Flutter로 만드는 chatGPT 음성번역앱

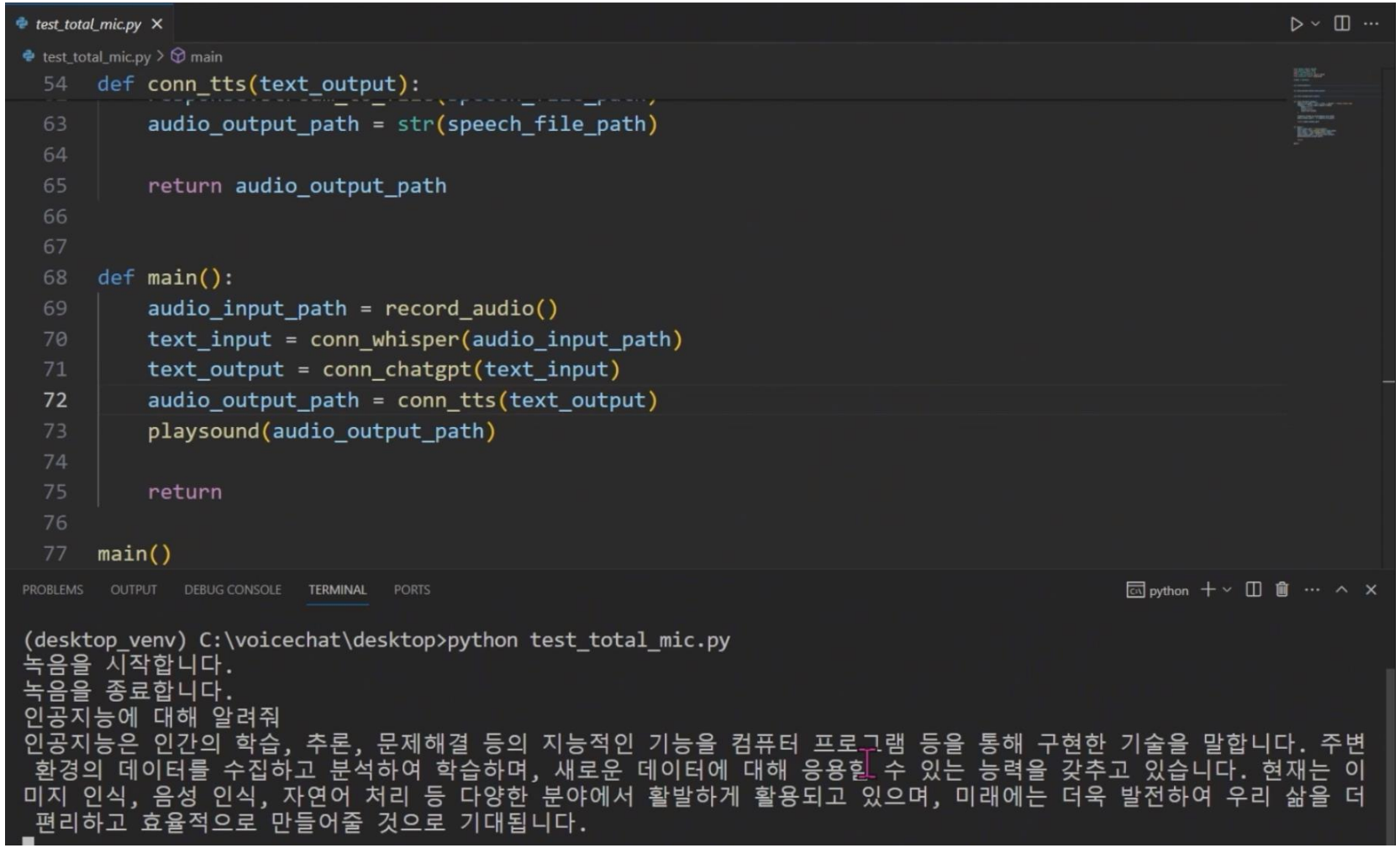
다비

- 강의소개

PC	Mobile
Python v3.9.0	Flutter(Dart) v3.19.x
초급지식 필요	X

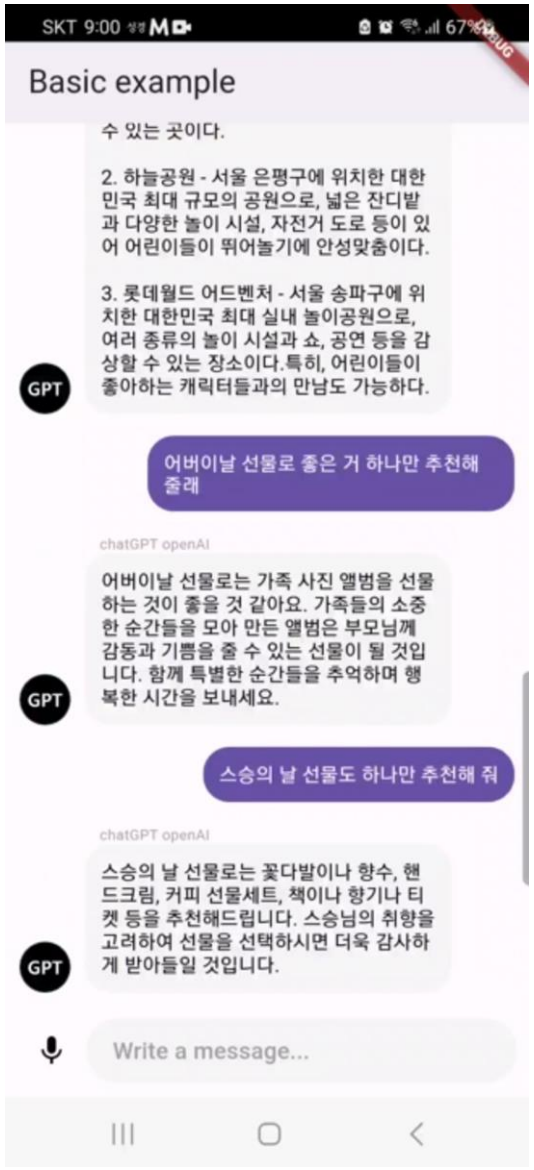
- 강의소개

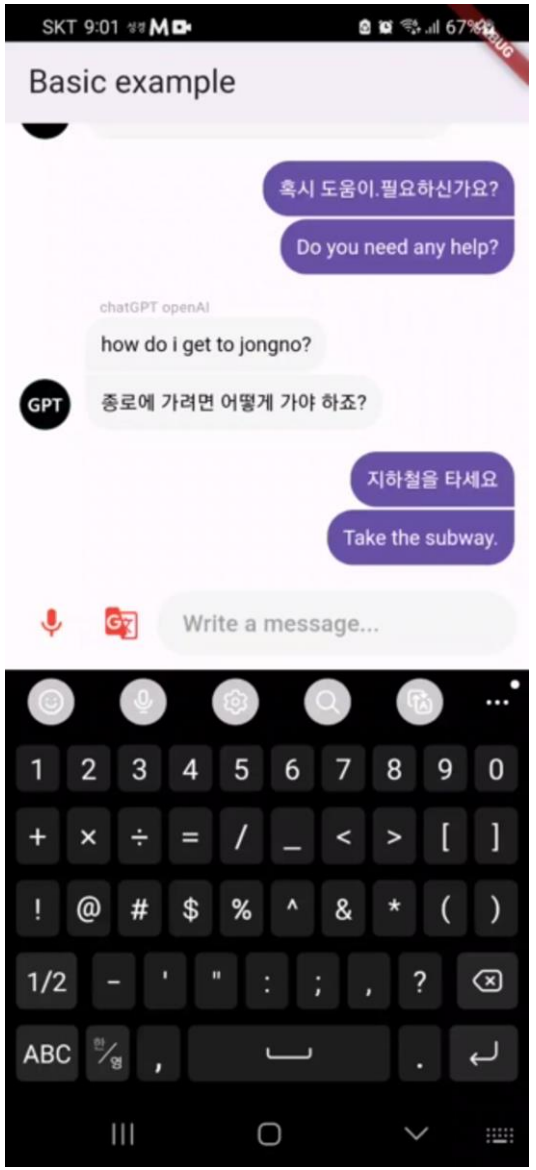
PC


test_total_mic.py
test_total_mic.py > main
54 def conn_tts(text_output):
63 audio_output_path = str(speech_file_path)
64
65 return audio_output_path
66
67
68 def main():
69 audio_input_path = record_audio()
70 text_input = conn_whisper(audio_input_path)
71 text_output = conn_chatgpt(text_input)
72 audio_output_path = conn_tts(text_output)
73 playsound(audio_output_path)
74
75 return
76
77 main()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python
(desktop_venv) C:\voicechat\desktop>python test_total_mic.py
녹음을 시작합니다.
녹음을 종료합니다.
인공지능에 대해 알려줘
인공지능은 인간의 학습, 추론, 문제해결 등의 지능적인 기능을 컴퓨터 프로그램 등을 통해 구현한 기술을 말합니다. 주변 환경의 데이터를 수집하고 분석하여 학습하며, 새로운 데이터에 대해 응용할 수 있는 능력을 갖추고 있습니다. 현재는 이미지 인식, 음성 인식, 자연어 처리 등 다양한 분야에서 활발하게 활용되고 있으며, 미래에는 더욱 발전하여 우리 삶을 더 편리하고 효율적으로 만들어줄 것으로 기대됩니다.

- 강의소개

Mobile


Basic example
수 있는 곳이다.
2. 하늘공원 - 서울 은평구에 위치한 대한민국 최대 규모의 공원으로, 넓은 잔디밭과 다양한 놀이 시설, 자전거 도로 등이 있어 어린이들이 뛰어놀기에 안성맞춤이다.
3. 롯데월드 어드벤처 - 서울 송파구에 위치한 대한민국 최대 실내 놀이공원으로, 여러 종류의 놀이 시설과 쇼, 공연 등을 감상할 수 있는 장소이다.특히, 어린이들이 좋아하는 캐릭터들과의 만남도 가능하다.
GPT
어버이날 선물로 좋은 거 하나만 추천해 줄래
chatGPT openAI
어버이날 선물로는 가족 사진 앨범을 선물하는 것이 좋을 것 같아요. 가족들의 소중한 순간들을 모아 만든 앨범은 부모님께 감동과 기쁨을 줄 수 있는 선물이 될 것입니다. 함께 특별한 순간들을 추억하며 행복한 시간을 보내세요.
스승의 날 선물도 하나만 추천해 줘
chatGPT openAI
스승의 날 선물로는 꽃다발이나 향수, 핸드크림, 커피 선물세트, 책이나 향기나 티켓 등을 추천해드립니다. 스승님의 취향을 고려하여 선물을 선택하시면 더욱 감사하게 받아들일 것입니다.
Write a message...


Basic example
혹시 도움이.필요하신가요?
Do you need any help?
chatGPT openAI
how do i get to jongno?
GPT
종로에 가려면 어떻게 가야 하죠?
지하철을 타세요
Take the subway.
Write a message...


Basic example
GPT
이번에는 귀여운 고양이 한 마리 그려 줄래
chatGPT openAI
Write a message...

- 강의소개

1	환경세팅
01. 강의 소개 02. 파이썬 설치 03. VSCode 설치 04. 가상환경 설치 05. HelloWorld	

2	Desktop API 연결
01. API 이해 02. OpenAI 계정 신청 03. 인증키 등록 04. chatGPT API 연결 05. chatGPT API 응용 06. 추가 API 07. Whisper API 연결 08. TTS API 연결	

3	Desktop 음성대화
01. 함수 구조 만들기 02. 함수 코드 구현 03. 마이크 음성 녹음 04. 음성통신 테스트	

4	Flutter 세팅
01. Flutter 환경세팅 소개 02. Git 설치 03. Android Studio 설치 04. Flutter 기본앱 실행 05. Flutter 기본앱 변경 06. Safearea 적용	

- 강의소개

5	Flutter 음성 대화	6	음성 번역기	7	이미지 생성	8	파인튜닝
01. 음성인식버튼 추가 02. STT 세팅 03. STT 기본코드 04. STT 음성인식 05. STT 환경설정 06. STT 테스트 07. TTS 세팅 08. TTS 구현 09. 최종 테스트		01. 프로젝트 복제 02. 번역기 구조 03. 번역기 기능 04. 번역기 테스트		01. Dall-e 구조 02. Dall-e 기능 03. Dall-e 테스트		9	부록

1. 환경세팅

- 파이썬 설치

Active Python Releases

For more information visit the Python Developer's Guide.

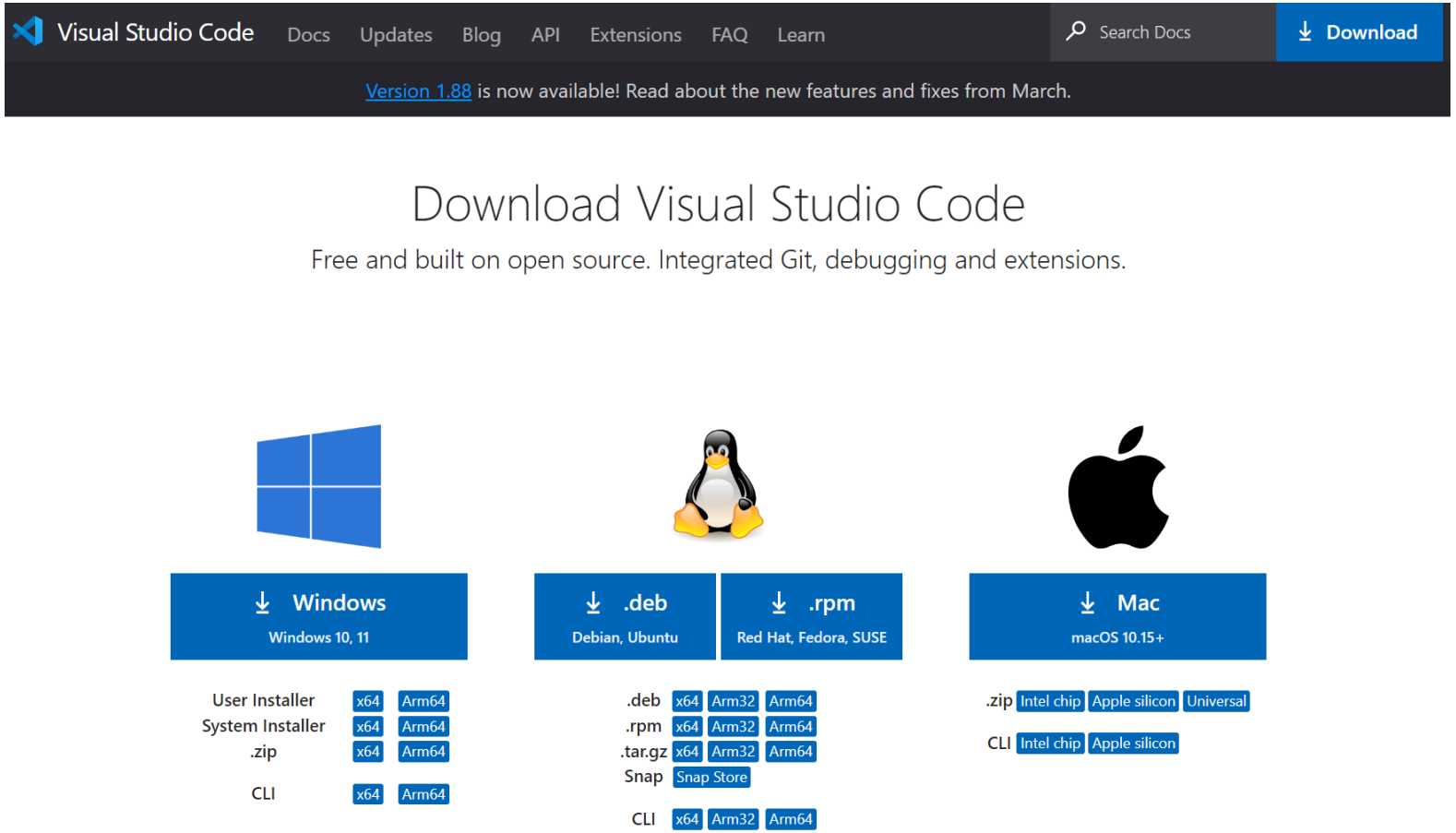
Python version	Maintenance status	First released	End of support	Release schedule
3.13	prerelease	2024-10-01 (planned)	2029-10	PEP 719
3.12	bugfix	2023-10-02	2028-10	PEP 693
3.11	bugfix	2022-10-24	2027-10	PEP 664

3.9.0 버전 권장

https://www.python.org/downloads/

1. 환경세팅

- VSCode 설치



1.87.2 이후 버전 권장

https://code.visualstudio.com/download

- 가상환경 설치

1
```
C:\voicechat\desktop>python -m venv desktop_venv
```

2
```
C:\voicechat\desktop>cd desktop_venv

C:\voicechat\desktop\desktop_venv>cd Scripts

C:\voicechat\desktop\desktop_venv\Scripts>activate
```

3
```
(desktop_venv) C:\voicechat\desktop\desktop_venv\Scripts>
```

4
```
(desktop_venv) C:\voicechat\desktop\desktop_venv\Scripts>cd..

(desktop_venv) C:\voicechat\desktop\desktop_venv>cd..

(desktop_venv) C:\voicechat\desktop>
```

5 해제시 : deactivate

1. 가상환경 생성 :

   python –m venv 가상환경이름
2. 가상환경 활성화 :

   cd 가상환경이름

   cd Scripts

   activate
3. 가상환경 활성화 상태 확인
4. 작업 폴더로 이동

   cd..

