**VSCode를 이용한 Speech To Text 및 번역 기능 구현**



|  |  |
| --- | --- |
| 과목명 | 운영체제 |
| 담당교수님 | 최경희 교수님 |
| Team Number | 16조 |
| 조원 | 산업공학과 201620239 최유림  미디어학과 202021109 김세훈 |

**목차**

1. 문제 정의

2. 본문

2.1. Visual Studio code와 파이썬 환경 구축

2.2. 구글 클라우드 플랫폼 API

2.3. 파일의 구성

2.4. 헤더 파일(header.py)

2.5. GUI(gui.py)

2.6. Wav 파일을 클라우드 버킷으로 전송(upload\_wav.py)

2.7. 영상을 음성 파일로 변환하여 텍스트로 반환(speech\_to\_text.py)

2.8. 텍스트를 네이버 Papago API를 통해 번역하기(translate\_test.py)

2.9. 위의 작업들을 영상과 자막으로 함께 반환(Video\_out.py)

2.10. Main.py

1. 결론

3.1. 테스트 결과

3.2. 팀원 역할 분담

3.3. 참고 사이트

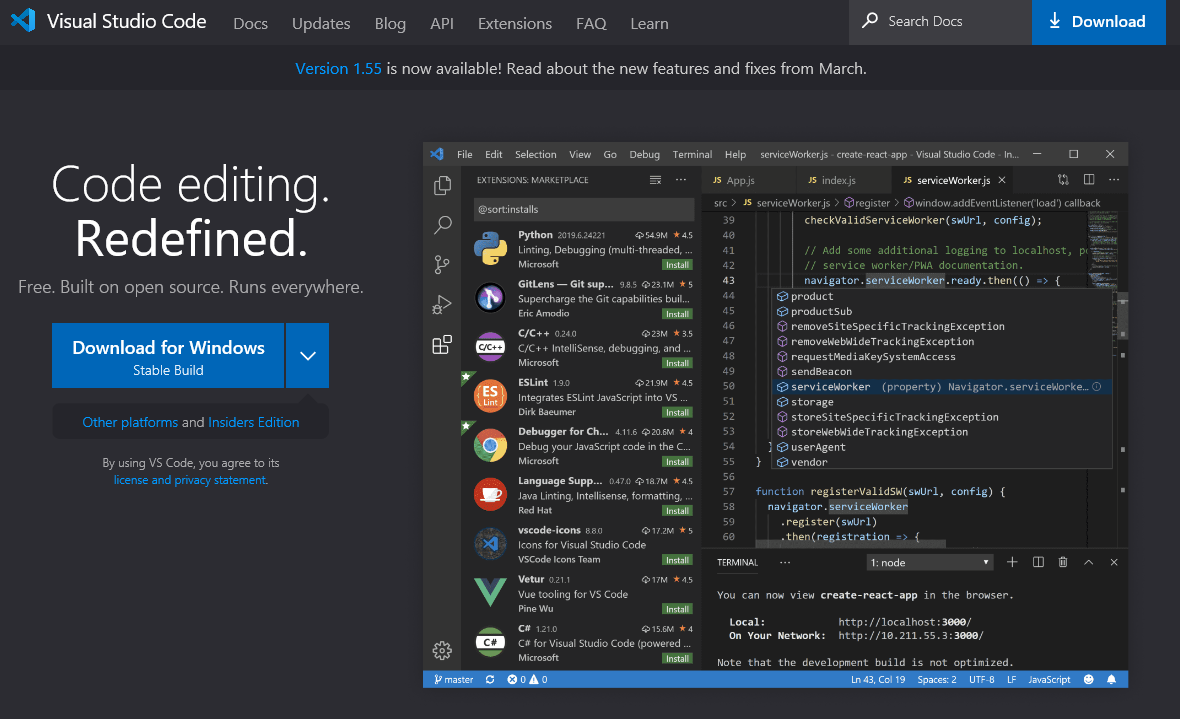
1. **문제 정의**

저희 조는 VSCode(Visual Studio Code)에 파이썬 환경을 구축하여 이를 통해 Speech-To-Text 과 이를 번역하여 영상의 자막으로 내보내는 기능을 구현해보았습니다. Speech-To-Text란 말 그대로 사람의 음성을 인식하여 이를 텍스트로 변환시켜주는 기능을 일컫는 말입니다. 우선 음성 인식을 할 1분 내외의 샘플 영상 4개를 구했습니다. 음성 인식을 제대로 하기 위해 발음이 보다 정확한 아나운서들의 영상으로 구했고, 이렇게 구한 영상들을 input 파일에 넣으면 버킷에 전송하도록 구현하였습니다. 구현 과정으로는 우선 사용자가 번역하고자 하는 영상과, 그 영상의 언어와 번역하고자 하는 언어를 선택할 수 있는 GUI를 구현했습니다. 사용자가 번역할 영상을 선택하면 그 영상이 버킷에 있는지 확인하고, 버킷에 있다면 이 영상을 음성 인식을 통해 텍스트로 변환하여 음성 인식을 한 텍스트 파일과 번역한 텍스트 파일을 반환하도록 하였습니다. 이렇게 음성 인식과 번역 작업을 마치면, gui를 통해 비디오 파일과 아래에 자막이 나오도록 하였습니다. 이어서 자세한 구현 과정에 대하여 설명 드리도록 하겠습니다.

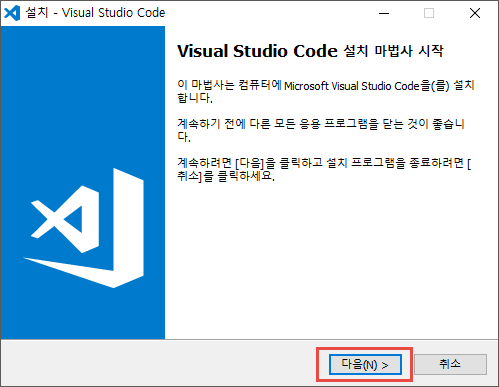
1. **본문**
   1. **Visual Studio code와 파이썬 환경 구축**

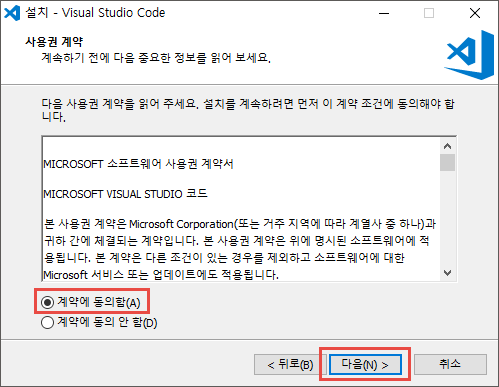
우선 다음 사이트에 들어가서 vscode를 다운로드 받습니다.

<https://code.visualstudio.com/>

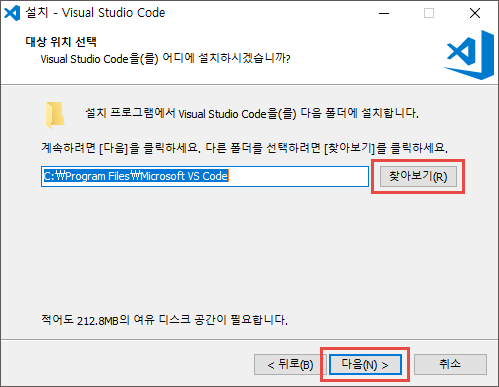


저희 같은 경우 이미 팀 프로젝트 이전에 VSCode를 다운로드 받은 상태였기에 다른 사이트를 참고하여 작성하였습니다.

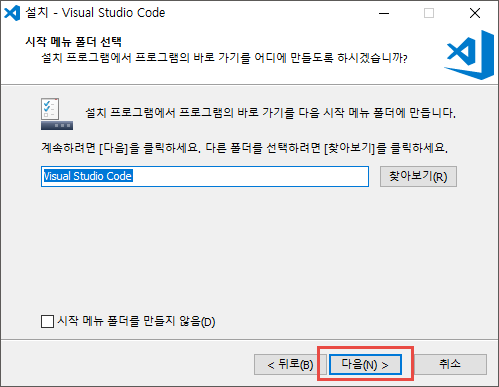


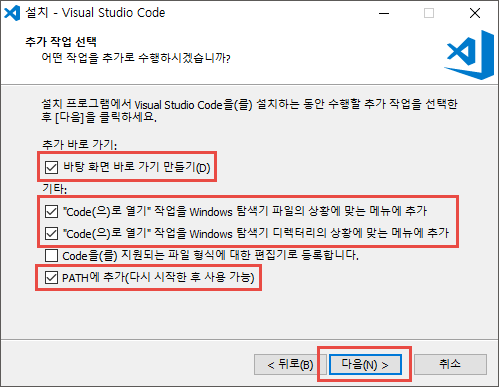


“계약에 동의함”을 클릭합니다.



원하는 위치에 다운로드 합니다.

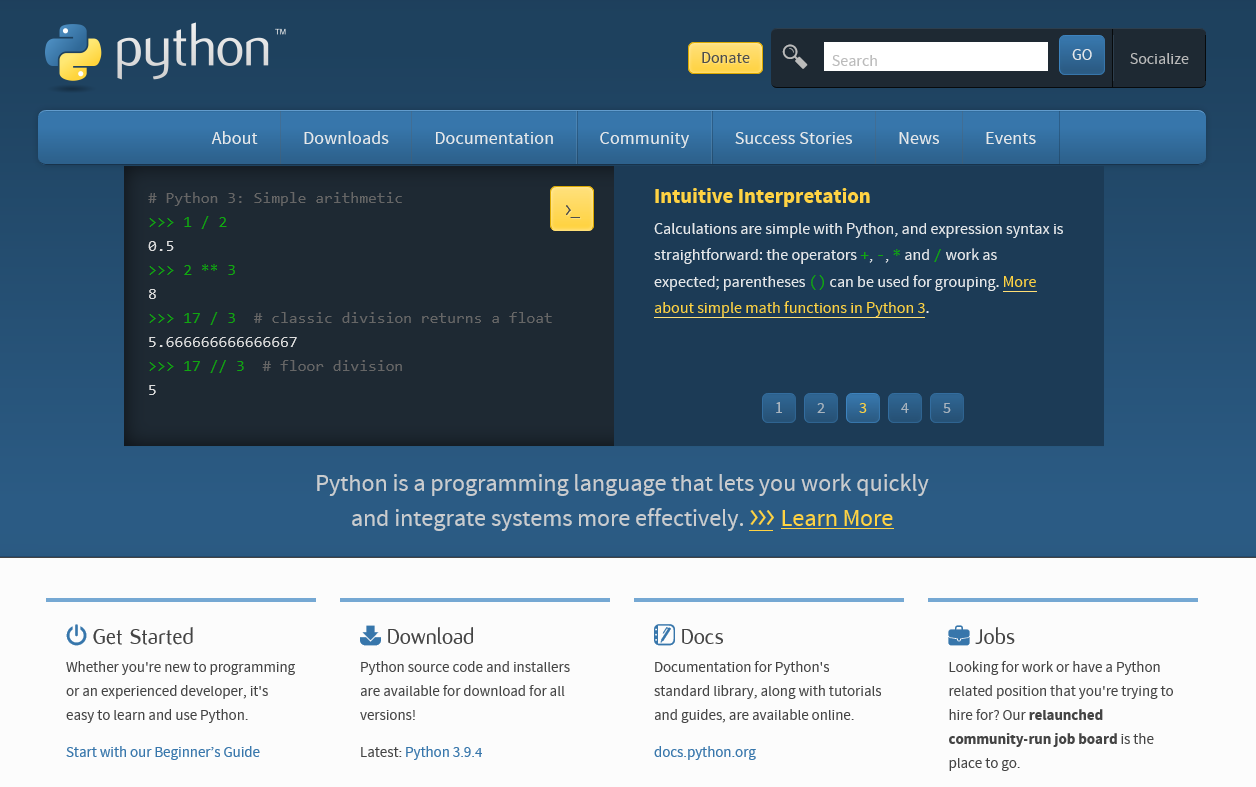




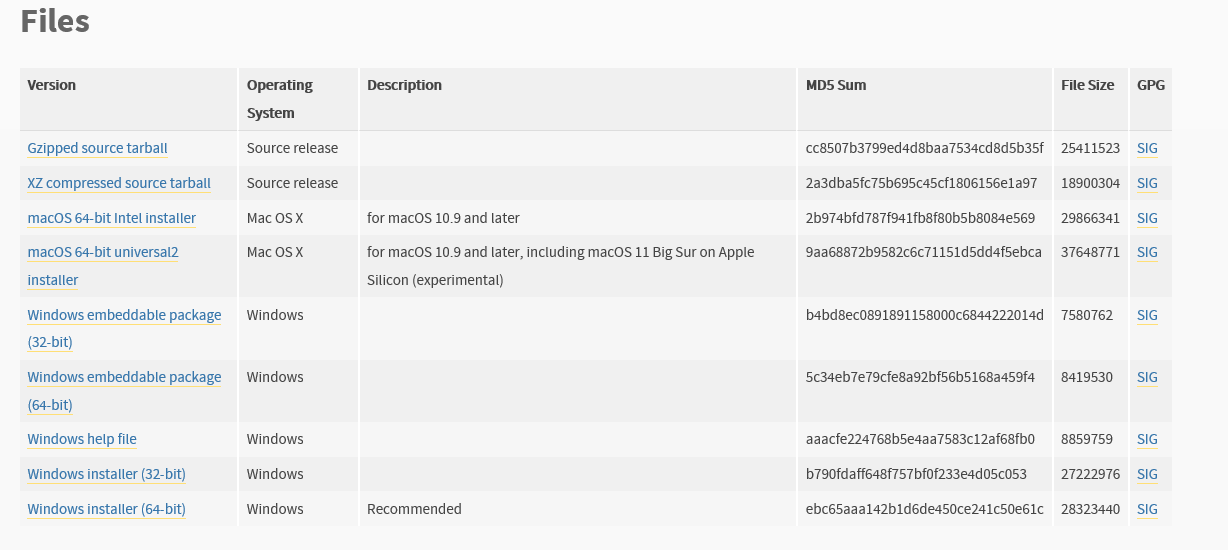
다음과 같은 사항들을 체크하고 다음을 클릭합니다. 설치를 한 뒤에 마침을 누르면 설치가 완료됩니다.

