



Full-Stack Service Programming

Lecture 4

Flutter 기반 웹 & 데스크톱 개발

2023. 09. 01

Sungwon Lee
Department of Software Convergence

교재 목차

- Volume.F Flutter로 Desktop App 개발
- Volume.G Flutter로 Web 서비스 개발



TEXT BOOK & SOURCE CODE

소스 코드

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

The screenshot shows a GitHub repository page for the user drsungwon named DART-FLUTTER-BOOK. The repository is public and has 1 branch and 0 tags. The 'Code' tab is selected. A recent commit by drsungwon, titled 'Update README.md', is shown, dated May 9, 2024, with 12 commits. Below the commit list, there are several folder entries under 'volume-B-chapter-02' through 'volume-B-chapter-08', all added via upload. To the right of the code area, there's an 'About' section with a brief description in Korean: '플스택 개발이 쉬워지는 다트 & 플러터 (영진단 캠)'. This section also lists the repository's stats: 1 watching, 1 fork, and 3 stars. It includes links for 'Readme', 'MIT license', and 'Report repository'. At the bottom, it says 'No releases published'.

GitHub - drsungwon/DART-FLUTTER-BOOK

Product Solutions Open Source Pricing

Search Sign in Sign up

drsunwon / DART-FLUTTER-BOOK Public

Notifications Fork 1 Star 3

Code Issues Pull requests Actions Projects Security Insights

main 1 branch 0 tags Go to file Code

drsunwon Update README.md 62b9a46 on May 9 12 commits

BOOKTITLE Add files via upload 2 months ago

volume-B-chapter-02 Add files via upload 2 months ago

volume-B-chapter-03 Add files via upload 2 months ago

volume-B-chapter-04 Add files via upload 2 months ago

volume-B-chapter-05 Add files via upload 2 months ago

volume-B-chapter-06 Add files via upload 2 months ago

volume-B-chapter-07 Add files via upload 2 months ago

volume-B-chapter-08 Add files via upload 2 months ago

About

플스택 개발이 쉬워지는 다트 & 플러터 (영진단 캠)

Readme MIT license 3 stars 1 watching 1 fork Report repository

No releases published

VOLUME.F CHAPTER.1

Flutter for Desktop 이해하기

```
drsungwon~$ flutter create mydesktopapp
Signing iOS app for device deployment using developer identity: "iPhone Developer: Sungwon Lee (552U84J2MQ)"
Creating project mydesktopapp...
Resolving dependencies in mydesktopapp... (1.1s)
Got dependencies in mydesktopapp.
Wrote 129 files.
```

All done!
You can find general documentation for Flutter at: <https://docs.flutter.dev/>
Detailed API documentation is available at: <https://api.flutter.dev/>
If you prefer video documentation, consider: <https://www.youtube.com/c/flutterdev>

In order to run your application, type:

```
$ cd mydesktopapp
$ flutter run
```

Your application code is in mydesktopapp/lib/main.dart.

```
drsungwon~$ cd mydesktopapp
drsungwon~$ flutter devices
2 connected devices:
```

```
macOS (desktop) • macos • darwin-arm64 • macOS 13.4.1 22F82 darwin-arm64 (Rosetta)
Chrome (web) • chrome • web-javascript • Google Chrome 114.0.5735.198
```

No wireless devices were found.

```
drsungwon~$ flutter run -d macos
Launching lib/main.dart on macOS in debug mode...
Building macOS application...
Syncing files to device macOS... 45ms
```

Flutter run key commands.

```
r Hot reload. 🔥🔥
R Hot restart.
h List all available interactive commands.
d Detach (terminate "flutter run" but leave application running).
c Clear the screen
q Quit (terminate the application on the device).
```

A Dart VM Service on macOS is available at: <http://127.0.0.1:49728/JEu2F3zUJ1M=/>
The Flutter DevTools debugger and profiler on macOS is available at: <http://127.0.0.1:9100?uri=http://127.0.0.1:49728/JEu2F3zUJ1M=/>

Lost connection to device.

```
drsungwon~$ flutter build macos
Building macOS application...
drsungwon~$
```

VOLUME.F CHAPTER.1

Flutter for Desktop 이해하기

```
drsungwon~$ flutter create mydesktopapp
Signing iOS app for device deployment using developer id...
Creating project mydesktopapp...
Resolving dependencies in mydesktopapp... (1.1s)
Got dependencies in mydesktopapp.
Wrote 129 files.
```

All done!
You can find general documentation for Flutter at: <https://flutter.dev>.
Detailed API documentation is available at: <https://api.flutter.dev>.
If you prefer video documentation, consider: <https://www.youtube.com/flutter>.

In order to run your application, type:

```
$ cd mydesktopapp
$ flutter run
```

Your application code is in mydesktopapp/lib/main.dart.

```
drsungwon~$ cd mydesktopapp
drsungwon~$ flutter devices
2 connected devices:

macOS (desktop) • macos • darwin-arm64 • macOS 13.4
Chrome (web)    • chrome • web-javascript • Google Chrome

No wireless devices were found.
```

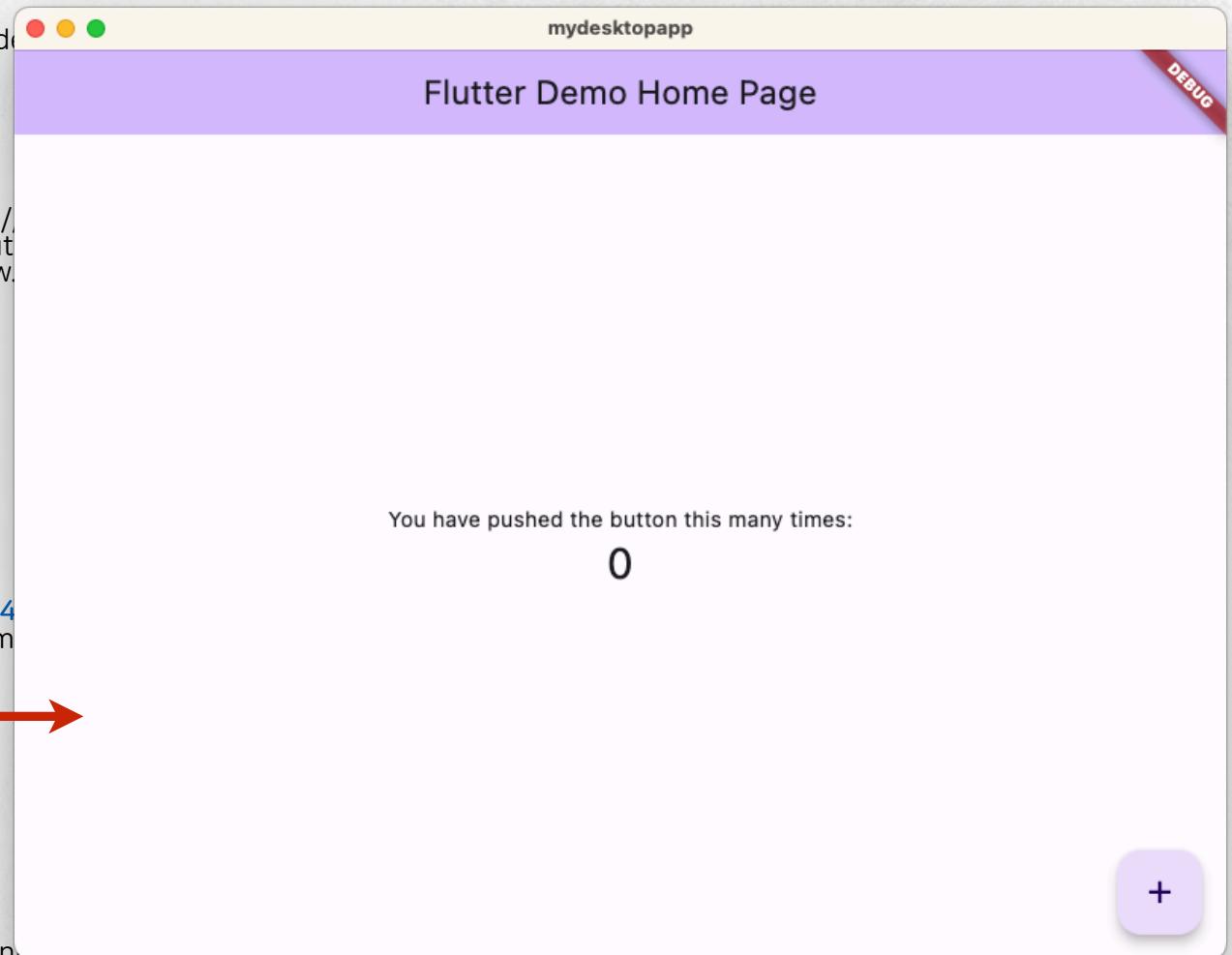
```
drsungwon~$ flutter run -d macos
Launching lib/main.dart on macOS in debug mode...
Building macOS application...
Syncing files to device macOS...          45ms
```

Flutter run key commands.