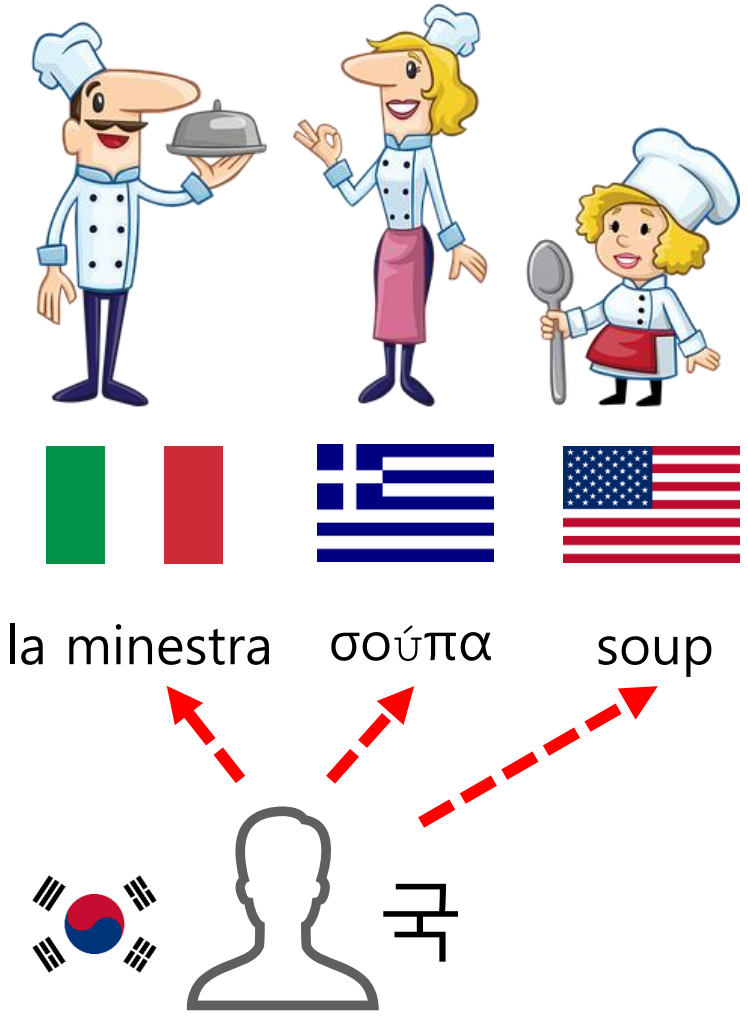
   cd..
5. 가상환경 활성화 해제 :

   deactivate

2. Desktop API 연결

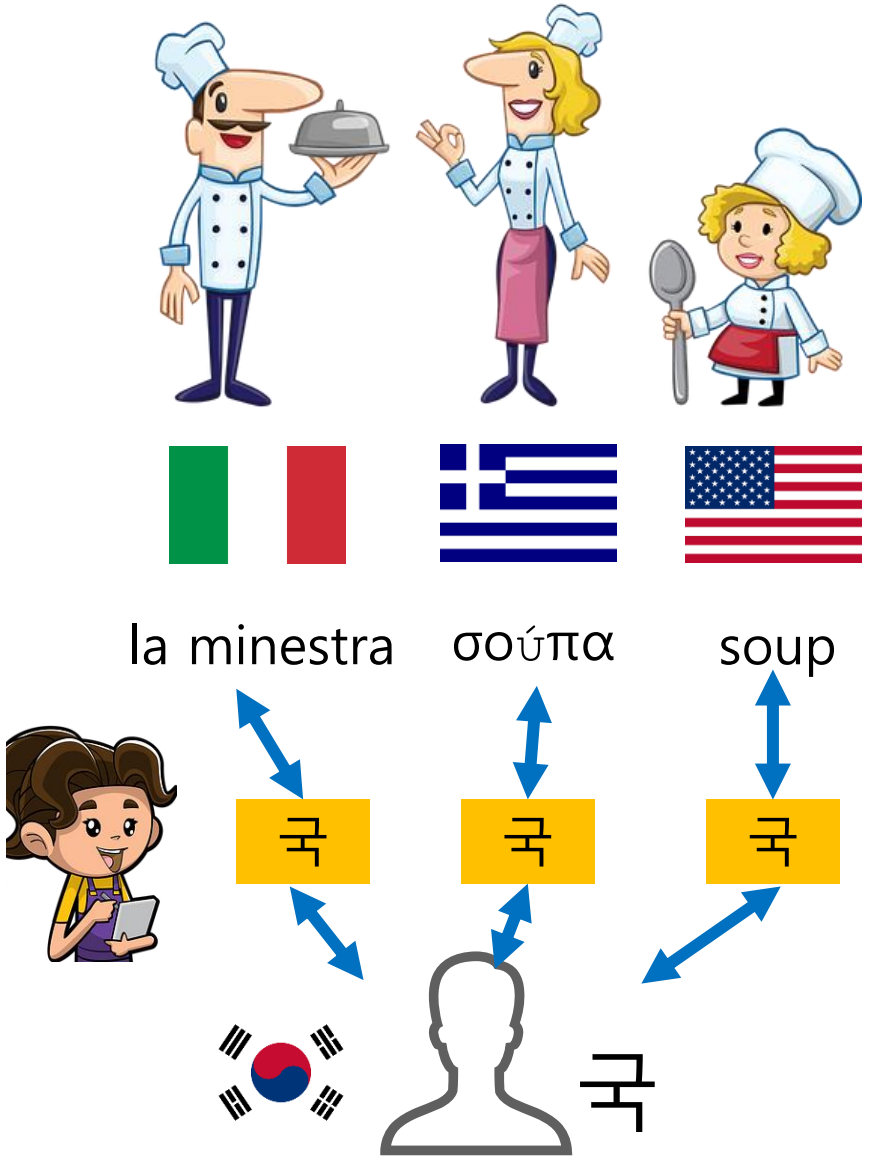
- API 이해

API(Application Programming Interface)



- 서로 다른 언어 체계
- 요리사와 직접 대면
  > 요리사 정보 공개
- 계산서도 각각 다른 방식

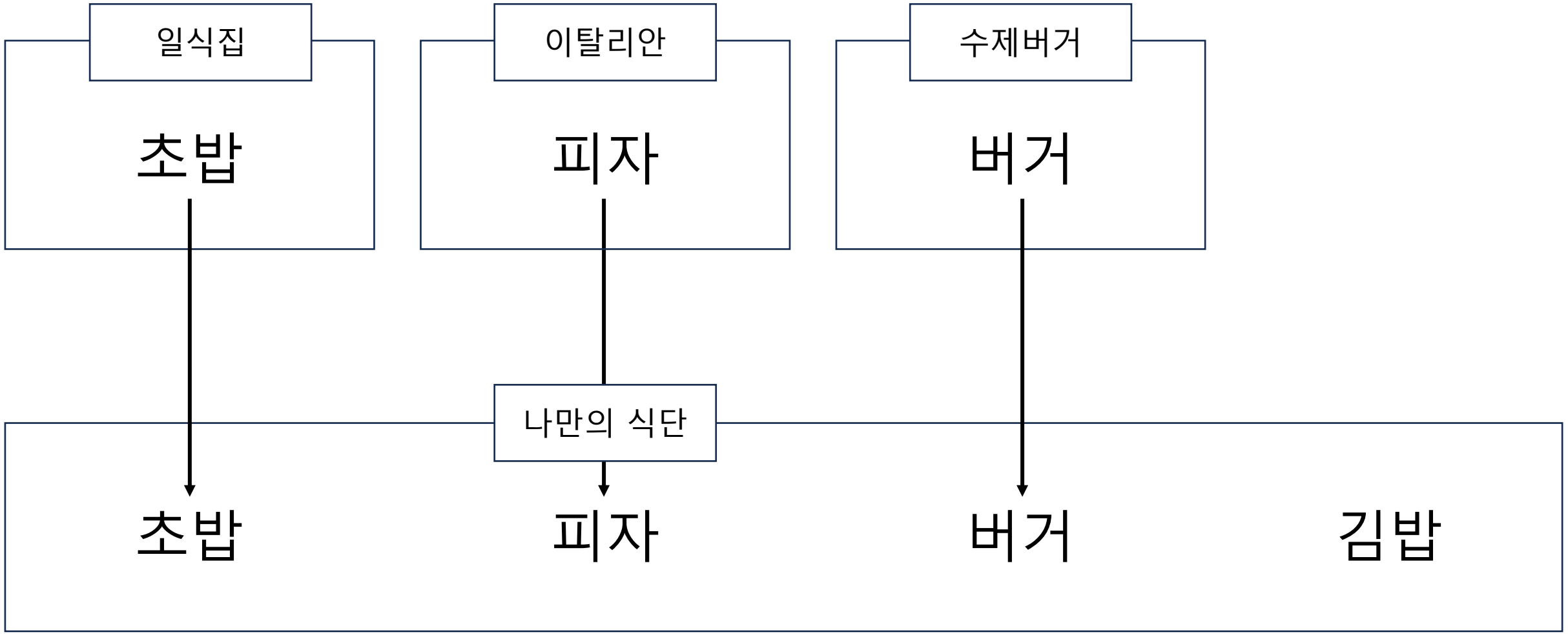
- API 이해



- 하나의 언어 체계
- 요리사와 간접 대면
- 통일된 계산서(결과물)

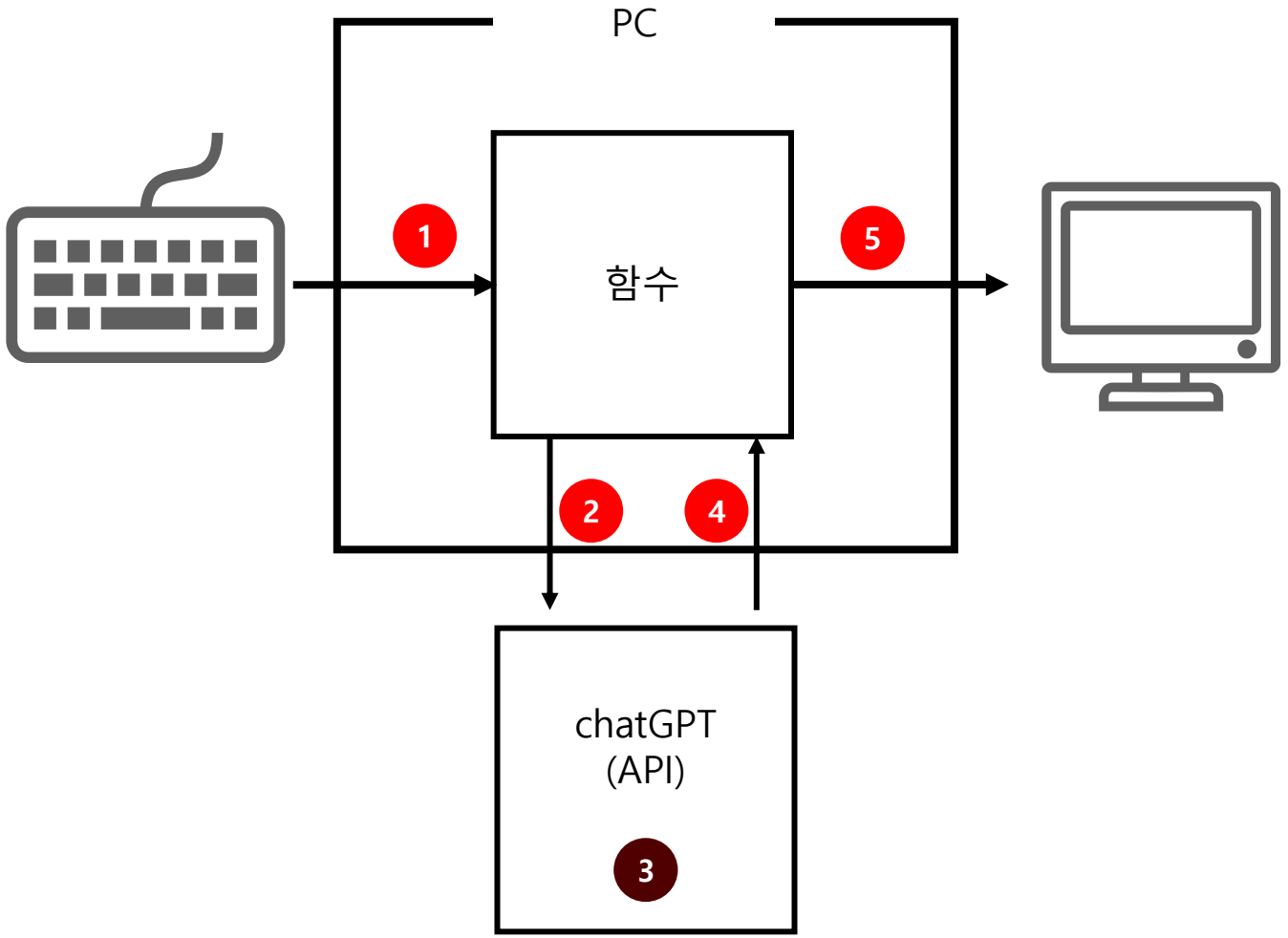
- 활용이 쉬워짐
- 데이터가 다양해짐
- 데이터 통합이 쉬워짐

- API 이해



- API 이해

Text 기반 대화 ChatGPT API 구조



2. Desktop API 연결

- OpenAI 계정 신청

0

Account setup

First, create an OpenAI account or sign in. Next, navigate to the API key page and "Create new secret key", optionally naming the key. Make sure to save this somewhere safe and do not share it with anyone.

1

Step 1: Setting up Python

> Install Python

> Set up a virtual environment (optional)

> Install the OpenAI Python library

Once you have Python 3.7.1 or newer installed and (optionally) set up a virtual environment, the OpenAI Python library can be installed. From the terminal / command line, run:

```
pip install --upgrade openai
```

Once this completes, running `pip list` will show you the Python libraries you have installed in your current environment, which should confirm that the OpenAI Python library was successfully installed.

0. 계정 세팅

0.1. 회원가입

0.2. API Key 생성

1. 파이썬 세팅

~~1.1. 파이썬 설치~~

~~1.2. 가상환경 만들기~~

1.3. 라이브러리 설치

https://platform.openai.com/docs/quickstart?context=python

2. Desktop API 연결

- 인증키 등록

0

Account setup

First, create an OpenAI account or sign in. Next, navigate to the API key page and "Create new secret key", optionally naming the key. Make sure to save this somewhere safe and do not share it with anyone.

1

Step 1: Setting up Python

> Install Python

> Set up a virtual environment (optional)

> Install the OpenAI Python library

Once you have Python 3.7.1 or newer installed and (optionally) set up a virtual environment, the OpenAI Python library can be installed. From the terminal / command line, run:

```
pip install --upgrade openai
```

Once this completes, running `pip list` will show you the Python libraries you have installed in your current environment, which should confirm that the OpenAI Python library was successfully installed.

0. 계정 세팅
    0.1. 회원가입
    0.2. API Key 생성
1. 파이썬 세팅
    ~~1.1. 파이썬 설치~~
    ~~1.2. 가상환경 만들기~~
    1.3. 라이브러리 설치

https://platform.openai.com/docs/quickstart?context=python

2. Desktop API 연결

- 인증키 등록

2

Step 2: Set up your API key

Set up your API key for all projects (recommended)

Windows

3 **Permanent setup**: To make the setup permanent, add the variable through the system properties as follows:

- Right-click on 'This PC' or 'My Computer' and select 'Properties'.
- Click on 'Advanced system settings'.
- Click the 'Environment Variables' button.
- In the 'System variables' section, click 'New...' and enter OPENAI_API_KEY as the variable name and your API key as the variable value.

4 **Verification**: To verify the setup, reopen the command prompt and type the command below. It should display your API key: `echo %OPENAI_API_KEY%`

2. API Key 세팅

2.1. 검색 > 시스템환경변수편집 > > 환경변수 > 시스템변수 > 새로만들기 > 변수이름(OPENAI_API_KEY)와 변수값(API Key값)입력 > 확인

2.2. 검증 : 명령프롬프트(cmd)에서 echo %OPENAI_API_KEY% 실행

- chatGPT API 연결

3

Step 3: Sending your first API request

Making an API request

After you have Python configured and set up an API key, the final step is to send a request to the OpenAI API using the Python library. To do this, create a file named `openai-test.py` using th terminal or an IDE.