그 다음에 파이썬 환경을 구축해야 합니다. 다음 사이트에서 파이썬을 다운로드 받습니다.

<https://www.python.org/>



빨간 네모로 표시한 하이퍼링크를 클릭한다.

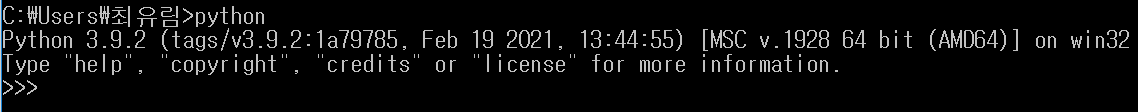


본인의 환경에 맞는 파이썬 파일(installer라고 쓰여 있는 것)을 다운로드 받는다.

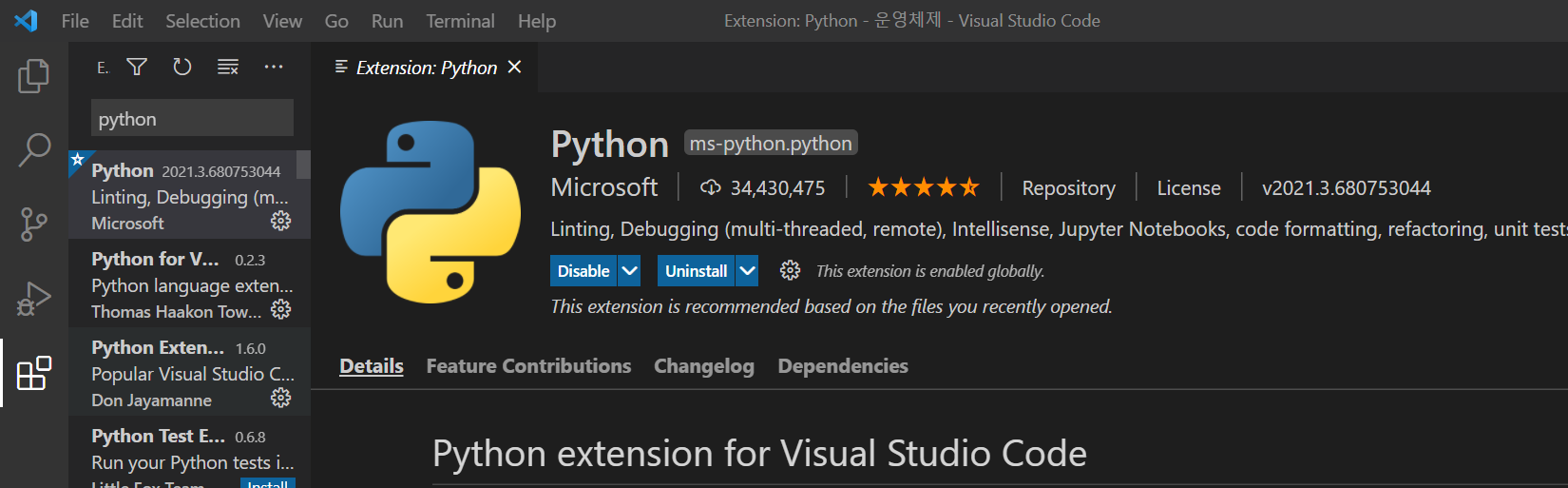


저희는 이미 다운로드를 한 상태이기 때문에 위와 같이 창이 뜨지를 않아서 다른 사이트를 참고하여 설명을 드리겠습니다. 아래에 “Add Python 3.7(또는 어떤 버전) to PATH” 를 클릭하면 파이썬이 환경 변수에 추가되는 것으로, 체크를 꼭 해야 합니다.

Cmd에 python이라고 입력하였을 때 다음과 같이 나온다면 VSCode와 연동할 수 있습니다.

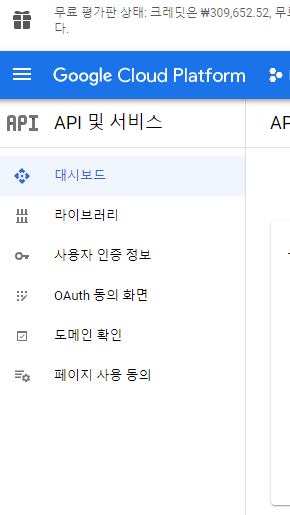


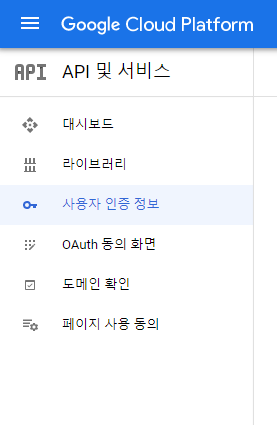
파이썬 설치를 한 뒤 아래 이미지(Extensions)를 클릭한 뒤 python을 검색하여서 나온 Python 을 install 합니다.



* 1. 구글 클라우드 플랫폼 API

다음은 speech-to-text를 위한 구글 클라우드 API를 받는 과정입니다. 우선 구글 클라우드 서비스에 접속을 하여 API 및 서비스로 들어갑니다.

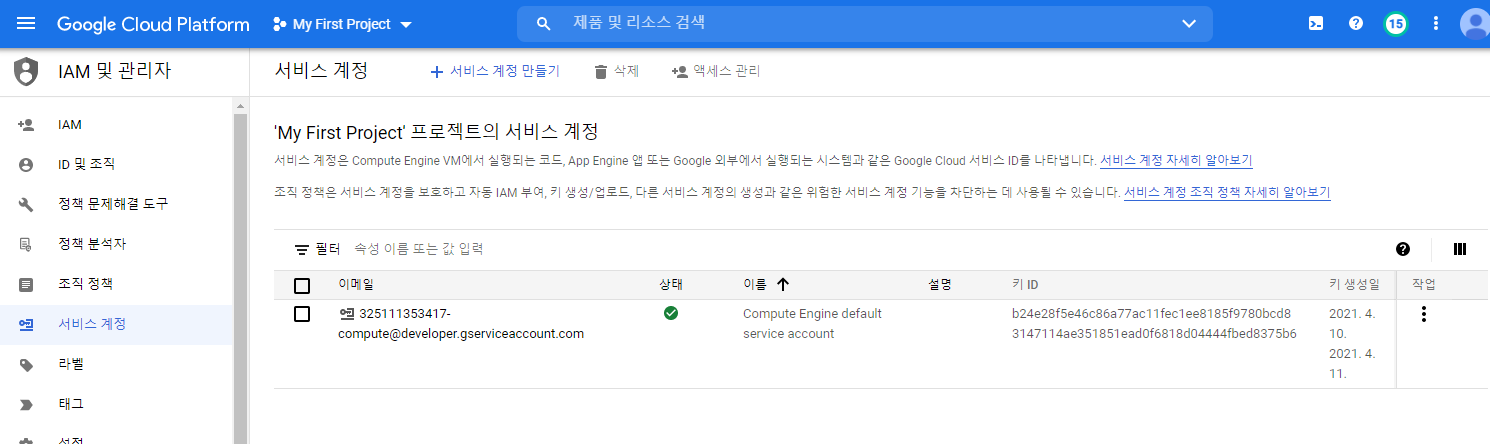


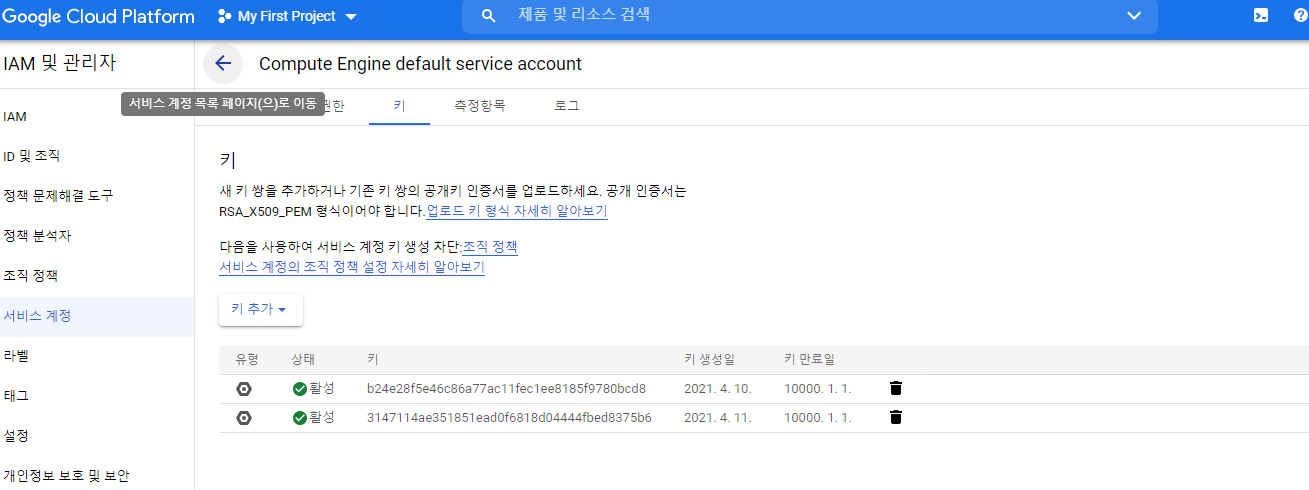


그렇게 해서 다음과 같이 speech-to-text api를 생성합니다.

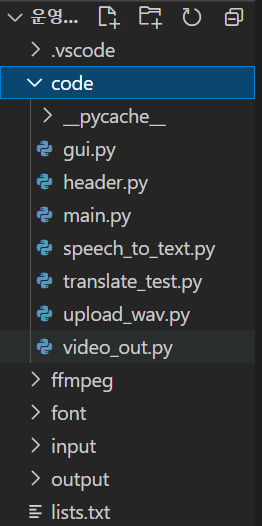


또한 구글 클라우드 서비스에서 IAM 및 관리자로 들어가서 서비스 계정으로 들어가서 다음과 같이 key를 생성합니다.