- r Hot reload. 🔥🔥
- R Hot restart.
- h List all available interactive commands.
- d Detach (terminate "flutter run" but leave application running...).
- c Clear the screen.
- q Quit (terminate the application on the device).

A Dart VM Service on macOS is available at: <http://127.0.0.1:49728/JEu2F3zUJ1M=/>
The Flutter DevTools debugger and profiler on macOS is available at: <http://127.0.0.1:9100?uri=http://127.0.0.1:49728/JEu2F3zUJ1M=/>
Lost connection to device.

```
drsungwon~$ flutter build macos
Building macOS application...
drsungwon~$
```



VOLUME.F CHAPTER.1

Flutter for Desktop 이해하기

```
drsungwon~$ flutter create mydesktopapp
```

```
Signed iOS app for device deployment using developer profile  
Creating project mydesktopapp...  
Resolving dependencies in mydesktopapp... (1.1s)  
Got dependencies in mydesktopapp.  
Wrote 129 files.
```

All done!

You can find general documentation for Flutter at: <https://flutter.dev/>
Detailed API documentation is available at: <https://api.flutter.dev/>
If you prefer video documentation, consider: <https://www.youtube.com/watch?v=IjwDyJzXGQk>

In order to run your application, type:

```
$ cd mydesktopapp  
$ flutter run
```

Your application code is in mydesktopapp/lib/main.dart

```
drsungwon~$ cd mydesktopapp  
drsungwon~$ flutter devices  
2 connected devices:
```

```
macOS (desktop) • macos • darwin-arm64 • macOS  
Chrome (web) • chrome • web-javascript • Google Chrome
```

No wireless devices were found.

```
drsungwon~$ flutter run -d macos  
Launching lib/main.dart on macOS in debug mode...  
Building macOS application...  
Syncing files to device macOS...
```

Flutter run key commands.

r Hot reload. 🔥🔥

R Hot restart.

h List all available interactive commands.

d Detach (terminate "flutter run" but leave application running).

c Clear the screen.

q Quit (terminate the application on the device).

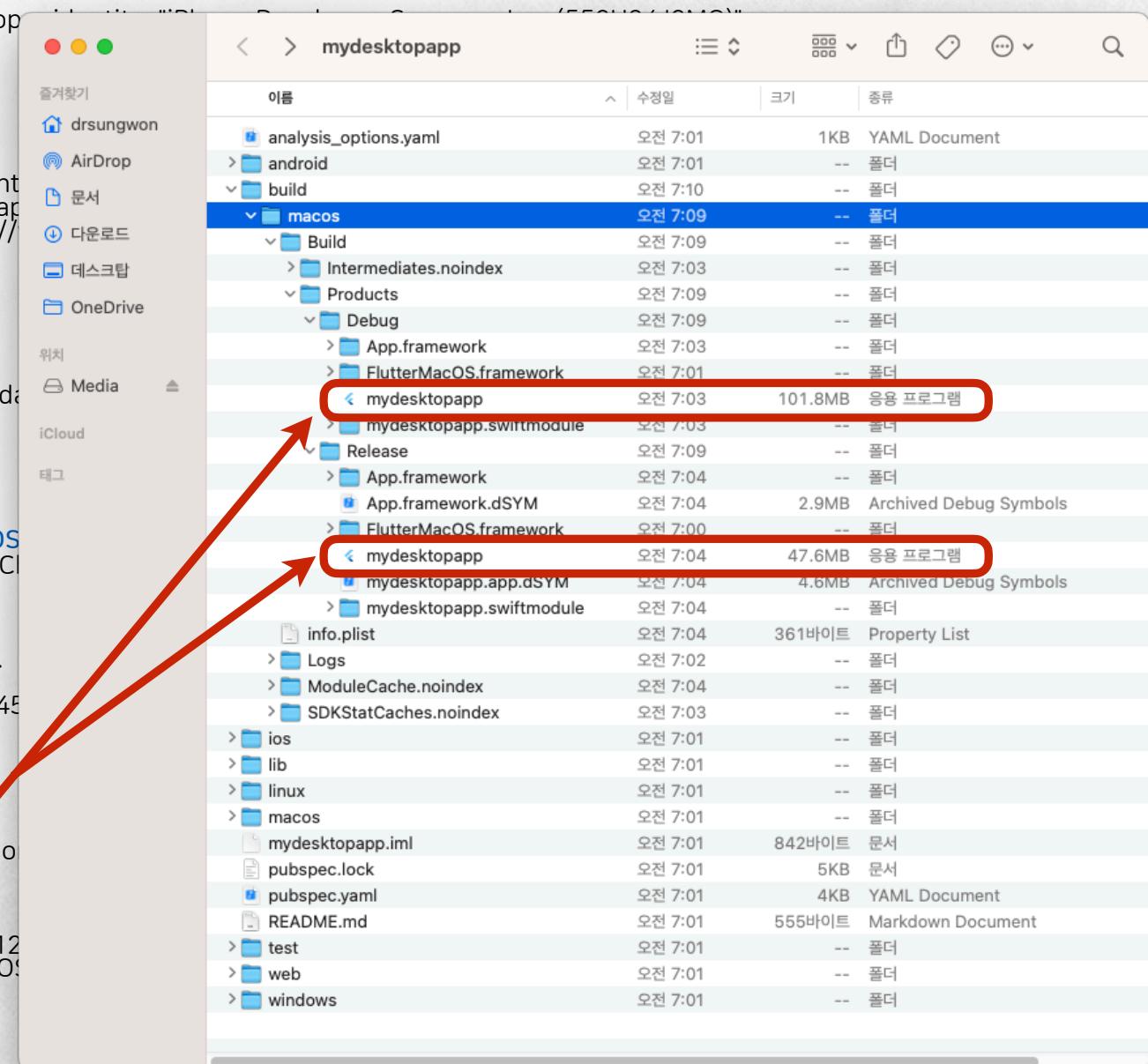
A Dart VM Service on macOS is available at: <http://127.0.0.1:4567>

The Flutter DevTools debugger and profiler on macOS

Lost connection to device.

```
drsungwon~$ flutter build macos  
Building macOS application...
```

```
drsungwon~$
```



VOLUME.F CHAPTER.1

Flutter for Desktop 이해하기

```
drsungwon~$ flutter create mydesktopapp
Signing iOS app for device deployment using developer identity: "iPhone Developer: Su
Creating project mydesktopapp...
Resolving dependencies in mydesktopapp... (1.1s)
Got dependencies in mydesktopapp.
Wrote 129 files.
```

```
All done!
You can find general documentation for Flutter at: https://docs.flutter.dev/
Detailed API documentation is available at: https://api.flutter.dev/
If you prefer video documentation, consider: https://www.youtube.com/c/flutterdev
```

In order to run your application, type:

```
$ cd mydesktopapp
$ flutter run
```

Your application code is in mydesktopapp/lib/main.dart.

```
drsungwon~$ cd mydesktopapp
drsungwon~$ flutter devices
2 connected devices:
```

```
macOS (desktop) • macos • darwin-arm64 • macOS 13.4.1 22F82 darwin-arm64 (Rosetta)
Chrome (web) • chrome • web-javascript • Google Chrome 114.0.5735.198
```

No wireless devices were found.

```
drsungwon~$ flutter run -d macos
Launching lib/main.dart on macOS in debug mode...
Building macOS application...
Syncing files to device macOS... 45ms
```

Flutter run key commands.

```
r Hot reload. 🔥🔥
R Hot restart.
h List all available interactive commands.
d Detach (terminate "flutter run" but leave application running).
c Clear the screen
q Quit (terminate the application on the device).
```

```
A Dart VM Service on macOS is available at: http://127.0.0.1:49728/JEu2F3zUJ1M=/
The Flutter DevTools debugger and profiler on macOS is available at: http://127.0.0.1:9100?uri=http://127.0.0.1:49728/JEu2F3zUJ1M=/
Lost connection to device.
drsungwon~$ flutter build macos
Building macOS application...
drsungwon~$
```