Inside the file, copy and paste one of the examples below:

ChatCompletions

```python
from openai import OpenAI
client = OpenAI()

completion = client.chat.completions.create(
  model="gpt-3.5-turbo",
  messages=[
    {"role": "system", "content": "You are a poetic assistant, skilled in explaining
    {"role": "user", "content": "Compose a poem that explains the concept of recursio
  ]
)

print(completion.choices[0].message)
```

3. API 통신 테스트

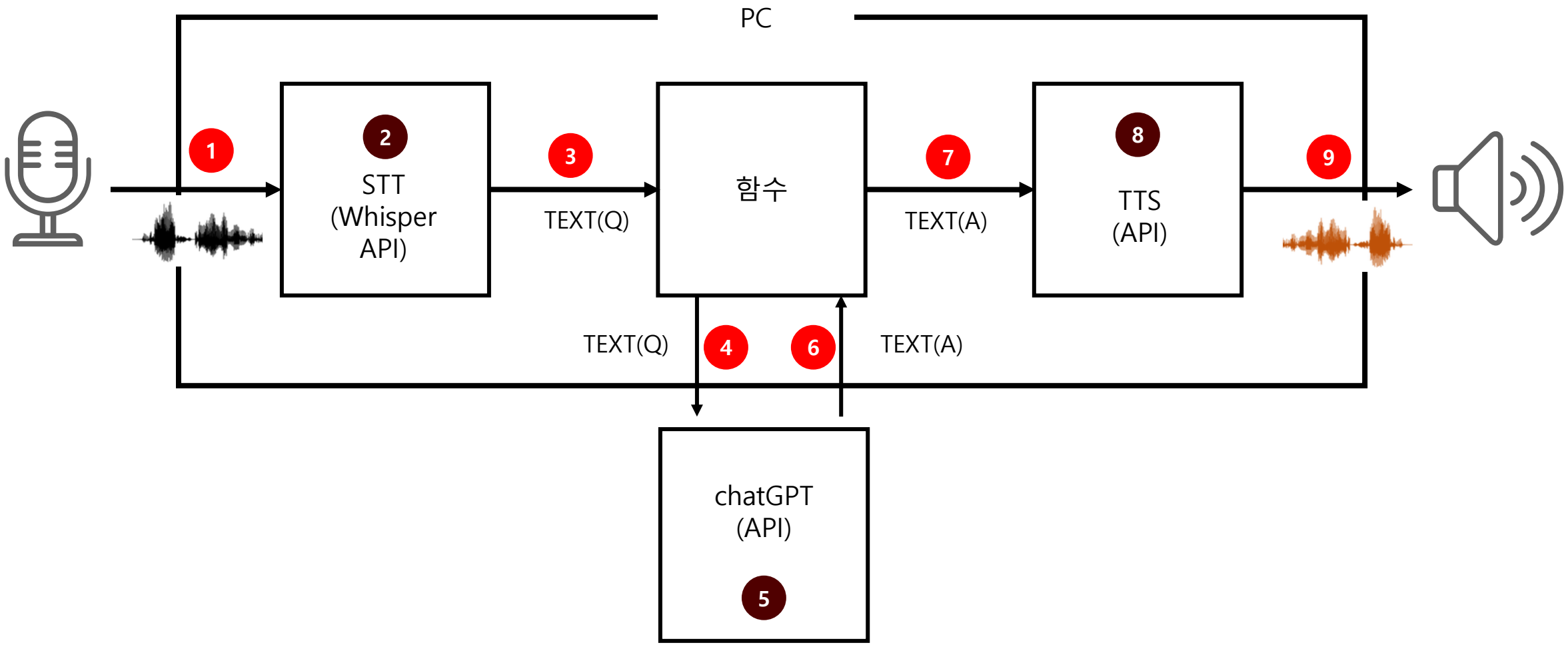
3.1. 예제 코드 복사

3.2. test_chatgpt.py 만들기

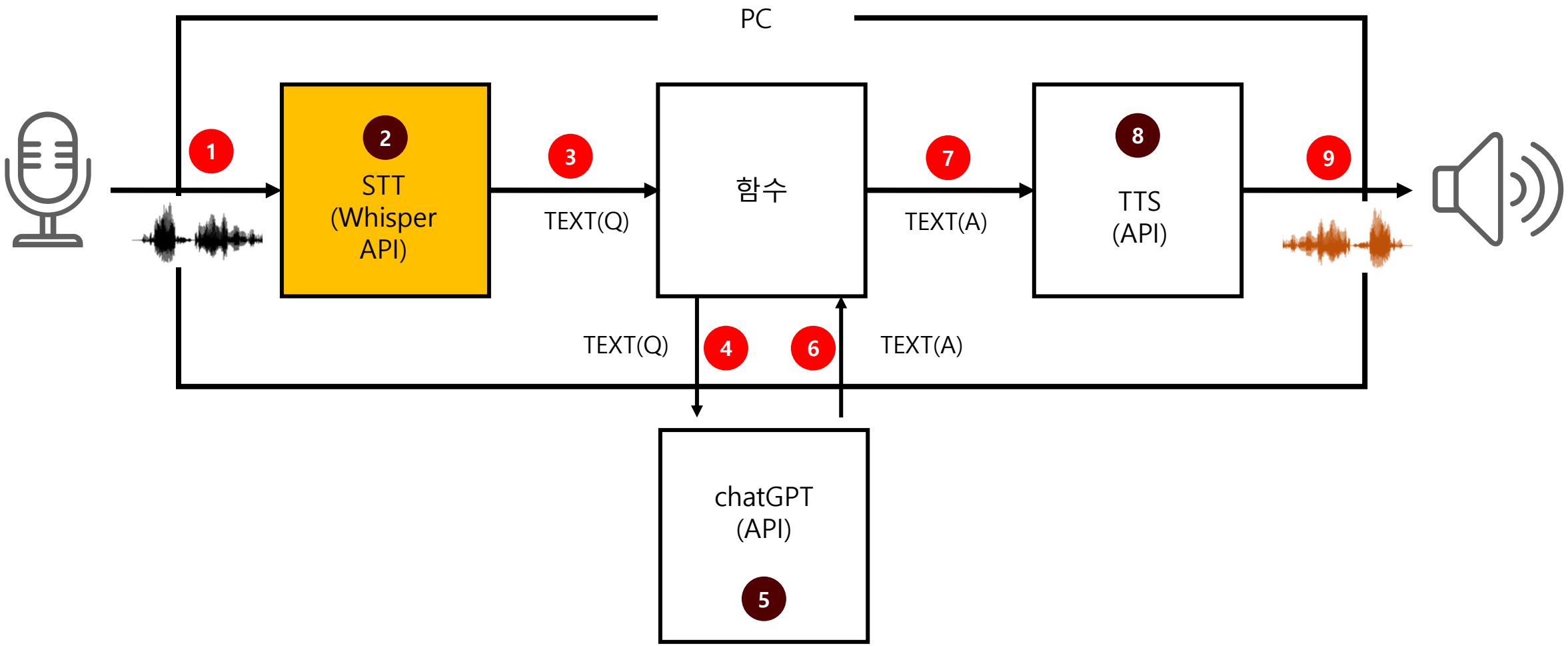
3.3. 붙여넣기

3.4. python test_chatgpt.py실행

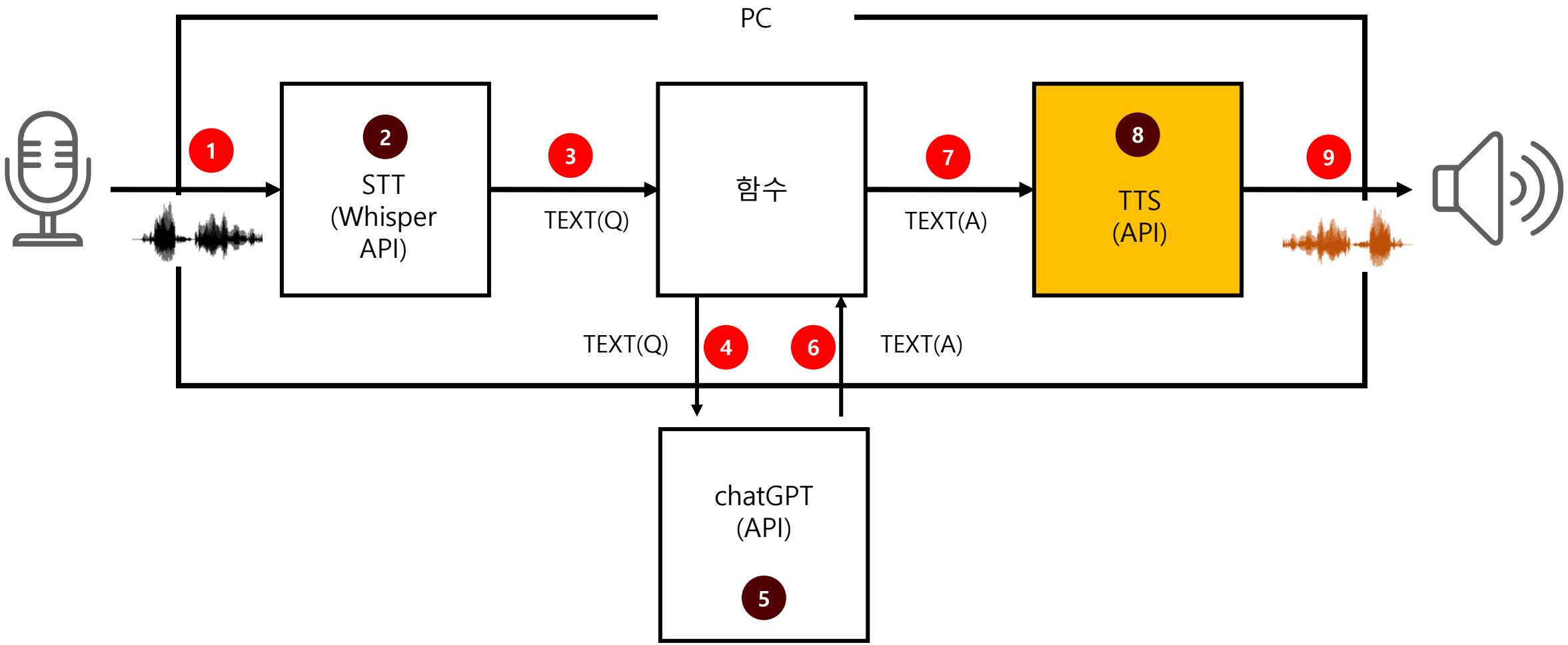
- 추가 API


PC
1
2
STT
(Whisper
API)
3
TEXT(Q)
함수
7
TEXT(A)
8
TTS
(API)
9
TEXT(Q)
4
6
TEXT(A)
chatGPT
(API)
5

- Whisper API

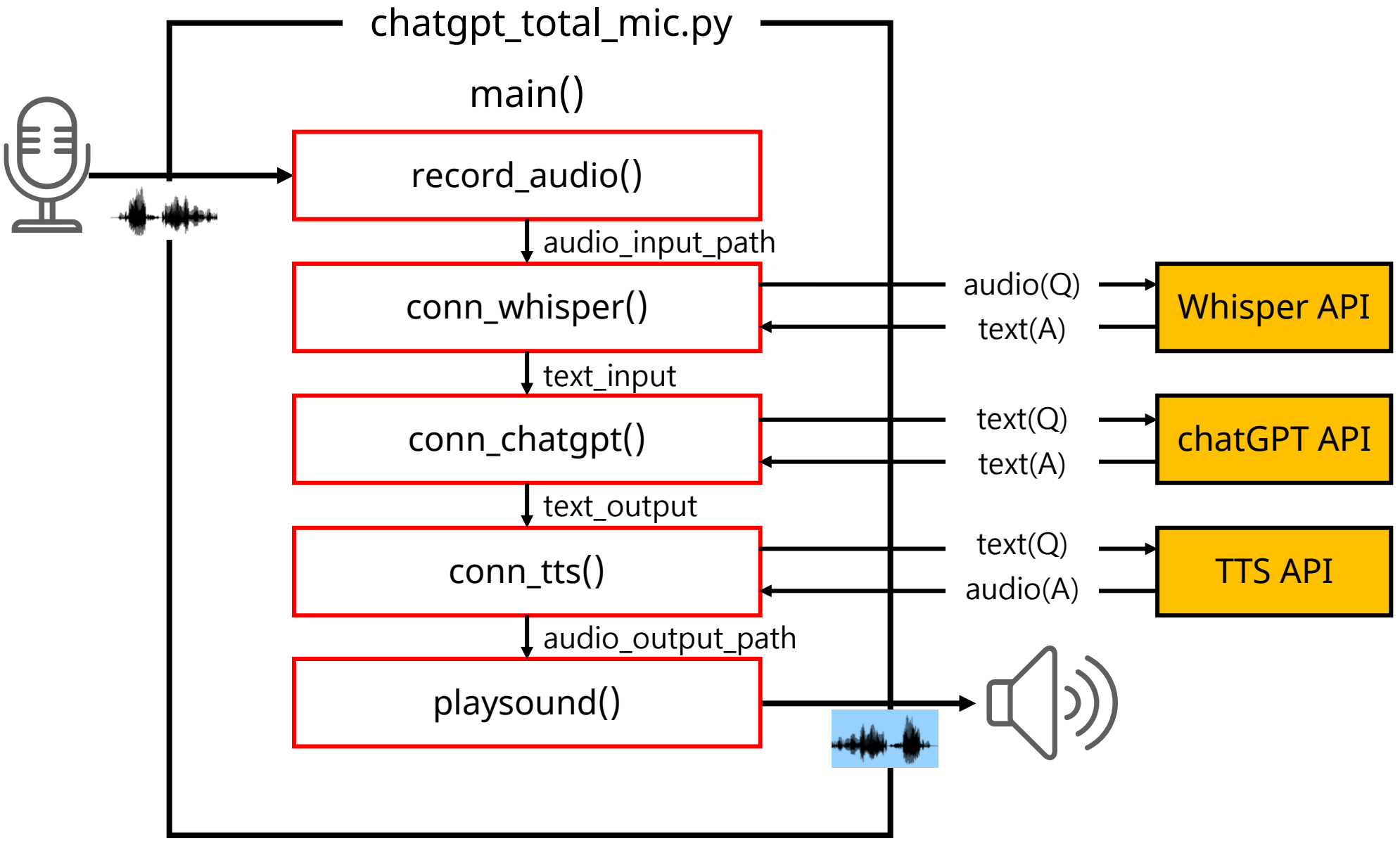

PC
1
2
STT
(Whisper
API)
3
TEXT(Q)
함수
7
TEXT(A)
8
TTS
(API)
9
TEXT(Q)
4
6
TEXT(A)
chatGPT
(API)
5

- TTS API 연결

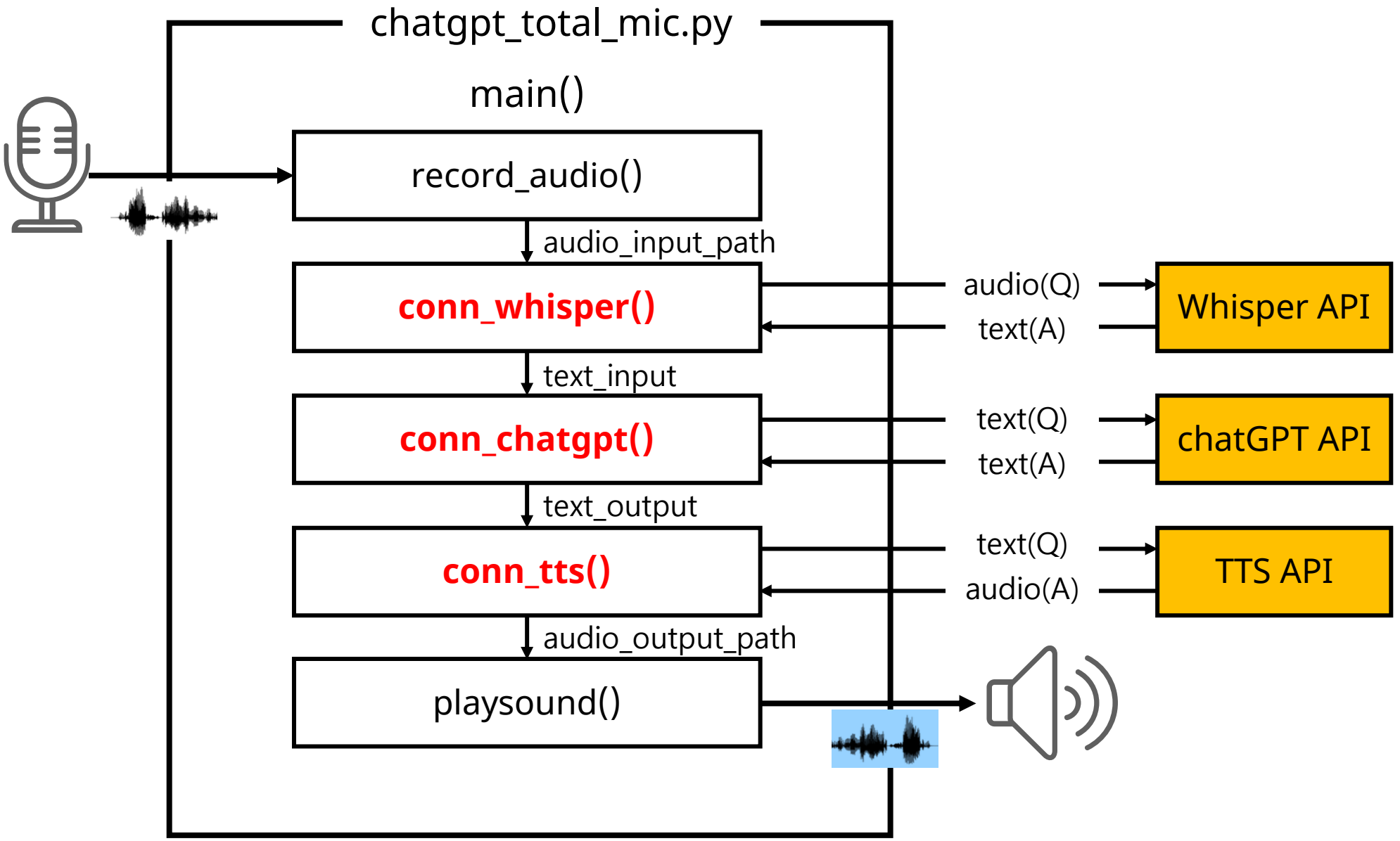

PC
1
2
STT
(Whisper
API)
3
TEXT(Q)
함수
7
TEXT(A)
8
TTS
(API)
9
TEXT(Q)
4
6
TEXT(A)
chatGPT
(API)
5

3. Desktop 음성대화

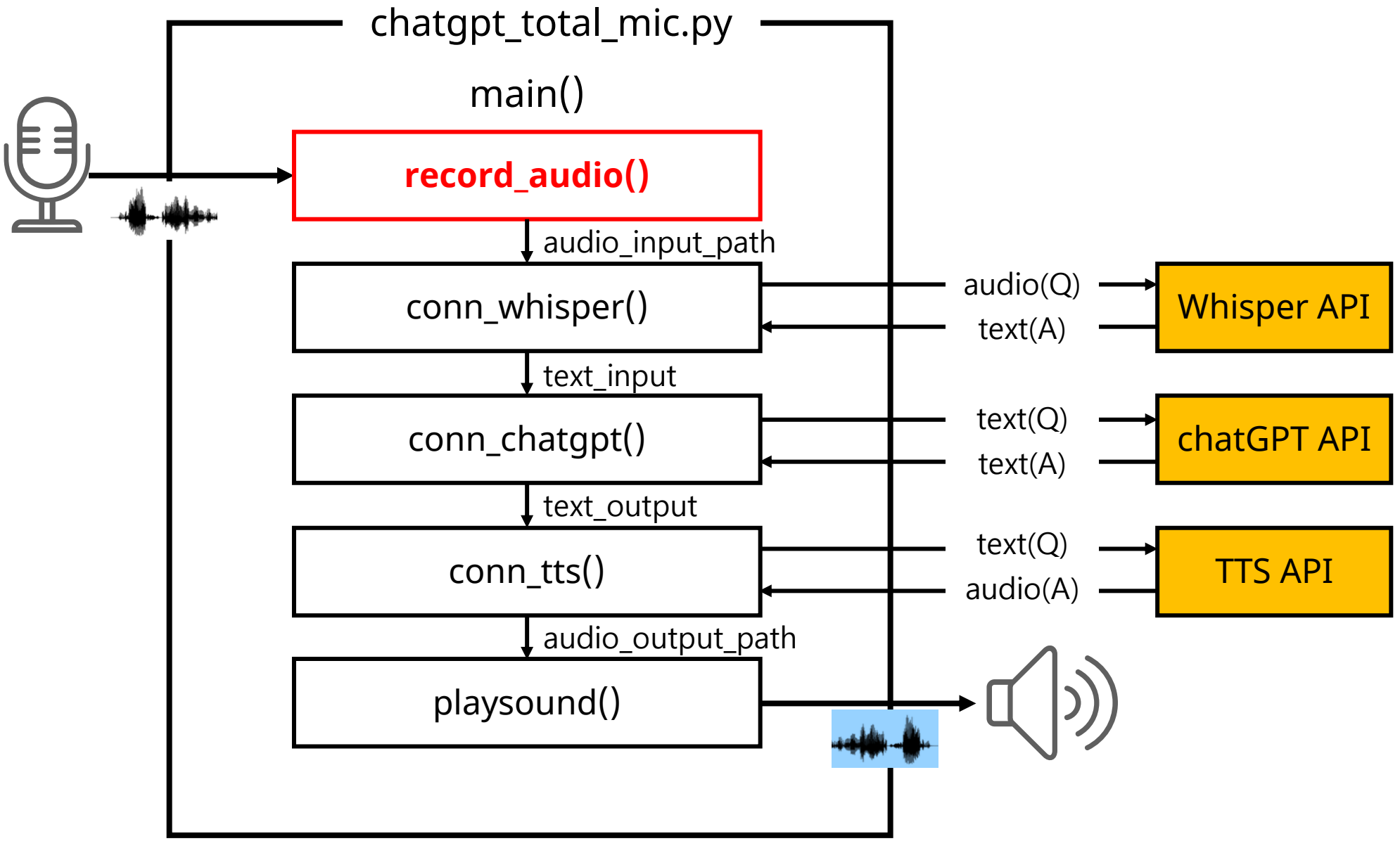
- 함수 구조 만들기


chatgpt_total_mic.py
main()
record_audio()
audio_input_path
conn_whisper()
text_input
conn_chatgpt()
text_output
conn_tts()
audio_output_path
playsound()
audio(Q)
text(A)
Whisper API
text(Q)
text(A)
chatGPT API
text(Q)
audio(A)
TTS API