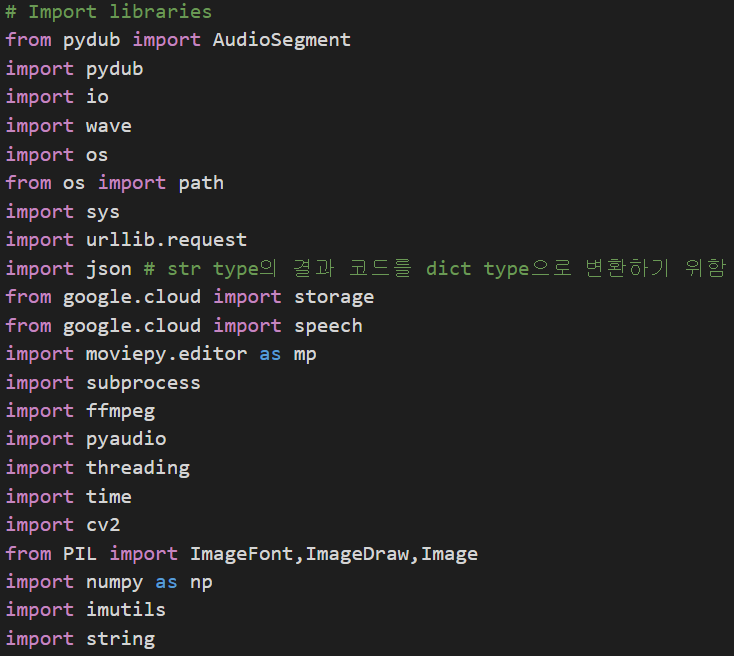
* 1. **파일의 구성**



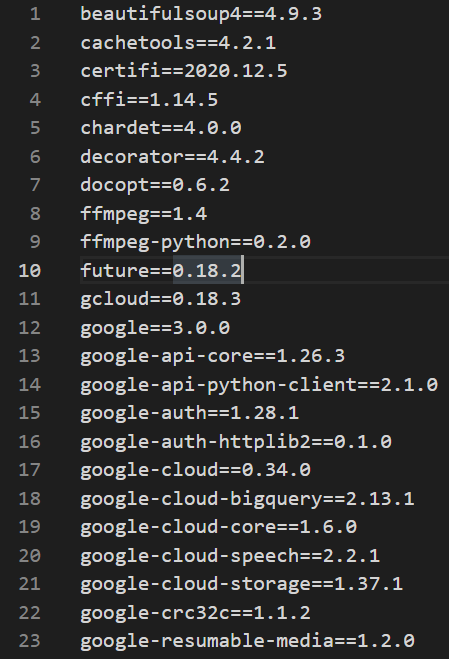
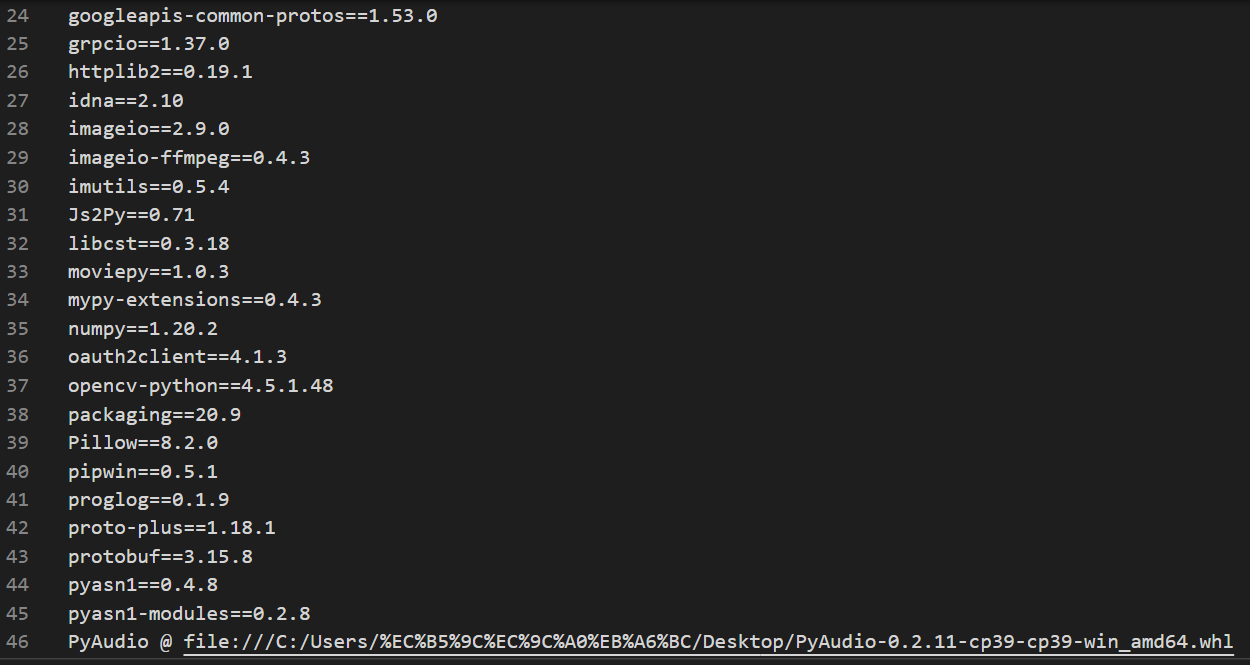
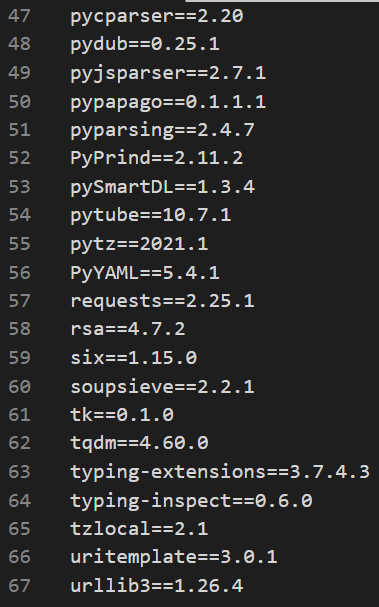
위와 같이 운영체제 폴더 내에 다음과 같은 폴더들이 있고, 파이썬 코드는 main을 포함하여 총 7개가 있습니다. Input 폴더에는 영상들이 들어있고, Output 폴더에는 음성 인식과 그것의 번역 텍스트 파일, mp3와 wav 음성 파일, 자막의 타임 스탬프에 대한 프로그램의 결과물이 저장되어 있습니다.

* 1. **헤더 파일(header.py)**

다른 파이썬 코드들에 앞서 헤더 파일부터 설명 드리겠습니다. 헤더 파일에는 나머지 6개 코드에서 사용되는 라이브러리들을 import 하고, 또한 6개의 파일에서 사용되는 변수 및 함수들에 대하여 정의했습니다.

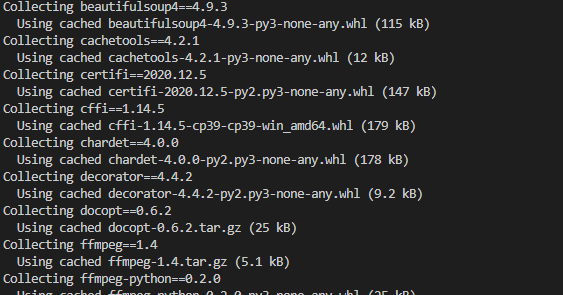


위의 사진이 저희가 영상을 음성 파일로 바꾼 뒤 이를 음성 인식하여 텍스트 파일로 나타내고, 이를 번역하여 텍스트 파일로 나타낸 뒤 영상의 자막으로 다는 과정까지 사용한 라이브러리들입니다.

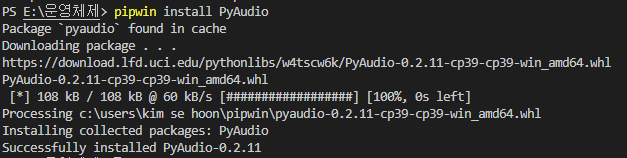
  

또한 위 사진은 이러한 라이브러리를 사용하기 위해 pip install 한 사항들입니다. 이러한 사항들을 텍스트 파일로 저장한 뒤에 다음과 같이 작성하면 한번에 install 할 수 있습니다.

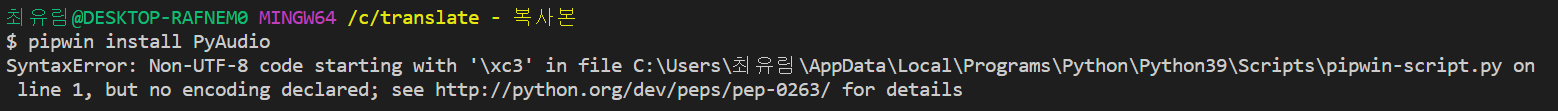




그리고 다음과 같이 PyAudio를 install 해줍니다.

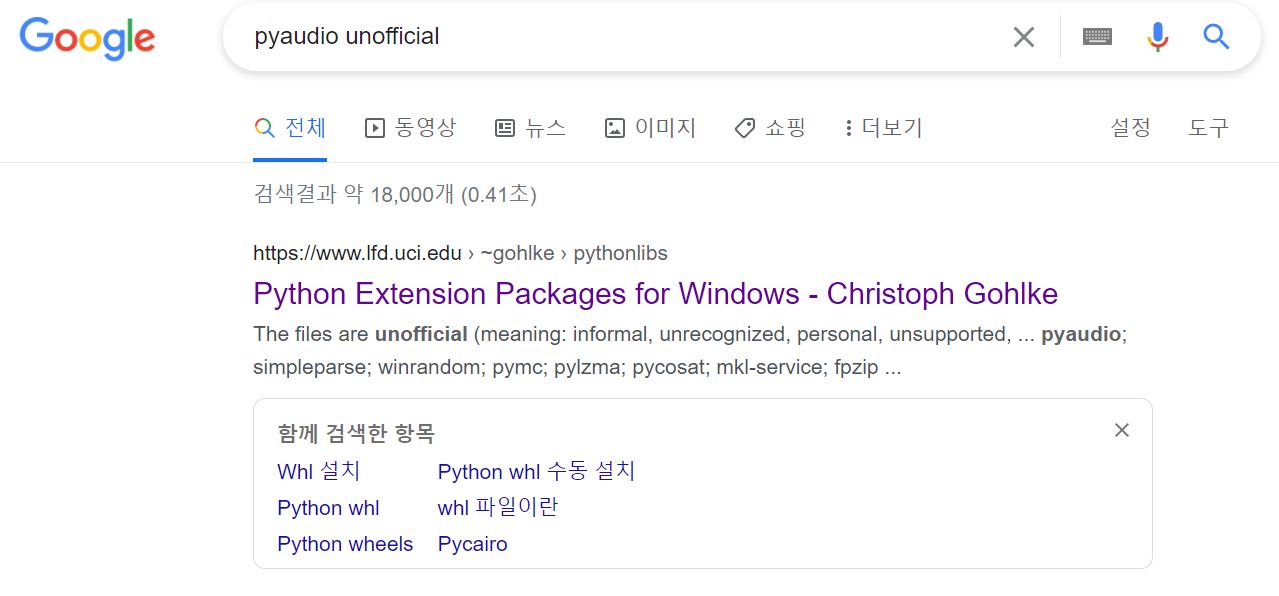


이 때, 위에서 PyAudio의 경우 다른 사항들과 예외적으로 “pipwin install PyAudio”를 해야 하는데, 이 부분을 vscode terminal에서 실행할 때 한 팀원분은 문제 없이 실행이 되었지만, 저 같은 경우에는 다음과 같은 오류가 발생하는 것을 볼 수가 있었습니다.

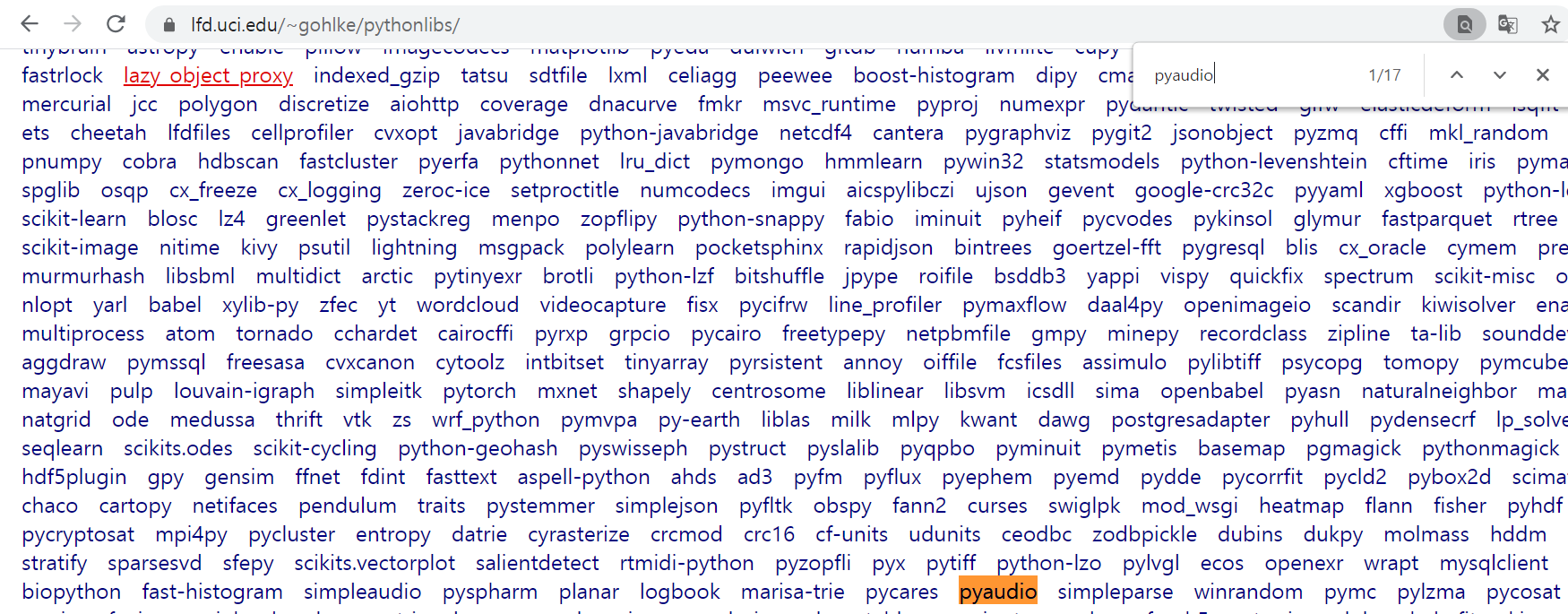


따라서 이런 오류가 났을 때 저는 vscode 터미널이 아닌, Windows 검색창에 “cmd”을 검색하여 명령 프롬프트 창에서 다음과 같이 해결했습니다.

구글에 “pyaudio unofficial”을 검색합니다. 검색하여 나온 맨 위의 사이트를 클릭하여 들어갑니다.



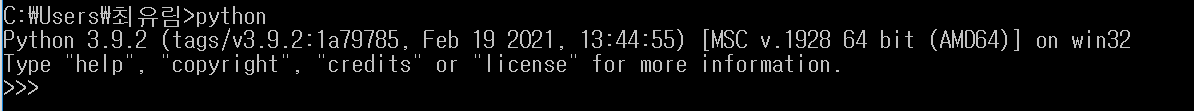
Ctrl+f 를 통해 “pyaudio”를 검색하고, pyaudio라 쓰여있는 하이퍼링크를 클릭합니다.



이러면 다음과 같이 여러 파일들이 나옵니다.



cmd창에서 “python”을 입력하여 파이썬 버전을 확인합니다.

