



실습 2

파이썬을 이용한 음성인식 구현



파이썬과 딥러닝

- 파이썬 : 최근 몇년간 딥러닝을 위한 도구로서 각광받고 있는 프로그래밍 언어
 - 다른 언어에 비해 코드가 간결하고 읽기 쉬움
 - 간단한 구문 덕분에 빠른 개발
 - 풍부한 머신러닝 딥러닝 라이브러리
 - TensorFlow, Keras, PyTorch, NumPy, Scikit-Learn, SciPy, Pandas, Matplotlib 등

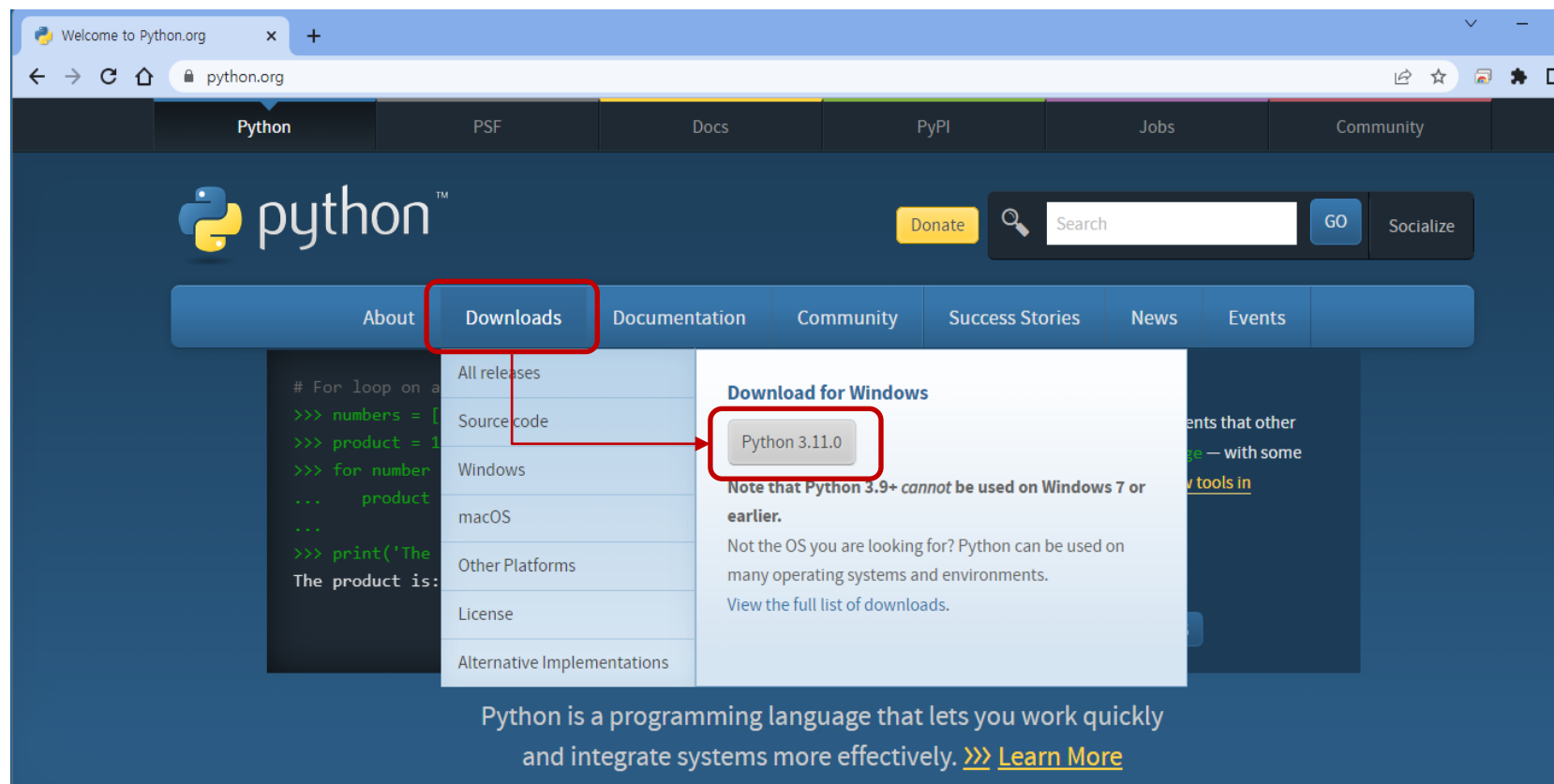


TensorFlow



파이썬 설치 (1/2)