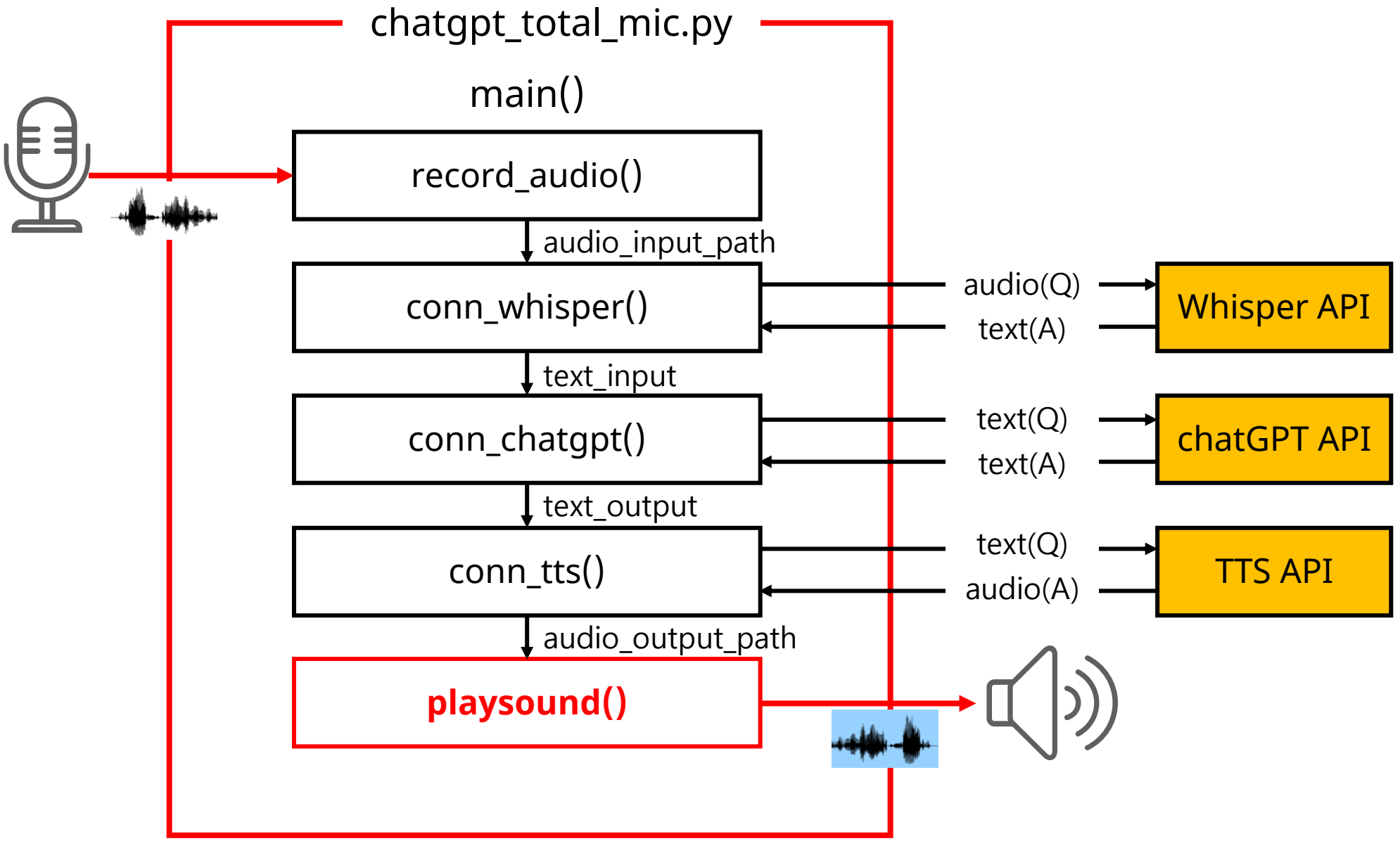
- 함수 코드 구현


chatgpt_total_mic.py
main()
record_audio()
audio_input_path
conn_whisper()
audio(Q)
text(A)
Whisper API
text_input
conn_chatgpt()
text(Q)
text(A)
chatGPT API
text_output
conn_tts()
text(Q)
audio(A)
TTS API
audio_output_path
playsound()

- 마이크 음성 녹음


chatgpt_total_mic.py
main()
record_audio()
audio_input_path
conn_whisper()
audio(Q)
text(A)
Whisper API
text_input
conn_chatgpt()
text(Q)
text(A)
chatGPT API
text_output
conn_tts()
text(Q)
audio(A)
TTS API
audio_output_path
playsound()

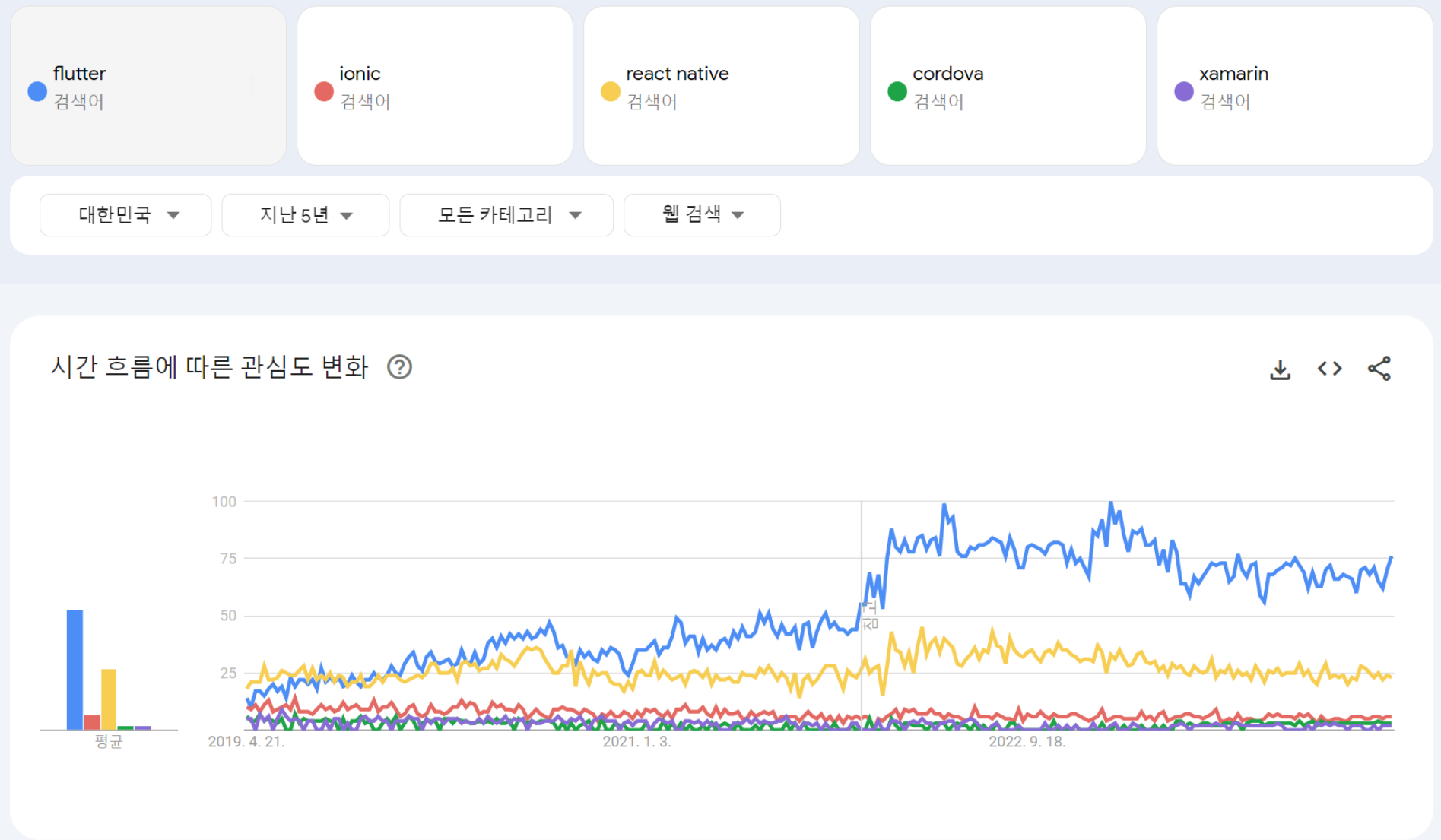
- 음성 통신 테스트


chatgpt_total_mic.py
main()
record_audio()
audio_input_path
conn_whisper()
audio(Q)
text(A)
Whisper API
text_input
conn_chatgpt()
text(Q)
text(A)
chatGPT API
text_output
conn_tts()
text(Q)
audio(A)
TTS API
audio_output_path
playsound()

4. Flutter세팅

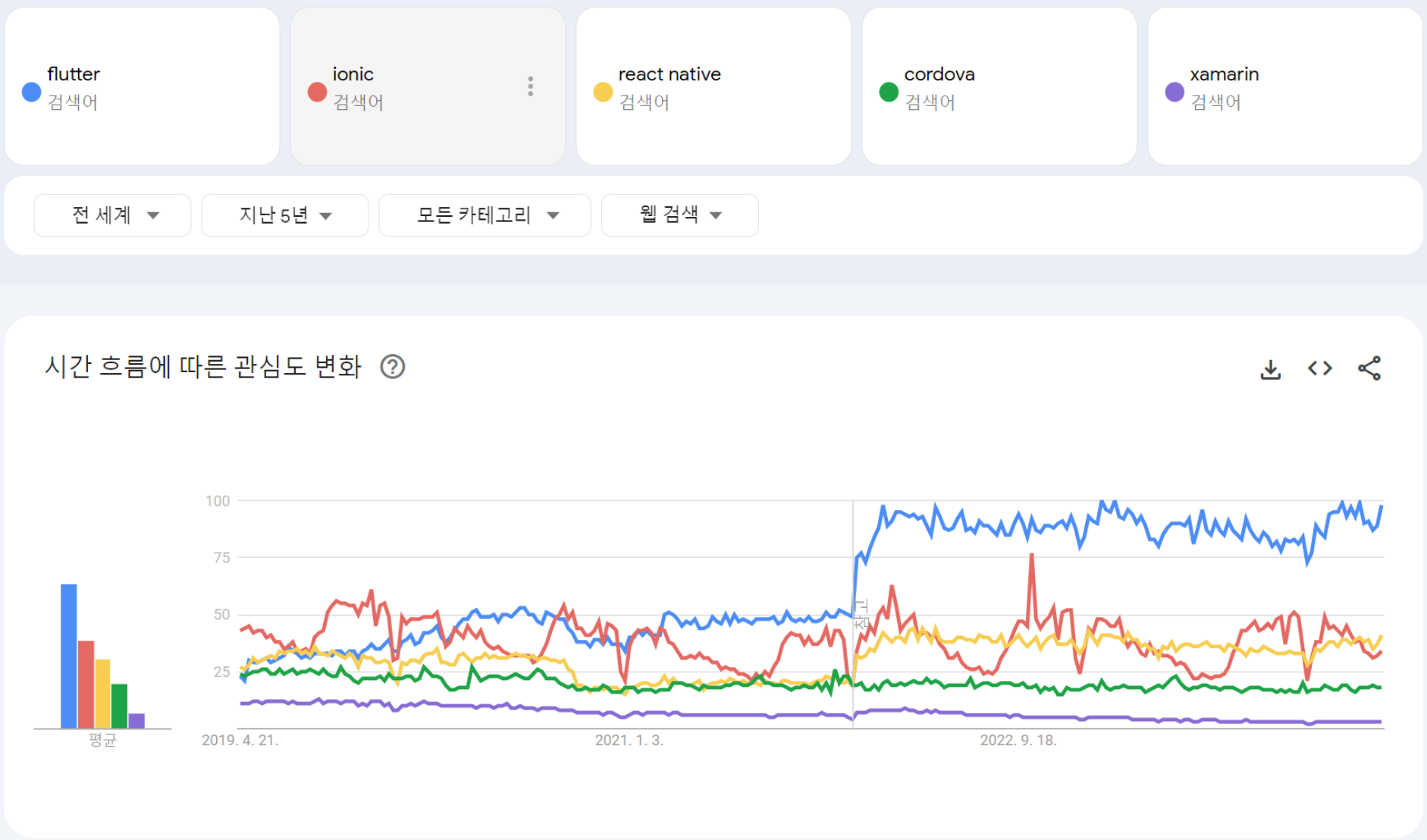
4. Flutter세팅

- Flutter 환경세팅 소개



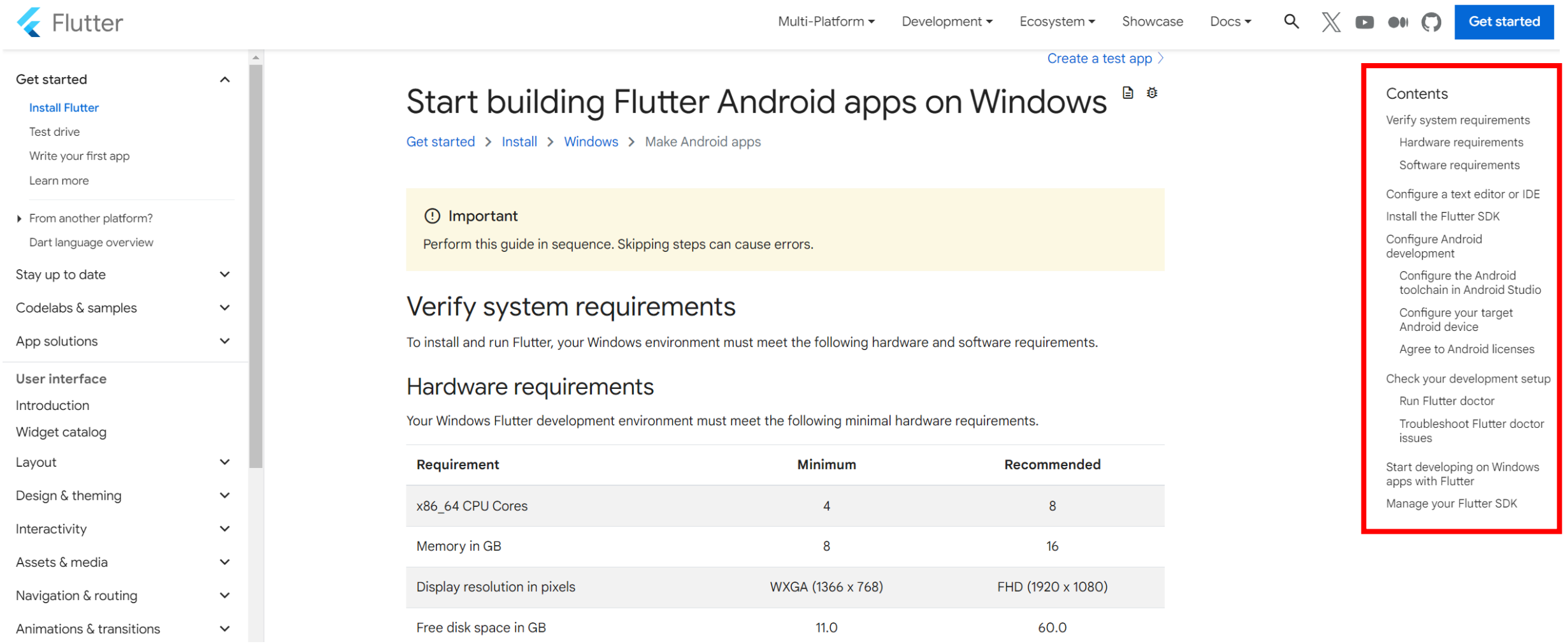
https://trends.google.com/trends/explore?date=today%205-y&q=flutter,ionic,react%20native,cordova,xamarin&hl=ko

- Flutter 환경세팅 소개



https://trends.google.com/trends/explore?date=today%205-y&q=flutter,ionic,react%20native,cordova,xamarin&hl=ko

- Flutter 환경세팅 소개



https://docs.flutter.dev/get-started/install/windows/mobile?tab=first-start

- Git설치

1. System
    1.1. Hardware
    1.2. **Software**
2. Config IDE
3. Flutter SDK
4. Android dev
    4.1. Android toolchain
    4.2. Android device
    4.3. Android licenses
5. Check setup
    5.1. Flutter doctor
    5.2. Troubleshoot

Software requirements

To write and compile Flutter code for Android, you must have the following version of Windows and the listed software packages.