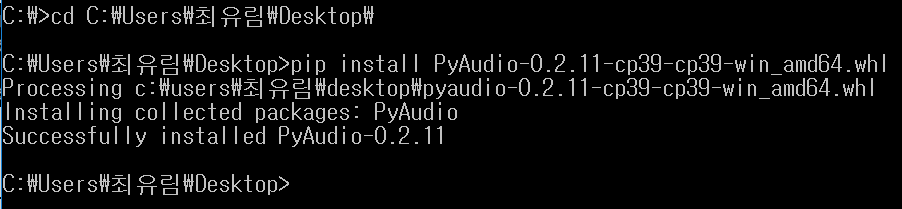


Python 3.9.2 버전을 사용함을 알 수 있고, 옆에 AMD64라고 쓰여 있는 것을 볼 수가 있습니다. 따라서 위에 네모 친 “cp39”에 “amd64”라고 명시되어 있는 파일을 다운 받습니다.

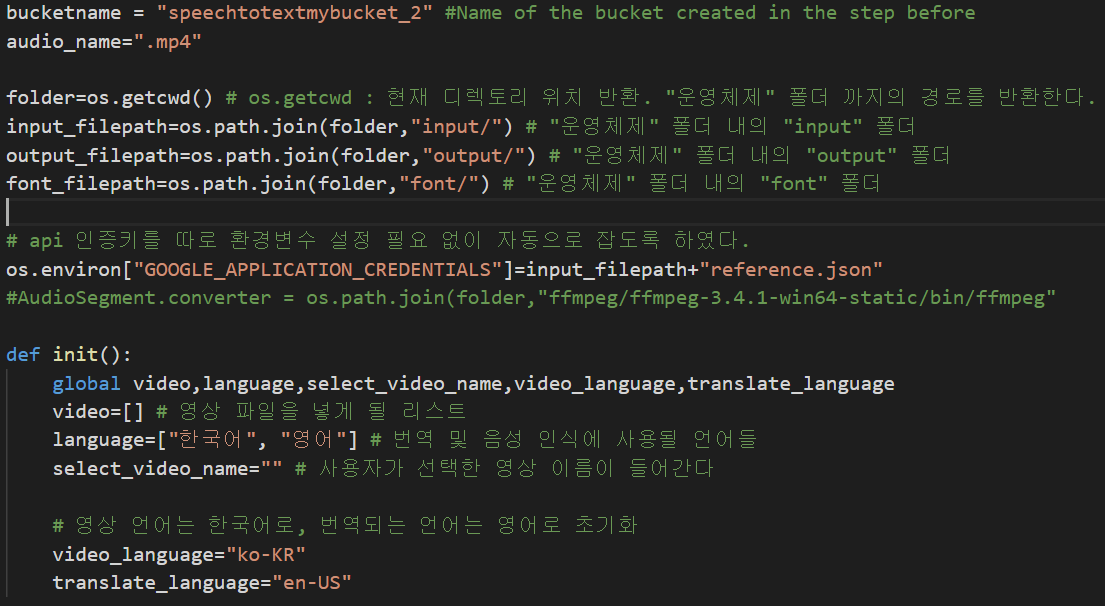
그러고 난 뒤, cmd창에 “cd 파일 위치”를 입력해줍니다. 파일 위치는 다음과 같이 파일에 오른쪽 cursor를 클릭한 뒤 속성을 클릭하여 볼 수 있습니다.



위치가 위와 같이 C:\Users\최유림\Desktop 이므로 다음과 같이 작성합니다. “cd 파일위치”를 입력한 뒤에 “pip install 파일명”을 입력하면 PyAudio가 install이 됩니다.



헤더파일 설명을 다시 이어서 하겠습니다. 다음은 앞으로 6개의 파일에서 사용될 변수들과 함수 입니다.

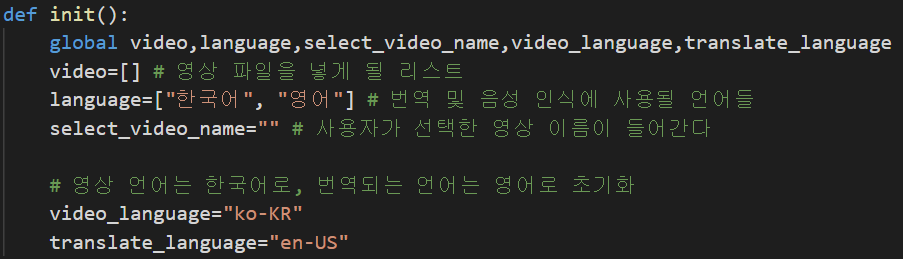


위의 내용들을 앞으로 설명할 6개의 파일에서 사용될 때 다시 설명 드리도록 하겠습니다.

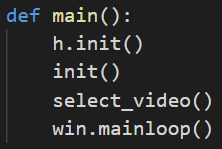
* 1. **GUI(gui.py)**

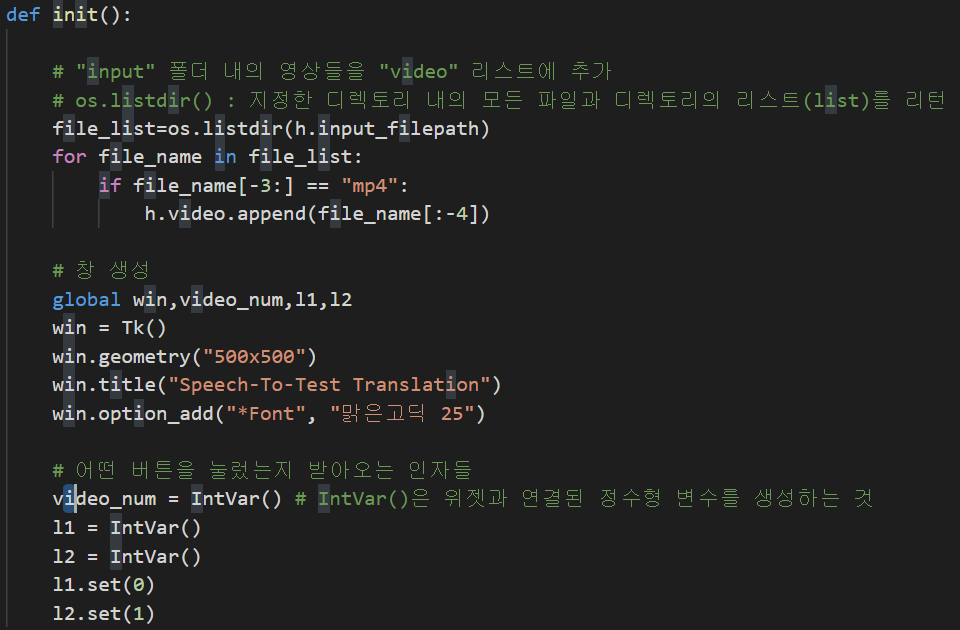
다음으로 사용자가 번역할 영상, 영상 언어와 자막(번역) 언어를 선택할 수 있도록 해주는 GUI를 구현하였습니다. Gui.py 파일을 보면 첫 줄에 import header as h 라고 쓰여 있는 것을 볼 수가 있습니다. GUI에서는 헤더 함수의 다음과 같은 변수들과 메소드가 사용이 됩니다.



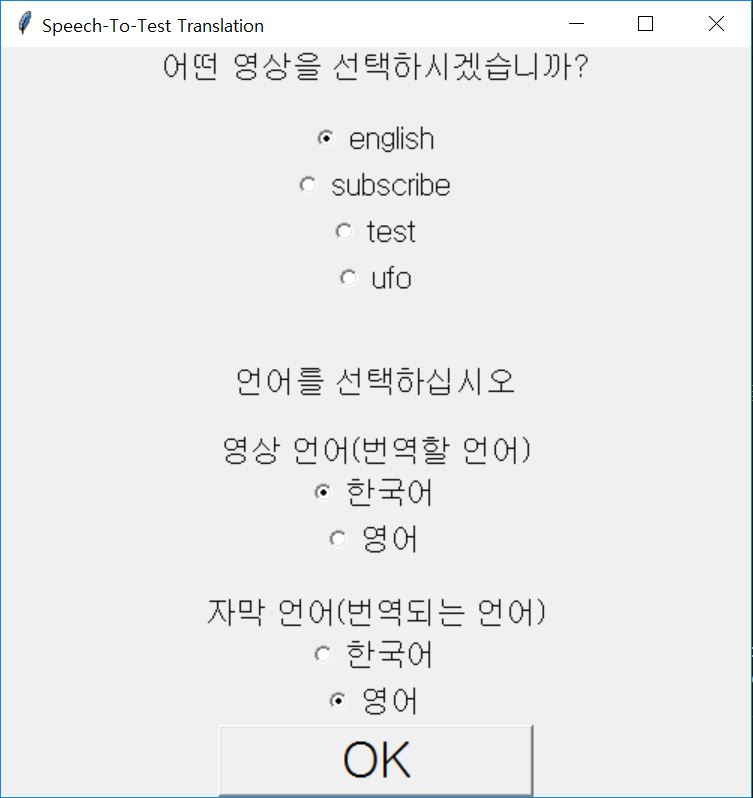


Gui.py 내의 메소드 실행 순서는 다음과 같습니다.

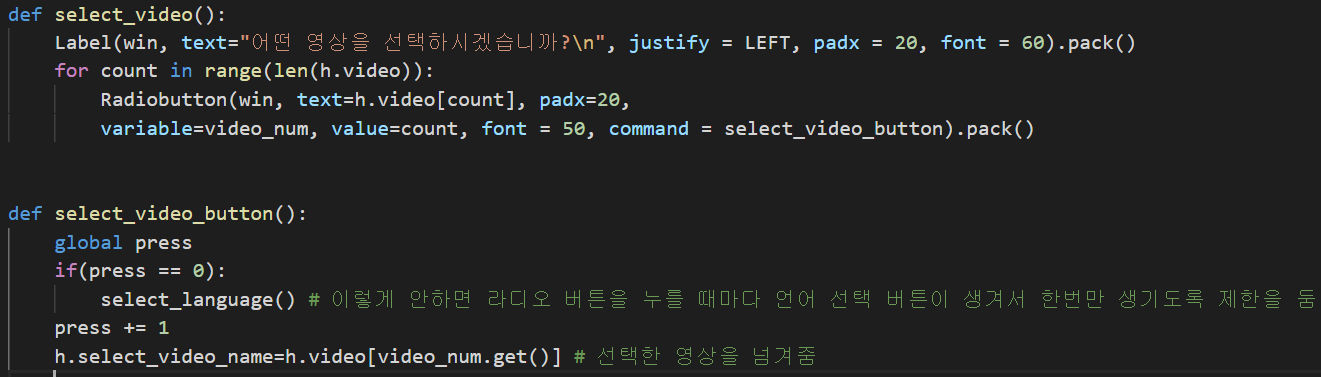




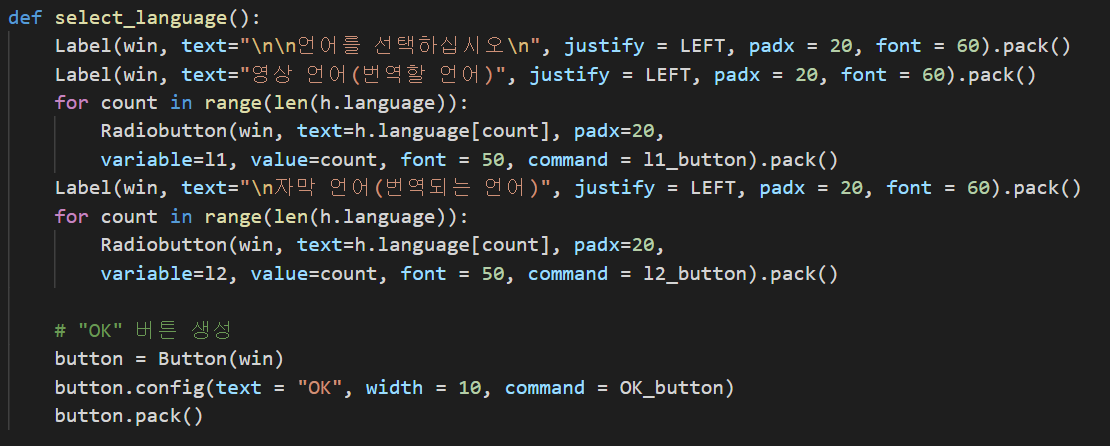
Def init() 메소드는 GUI 초기 상태에 대한 것으로, “input” 파일 내의 영상들을 video 라는 리스트에 넣고, GUI 창의 크기와 제목, 제목 글씨체 등을 다음과 같은 초기 상태로 설정해줍니다. 또한 아래 이미지에서도 볼 수 있듯이, 영상 언어와 자막 언어를 선택하는 문항에서 영상 언어는 첫 번째(한국어), 자막 언어는 두 번째(영어)로 설정합니다.



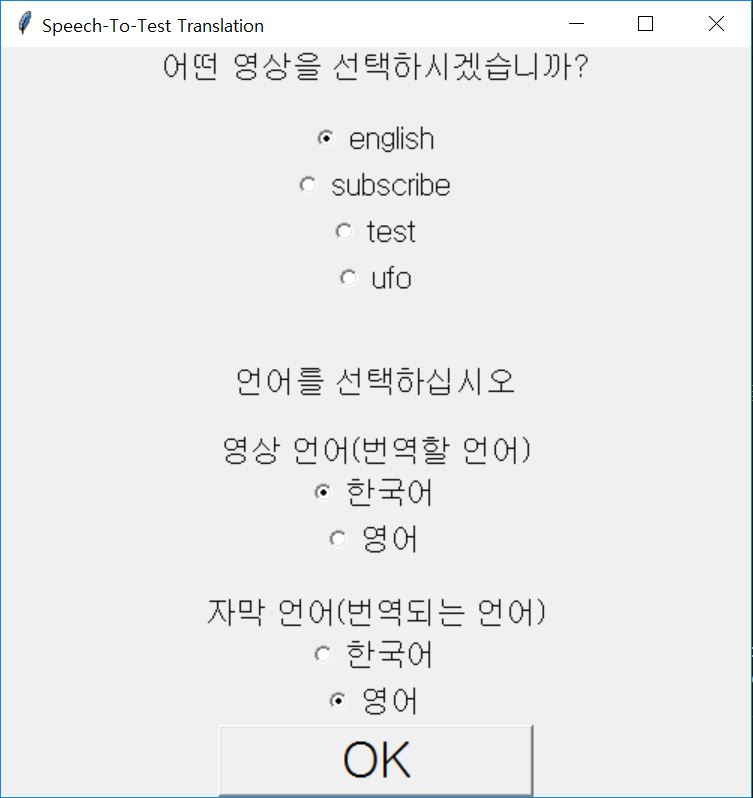
다음 메소드인 select\_video()에 대하여 설명 드리겠습니다.