- <http://www.python.org>
[Downloads] - [Python 3.x]



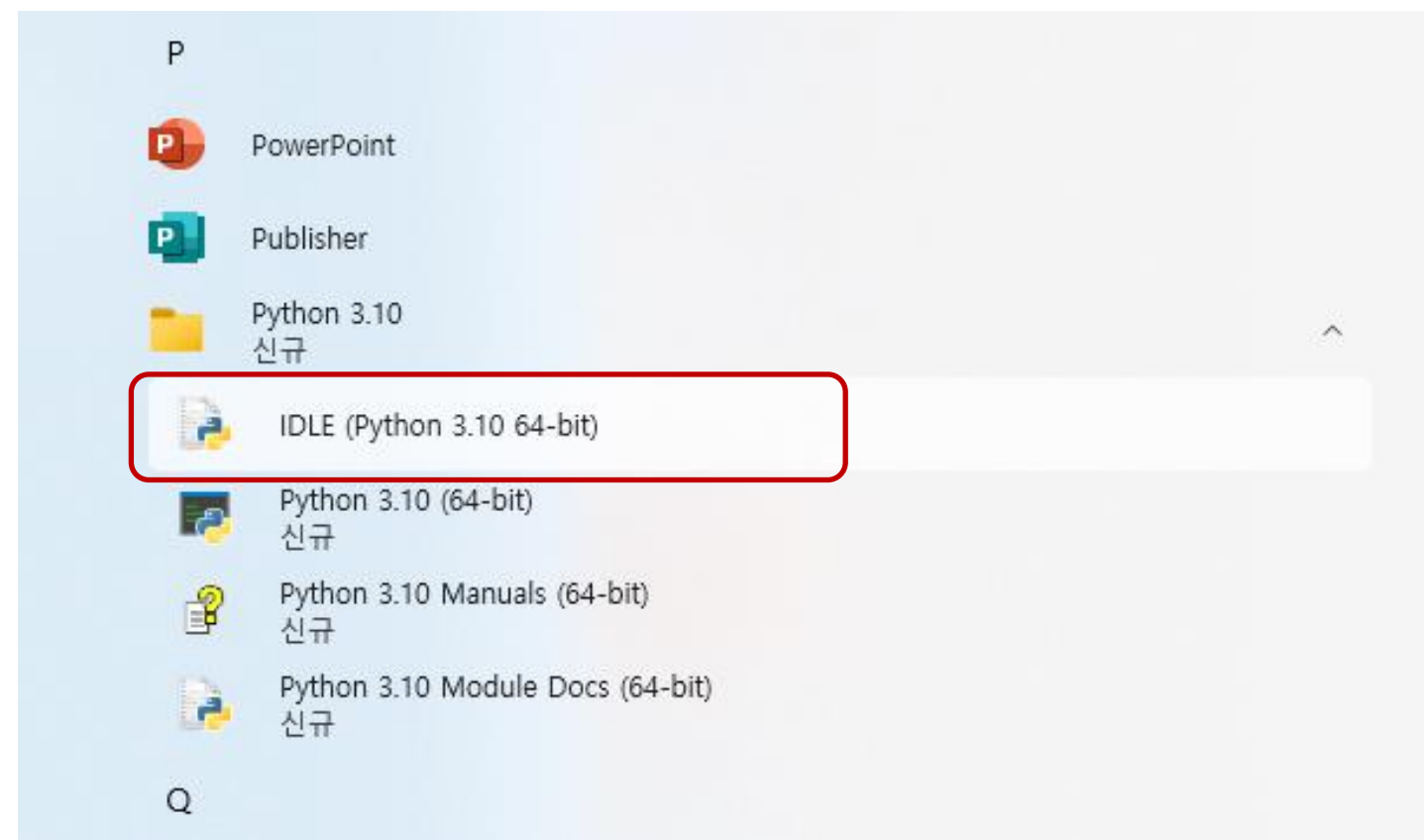
파이썬 설치 (2/2)