Operating system

Flutter supports 64-bit version of Microsoft Windows 10 or later. These versions of Windows should include the required

1 Windows PowerShell 5 or later.

Development tools

Download and install the Windows version of the following packages:

2 • Git for Windows 2.27 or later to manage source code.

3 Visual Studio 2022 with the the **Desktop development with C++** workload or Build Tools for Visual Studio 2022. This enables building Windows app including all of its default components. **Visual Studio** is an IDE separate from **Visual Studio *Code***.

4 • Android Studio 2023.1 (Hedgehog) to debug and compile Java or Kotlin code for Android. Flutter requires the full version of Android Studio.

5 • The latest version of Google Chrome to debug JavaScript code for web apps.

1. Windows PowerShell : windows10이상이면 이미 설치됨. 필요시 Windows Management Framework 설치 (https://www.microsoft.com/en-us/download/details.aspx?id=54616)
2. **Git : Dart등 설치시 필요할 수 있으므로 설치 권장**
3. Visual Studio 2022 : 윈도우앱을 만들 때 필요. 안드로이드 앱을 만들기 때문에 제외해도 됨
4. Android Studio : 안드로이드 가상 디바이스를 만들 때 필요하므로 설치 권장
5. Chrome : 웹용 앱을 만들 때 필요. 대부분 설치되어 있으므로 별도 설치 필요없음.

- Android Studio 설치

1. System
    1.1. Hardware
    1.2. **Software**
2. Config IDE
3. Flutter SDK
4. Android dev
    4.1. Android toolchain
    4.2. Android device
    4.3. Android licenses
5. Check setup
    5.1. Flutter doctor
    5.2. Troubleshoot

Software requirements

To write and compile Flutter code for Android, you must have the following version of Windows and the listed software packages.

Operating system

Flutter supports 64-bit version of Microsoft Windows 10 or later. These versions of Windows should include the required

1 Windows PowerShell 5 or later.

Development tools

Download and install the Windows version of the following packages:

2 • Git for Windows 2.27 or later to manage source code.

3 Visual Studio 2022 with the the **Desktop development with C++** workload or Build Tools for Visual Studio 2022. This enables building Windows app including all of its default components. **Visual Studio** is an IDE separate from **Visual Studio *Code***.

4 • Android Studio 2023.1 (Hedgehog) to debug and compile Java or Kotlin code for Android. Flutter requires the full version of Android Studio.

5 • The latest version of Google Chrome to debug JavaScript code for web apps.

1. Windows PowerShell : windows10이상이면 이미 설치됨. 필요시 Windows Management Framework 설치 (https://www.microsoft.com/en-us/download/details.aspx?id=54616)
2. Git : Dart등 설치시 필요할 수 있으므로 설치 권장
3. Visual Studio 2022 : 윈도우앱을 만들 때 필요. 안드로이드 앱을 만들기 때문에 제외해도 됨
4. **Android Studio : 안드로이드 가상 디바이스를 만들 때 필요하므로 설치 권장**
5. Chrome : 웹용 앱을 만들 때 필요. 대부분 설치되어 있으므로 별도 설치 필요없음.

- 기타 환경 설정

1. System
    1.1. Hardware
    1.2. Software
2. **Config IDE**
3. Flutter SDK
4. Android dev
    4.1. Android toolchain
    4.2. Android device
    4.3. Android licenses
5. Check setup
    5.1. Flutter doctor
    5.2. Troubleshoot

Configure a text editor or IDE

You can build apps with Flutter using any text editor or integrated development environment (IDE) combined with Flutter's command-line tools.

Using an IDE with a Flutter extension or plugin provides code completion, syntax highlighting, widget editing assists, debugging, and other features.

Popular options include:

- Visual Studio Code 1.77 or later with the Flutter extension for VS Code.
- Android Studio 2023.1 (Hedgehog) or later with the Flutter plugin for IntelliJ.
- IntelliJ IDEA 2023.1 or later with the Flutter plugin for IntelliJ.

Recommended

The Flutter team recommends installing Visual Studio Code 1.77 or later and the Flutter extension for VS Code. This combination simplifies installing the Flutter SDK.

- VSCode > 왼쪽 Tab 중 'Extensions' 선택(Ctrl+Shift+X) > flutter 검색 > install

- 기타 환경 설정

1. System
    1.1. Hardware
    1.2. Software
2. Config IDE
3. **Flutter SDK**
4. Android dev
    4.1. Android toolchain
    4.2. Android device
    4.3. Android licenses
5. Check setup
    5.1. Flutter doctor
    5.2. Troubleshoot

Install the Flutter SDK

To install the Flutter SDK, you can use the VS Code Flutter extension or download and install the Flutter bundle yourself.

Use VS Code to install | Download and install

Use VS Code to install Flutter

To install Flutter using these instructions, verify that you have installed Visual Studio Code 1.77 or later and the Flutter extension for VS Code.

Prompt VS Code to install Flutter

1. Launch VS Code.
2. To open the **Command Palette**, press `Control` + `Shift` + `P`.
3. In the **Command Palette**, type `flutter`.
4. Select **Flutter: New Project**.
5. VS Code prompts you to locate the Flutter SDK on your computer.
    1. If you have the Flutter SDK installed, click **Locate SDK**.
    2. If you do not have the Flutter SDK installed, click **Download SDK**.

       This option sends you the Flutter install page if you have not installed Git for Windows as directed in the development tools prerequisites.
6. When prompted **Which Flutter template?**, ignore it. Press `Esc`. You can create a test project after checking your development setup.

- VSCode > Ctrl+Shift+P > flutter 검색 > Flutter: New Project 선택
  (처음 설치) Download SDK 선택 > 폴더 지정
  (이미 설치) Locate SDK 선택 > 폴더 지정

- 기타 환경 설정

1. System
    1.1. Hardware
    1.2. Software
2. Config IDE
3. Flutter SDK
4. Android dev
    4.1. **Android toolchain**
    4.2. Android device
    4.3. Android licenses
5. Check setup
    5.1. Flutter doctor
    5.2. Troubleshoot

Configure Android development

Configure the Android toolchain in Android Studio

To create Android apps with Flutter, verify that the following Android components have been installed.

- **Android SDK Platform, API 34.0.0**
- **Android SDK Command-line Tools**
- **Android SDK Build-Tools**
- **Android SDK Platform-Tools**
- **Android Emulator**

- (처음 설치) 설치마법사 > 컴포넌트 선택
  (이미 설치) 오선 상단메뉴 > Tools > SDK Manager > SDK Platform 탭에서 API Level 34 체크
  SDK Tools 탭 > Android SDK Build-Tools, Android SDK Command-line Tools, Android Emulator, Android SDK Platform-Tools 체크 > OK

- 기타 환경 설정

1. System
    1.1. Hardware
    1.2. Software
2. Config IDE
3. Flutter SDK
4. Android dev
    4.1. Android toolchain
    4.2. **Android device**
    4.3. Android licenses
5. Check setup
    5.1. Flutter doctor
    5.2. Troubleshoot

Configure your target Android device

Virtual Device | Physical Device

Set up the Android emulator

Help

To configure your Flutter app to run in an Android emulator, follow these steps to create and select an emulator.

1. Enable VM acceleration on your development computer.
2. Start **Android Studio**.
3. Go to the **Settings** dialog to view the **SDK Manager**.
    1. If you have a project open, go to **Tools** > **Device Manager**.
    2. If the **Welcome to Android Studio** dialog displays, click the **More Options** icon that follows the **Open** button and click **Device Manager** from the dropdown menu.
4. Click **Virtual**.
5. Click **Create Device**.

   The **Virtual Device Configuration** dialog displays.
6. Select either **Phone** or **Tablet** under **Category**.
7. Select a device definition. You can browse or search for the device.
8. Click **Next**.
9. Click **x86 Images**.
10. Click one system image for the Android version you want to emulate.
    1. If the desired image has a **Download** icon to the right of the **Release Name**, click it.

       The **SDK Quickfix Installation** dialog displays with a completion meter.
    2. When the download completes, click **Finish**.
11. Click **Next**.

    The **Virtual Device Configuration** displays its **Verify Configuration** step.
12. To rename the Android Virtual Device (AVD), change the value in the **AVD Name** box.
13. Click **Show Advanced Settings** and scroll to **Emulated Performance**.
14. From the **Graphics** dropdown menu, select **Hardware - GLES 2.0**.

    This enables hardware acceleration and improves rendering performance.
15. Verify your AVD configuration. If it is correct, click **Finish**.

    To learn more about AVDs, check out Managing AVDs.
16. In the **Device Manager** dialog, click the **Run** icon to the right of your desired AVD. The emulator starts up and displays the default canvas for your selected Android OS version and device.

- 기타 환경 설정

1. System
    1.1. Hardware
    1.2. Software
2. Config IDE
3. Flutter SDK
4. Android dev
    4.1. Android toolchain
    4.2. **Android device**
    4.3. Android licenses
5. Check setup
    5.1. Flutter doctor
    5.2. Troubleshoot

[실제 스마트폰 연결]

1. USB 연결
2. 스마트폰 > 설정 > 휴대전화정보(디바이스정보) > 소프트웨어정보
3. 빌드번호 부분을 여러 번 터치
4. 비밀번호 입력
5. 개발자모드 활성화됨 : 설정 제일 아래 부분에 새로 생김
6. 개발자모드 중간에 있는 'USB 디버깅'이 On되어야 함

- 기타 환경 설정

1. System
    1.1. Hardware
    1.2. Software
2. Config IDE
3. Flutter SDK
4. Android dev
    4.1. Android toolchain
    4.2. Android device
    4.3. **Android licenses**
5. Check setup
    5.1. Flutter doctor
    5.2. Troubleshoot

Agree to Android licenses

Help

Before you can use Flutter and after you install all prerequisites, agree to the licenses of the Android SDK platform.

1. Open an elevated console window.
2. Run the following command to enable signing licenses.

```
C:> flutter doctor --android-licenses
```

If you accepted the Android Studio licenses at another time, this command returns:

```
[======================================] 100% Computing updates...
All SDK package licenses accepted.
```

You can skip the next step.

3. Before agreeing to the terms of each license, read each with care.

- flutter doctor –android-licenses 실행 > y 누르고 엔터

- 기타 환경 설정

1. System
    1.1. Hardware
    1.2. Software
2. Config IDE
3. Flutter SDK
4. Android dev
    4.1. Android toolchain
    4.2. Android device
    4.3. Android licenses
5. Check setup
    5.1. **Flutter doctor**
    5.2. Troubleshoot

Check your development setup

Help

Run Flutter doctor

The `flutter doctor` command validates that all components of a complete Flutter development environment for Windows.

1. Open PowerShell.
2. To verify your installation of all the components, run the following command.

```
C:> flutter doctor
```

As you chose to develop for , you do not need *all* components. If you followed this guide, the result of your command should resemble:

```
Running flutter doctor...
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.19.3, on Microsoft Windows 11 [Version 10.0.22621.3155], locale en)
[✓] Windows version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
[!] Chrome - develop for the web
[!] Visual Studio - develop Windows apps
[✓] Android Studio (version 2023.1 (Hedgehog) or later)
[✓] VS Code (version 1.86)
[✓] Connected device (1 available)
[✓] Network resources
```

- flutter doctor 엔터
- 웹용 어플 개발을 할 때 필요한 Chrome과 PC용 어플 개발을 할 때 필요한 Visual Studio는 설치되지 않아도 수업 진행 가능

- 기타 환경 설정

1. System
    1.1. Hardware
    1.2. Software
2. Config IDE
3. Flutter SDK
4. Android dev
    4.1. Android toolchain
    4.2. Android device
    4.3. Android licenses
5. Check setup
    5.1. Flutter doctor
    5.2. **Troubleshoot**

Troubleshoot Flutter doctor issues

When the `flutter doctor` command returns an error, it could be for Flutter, VS Code, , the connected device, or network resources.

If the `flutter doctor` command returns an error for any of these components, run it again with the verbose flag.

```
C:> flutter doctor -v
```

Check the output for other software you might need to install or further tasks to perform.

If you change the configuration of your Flutter SDK or its related components, run `flutter doctor` *again* to verify the installation.

- flutter doctor 실행 결과에서 다른 오류가 발생하면,
- flutter doctor –v를 실행해서 세부 오류 사항을 확인한 후 오류 부분을 해결. Flutter 환경세팅 부분을 천천히 다시 따라해보세요.
- Flutter를 삭제하거나 업그레이드할 경우에는 아래 페이지 부분을 참고하세요.

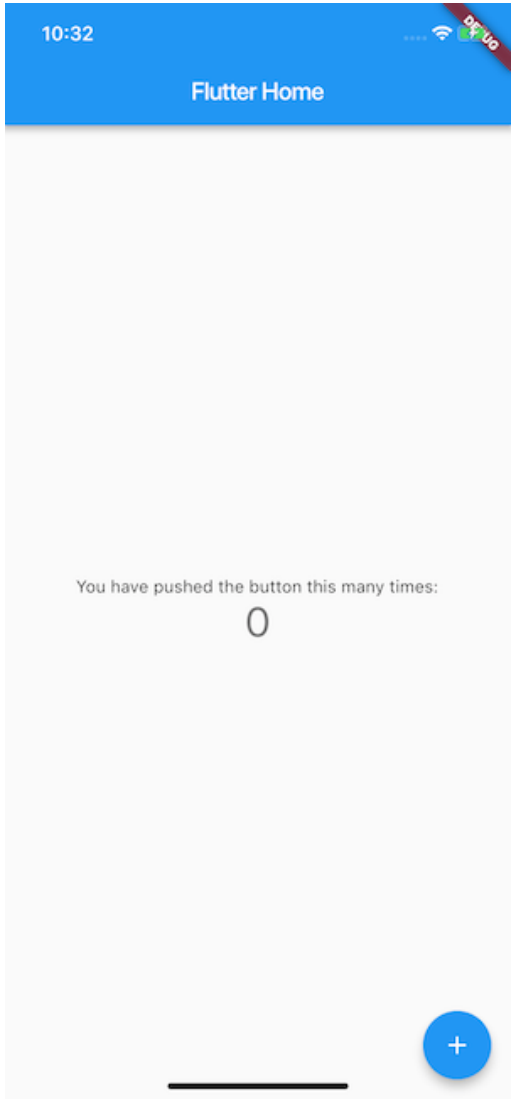
Manage your Flutter SDK

To learn more about managing your Flutter SDK install, consult the following resources.