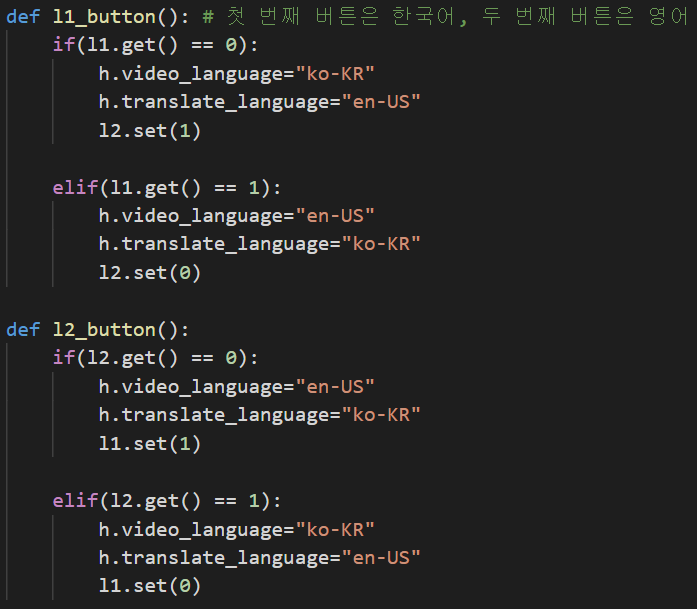


목록에서 원하는 영상을 하나 선택을 합니다. 영상 선택 버튼은 radio button으로 나타나며, 버튼을 누르면 select\_video\_button 메소드가 실행이 되도록 하였습니다. 여러 버튼 중 하나를 눌러 select\_video\_button 메소드가 실행이 되면 이어서 select\_language 메소드가 저절로 실행이 되도록 하였습니다. 또한 select\_language 메소드가 모두 실행이 된 후 선택한 영상을 select\_video\_name으로 넘겨주도록 하였습니다.



select\_language 메소드는 영상을 선택한 뒤 언어 선택을 하도록 선택지를 제공해주는 메소드입니다. 언어 선택 버튼 또한 radio button으로 나타나며, 영상 언어 선택 버튼, 번역 언어 선택 버튼을 누르면 l1\_button, l2button 메소드가 각각 실행이 되도록 하였습니다. 영상 언어, 자막 언어 선택하는 버튼이 나온 뒤, 아래 사진에서도 볼 수 있듯이 OK 버튼이 나올 수 있도록 하였습니다. OK 버튼을 누르면 OK\_button 메소드가 이어서 저절로 실행이 됩니다.

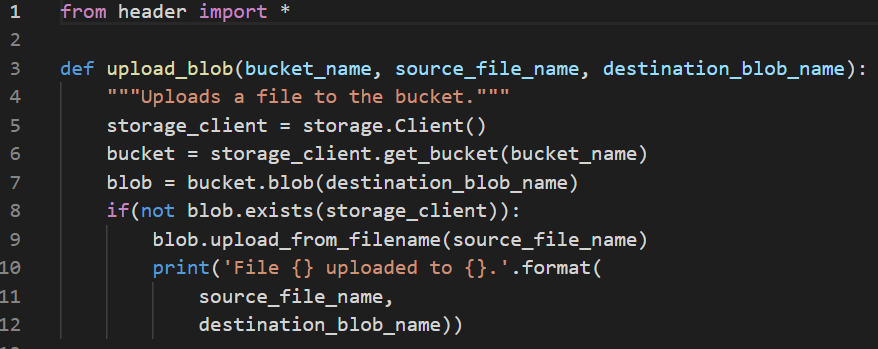




위 사진은 언어 선택 버튼을 누르면 실행되는 메소드들로, l1\_button은 영상 언어 선택 버튼에 관한 것이고, l2\_button은 번역 언어에 관한 메소드입니다. 두 메소드 다 첫번째 버튼은 영상이 한국어 번역 언어가 영어임을 의미하고, 두 번째 버튼은 그 반대의 경우를 의미합니다. 영상 언어 버튼이나 번역 언어 버튼 중 하나를 누르면, 나머지 언어 버튼에 대해서는 자동으로 다른 언어로 선택되도록 하였습니다. 예를 들어, 영상 언어를 한국어로 선택할 경우, 번역 언어는 자동으로 영어로 선택이 됩니다.

* 1. **Wav 파일을 클라우드 버킷으로 전송(upload\_wav.py)**

Wav 파일을 클라우드 버킷으로 전송하는 과정 이전에 mp4 파일을 wav로 변환하는 과정을 거쳐야 하지만, mp4에서 wav로 변환하는 과정을 담은 speech\_to\_text.py에서 이 파트(upload\_wav.py)를 import하여 사용하기 때문에 앞서 이 코드에 대해 먼저 설명 드리고자 합니다.



2.6 부분에서 설명을 드리겠지만, 영상을 사용자가 선택을 하면 mp4 파일을 wav로 변환하는 과정을 거치게 됩니다. 그 다음에 클라우드 버킷에 wav 파일이 있어야 번역 작업을 할 수 있기 때문에 wav 파일이 버킷에 존재하지 않는다면 버킷 주소로 wav 파일을 전송 해주어야 합니다. 위의 코드에서는 if 문을 통해 만약 wav 파일이 클라우드 버킷에 존재하지 않는다면 버킷에 업로드하는 과정을 담은 코드입니다.

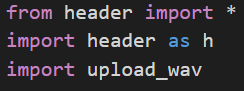
* 1. **영상을 음성 파일로 변환하여 텍스트로 반환(speech\_to\_text.py)**

GUI에서 사용자가 원하는 영상과 언어들을 설정하면 이를 wav로 변환하고, wav 음성 파일을 텍스트로 변환하여 텍스트 파일로 대본을 반환하게 됩니다. Mp4 파일을 wav로 바로 변환하지는 않고, mp4 파일에서 mp3로 변환한 뒤, mp3 파일을 wav로 다시 변환한 뒤, 이를 텍스트로 변환하는 것입니다. 즉, 과정이 “mp4 🡪 mp3 🡪 wav 🡪 wav가 bucket에 있는지 확인 🡪 텍스트로 변환”과 같이 이루어집니다.

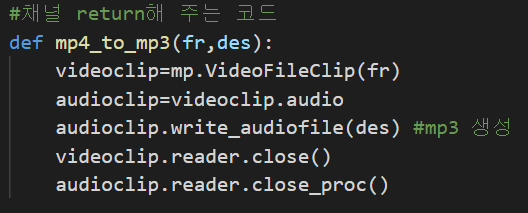
* Mp4 파일에서 wav로 바로 변환하지 않는 이유 ※

일단, mp4에서 wav로 바로 변환이 불가능합니다. 그래서 중간에 mp3 변환 과정을 거치는 것입니다. 그러면 왜 음성 인식에 mp3 파일을 사용하지 않느냐, 이에 대한 답변은 speech-to-text가 구글에서 지원하는 것인데, mp3 파일이 구글에서 지원하지 않아 mp3 파일로 텍스트로 변환하는 작업을 할 수가 없어서 다시 wav로 변환하는 과정을 거치는 것입니다.

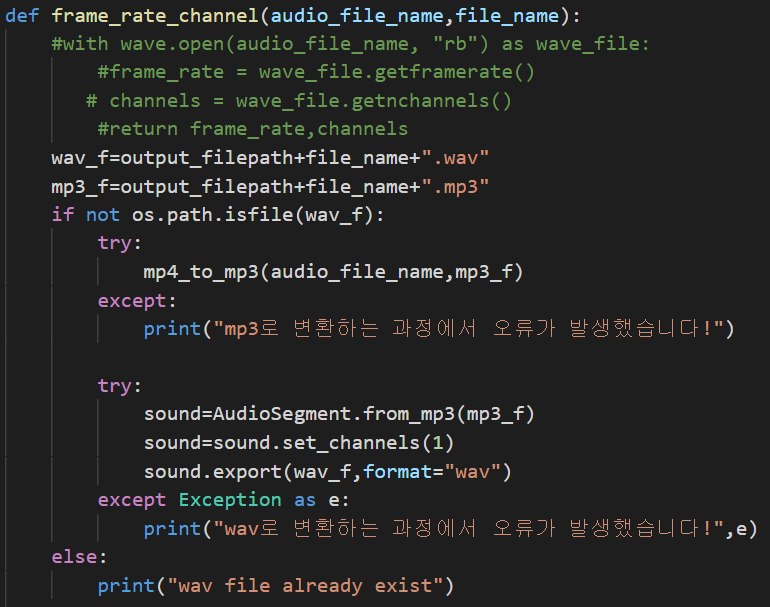
Speech\_to\_text.py 에서는 다음과 같이 헤더 파일과 2.5에서 설명 드린 upload\_wav.py를 import하여 Speech\_to\_text.py를 실행하는 데에 두 파일을 사용합니다.

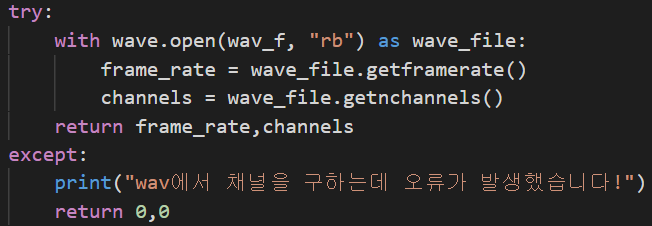


이어서 보이는 다음 코드는 mp4 파일을 mp3로 변환하여 채널을 반환하여 주는 코드입니다.

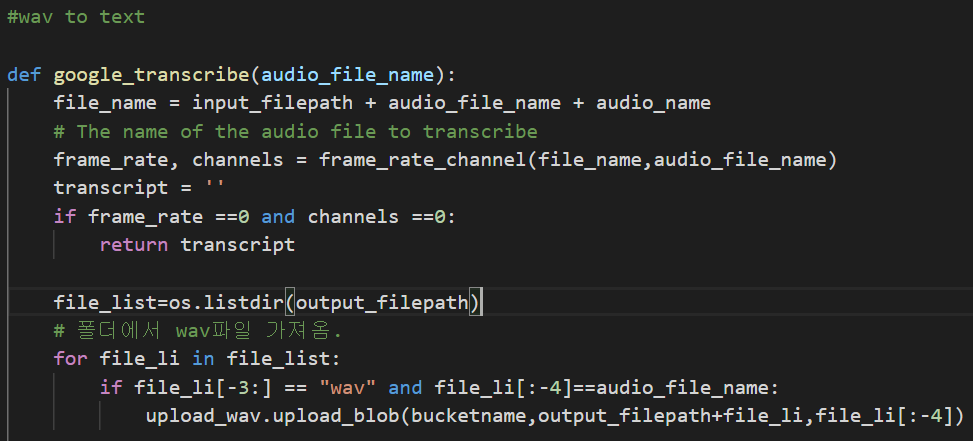


그리고 다음 코드에서 위의 메소드를 실행하여 mp3로 변환하고, 이것을 다시 wav 파일로 변환하고, 채널을 반환합니다.

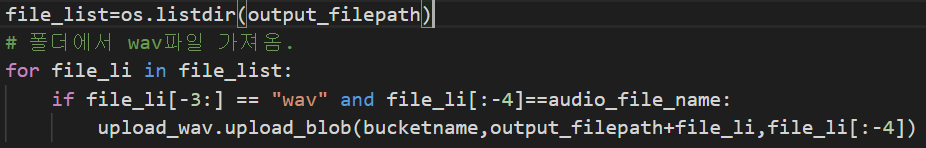




이 다음 코드를 보면 google\_transcribe라는 메소드를 볼 수 있는데, 이것은 앞서 변환한 wav 파일을 클라우드 버킷으로 업로드하고 이를 텍스트로 변환하는 과정을 담고 있습니다.

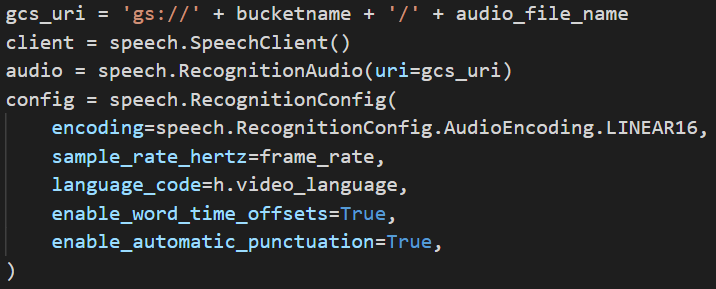


이때 위의 사진의 아래에서 5번째 줄을 보도록 하겠습니다.