- 윈도우 환경변수 PATH에 파이썬 실행파일을 추가해야 함.
- 체크박스를 체크하면 자동으로 설정됨. 잊지 말고 체크할 것

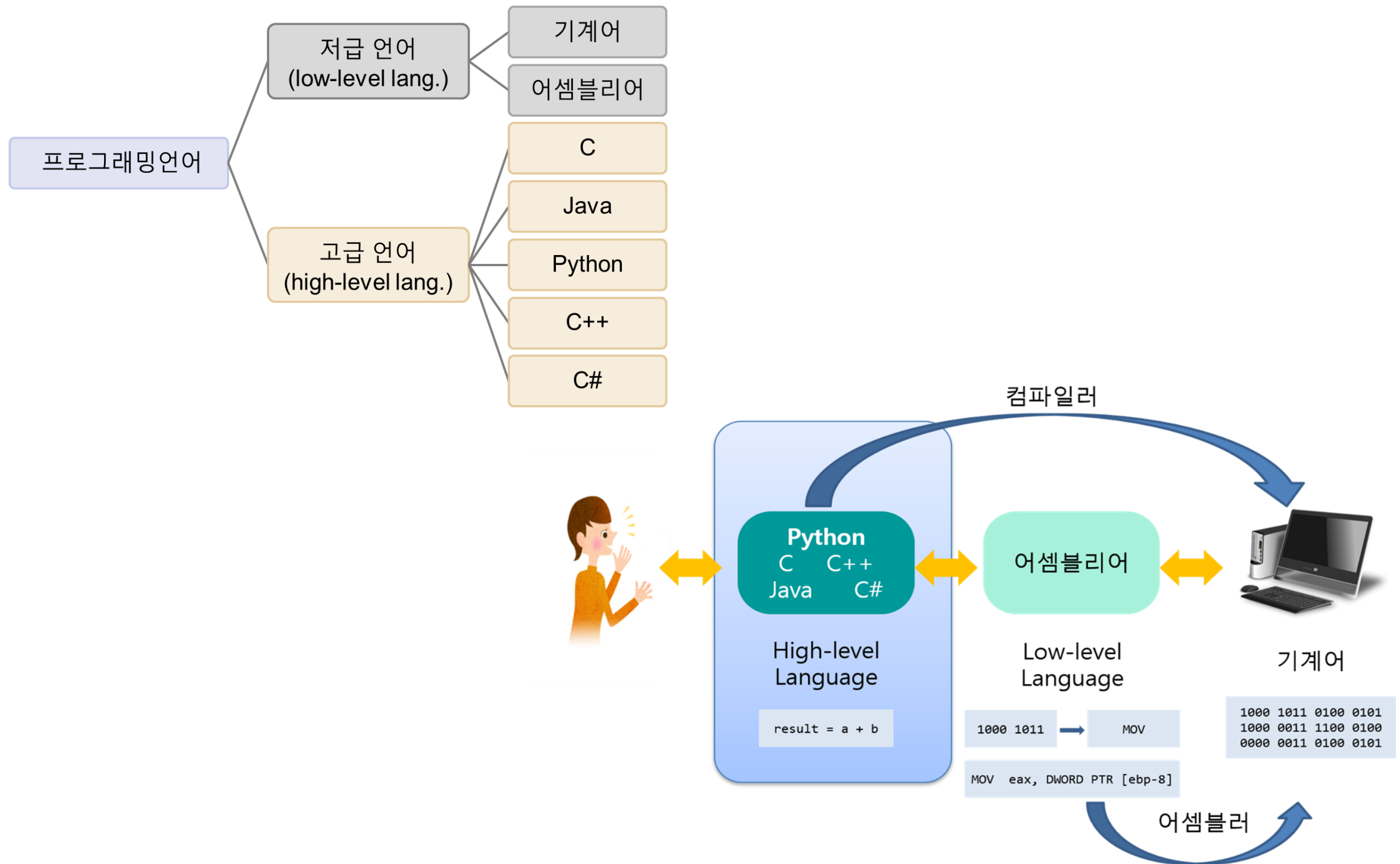


설치 확인

- [시작메뉴]-[모든 앱]를 통해 설치 확인



프로그래밍언어와 컴파일

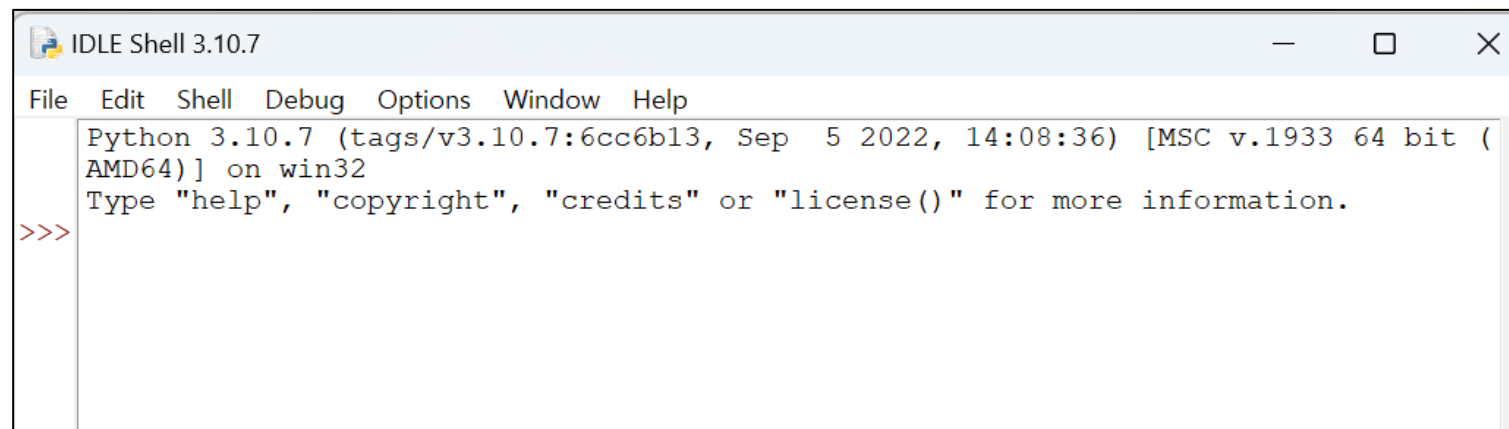


파이썬 IDLE 실행

- IDLE(Integrated Development Environment)
 - = 통합 개발환경
 - = 에디터 + 컴파일러 + 실행 + 디버거
- 프로그램 개발 과정을 돕기 위한 소프트웨어
 - PyCharm, spyder, Jupyter Notebook, VS Code 등 다양한 툴들이 있음
 - 코랩: 구글이 제공하는 클라우드 기반 Jupyter Notebook 환경
(<https://colab.research.google.com/>)

파이썬 셸(shell)

