 개발하기

- volume-E-chapter-08-code-02.dart (continued)







새롭게 등장한 위젯들

- PageController :

  https://api.flutter.dev/flutter/widgets/PageController-class.html

- PageView :

  https://api.flutter.dev/flutter/widgets/PageView-class.html

- BottomNavigationBarItem :

  https://api.flutter.dev/flutter/material/BottomNavigationBar-class.html

두고 두고 활용할 레퍼런스 프로그램 (리뷰 & 실습)

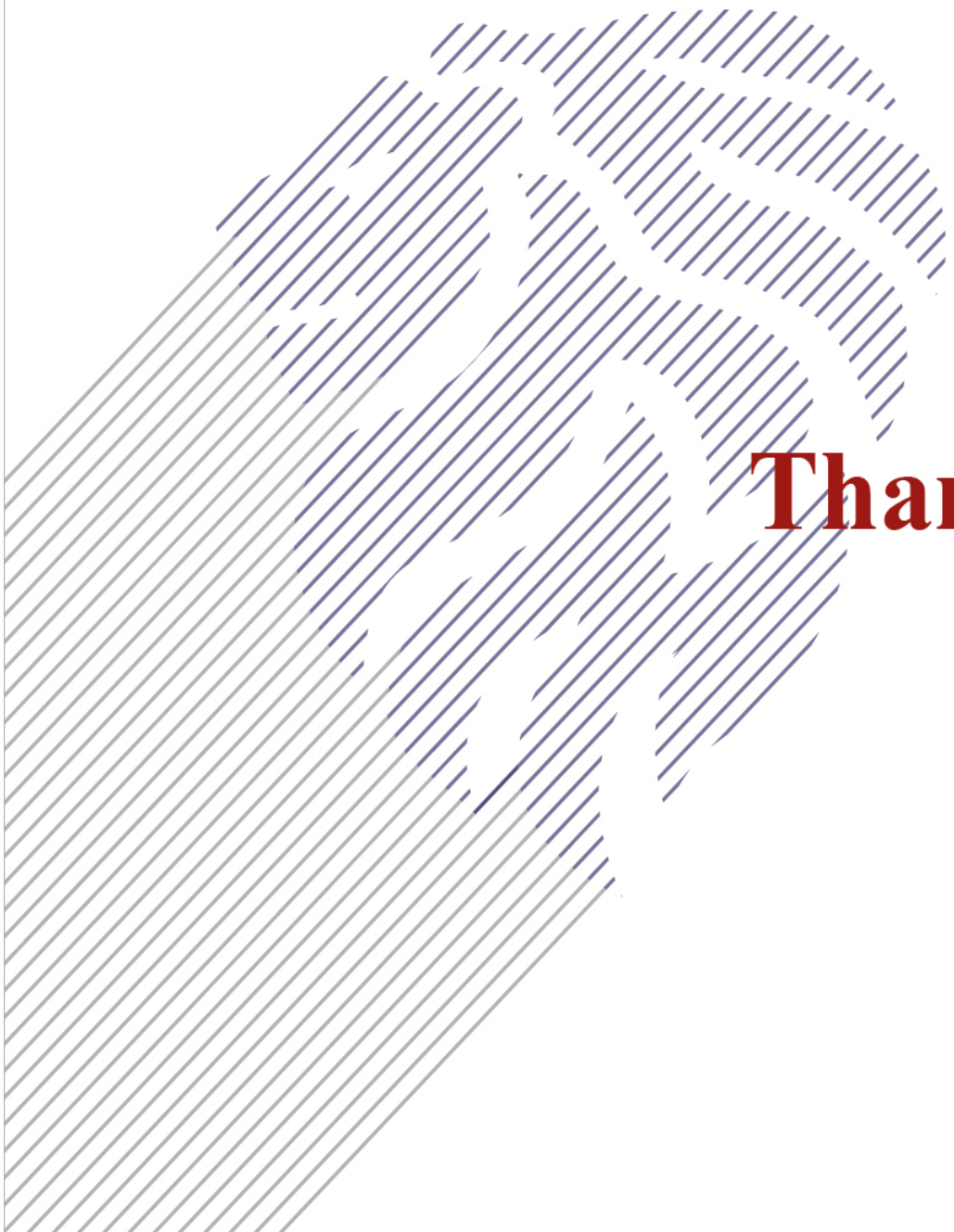
- volume-E-chapter-08-code-01.dart
- volume-E-chapter-08-code-02.dart

RECOMMENDED : Run @ Smart Phone & Tablet

- iOS, Android 등 모바일 환경에서 반드시 레퍼런스 프로그램을 실행하도록 함

참조: State 관리를 위한 진화된 기술들

- Riverbed : https://riverpod.dev/
- Provider : https://pub.dev/packages/provider
- BLOC : https://pub.dev/packages/flutter_bloc
- GetX : https://pub.dev/packages/get



Thank you

경희대학교
KYUNG HEE UNIVERSITY