위의 코드를 보면 2.5에서 설명 드렸던 upload\_wav 파일의 upload\_blob 메소드를 사용하는 것을 볼 수가 있습니다. 만약 앞서 변환한 wav 파일의 이름과 클라우드 버킷 내의 wav 파일과 이름이 같다면 기존에 버킷에 있던 wav 파일로 text 변환에 들어가면 되고, 만약 같은 이름의 wav 파일이 없다면 이 wav 파일을 버킷으로 전송하는 방식으로 구현하였습니다.

그 다음에 이어서 나오는 코드들을 보겠습니다.



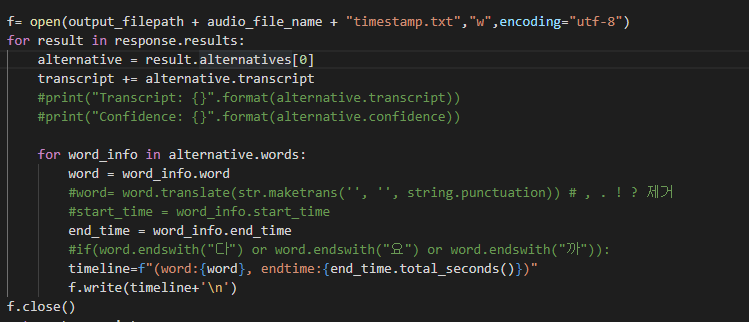
이 코드는 위에서 wav 파일을 버킷으로 전송한 뒤, api를 받아서 버킷 내의 wav 파일을 text로 변환하는 과정입니다. Wav 음성 파일을 speech-to-text를 통해 텍스트 파일로 돌려보면 말 그대로 음성이 들리는 대로만 받아 적고, 문장의 끝맺음이 이루어지지 않은 것을 볼 수가 있었습니다. 이렇게 되면 번역을 하는 데에 문제가 있어 처음에는 다음과 같이 텍스트로 받은 글에서 “다, 까, 요”로 단어가 끝나면 “.”이나 “?”를 뒤에 작성하도록 하였습니다(주로 문장의 끝이 ‘다, 까, 요’로 끝나므로).



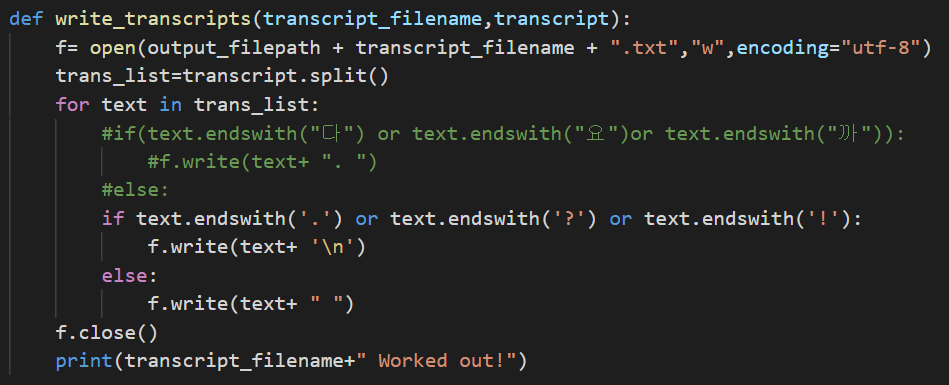
하지만 이후에 좀더 조사를 진행하다 보니 구글에서 제공하는 자동 구두점 기능인 enable\_automatic\_punctuation을 발견하였고, 보다 정확하게 하기 위해 이를 사용하여 문장의 끝맺음을 이뤄주었습니다.



그렇게 해서 얻은 텍스트 파일을 갖고 아래 사진과 같이 단어와 말하는 시간이 끝나는 지점을 timestamp에 저장해서 이후에 자막이 넘어갈 때 이곳에 저장된 timestamp를 기준으로 넘어갑니다.



위의 google\_transcribe 메소드를 통해 wav 파일을 텍스트로 변환을 마치면 아래에 write\_transcripts 메소드를 볼 수 있습니다.

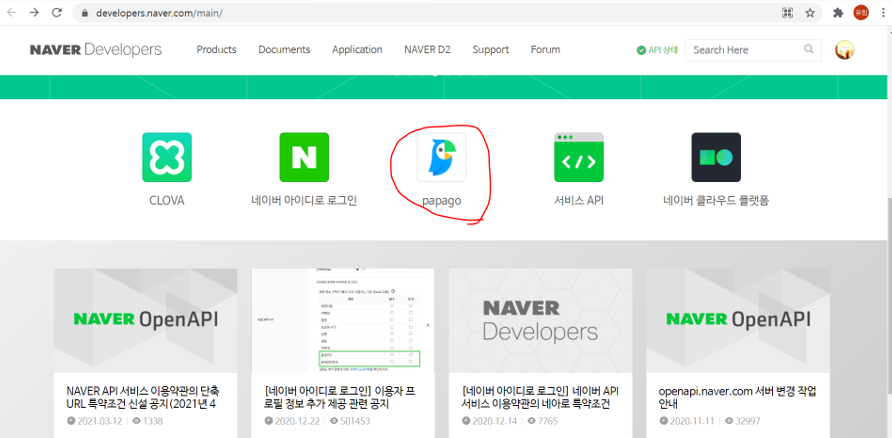


이 메소드는 구두점을 만나면 한 칸 띄우고 저장하는 방식으로 구현되었습니다. 여기서 String.endwith가 끝자리를 비교하는데 쓰이는데, 만약 문장이 “감사합니다.”로 끝났을 때 단어에 “감사합니다.”가 들어가고, 이 때 endtime 20.0이면 다음 자막이 넘어가게 되는 시점이 20.0이 되는 것입니다.

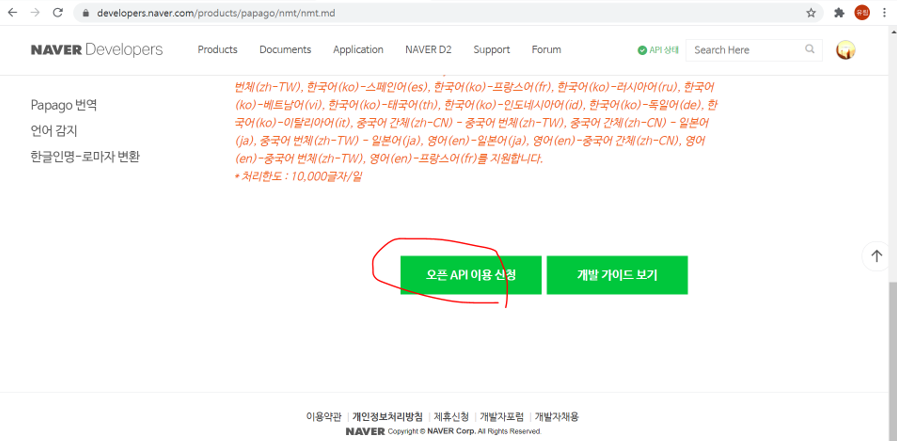
* 1. **텍스트를 네이버 Papago API를 통해 번역하기(translate\_test.py)**

앞서 음성 파일을 텍스트로 변환하는 작업을 거쳤다면, 이번에는 이 텍스트 파일을 번역하는 작업을 거치게 됩니다. 저희는 번역 작업에 네이버 파파고 api를 사용하였습니다. 네이버 파파고 api를 사용하기에 앞서 우선 명령 프롬프트 창에서 “pip install pypapago”를 해주어야 합니다.

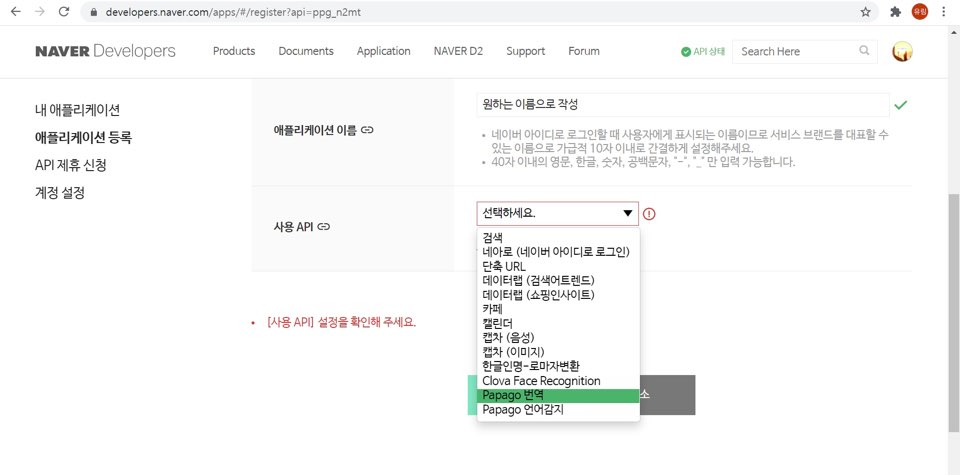
그 이후 네이버 개발자 센터(<https://developers.naver.com/main/>) 에 접속하여 papago를 클릭합니다.



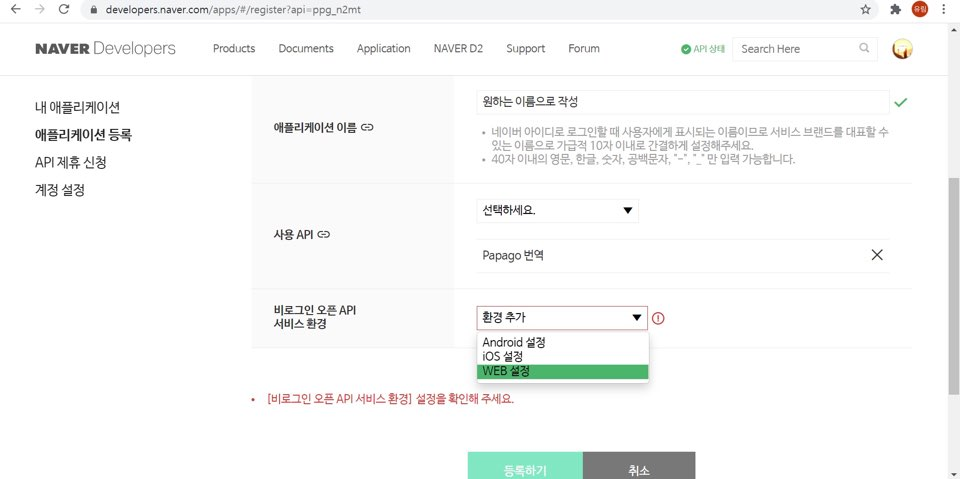
클릭한 뒤 스크롤을 내려 오픈 api 이용 신청을 클릭합니다.



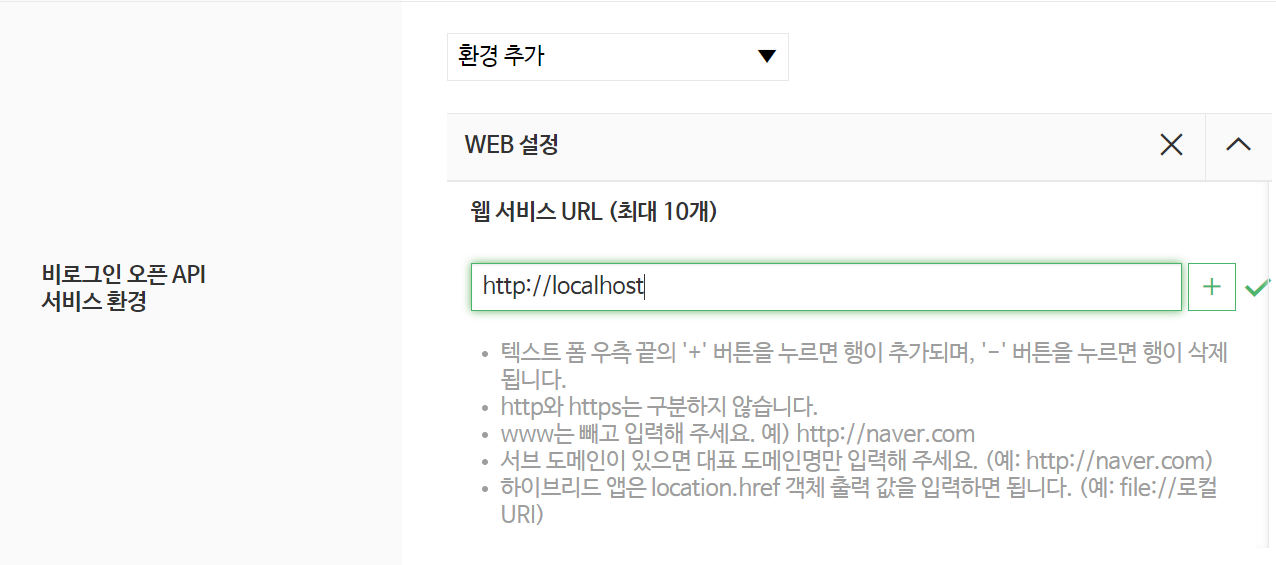
그렇게 하면 아래와 같이 애플리케이션 이름과 사용 API를 적도록 되어 있는데, 애플리케이션 이름은 원하는 것으로 작성하고, 사용 API는 Papago 번역으로 설정합니다.