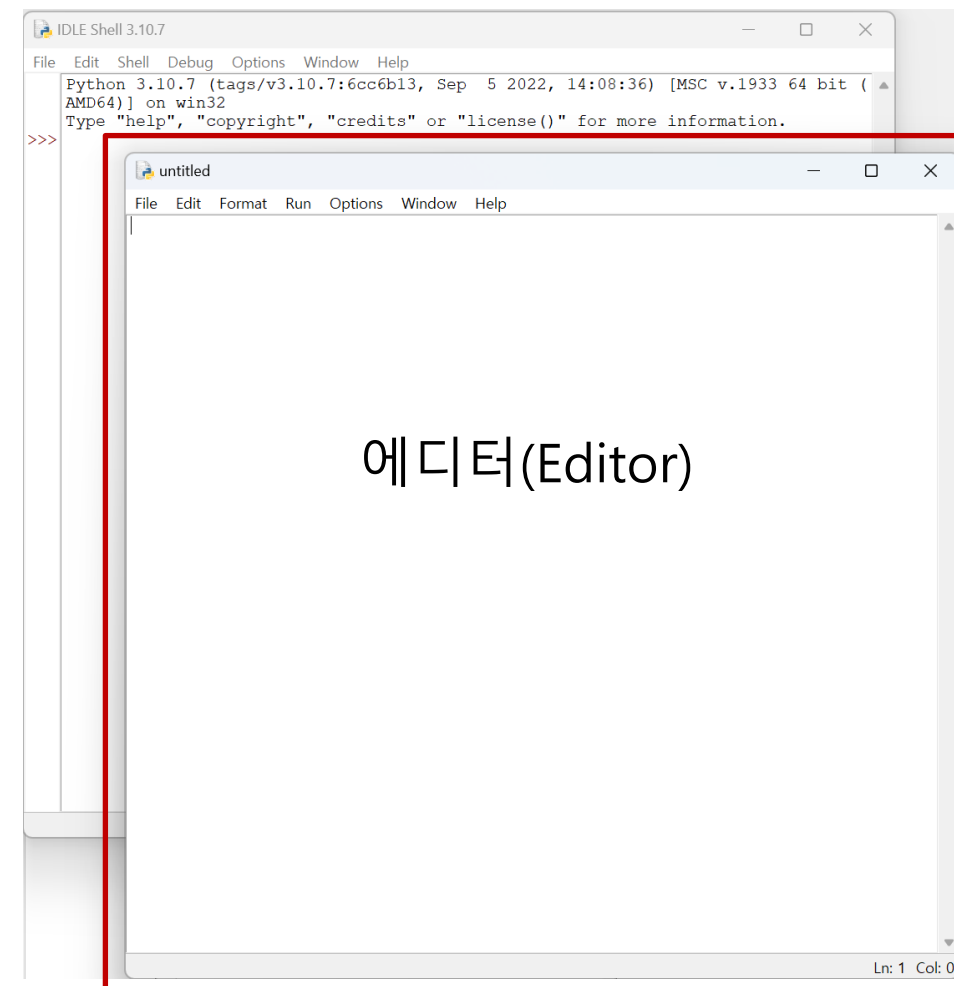
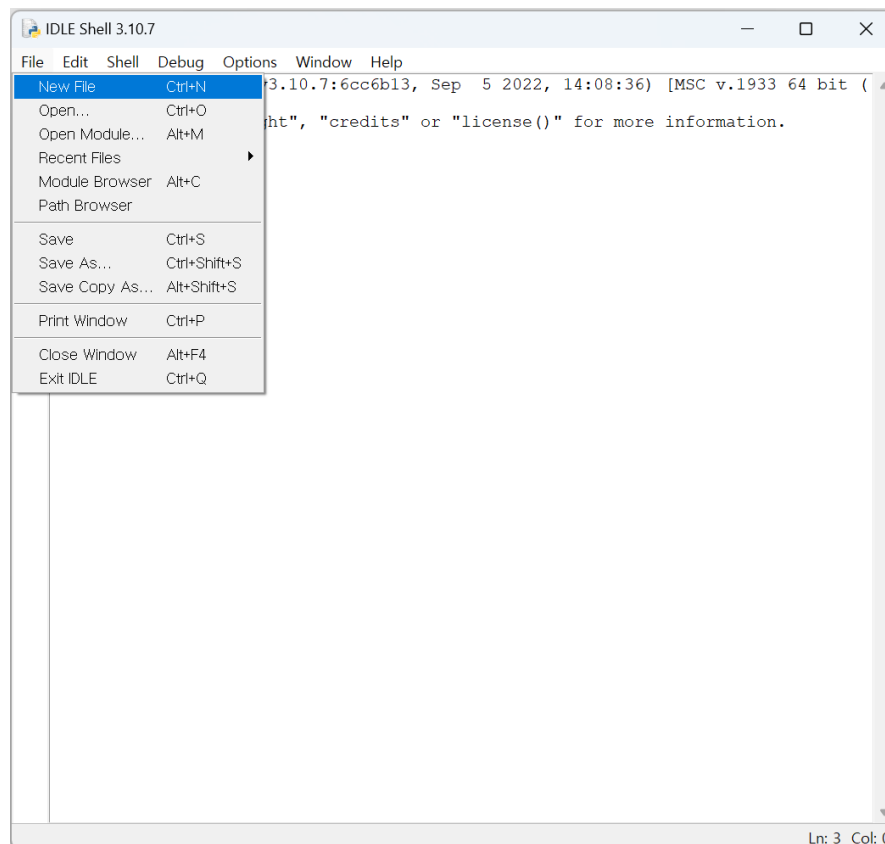
- IDLE를 실행해서 파이썬 shell 환경열기
 - Shell : 사용자가 명령어 및 텍스트를 입력해서, 컴퓨터에 명령할 수 있도록 해주는 프로그램
 - **>>>**는 프롬프트(prompt) : 파이썬 명령어를 실행할 수 있는 곳