MS Windows 경우,

```
flutter run -d windows
flutter build windows
/build/windows/runner/release/
```

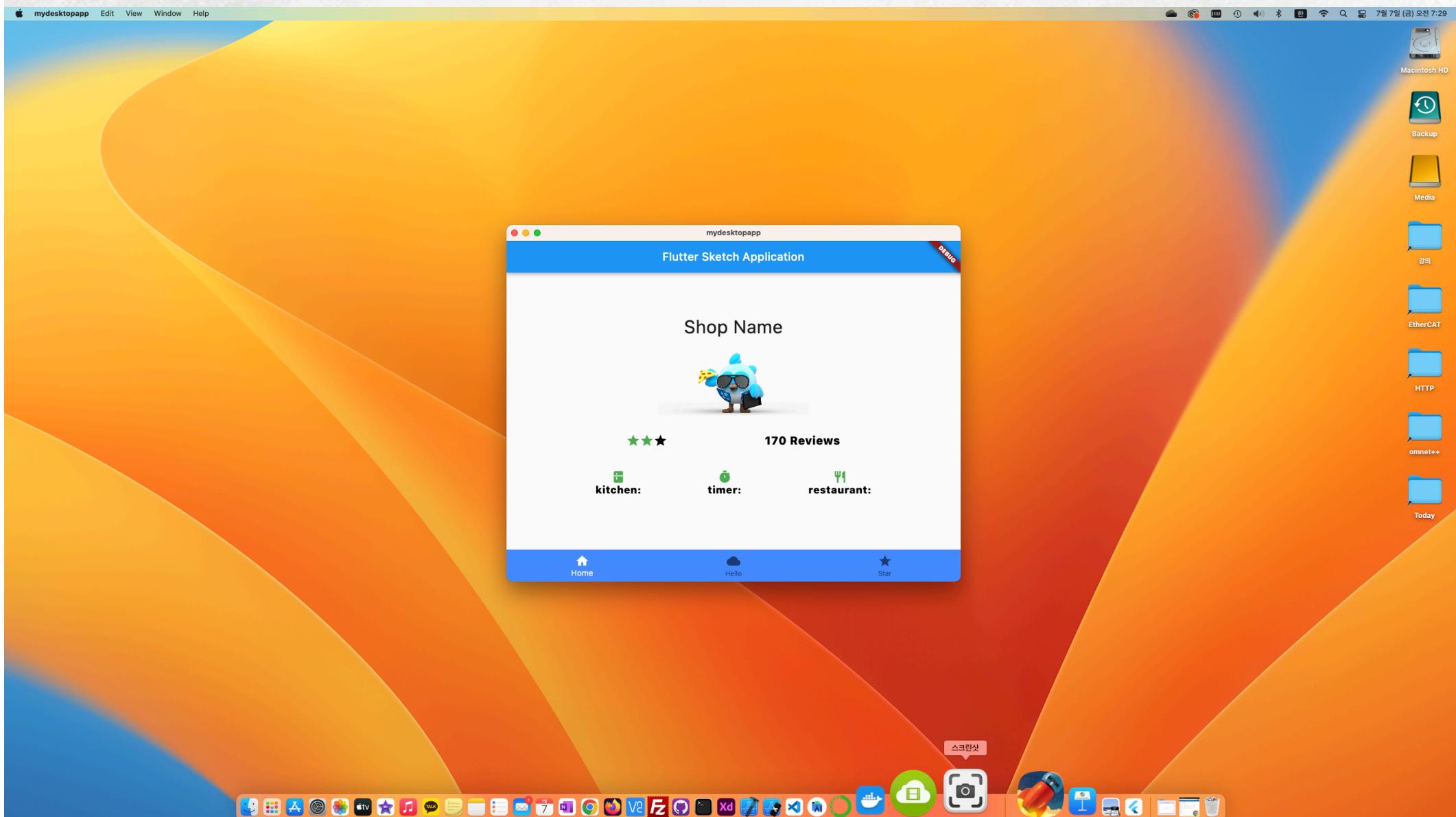
Flutter for Desktop 이해하기 (리뷰 및 실습)

- 신규 프로젝트 생성
- {프로젝트폴더}/images/flutter_00.png 복사
- {프로젝트폴더}/images/flutter_01.png 복사
- {프로젝트폴더}/lib/main.dart 복사
- {프로젝트폴더}/pubspec.yaml 수정
 - ▣ 프로젝트에 이미지 등 리소스를 포함하는 방법 예시

```
flutter:  
  assets:  
    - images/flutter_00.png  
    - images/flutter_01.png
```

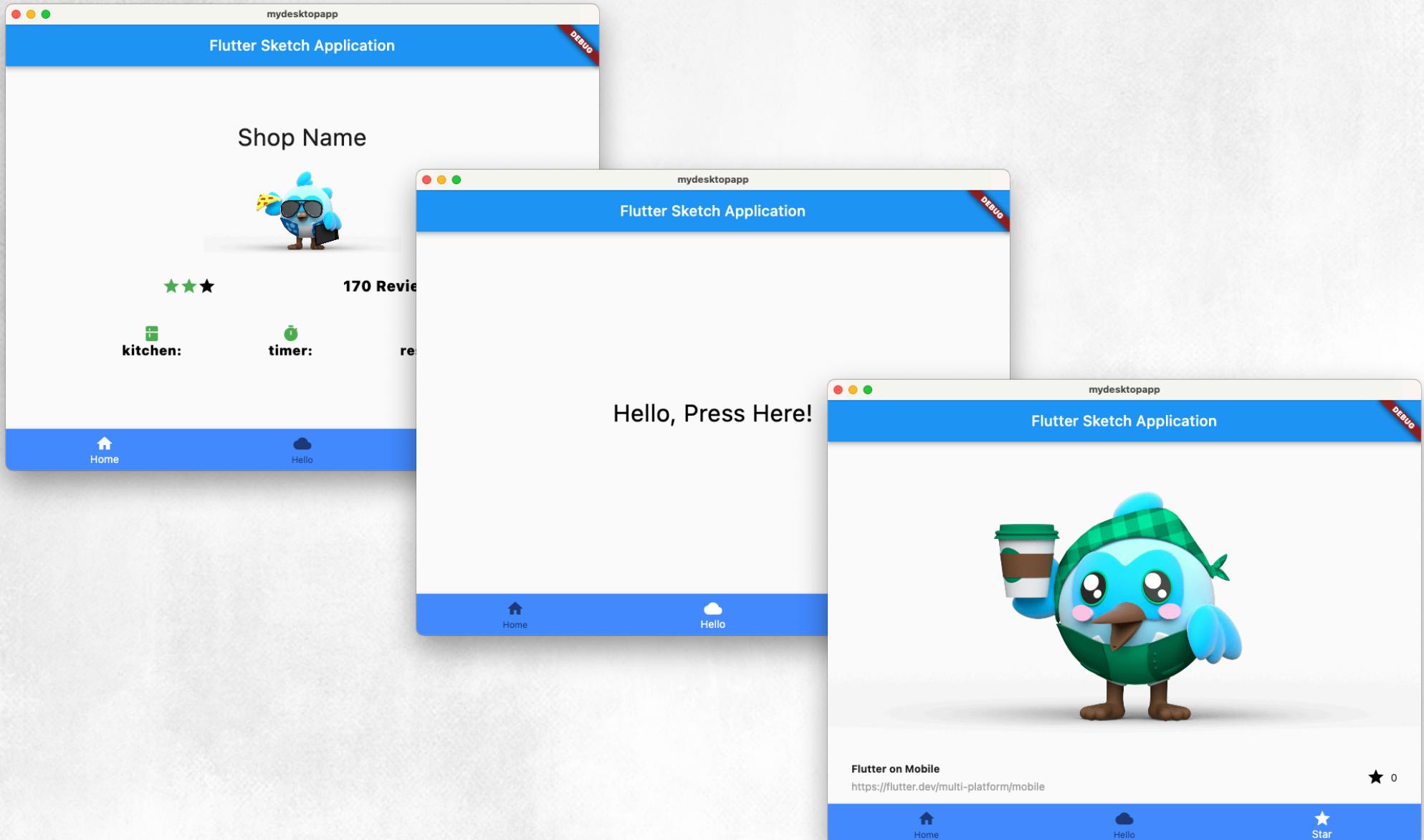
- 프로젝트 실행

Flutter for Desktop 이해하기 (리뷰 및 실습)