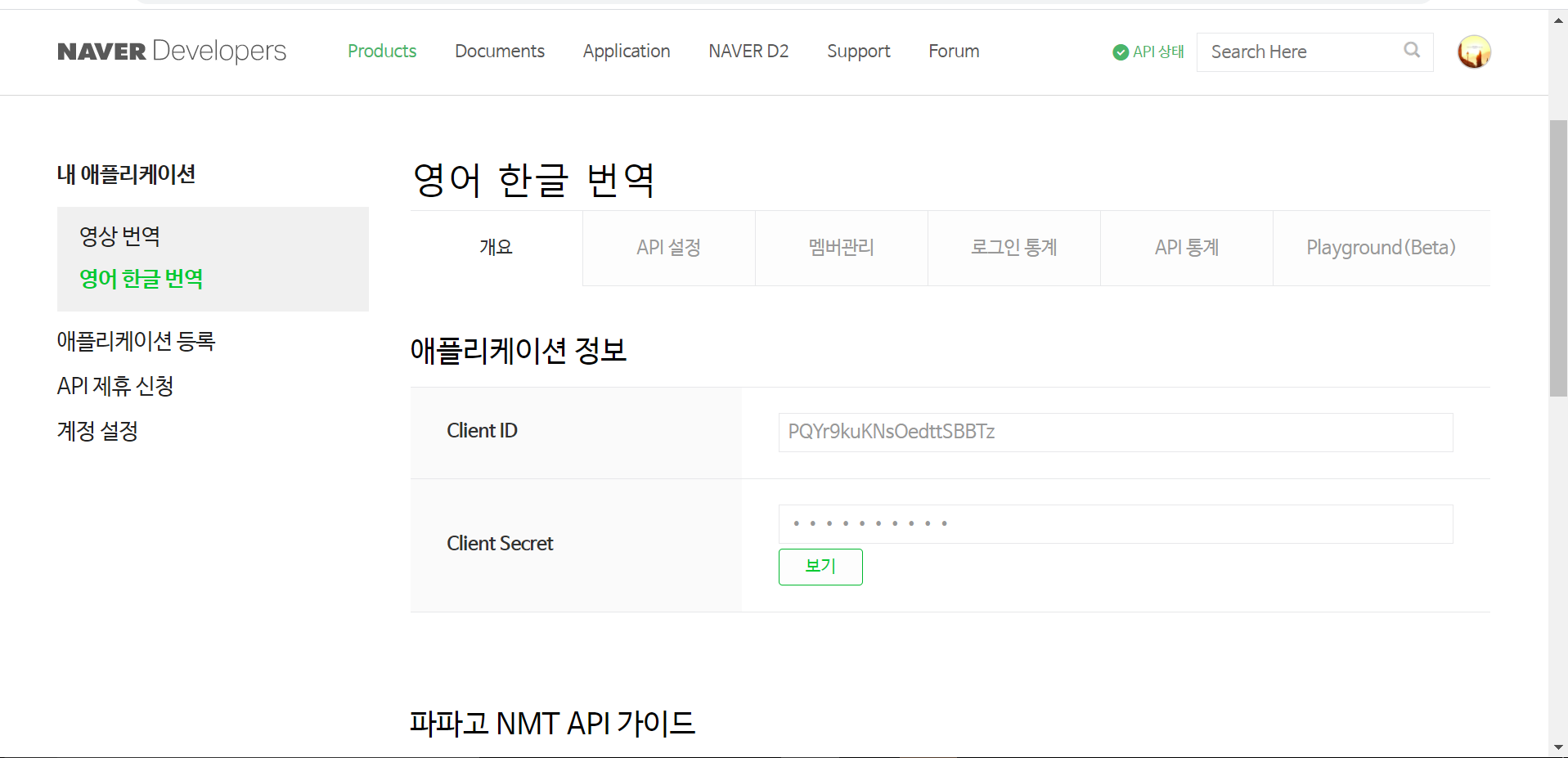
비로그인 오픈 API 서비스 환경은 WEB 설정으로 합니다.



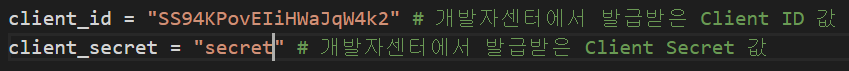
그러면 아래에 웹 서비스 URL 입력란이 나오는데, 우리는 실제 서비스를 만드는 것이 아닌, 테스트용으로 만드는 것이기 때문에 http://localhost 로 입력해주었습니다.



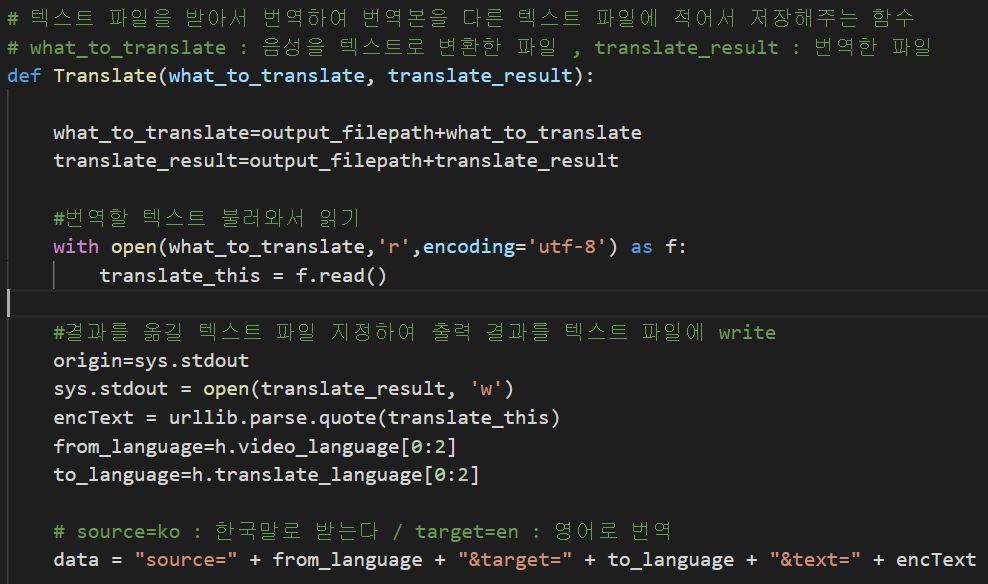
이렇게 모두 작성을 완료하고 등록하기를 누르면 Client ID와 Client Secret 을 얻을 수 있습니다.



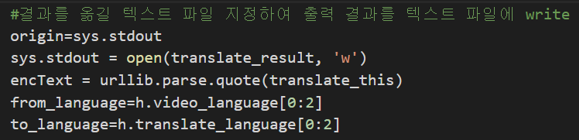
그러면 본격적으로 코드를 살펴보도록 하겠습니다. 먼저 앞서 제공 받은 client ID와 client secret code를 다음과 같이 작성해줍니다.



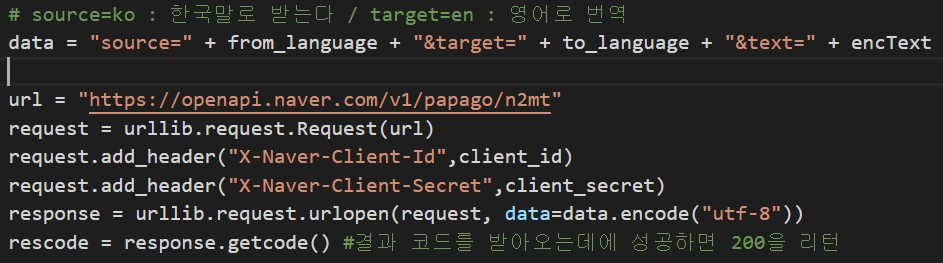
이어서 나오는 다음 메소드는 바로 텍스트 파일을 받아서 한글 파일을 받았다면 영어로, 영어 파일을 받았다면 한글로 번역해주는 메소드입니다.



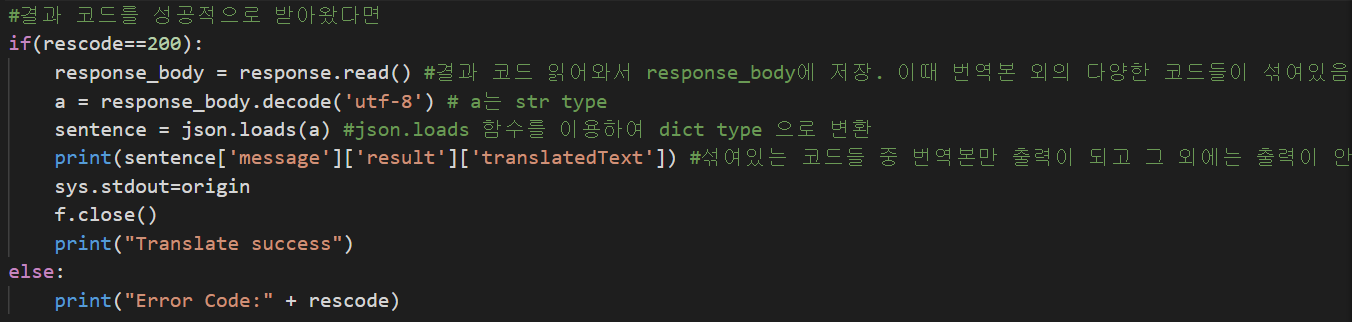
What\_to\_translate 인수는 번역하고자 하는 텍스트 파일을 전달 받습니다. 이 변수는 이 메소드 내에서 output\_filepath와 결합해서 텍스트 파일의 경로로 수정이 됩니다. Translate\_result는 번역 결과를 적을 빈 텍스트 파일을 전달해줍니다. 마찬가지로 메소드 내에서 output\_filepath와 결합해서 번역본 파일의 경로로 수정이 됩니다. 지금은 빈 텍스트 파일이지만 코드를 실행하면서 번역이 되면 그 번역 결과가 빈 텍스트 파일에 작성됩니다. Open 함수를 통해서 텍스트 파일을 읽습니다.



위의 코드는 번역 결과를 작성할 번역본을 지정하여 출력 결과를 텍스트 파일에 작성하도록 지정해주는 과정입니다. 처음 텍스트 파일 언어를 from\_language, 번역할 언어를 to\_language에 넣습니다.



이렇게 처음 언어와 번역할 언어를 data에 전달해줍니다. Data 변수와 파파고 api를 통해 번역 과정을 거쳐서 나온 결과 코드를 response 변수에 넣고, 결과 코드를 받는 데에 성공하면 rescode는 200을 전달받게 됩니다.



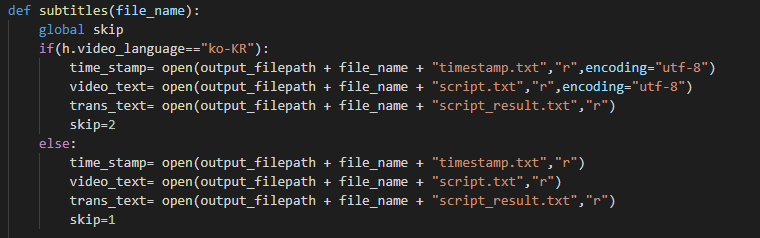
결과 코드를 성공적으로 받아왔다면, 결과 코드를 읽어 와서 response\_body에 할당합니다. 이 때 결과 코드를 보면 번역본 외의 여러 불순물들이 dict 형태로 섞여 있는데 형태는 dict 형태이지만 str 타입이라 이 결과 코드를 json.load를 통해 dict type으로 변환하는 과정을 거쳐야 합니다. 이렇게 dict 타입으로 변환하여 딱 원하는 번역본만 추출해내서 번역본에 작성을 합니다.

* 1. **위의 작업들을 영상과 자막으로 함께 반환(Video\_out.py)**

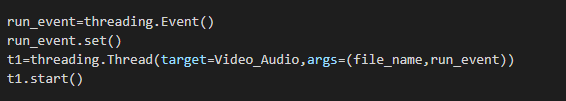
번역 과정까지 모두 마치면 음성 인식하여 텍스트로 변환하고, 이를 번역한 과정들을 토대로 영상과 함께 자막으로 보여지게 됩니다.



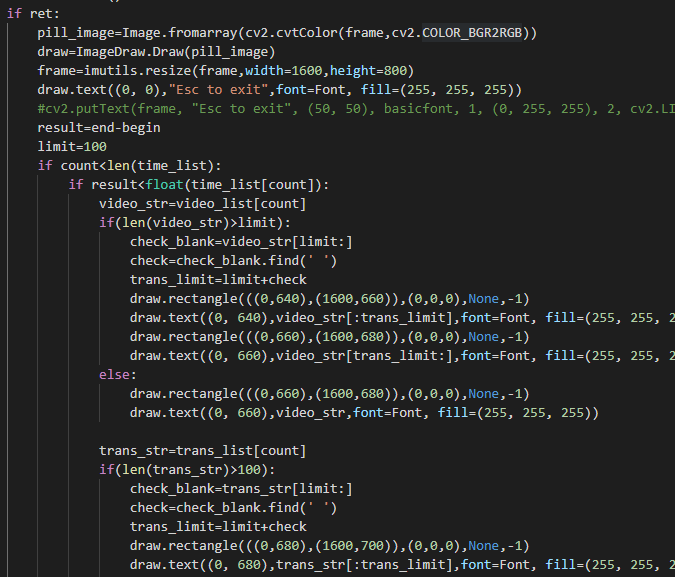
위의 코드는 Thread로 만들어서 음성 데이터를 출력하는 과정을 보여주는 코드입니다.