```
IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

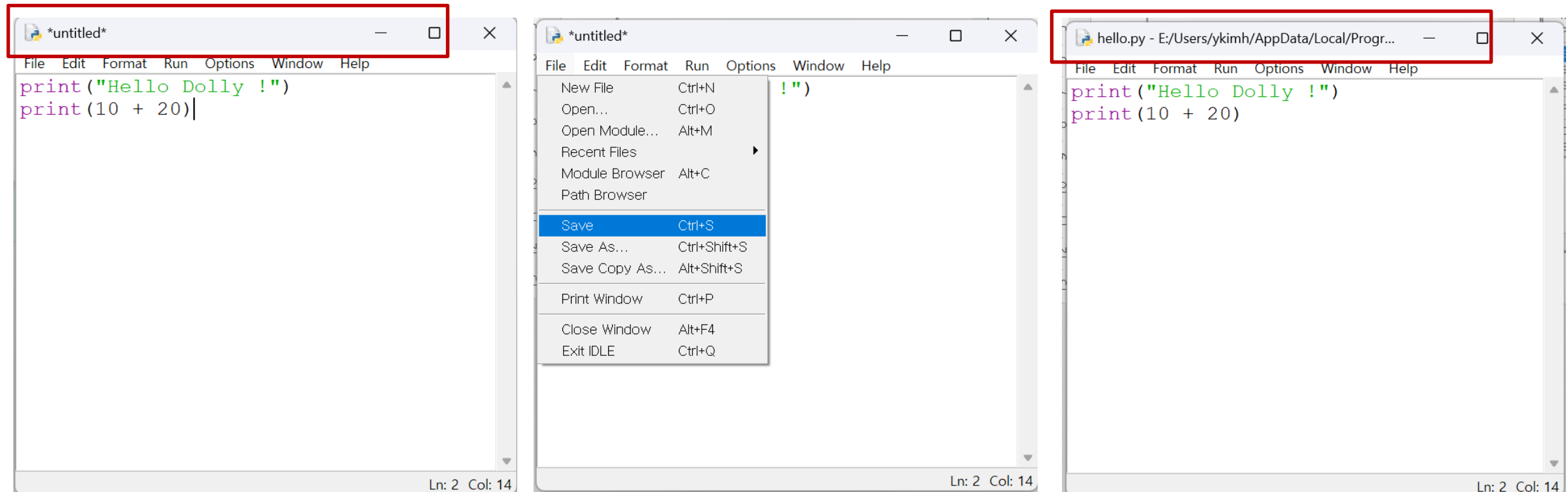

파이썬 실행하기 (1/3)

- 여러 줄의 코드를 파일로 저장해서 실행하는 방법
- IDLE 메뉴의 [File]-[New File]을 선택해서 에디터 실행