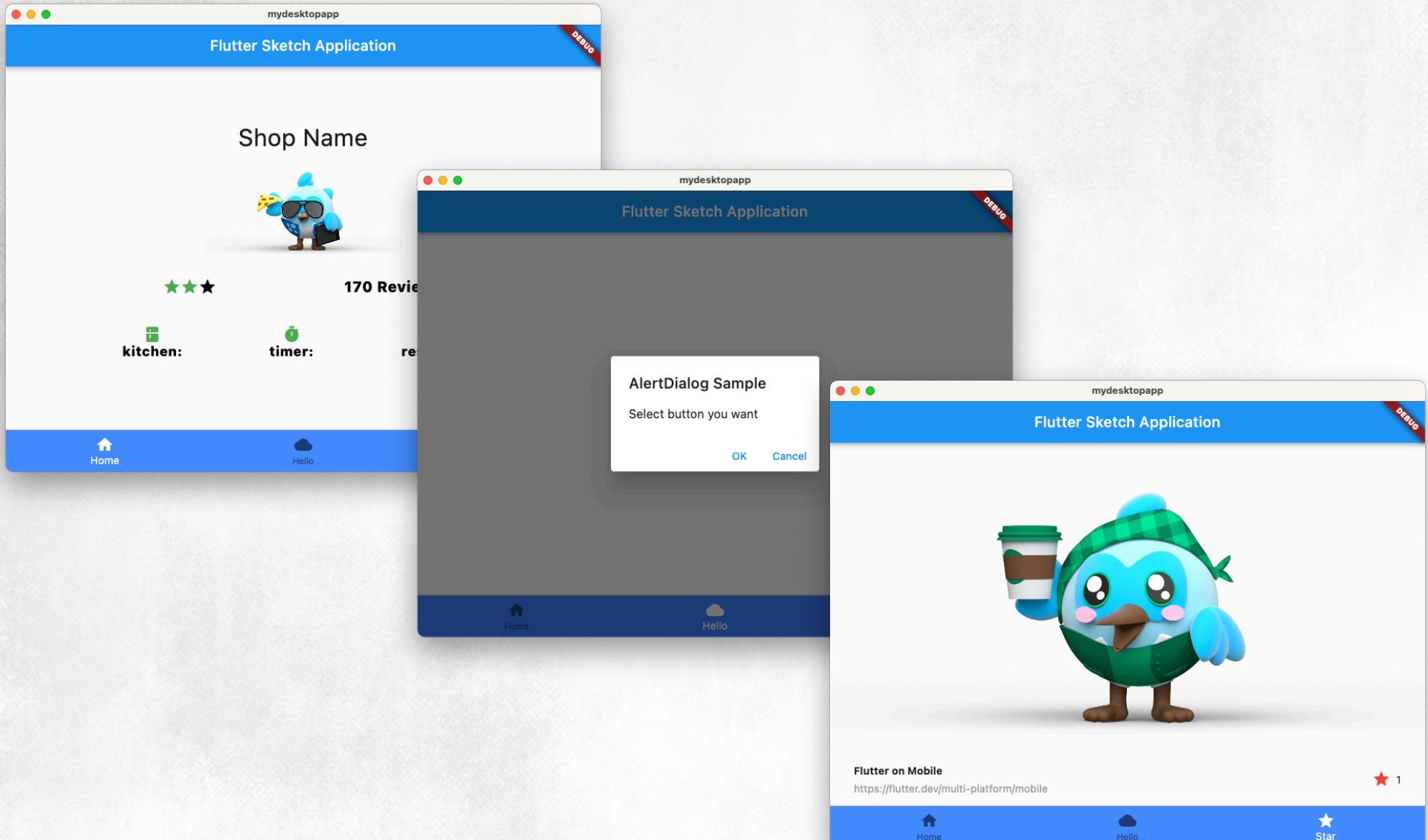
VOLUME.F CHAPTER.1

Flutter for Desktop 이해하기 (리뷰 및 실습)



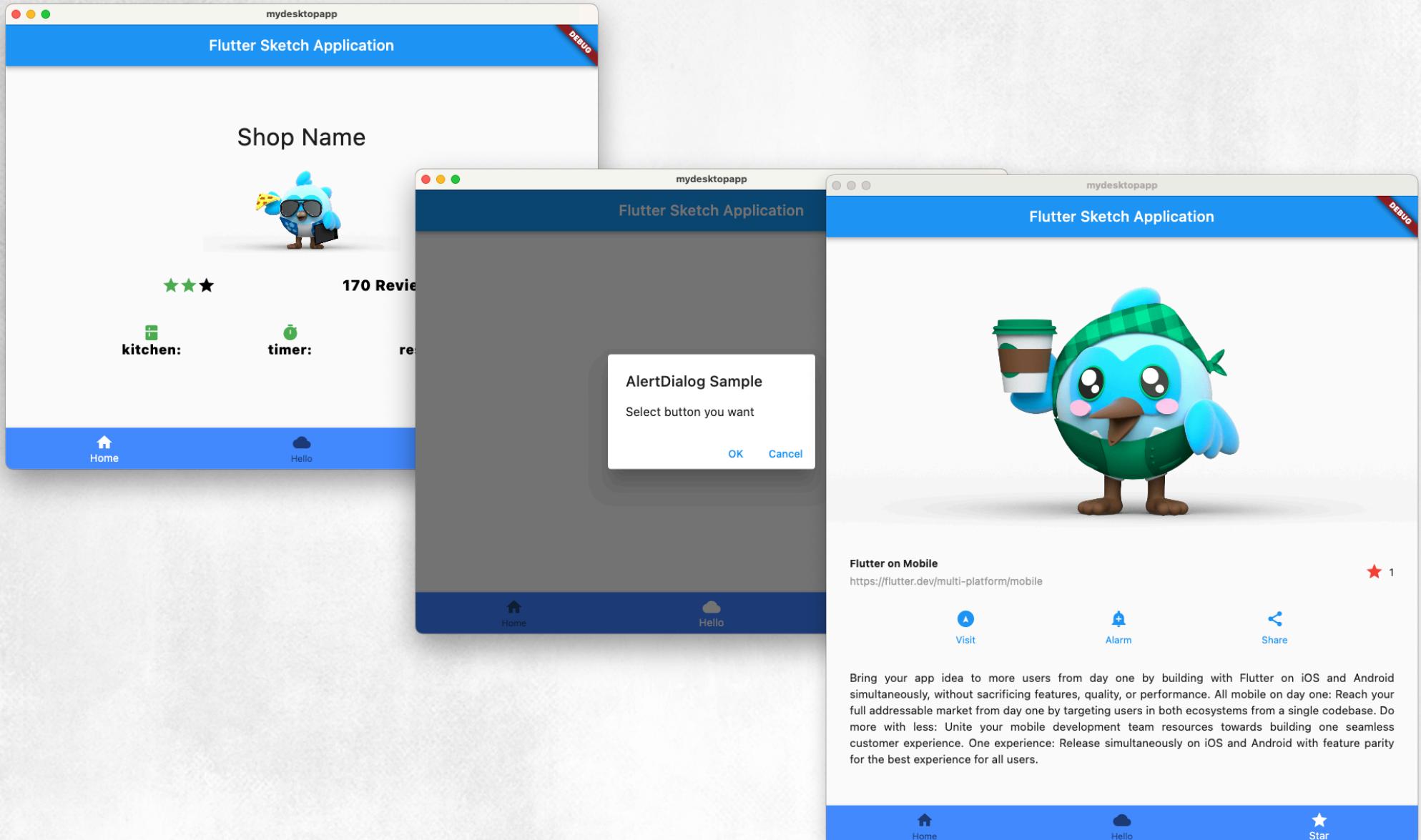
VOLUME.F CHAPTER.1

Flutter for Desktop 이해하기 (리뷰 및 실습)



VOLUME.F CHAPTER.1

Flutter for Desktop 이해하기 (리뷰 및 실습)



Flutter 아키텍처 이해하기

- **Build performantly :**

Get native-compiled performance without large browser engine dependencies.

- **Target more users**

Reach more users across the Windows, Mac App, and Linux Snap stores.