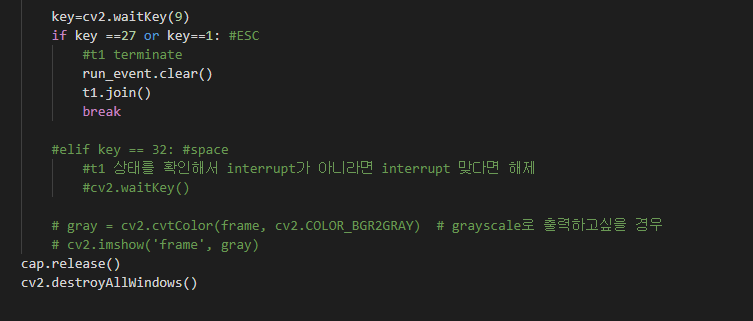
위의 코드는 언어 별로 설정이 다르게 들어가는 과정을 보여줍니다. 영상 언어가 한국어일 경우 encoding = ‘utf-8’이 추가로 들어가게 됩니다.



위의 코드는 txt에 저장되어 있는 번역을 저장하고, 언제 자막이 넘어갈지 설정하는 과정입니다. 이 과정에서 음성데이터 쓰레드화가 이루어집니다.



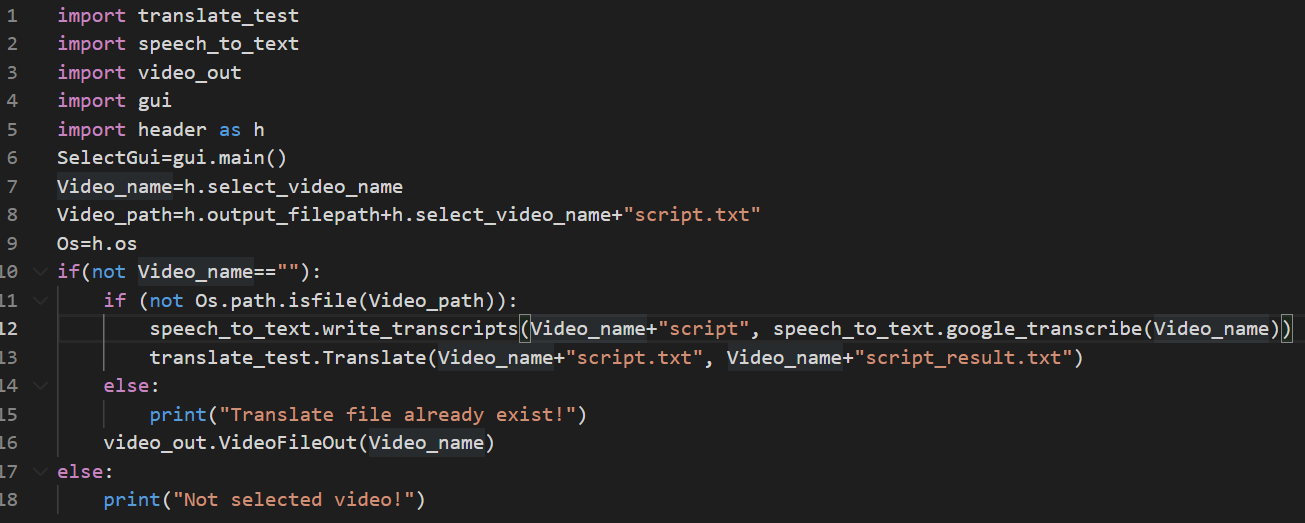
위의 코드는 limit을 설정하여 자막이 많이 길 경우 두 줄로 나올 수 있도록 해주는 코드입니다.



Esc를 누르면 시청하던 영상을 나갈 수 있습니다.

* 1. Main.py

마지막으로 Main.py는 지금까지 나왔던 파일들을 통틀어서 실행을 하는 파일로, 일반적인 프로그램에서의 main 함수와 유사합니다.

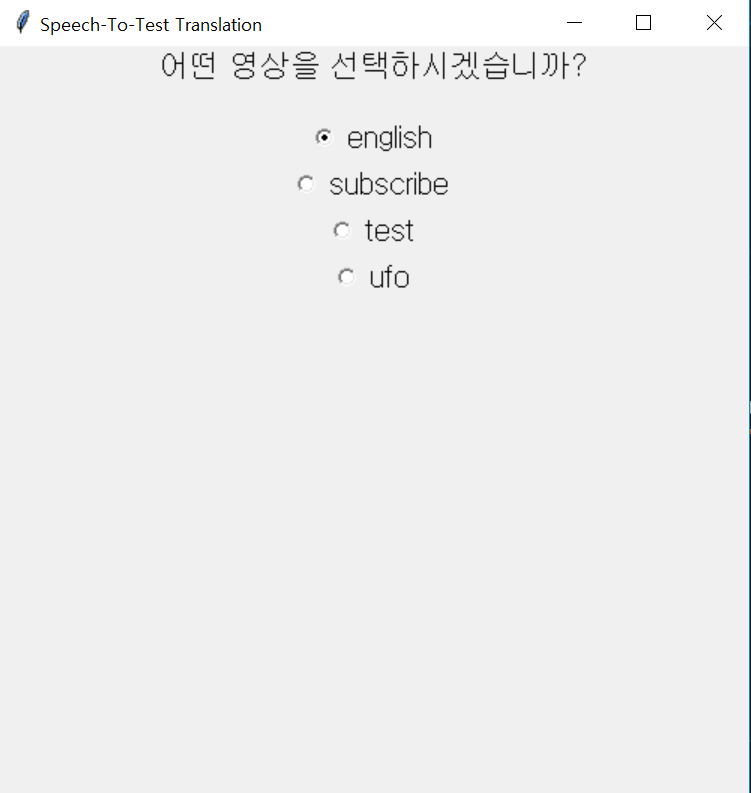


앞서 설명 드린 순서대로 gui를 먼저 생성하고, 사용자가 영상과 언어를 선택했을 시에 speech\_to\_test와 translate\_test를 통해 음성을 테스트로 변환하고 번역하는 과정들을 작성한 코드입니다.

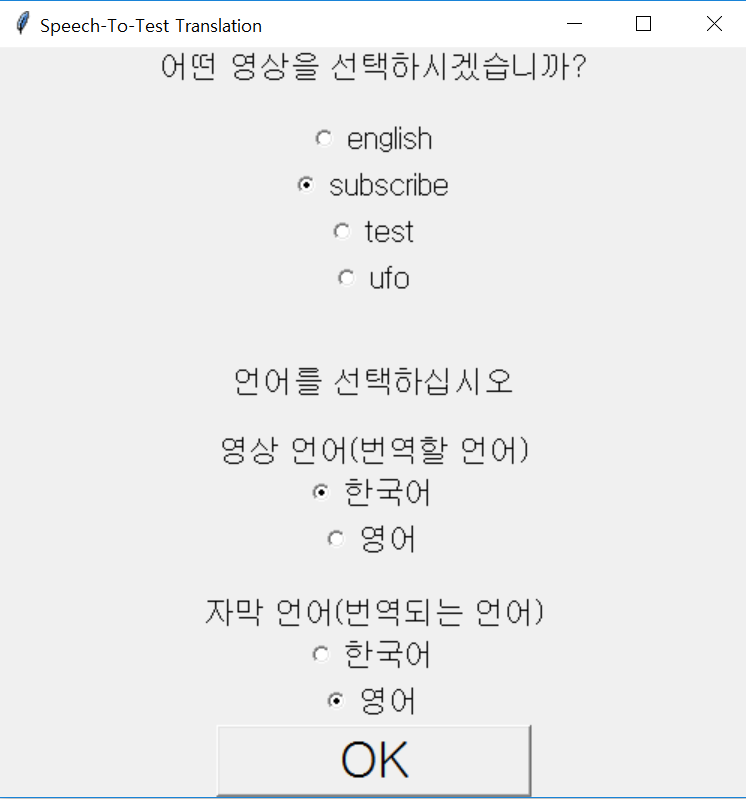
1. 결론
   1. 테스트 결과

위의 Main.py를 실행하면 다음과 같은 결과를 볼 수가 있습니다.

우선 영상을 선택합니다.

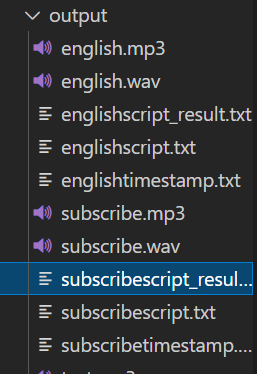


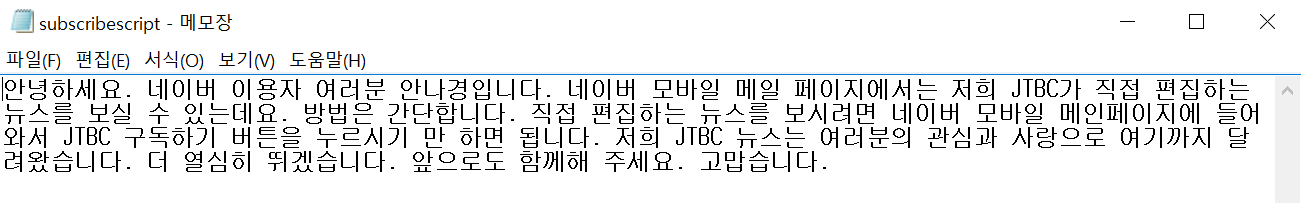
영상 선택 버튼을 누르면 언어를 선택할 수가 있습니다.



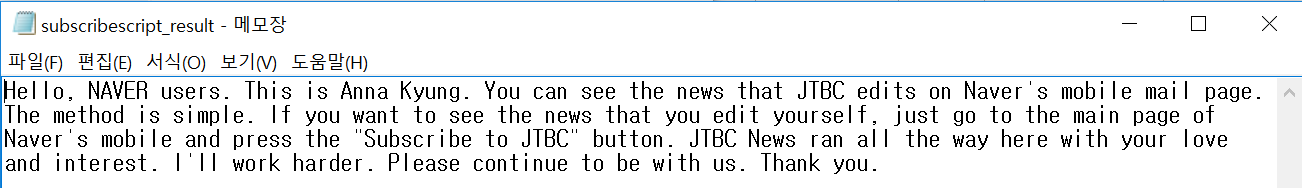
OK 버튼을 누르면 영상과 함께 자막이 나옵니다. 영상을 내보내느라 gui 사이즈가 커져서 자막이 보시기 힘드실 수 있지만, 화면을 움직이다 보면 자막이 나오는 것을 보실 수 있습니다.

또한 output 파일을 보면 다음과 같이 음성 파일과 timestamp, 음성을 텍스트로 변환한 것과 번역본이 저장되어 있는 것을 보실 수 있습니다.

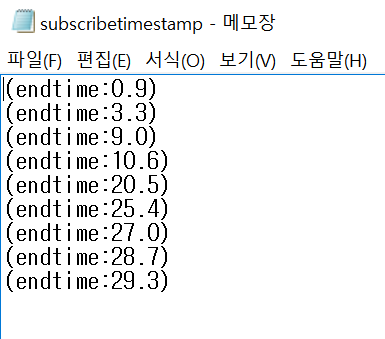




(음성 🡪 텍스트)



(번역본)



(timestamp)

* 1. 팀원 역할 분담

김세훈: 음성을 텍스트로 변환하는 과정, 코드 수정 및 분담한 코드들을 통합

(speech\_to\_text.py / header.py / main.py / upload\_wav.py / video\_out.py)

최유림: GUI 생성, 번역본 작성 과정, 보고서 작성

(gui.py / translate\_test.py)

* 1. 참고 사이트

① gui.py

창 만들기

https://www.youtube.com/watch?v=HbptDiBNPR4&t=845s

버튼 만들기

https://www.youtube.com/watch?v=Eja3D6uran8&t=709s

radio button 만들기

https://www.youtube.com/watch?v=Uhb4A3nK8lU&t=238s

② speech\_to\_text.py

mp4 -> mp3

<https://stackoverflow.com/questions/55032551/moviepy-add-audio-to-a-video>

mp3-> wav(Speech recognition 라이브러리에서 지원하는 오디오 파일 포맷이 wav 라서. mp3는 지원 안 함)

<https://seulcode.tistory.com/182>

TTS예제

<https://stackoverflow.com/questions/61092226/how-to-convert-long-speech-to-text-google-asyncronous-speech-recognition-in-pyt>

자동 구두점

<https://cloud.google.com/speech-to-text/docs/automatic-punctuation?hl=ko>

타임스탬프

<https://cloud.google.com/speech-to-text/docs/async-time-offsets?hl=ko>

Google Cloud bucket 전송

https://www.thecodebuzz.com/python-upload-files-download-files-google-cloud-storage/

③ translate\_test.py

파파고 구현 예제

https://developers.naver.com/docs/papago/papago-nmt-example-code.md#python

json.loads() 함수를 사용하여 번역본만 출력하도록 하는 과정

https://www.daleseo.com/python-json/

파이썬 결과를 txt 파일에 저장하는 방법

https://opentutorials.org/module/2980/17644

파파고 텍스트 파일 읽어오는 예시

https://wikidocs.net/35983

④ VSCode 및 python 다운로드

VSCode 다운로드

<https://blog.naver.com/arumizz/221399938287>

Python 다운로드

<https://wikidocs.net/8>

---------------------------------------

오디오 재생

https://ungodly-hour.tistory.com/35

Opencv예제(동영상재생)

http://www.gisdeveloper.co.kr/?p=6317

https://076923.github.io/posts/Python-opencv-4/

https://stackoverflow.com/questions/30509573/writing-an-mp4-video-using-python-opencv

PIL예제 (글씨쓰기)

https://wikidocs.net/73764