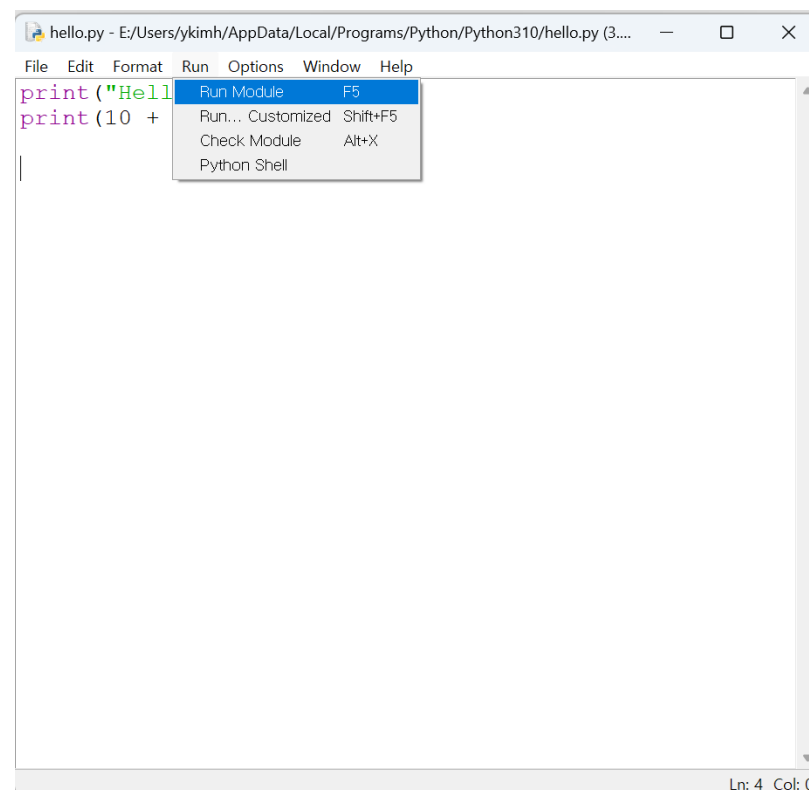
파이썬 실행하기 (2/3)

- 코드 입력 후 [File]메뉴의 [Save]로 저장
 - 파일이름을 hello라고 입력하면, 자동으로 .py 확장명이 붙음
 - 파일명 = hello.py



파이썬 실행하기 (3/3)

- [Run] 메뉴의 [Run Module] 또는 <F5>를 눌러서 실행



예제 실행 시켜보기

- 구구단 출력하기