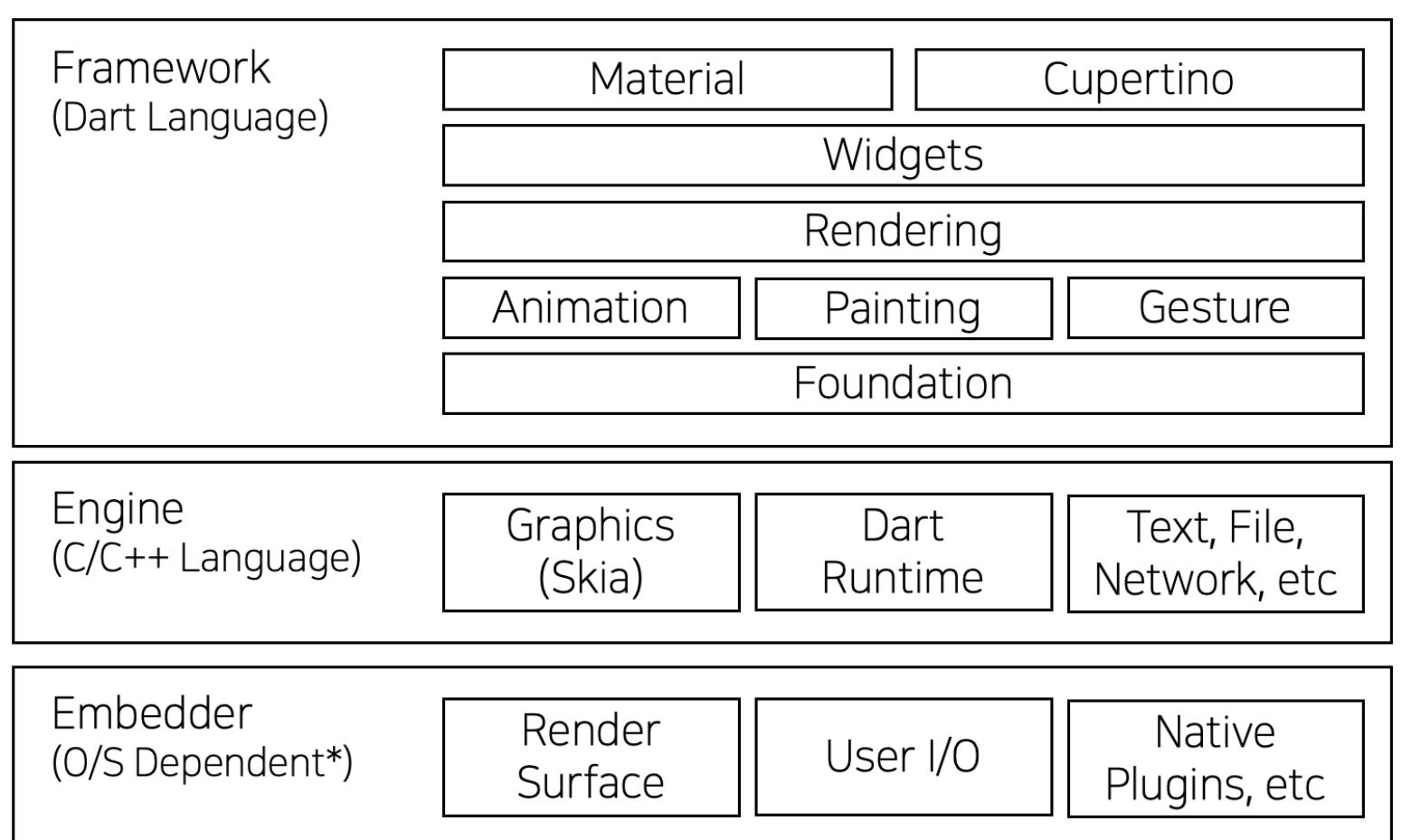
- **Native functionality**

Get full access to the underlying Win32, Cocoa, or UNIX platform APIs.

Flutter 아키텍처 이해하기

* 참고: Flutter 3.10 부터 Skia는 Impeller로 대체됨

FLUTTER ARCHITECTURE



*Java/C++ for Android, Objective-C/Objective-C++ for iOS & macOS, and C++ for Windows & Linux

Flutter for Desktop 참조자료

- Flutter on Desktop

<https://flutter.dev/multi-platform/desktop>

- Desktop support for Flutter

<https://docs.flutter.dev/platform-integration/desktop>

- Impeller redering engine

<https://docs.flutter.dev/perf/impeller>

Flutter for Web 이해하기

```
drsungwon~$ flutter create mydesktopapp
```

Signing iOS app for device deployment using developer identity: "iPhone Developer: Sungwon Lee (552U84J2MQ)"

Creating project mydesktopapp...

Resolving dependencies in mydesktopapp... (1.1s)

Got dependencies in mydesktopapp.

Wrote 129 files.

All done!

You can find general documentation for Flutter at: <https://docs.flutter.dev/>

Detailed API documentation is available at: <https://api.flutter.dev/>

If you prefer video documentation, consider: <https://www.youtube.com/c/flutterdev>

In order to run your application, type:

```
$ cd mydesktopapp  
$ flutter run
```

Your application code is in mydesktopapp/lib/main.dart.

```
drsungwon~$ cd mydesktopapp
```

```
drsungwon~$ flutter devices
```

2 connected devices:

macOS (desktop) • macos • darwin-arm64 • macOS 13.4.1 22F82 darwin-arm64 (Rosetta)

Chrome (web) • chrome • web-javascript • Google Chrome 114.0.5735.198

No wireless devices were found.

```
drsungwon~$ flutter run -d chrome
```

Launching lib/main.dart on macOS in debug mode...

Building macOS application...

Syncing files to device macOS... 45ms

Flutter run key commands.

r Hot reload. 🔥🔥

R Hot restart.

h List all available interactive commands.

d Detach (terminate "flutter run" but leave application running).

c Clear the screen

q Quit (terminate the application on the device).

A Dart VM Service on macOS is available at: <http://127.0.0.1:49728/JEu2F3zUJ1M=/>

The Flutter DevTools debugger and profiler on macOS is available at: <http://127.0.0.1:9100?uri=http://127.0.0.1:49728/JEu2F3zUJ1M=/>

Lost connection to device.

```
drsungwon~$ flutter build web
```

Building macOS application...

```
drsungwon~$
```

VOLUME.G CHAPTER.1~4

Flutter for Web 이해하기

```
drsungwon~$ flutter create mydesktopapp
```

Siging iOS app for device deployment using developer identity: "iPhone Developer: Sungwon Lee (552U84J2MQ)"

Creating project mydesktopapp...

Resolving dependencies in mydesktopapp... (1.1s)

Got dependencies in mydesktopapp.

Wrote 129 files.

All done!

You can find general documentation for Flutter at: <https://docs.flutter.dev>

Detailed API documentation is available at: <https://api.flutter.dev>

If you prefer video documentation, consider: <https://www.youtube.com/>

In order to run your application, type:

```
$ cd mydesktopapp  
$ flutter run
```

Your application code is in mydesktopapp/lib/main.dart.

```
drsungwon~$ cd mydesktopapp
```

```
drsungwon~$ flutter devices
```

2 connected devices:

```
macOS (desktop) • macos • darwin-arm64 • macOS 13.4.1 22F82 darw  
Chrome (web) • chrome • web-javascript • Google Chrome 114.0.5735.
```

No wireless devices were found.

```
drsungwon~$ flutter run -d chrome
```

Launching lib/main.dart on macOS in debug mode...

Building macOS application...

Syncing files to device macOS...

45ms

Flutter run key commands.

r Hot reload. 🔥🔥

R Hot restart.

h List all available interactive commands.

d Detach (terminate "flutter run" but leave application running).

c Clear the screen

q Quit (terminate the application on the device).