- Upgrade Flutter
- Uninstall Flutter

4. Flutter세팅

- Flutter 기본앱 실행



main.dart

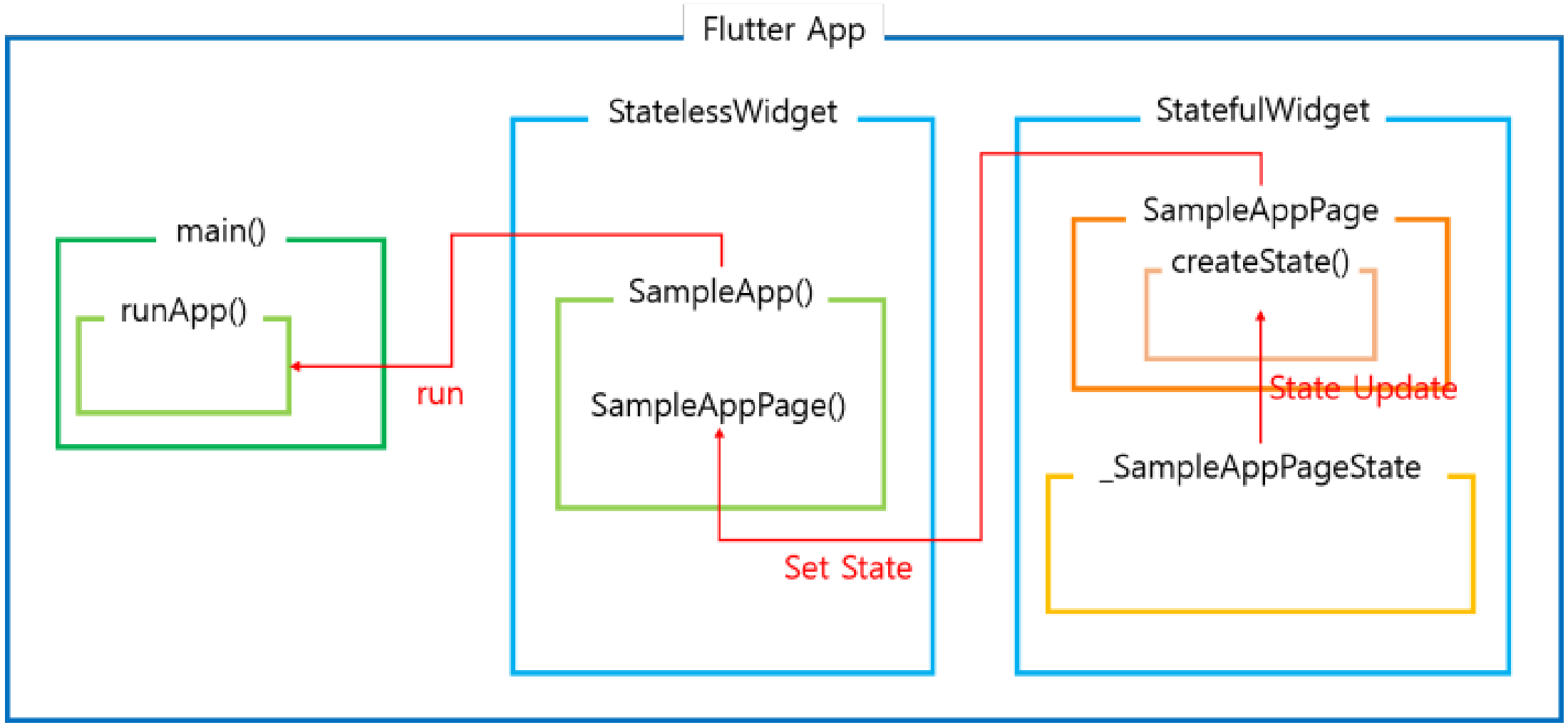
lib > main.dart

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

Run | Debug | Profile
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {···

class MyHomePage extends StatefulWidget {···

class _MyHomePageState extends State<MyHomePage> {···
```

- Main > MyApp(Stateless) > MyHomePage(Stateful) > _MyHomePageState(State)

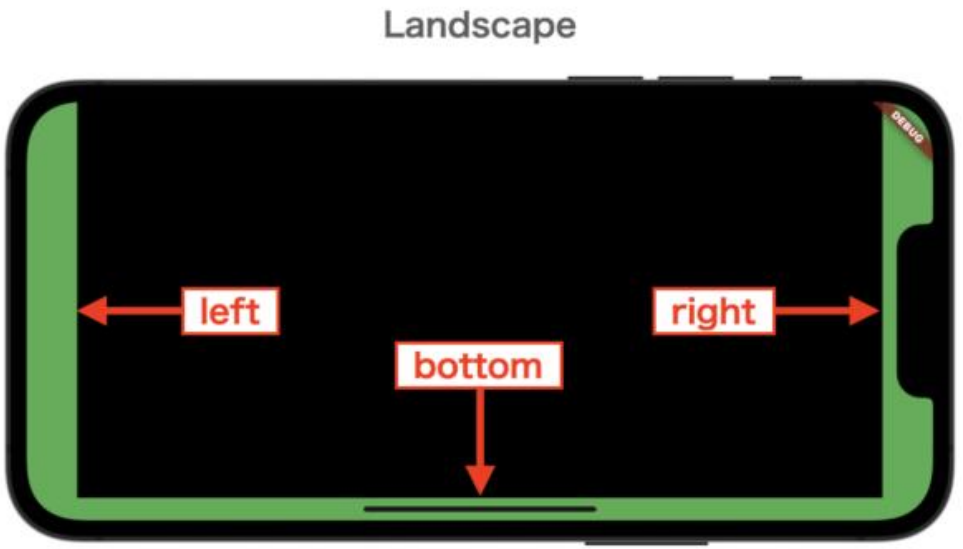
https://blog.codemagic.io/flutter-step-by-step-tutorial/

- Flutter 기본앱 변경



https://blog.naver.com/PostView.naver?blogId=minsuuuus&logNo=222384129853&categoryNo=34&parentCategoryNo=0

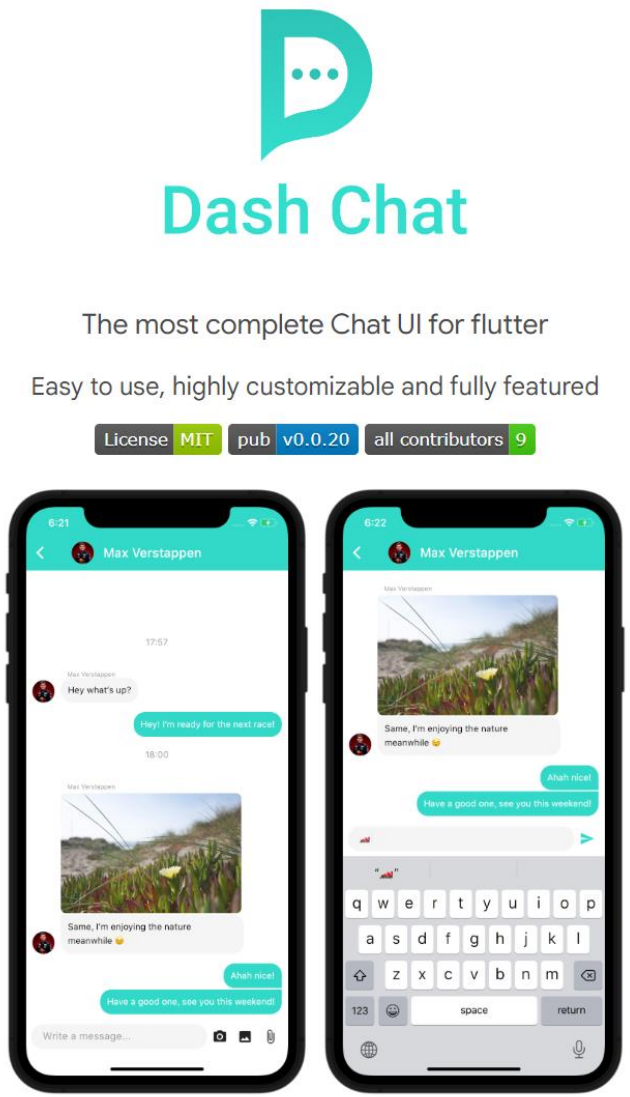
- Safearea 적용



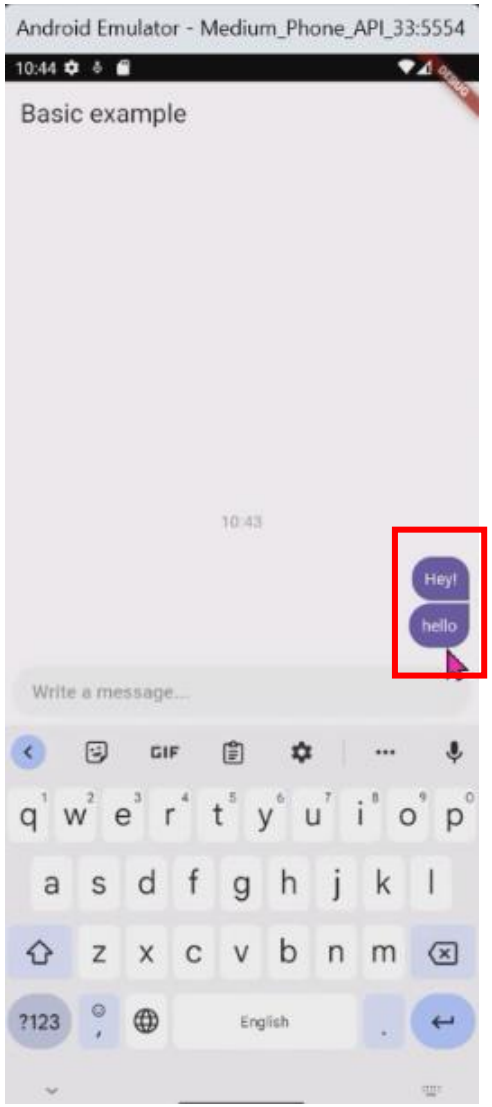
https://hiyoko-programming.com/1550/

5. Flutter문자대화

- Dash Chat 2 설치

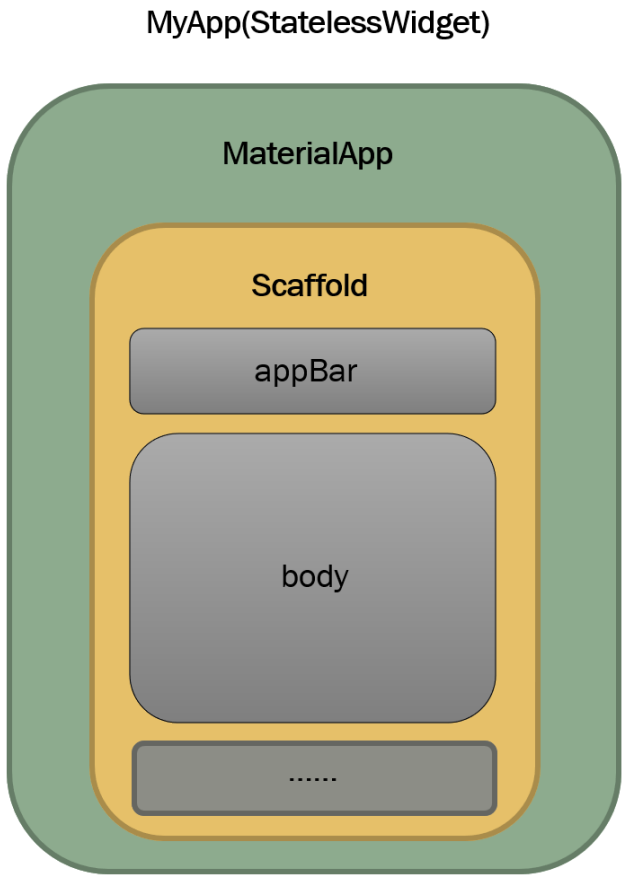


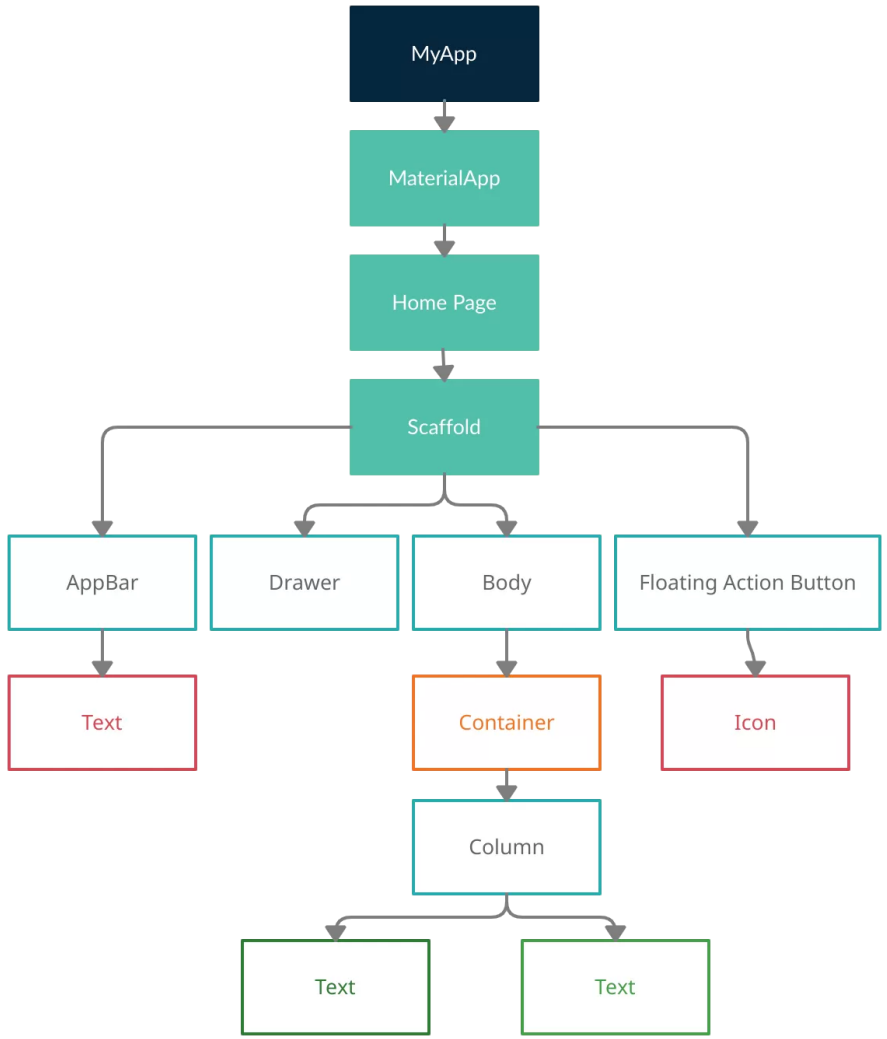
https://pub.dev/packages/dash_chat_2



user

- Dash Chat 2 설치





https://sothecode.tistory.com/13
https://www.fluttercampus.com/tutorial/5/flutter-widgets/

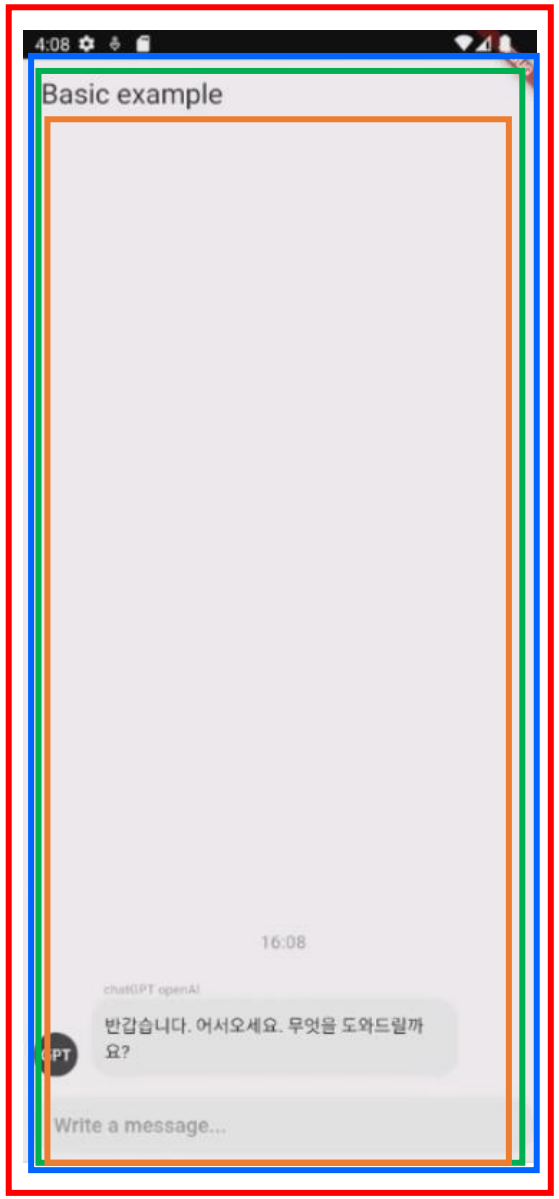
- Dash Chat 2 설치

MaterialApp

SafeArea

Scaffold

body

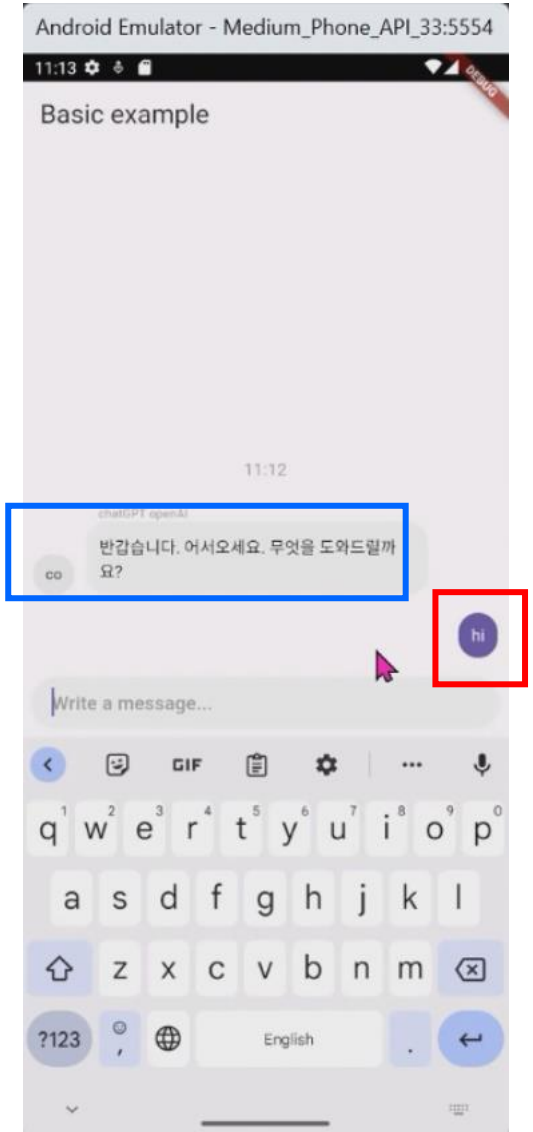


DashChat

ChatUser

ChatMessage

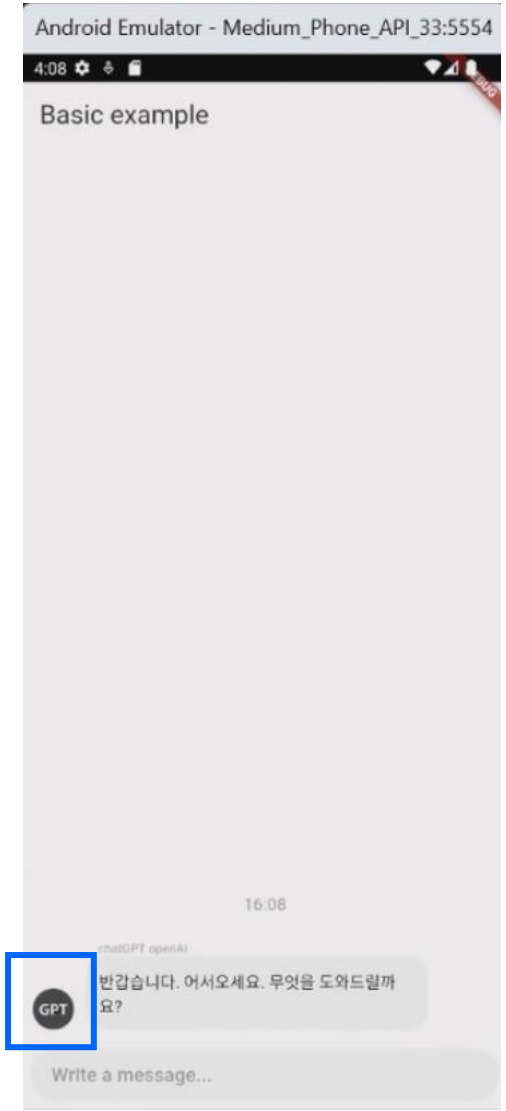
- 사용자 구분



user2

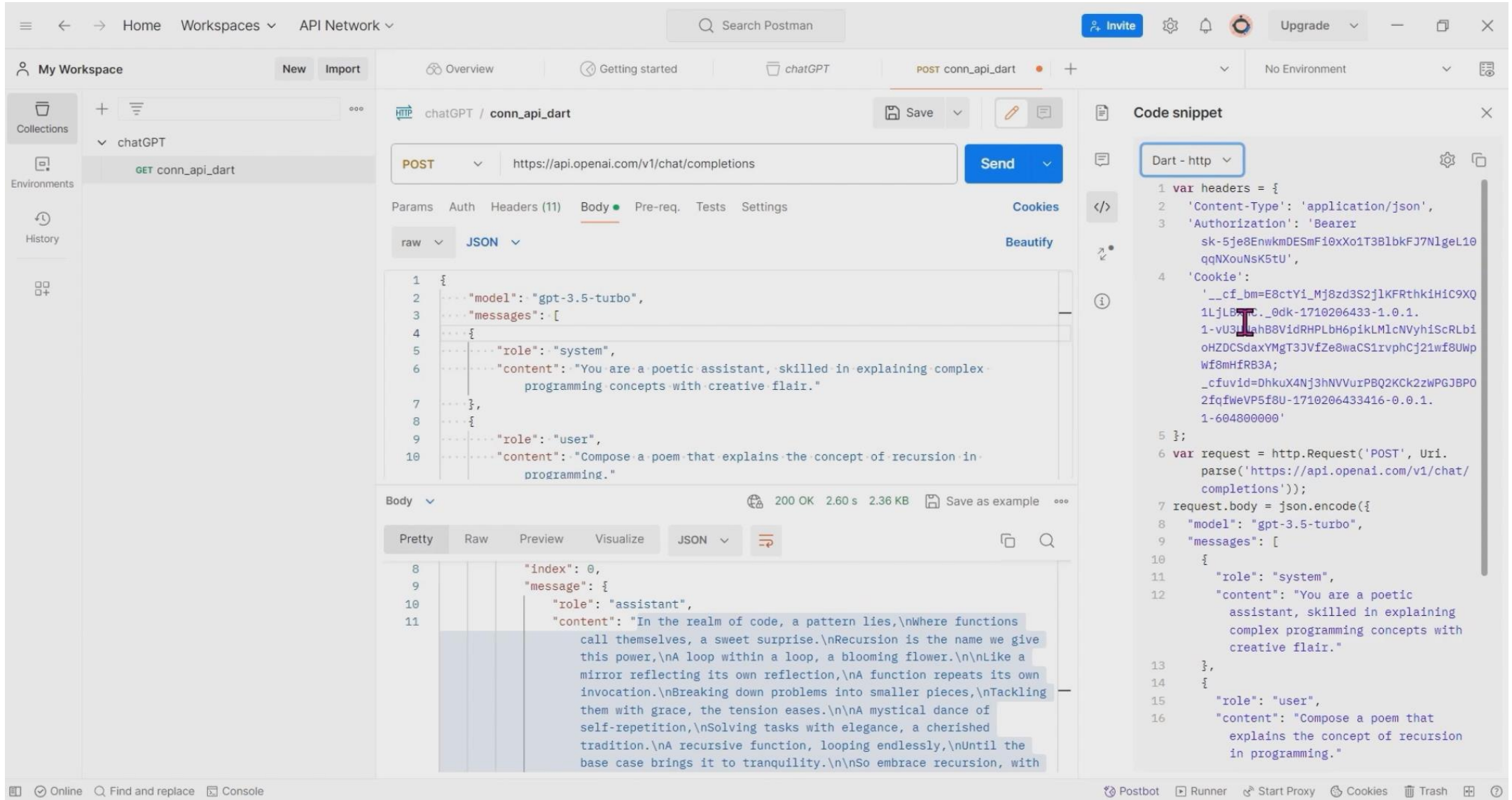
user1

- ProfileImage 적용

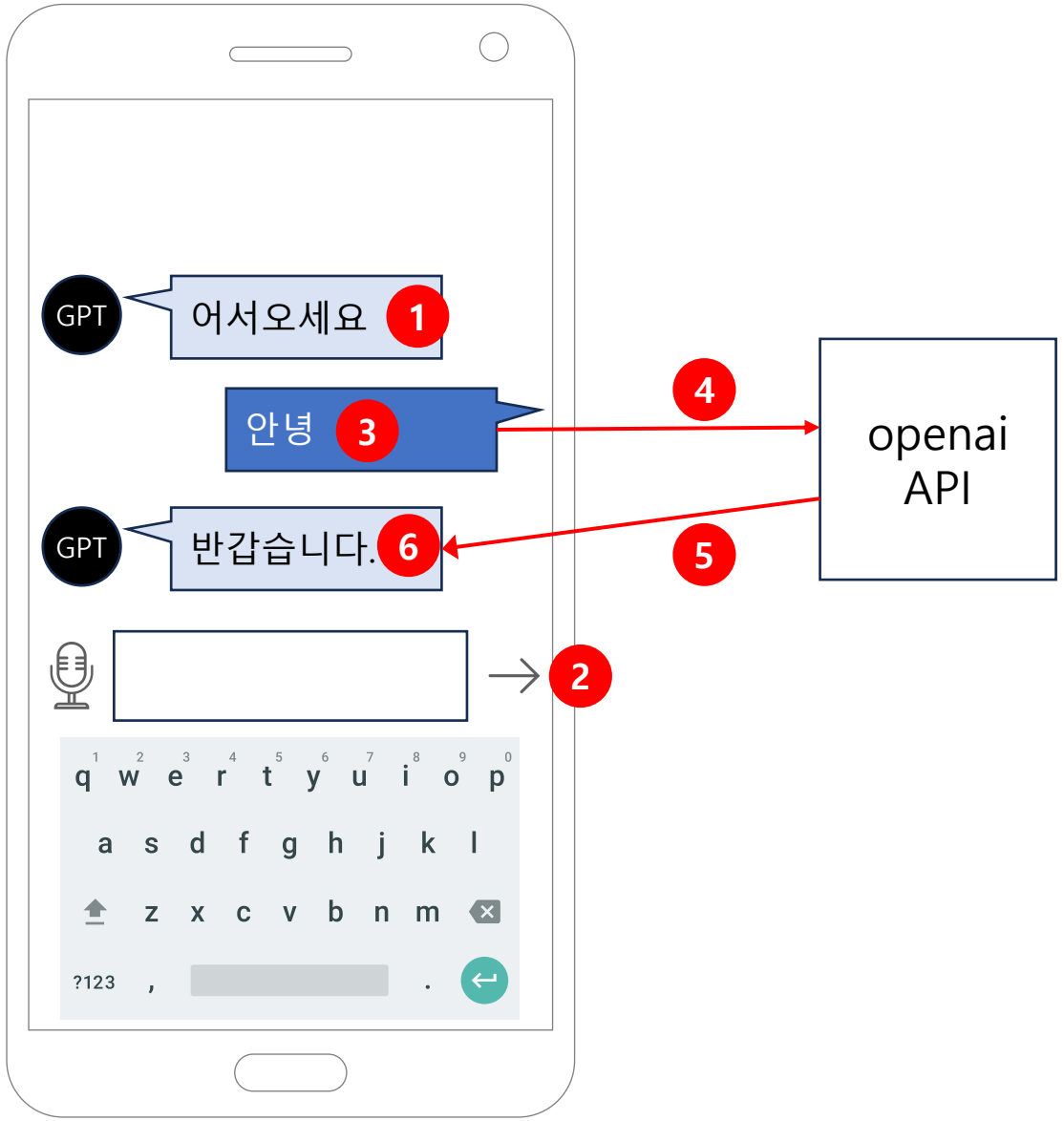


user2
ProfileImage

- Postman 통신 테스트



- chatGPT 통신 구조



1. 웰컴 메시지
2. 사용자가 입력한 메시지(ChatMessage.text)를 메시지창에 전송
3. 사용자메시지를 메시지창에 표시
4. sendMessageToServer()함수가 사용자메시지를 openai에 전송
5. API를 통해 응답 수신
6. 응답한 내용(value)을 메시지창에 표시

5. Flutter문자대화

- chatGPT 통신 함수