```
# 구구단 한단 출력
```

```
i, dan = 0, 0
```

```
dan = int(input("몇단 ?"))
```

```
for i in range(1, 10) :
```

```
    print(dan, "X", i, "=", dan*i)
```

➤ 파이썬은 들여쓰기가 매우 중요함

라이브러리 (1/2)

- 프로그래밍 언어들이 제공하는 표준 라이브러리
 - 라이브러리(library)는 소프트웨어를 개발할 때 프로그램이 사용하는 비휘발성 자원의 모임
 - 구성 데이터, 미리 작성된 코드, 서브루틴 (함수), 클래스, 값, 자료형 등
 - 개발의 필수적 기능을 라이브러리로 제공
 - 모든 기능을 직접 코딩하지는 못하기 때문에

라이브러리 (2/2)

- 표준 라이브러리

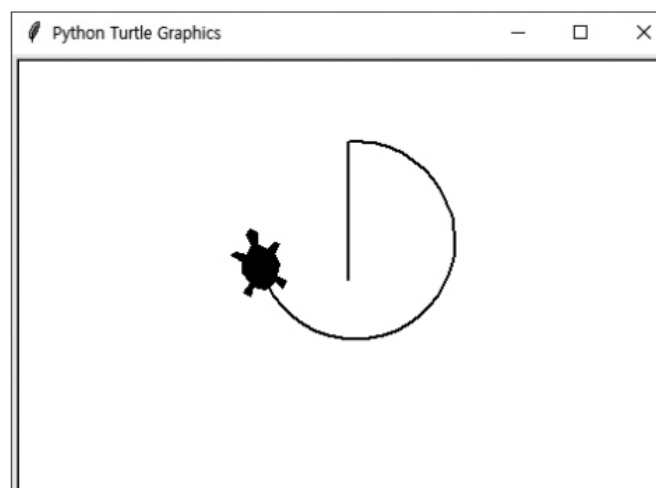
- 파이썬을 설치할 때 자동으로 컴퓨터에 설치
- `import`를 이용해서 로드
- 내장(Built-in) 함수 : `import` 없이 바로 사용
- 이 API들은 응용 프로그램이 시스템에 접근 할 수 있는 필수적인 기능을 제공

- 외부 라이브러리

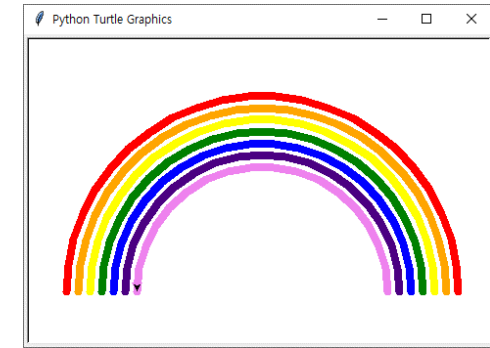
- `pip` 도구를 이용해서 설치 : 의존성 있는 모듈이나 패키지를 함께 설치해 주기 때문에 매우 편리

터틀 그래픽 (1/3)

- 화면에 쉽게 그림을 그릴 수 있도록 지원하는 라이브러리
 - 그래픽 객체(“Turtle”)를 이용하여 (x, y) 좌표로 구성된 그래픽 윈도우에서 그림 그리기
 - 다양한 선들을 이용하여 그림 그릴 수 있음
 - turtle 이라는 표준 라이브러리의 함수들을 이용



터틀 그래픽 (2/3)