A Dart VM Service on macOS is available at: <http://127.0.0.1:49728/JEui>

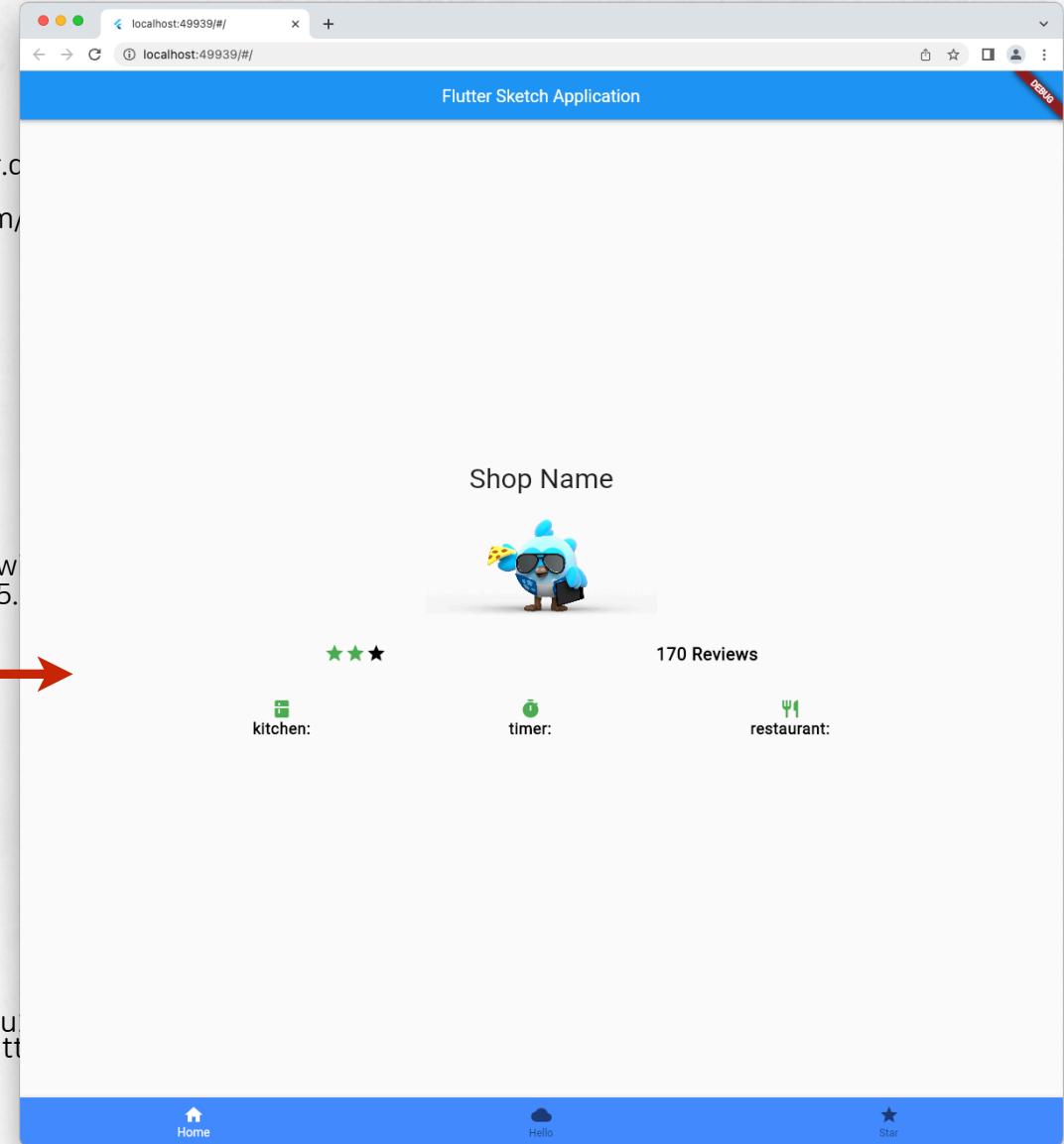
The Flutter DevTools debugger and profiler on macOS is available at: <http://127.0.0.1:49253/flutter>

Lost connection to device.

```
drsungwon~$ flutter build web
```

Building macOS application...

```
drsungwon~$
```



VOLUME.G CHAPTER.1~4

Flutter for Web 이해하기

```
drsungwon~$ flutter create mydesktopapp
```

Siging iOS app for device deployment using developer identity: "iPhone Developer: Sungwon Lee (552U84J2MQ)"

Creating project mydesktopapp...

Resolving dependencies in mydesktopapp... (1.1s)

Got dependencies in mydesktopapp.

Wrote 129 files.

All done!

You can find general documentation for Flutter at: <https://flutter.dev/>

Detailed API documentation is available at: <https://api.flutter.dev/>

If you prefer video documentation, consider: <https://www.youtube.com/flutter>

In order to run your application, type:

```
$ cd mydesktopapp  
$ flutter run
```

Your application code is in mydesktopapp/lib/main.dart

```
drsungwon~$ cd mydesktopapp
```

```
drsungwon~$ flutter devices
```

2 connected devices:

```
macOS (desktop) • macos • darwin-arm64 • macOS  
Chrome (web) • chrome • web-javascript • Google Cl
```

No wireless devices were found.

```
drsungwon~$ flutter run -d chrome
```

Launching lib/main.dart on macOS in debug mode...

Building macOS application...

Syncing files to device macOS...

Flutter run key commands.

r Hot reload. 🔥🔥

R Hot restart.

h List all available interactive commands.

d Detach (terminate "flutter run" but leave application running).

c Clear the screen.

q Quit (terminate the application on the device).

A Dart VM Service on macOS is available at: <http://127.0.0.1:41983>

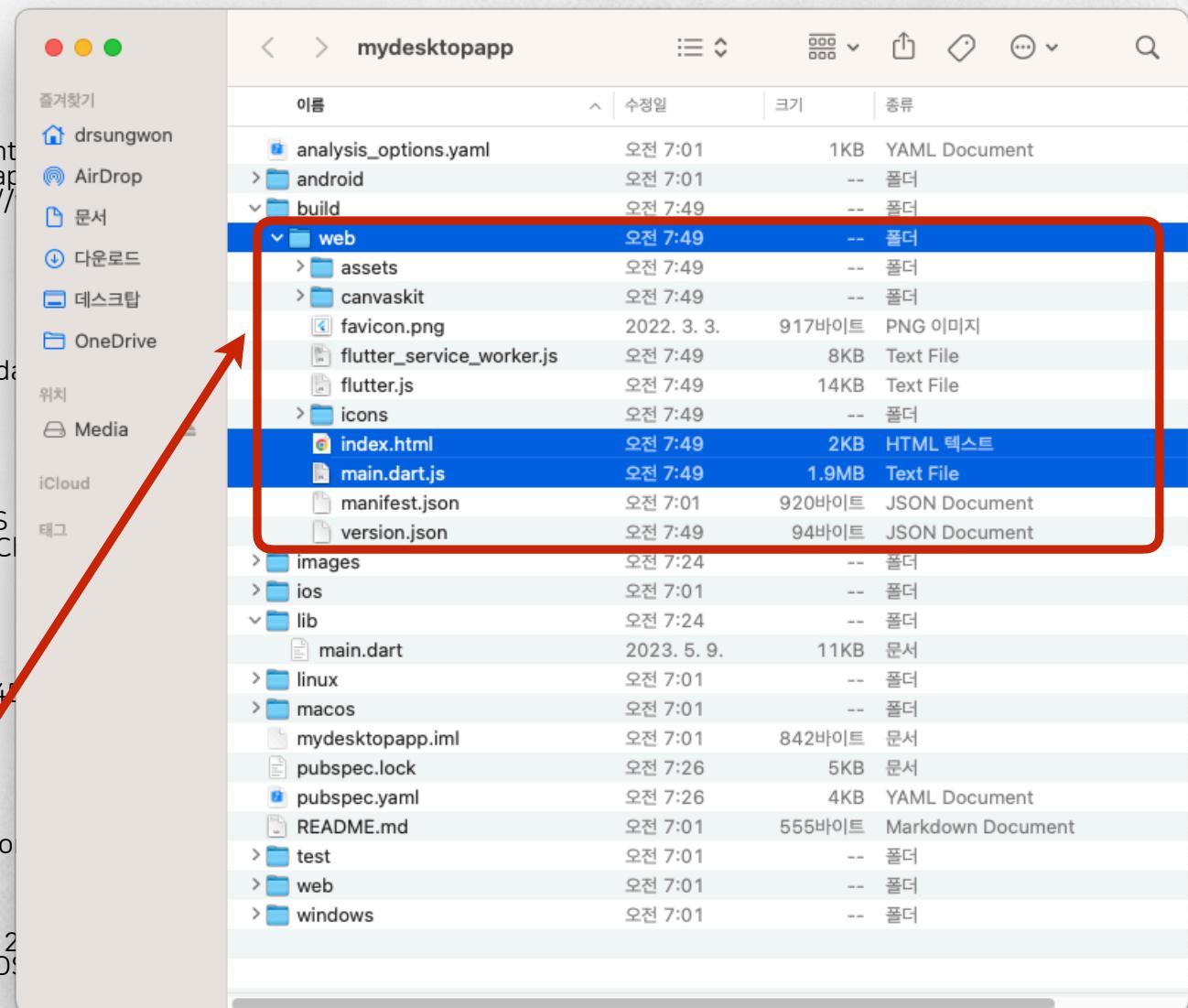
The Flutter DevTools debugger and profiler on macOS