```dart
class _BasicState extends State<BasicScreen> {
  Future<String> sendMessageToServer(String message) async{
    var headers = {
      'Content-Type': 'application/json',
      'Authorization': 'Bearer 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',
    };
    var request = http.Request('POST', Uri.parse('https://api.openai.com/v1/chat/completions'));
    request.body = json.encode({
      "model": "gpt-3.5-turbo",
      "messages": [
        {
          "role": "user",
          "content": message,
        }
      ]
    });
    request.headers.addAll(headers);
    http.StreamedResponse response = await request.send();
    if (response.statusCode == 200) {
      String responseString = await response.stream.bytesToString();
      Map<String, dynamic> jsonResponse = json.decode(responseString);
      String result = jsonResponse['choices'] != null? jsonResponse['choices'][0]['message']['content']: "N
      print(responseString);
      return result;
    }
    else {
      print(response.reasonPhrase);
      return "ERROR";
    }
  }
```

message

↓

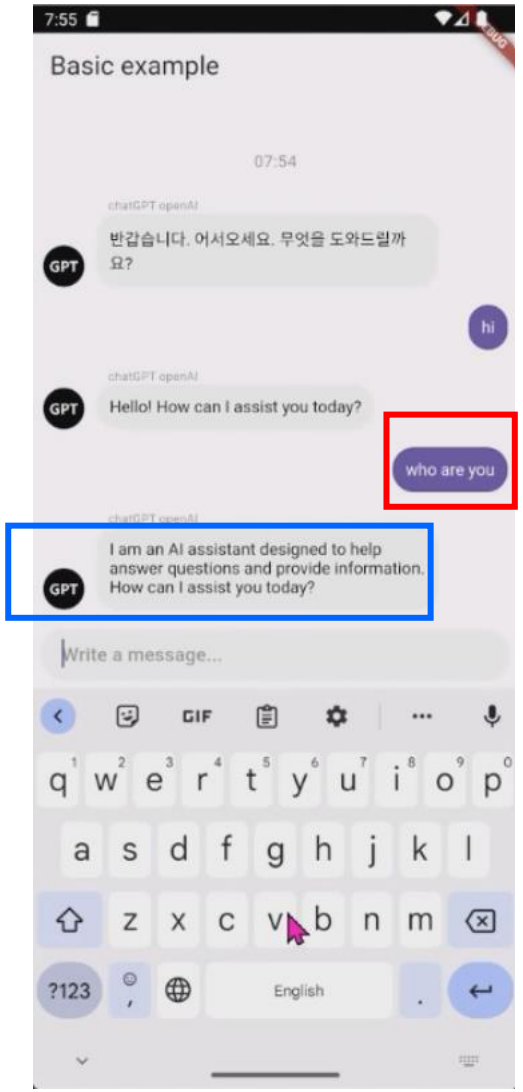
POST통신 설정
- URL
- Header
- Body(json)

통신 실행

예외처리
+
결과값 json 변환
+
content 추출
+
결과 return

- 문자 통신 테스트

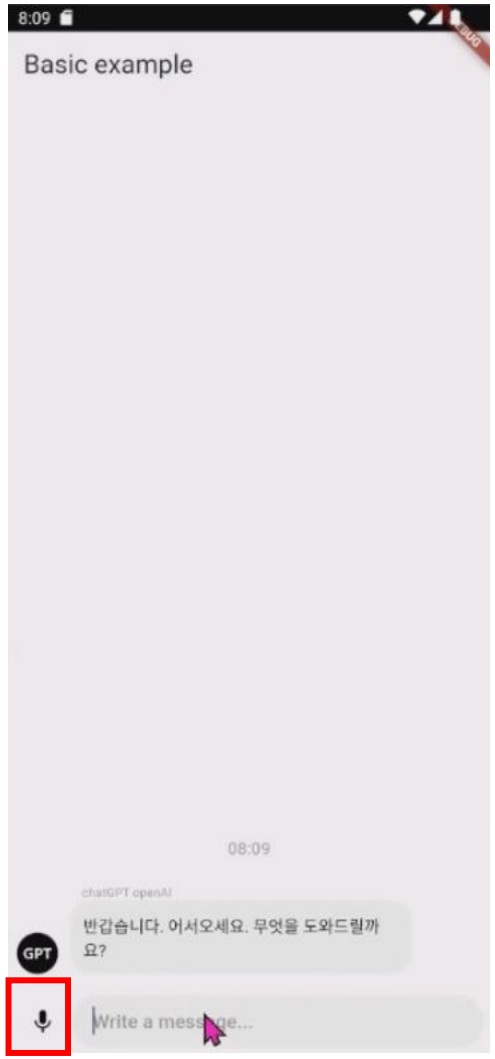
User2(chatGPT)의 답변



user1의 질문

6. Flutter음성대화

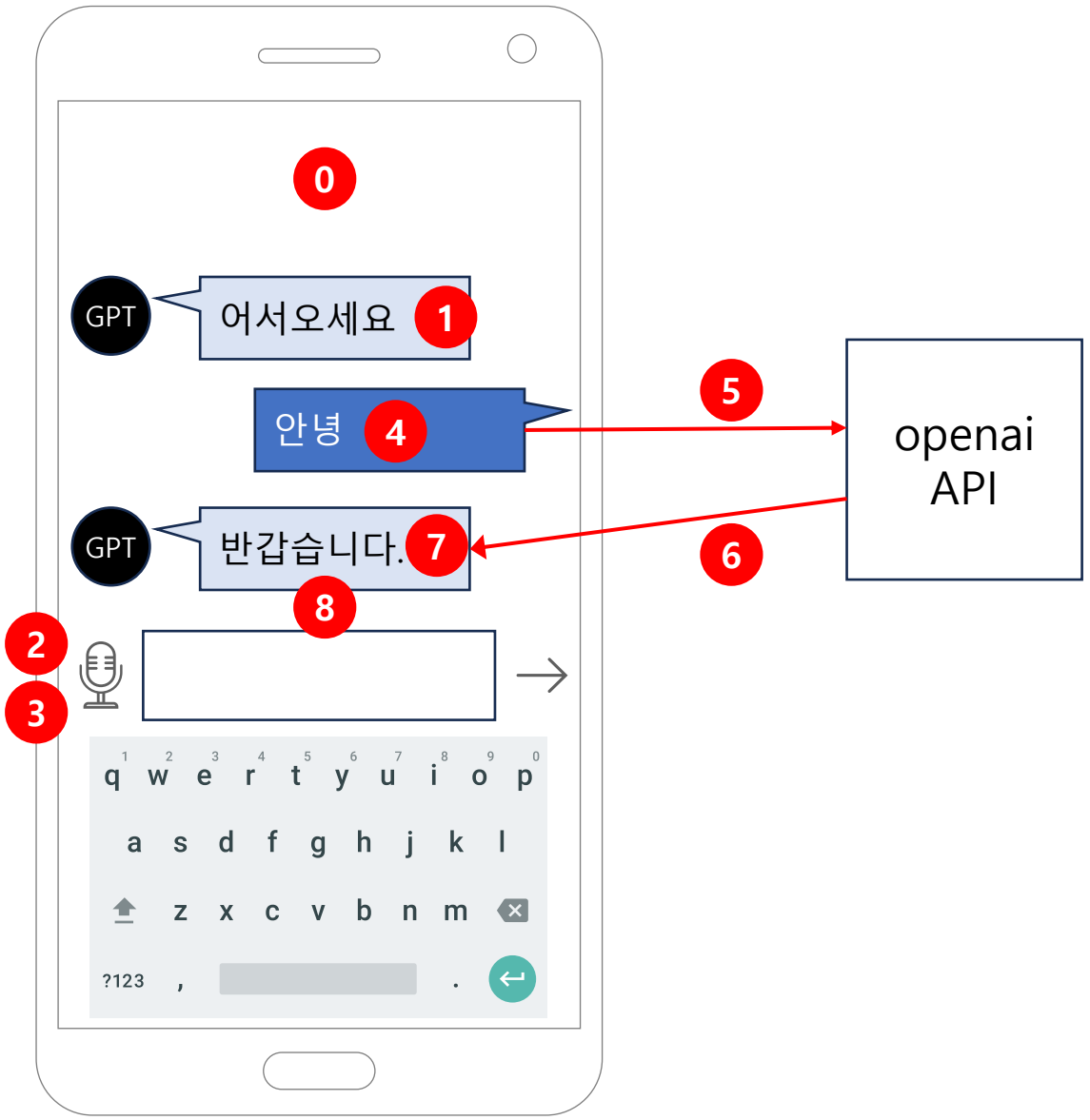
- 음성인식 버튼 추가



음성인식 버튼

6. Flutter음성대화

- STT 세팅



0 _initSpeech()

1 웰컴 메시지

2 STT On, 마이크활성화하여 음성입력 대기
_startListening()

3 STT Off, 마이크비활성화하여 음성입력 종료
_stopListening()

4 STT가 인식한 내용(_lastWords)을 메시지창에 표시

5 sendMessageToServer()함수가 사용자메시지를 openai에 전송

6 API를 통해 응답 수신

7 응답한 내용을 메시지창에 표시

8 TTS를 이용하여 응답 메시지를 음성으로 재생
(FlutterTts.speak(value))

- STT 기본코드

```dart
class _MyHomePageState extends State<MyHomePage> {
  SpeechToText _speechToText = SpeechToText();
  bool _speechEnabled = false;
  String _lastWords = '';

  @override
  void initState() {
    super.initState();
    _initSpeech();
  }

  /// This has to happen only once per app
  void _initSpeech() async {
    _speechEnabled = await _speechToText.initialize();
    setState(() {});
  }

  /// Each time to start a speech recognition session
  void _startListening() async {
    await _speechToText.listen(onResult: _onSpeechResult);
    setState(() {});
  }

  /// Manually stop the active speech recognition session
  /// Note that there are also timeouts that each platform enforces
  /// and the SpeechToText plugin supports setting timeouts on the
  /// listen method.
  void _stopListening() async {
    await _speechToText.stop();
    setState(() {});
  }

  /// This is the callback that the SpeechToText plugin calls when
  /// the platform returns recognized words.
  void _onSpeechResult(SpeechRecognitionResult result) {
    setState(() {
      _lastWords = result.recognizedWords;
    });
  }
```

변수 정의

초기화

음성인식 시작

음성인식 끝

음성인식 처리

https://pub.dev/packages/speech_to_text

- STT 음성인식

```
void _onSpeechResult(SpeechRecognitionResult result) {
  lastWords = "";
  if(result.finalResult){
    _lastWords = result.recognizedWords;
    print("최종 인식된 문장: $_lastWords");

    setState(() {
      messages.insert(0, ChatMessage(
        text: _lastWords,
        user: user1,
        createdAt: DateTime.now(),
      )); // ChatMessage
    });

    Future<String> data = sendMessageToServer(_lastWords);
    data.then((value){
      setState(() {
        messages.insert(0, ChatMessage(
          text: value,
          user: user2,
          createdAt: DateTime.now(),
        )); // ChatMessage
      });

    });

  }

}
```

최종 인식된 음성을 텍스트로 저장

인식된 텍스트를 채팅창에 출력

인식된 텍스트를 chatGPT API에 전송하여 답을 받음

- STT 환경설정

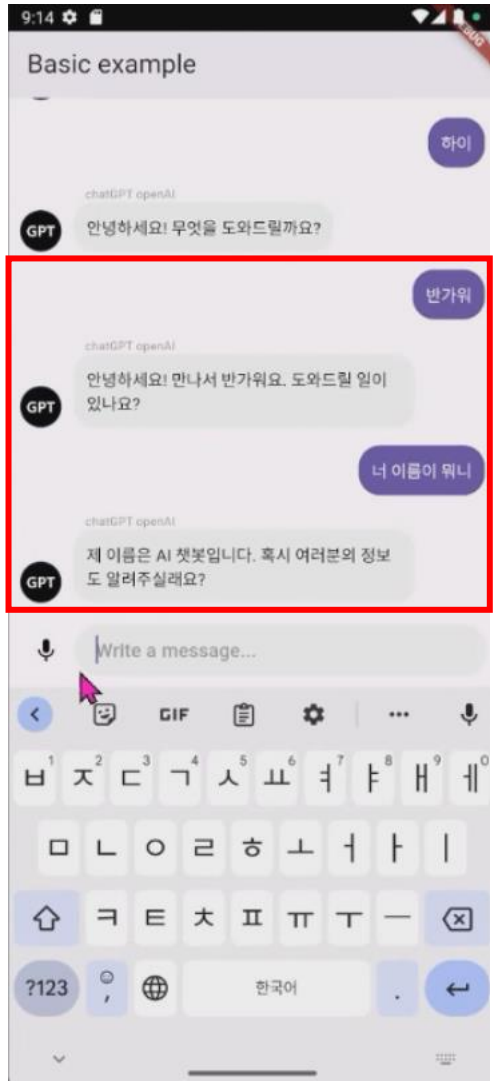
Software : /android/app/src/main/AndroidManifest.xml

Hardware

- 가상디바이스 오른쪽 … 클릭 > Microphone > Vitual microphone uses host audio input > check
- 가상디바이스 Settings(아래로 스와이프) > System > Languages > Add a language > 한국어 > 한국어를 1번째로 이동

minSDKVersion 수정 :
/android/app/build.gradle > minSDKVersion flutter.minSDKVersion을 minSDKVersion 21로 변경

- STT 테스트



텍스트 입력에는 텍스트 답변

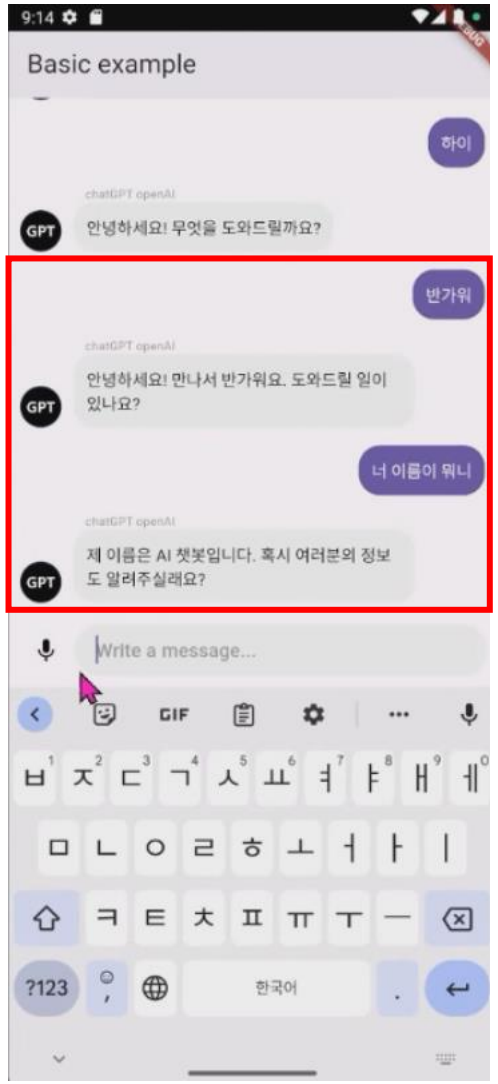
음석 입력시 텍스트로 변환하여
텍스트로 답변
(음성 답변은 추후 TTS추가로 구현)

6. Flutter음성대화

- TTS 세팅

FlutterTts flutterTts = FlutterTts();	인스턴스 생성
flutterTts.setLanguage("en-US"); flutterTts.setSpeechRate(1.0);	설정
flutterTts.speak("Hello World")	텍스트를 음성으로 재생

- 최종 테스트



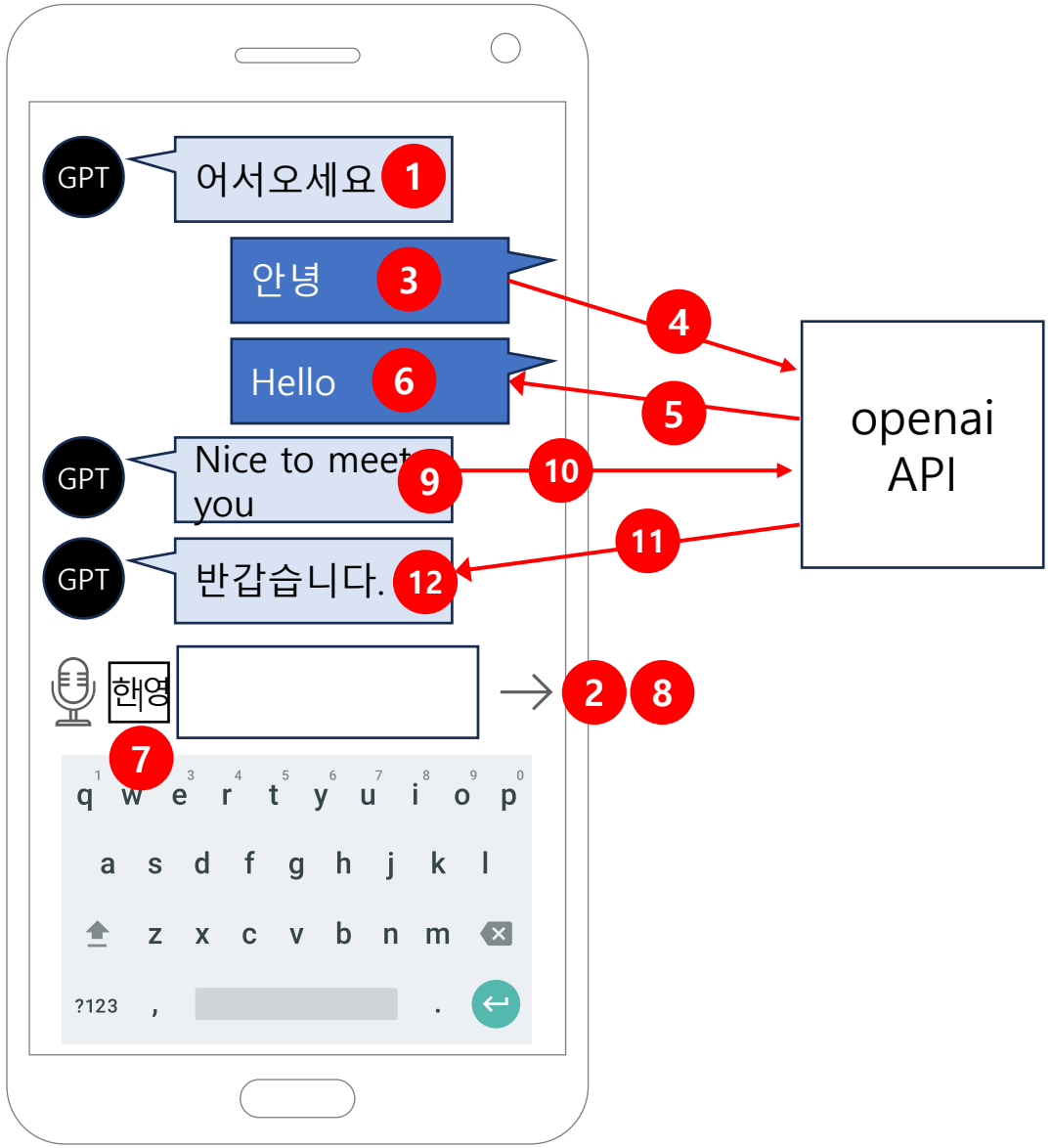
텍스트 입력에는 텍스트 답변

음석 입력시 텍스트로 변환하여 채팅창에 출력
텍스트 답변을 채팅창에 출력
텍스트 내용을 음성으로 읽어줌

7. 음성번역기

7. 음성번역기

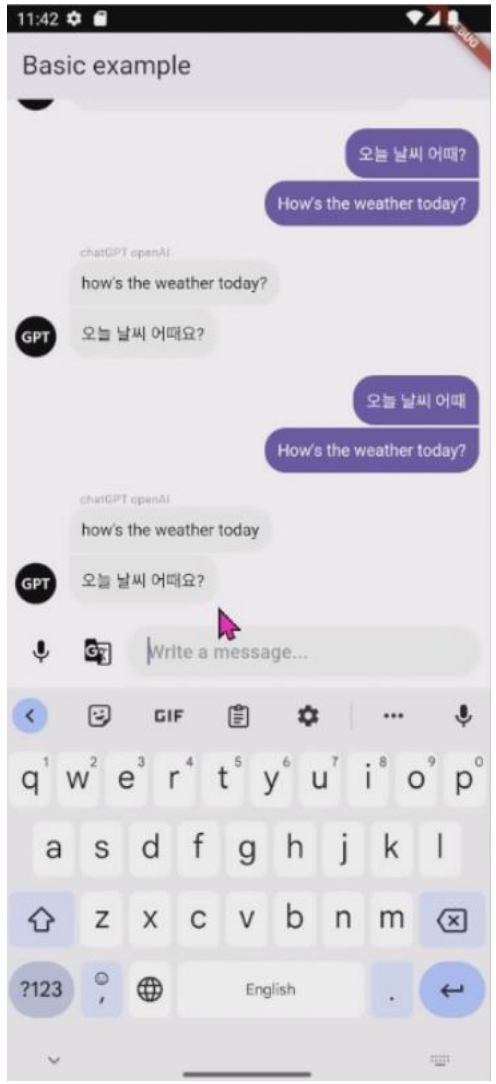
- 번역기 구조



1. 웰컴 메시지
2. 사용자1이 입력한 메시지를 메시지창에 전송
3. 사용자1의 메시지를 메시지창에 표시
4. 사용자1의 메시지를 API를 통해 openai에 전송(번역)
5. API를 통해 번역 내용 수신(한글 -> 영어)
6. 응답한 번역 내용을 메시지창에 표시
7. 사용자2를 위한 2번째 언어(영어) 선택
8. 사용자2가 입력한 메시지를 메시지창에 전송
9. 사용자2의 메시지를 메시지창에 표시
10. 사용자2의 메시지를 API를 통해 openai에 전송(번역)
11. API를 통해 번역 내용 수신(영어 -> 한글)
12. 응답한 번역 내용을 메시지창에 표시

7. 음성번역기

- 번역기 테스트



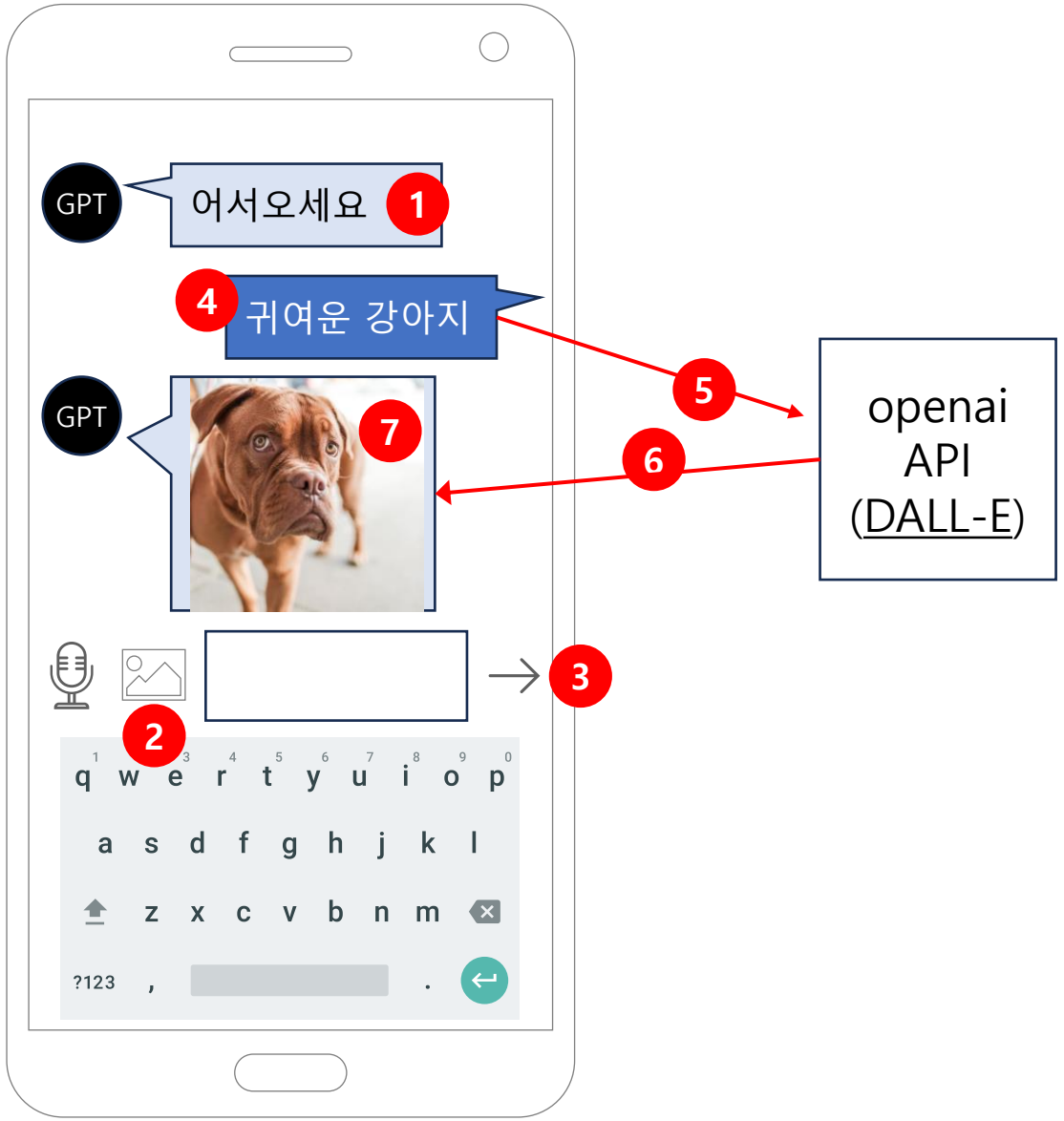
[텍스트]
입력내용 채팅창에 출력
chatGPT 통신 후 번역내용을 같은 사용자의 채팅창에 출력

[음성]
음성인식한 후
인식내용을 채팅창에 출력
chatGPT 통신 후 번역내용을 같은 사용자의 채팅창에 출력
출력내용을 음성으로 읽어줌

[번역기능 사용시]
번역 대상 언어와 입력 사용자가 반대로 바뀜

8. 이미지생성

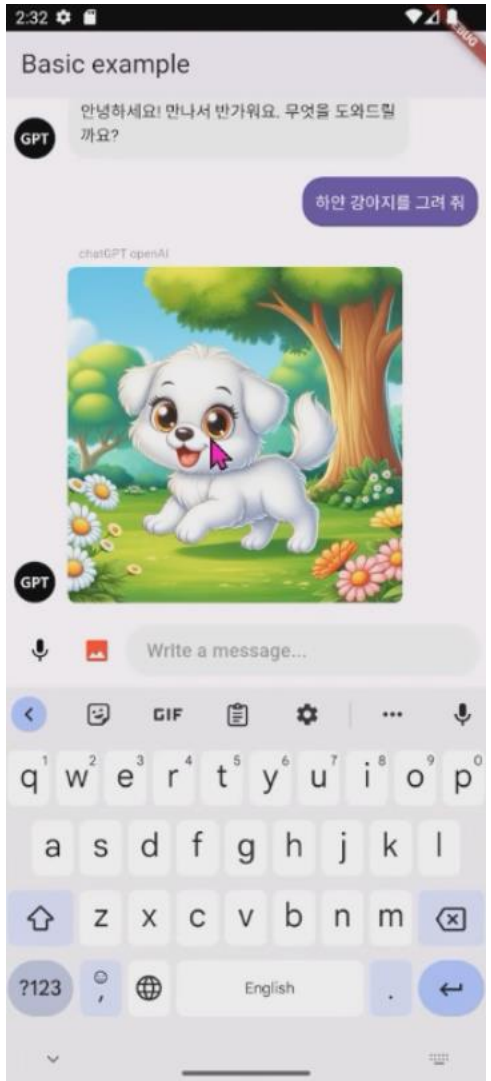
- Dall-e 구조



1. 웰컴 메시지
2. 이미지 생성 On
3. 사용자가 입력한 메시지를 메시지창에 전송
4. 사용자메시지를 메시지창에 표시
5. 사용자메시지를 API를 통해 openai(DALL-E)에 전송
6. API를 통해 이미지 URL 수신
6. URL의 이미지를 메시지창에 표시

8. 이미지생성

- Dall-e 테스트



[텍스트]
입력내용 채팅창에 출력
chatGPT 통신 후 번역내용을 같은 사용자의 채팅창에 출력

[음성]
음성인식한 후
인식내용을 채팅창에 출력
chatGPT 통신 후 번역내용을 같은 사용자의 채팅창에 출력
출력내용을 음성으로 읽어줌

[Dall-e기능 사용시]
응답결과가 생성된 이미지로 바뀜

9.파인튜닝

- 파인튜닝

1. 웰컴 메시지
2. 사용자가 입력한 메시지를 메시지창에 전송
3. 사용자메시지를 메시지창에 표시
4. 사용자메시지를 API를 통해 openai(파인튜닝모델)에 전송
   (미리 아파트 실거래가 데이터셋을 통해 파인튜닝을 실행해두어야 함)
5. API를 통해 응답 수신
6. 응답한 내용을 메시지창에 표시

감사합니다