Lost connection to device.

```
drsungwon~$ flutter build web
```

Building macOS application...

```
drsungwon~$
```



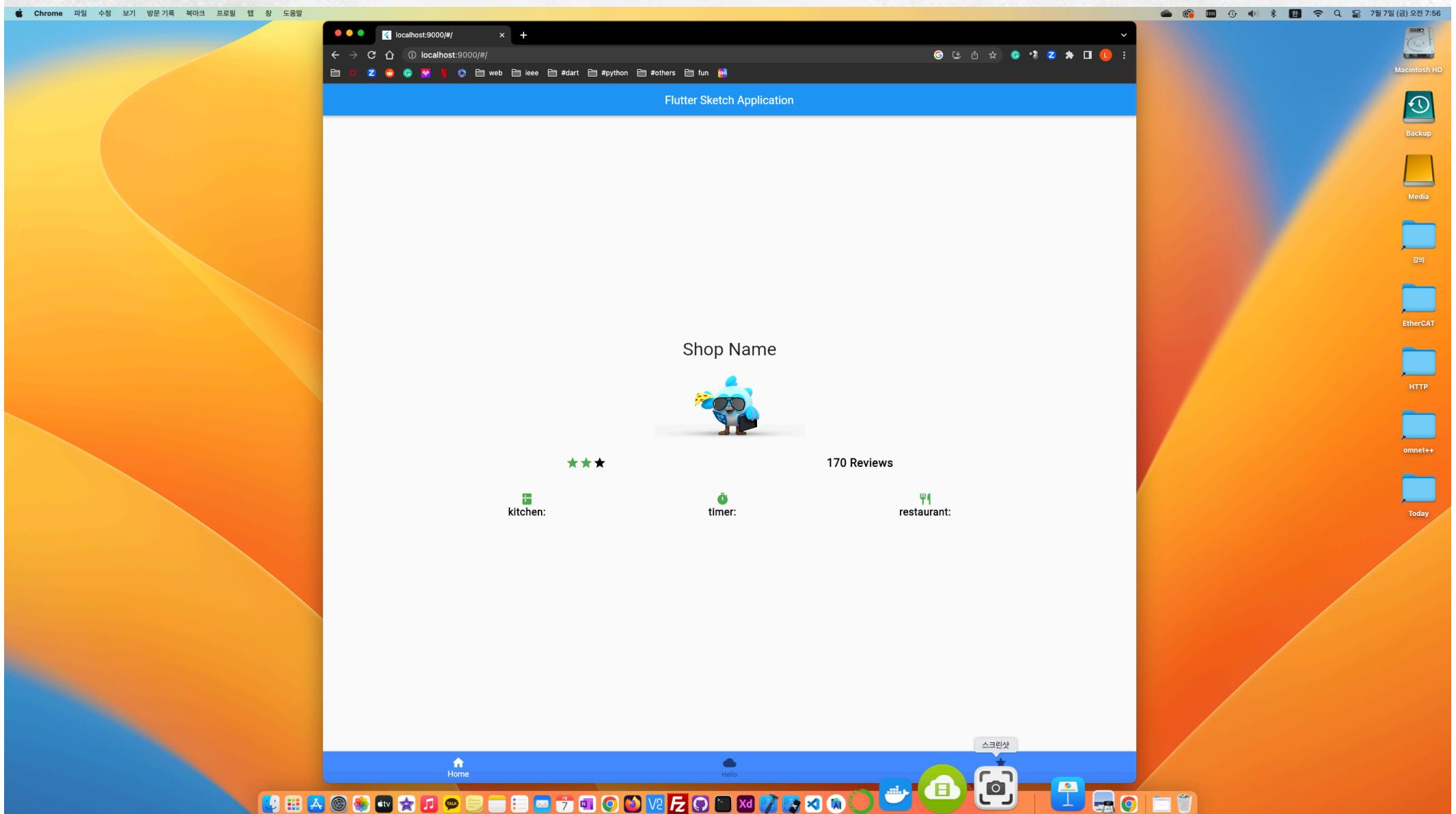
Flutter for Web 이해하기 (리뷰 및 실습)

- 신규 프로젝트 생성
- {프로젝트폴더}/images/flutter_00.png 복사
- {프로젝트폴더}/images/flutter_01.png 복사
- {프로젝트폴더}/lib/main.dart 복사
- **{프로젝트폴더}/pubspec.yaml 수정**
 - 프로젝트에 이미지 등 리소스를 포함하는 방법 예시

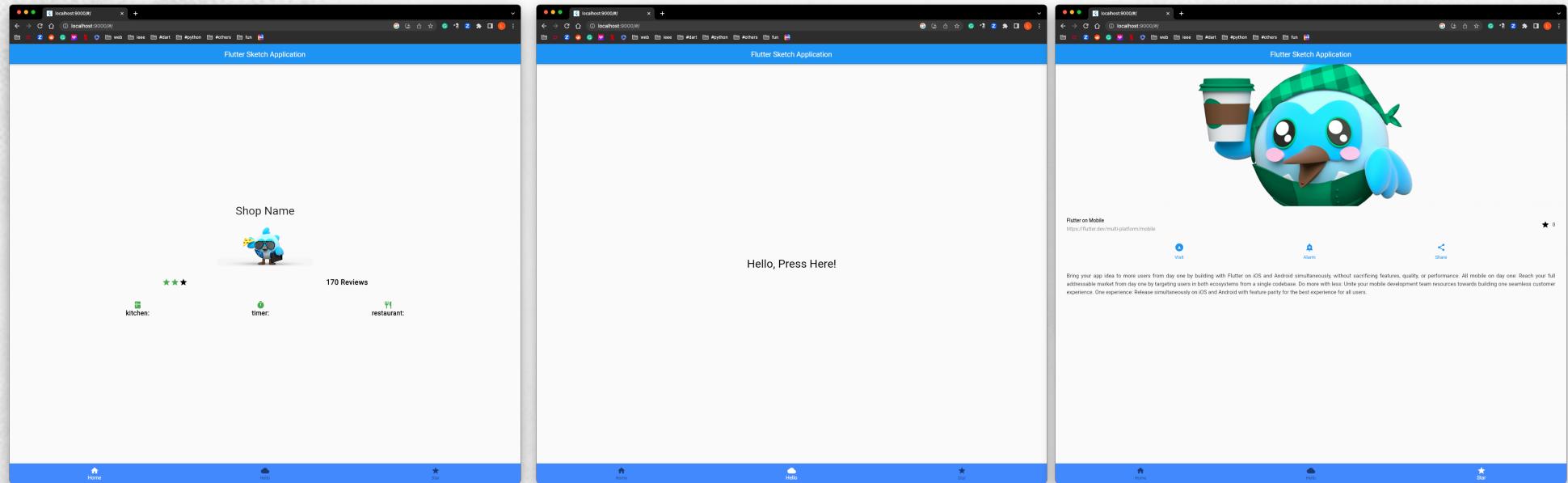
```
flutter:  
  assets:  
    - images/flutter_00.png  
    - images/flutter_01.png
```

- Web 프로젝트 Build (flutter build web)
- cd {프로젝트폴더}/build/web
- python -m http.server 9000
- 웹 브라우저로 http://localhost:9000 접속

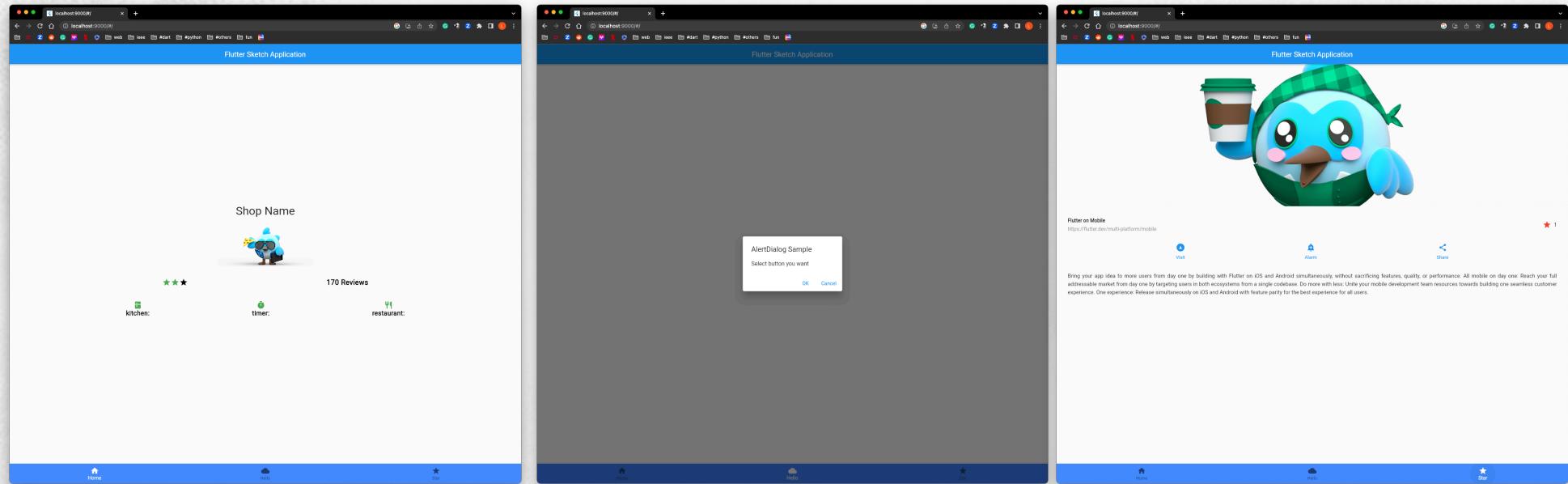
Flutter for Web 이해하기 (리뷰 및 실습)



Flutter for Web 이해하기 (리뷰 및 실습)



Flutter for Web 이해하기 (리뷰 및 실습)



FLUTTER ARCHITECTURE

Framework
(Dart Language)

Material

Cupertino

Widgets

Rendering

Animation

Painting

Gesture

Foundation

Browser
(JavaScript/C++)

HTML/CSS

Canvas

WebGL

WebAssembly

Flutter for Web 참조자료

- Flutter on Web

<https://flutter.dev/multi-platform/web>

- Dart compile (to JavaScript)

<https://dart.dev/tools/dart-compile#js>

Dart compile

- Compile a Dart program to a [target platform](#)

<https://dart.dev/tools/dart-compile>

Subcommands

The following table shows the subcommands of `dart compile`.

Subcommand	Output	More information
<code>exe</code>	Self-contained executable	A standalone, architecture-specific executable file containing the source code compiled to machine code and a small Dart runtime . Learn more .
<code>aot-snapshot</code>	AOT module	An architecture-specific file containing the source code compiled to machine code, but no Dart runtime . Learn more .
<code>jit-snapshot</code>	JIT module	An architecture-specific file with an intermediate representation of all source code, plus an optimized representation of the source code that executed during a training run of the program. JIT-compiled code can have faster peak performance than AOT code if the training data is good. Learn more .
<code>kernel</code>	Kernel module	A portable, intermediate representation of the source code. Learn more .
<code>js</code>	JavaScript	A deployable JavaScript file, compiled from the source code. Learn more .

● Dart & Flutter Support for WebAssembly (Wasm)

<https://docs.flutter.dev/platform-integration/web/wasm>

The screenshot shows the Flutter documentation website. The top navigation bar includes links for Multi-Platform, Development, Ecosystem, Showcase, Docs, a search bar, and social media icons. A blue "Get started" button is on the right. On the left, there's a sidebar with a tree view of documentation categories like Navigation & routing, Data & backend, Accessibility & localization, Platform integration (which is expanded to show Supported platforms, Build desktop apps with Flutter, Write platform-specific code, and lists for Android, iOS, Linux, macOS, Web, and Windows). The main content area has a large title "Support for WebAssembly (Wasm)" and a breadcrumb trail: Platform integration > Web > Wasm. Below the title, it says: "The Flutter and Dart teams are excited to add WebAssembly as a compilation target when building applications for the web." A yellow callout box contains a warning: "⚠ Warning: Development of WebAssembly support for Dart and Flutter remains ongoing. You can preview Wasm and WebAssembly garbage collection (WasmGC) in the master channel. As Flutter provides these features as previews, ongoing development might result in frequent changes. Revisit this page for the latest updates." It also notes the last update date: "Last updated May 23, 2023". To the right of the main content, there's a sidebar with links for Contents, Background, Try it out, Known limitations, and Learn more, along with links to specific Wasm-related topics.

Support for WebAssembly (Wasm)

Platform integration > Web > Wasm

The Flutter and Dart teams are excited to add [WebAssembly](#) as a compilation target when building applications for the web.

⚠ Warning: Development of WebAssembly support for Dart and Flutter remains ongoing. You can preview Wasm and WebAssembly garbage collection (WasmGC) in the [master channel](#). As Flutter provides these features as previews, ongoing development might result in frequent changes. Revisit this page for the latest updates.

Last updated May 23, 2023

Background

To compile Dart and Flutter to Wasm, you need a browser that supports [WasmGC](#). The Wasm standard plans to add WasmGC to help garbage-collecting languages like Dart execute code in an efficient manner.

Both the [Chromium V8](#) and Firefox teams continue to work on WasmGC. To see the current status on the WasmGC and other proposals, check out the [WebAssembly roadmap](#).

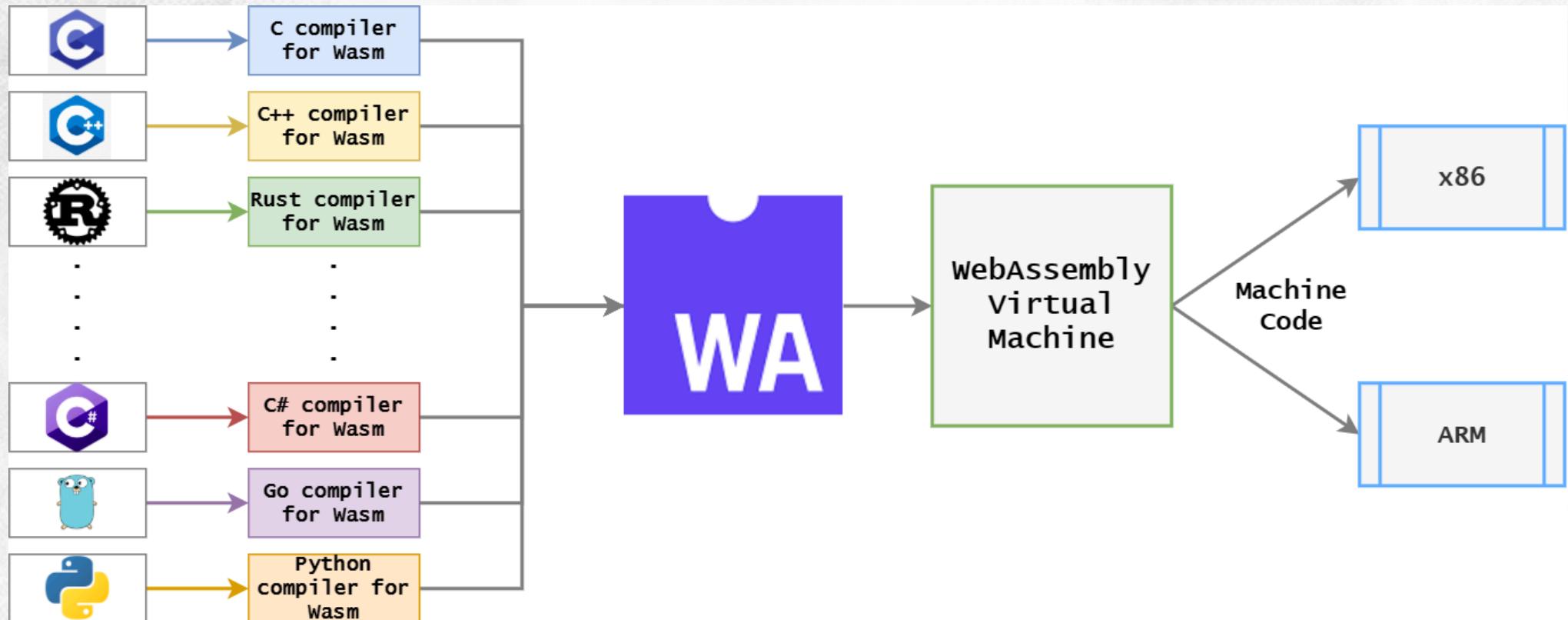
Contents
Background
Try it out
Known limitations
Learn more

Switch to the Flutter master channel.
Pick a (simple) Flutter web application
Run flutter build web --wasm
Serve the output locally with an HTTP server
Load it in a browser

WASM (Web Assembly)

● WASM?

<https://developer.mozilla.org/ko/docs/WebAssembly>





Thank you