```
import turtle
turtle.setup(480, 320) # 캔버스 크기 설정
my_ttl = turtle.Turtle() # 터틀 생성, turtle.getturtle()도 가능

color = ('red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet')
my_ttl.pensize(8)

radius = 200 # 무지개 반경 200에서 시작
for x in range(7):
    my_ttl.penup()
    my_ttl.setposition(radius, -100) # 오른쪽 아래 시작점으로 이동
    my_ttl.setheading(90)
    my_ttl.pendown()
    my_ttl.pencolor(color[x]) # 무지개 색상 선택
    my_ttl.circle(radius, 180) # 반원 그리기
    radius = radius - 12 # 반경을 -12씩 축소
```

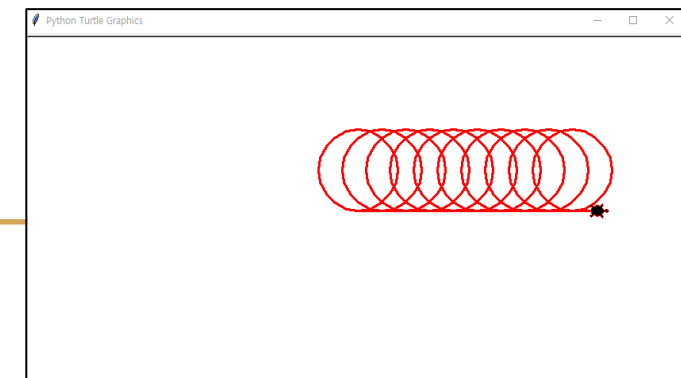

터틀 그래픽 (3/3)

```
import turtle

my_ttl = turtle.getturtle()
my_ttl.shape('turtle')

my_ttl.pensize(3)
my_ttl.pencolor('red')
my_ttl.speed(10)      # my_ttl.speed('fast')

for i in range (10) :
    my_ttl.circle(50)
    my_ttl.forward(30)
```



음성인식(STT)


- 컴퓨터 음성 인식 또는 음성-텍스트 변환(speech-to-text)이라고도 부르는 음성 인식은 프로그램이 사람의 음성을 텍스트 형식으로 처리할 수 있도록 해주는 기능

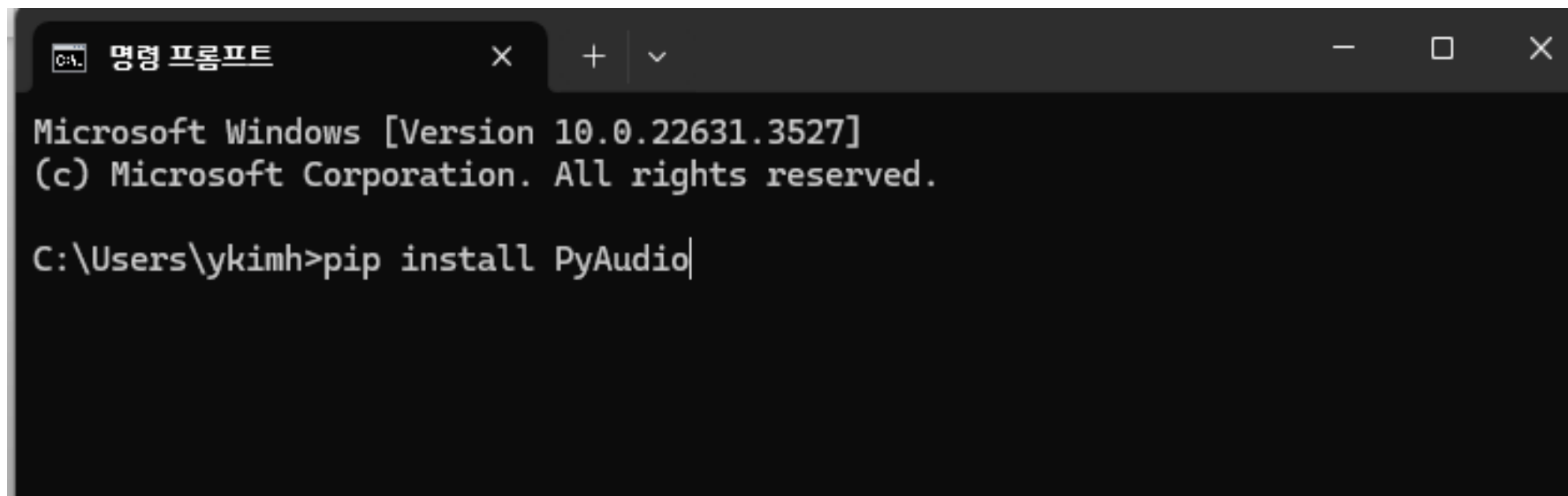


음성합성(TTS)

- 말소리의 음파를 기계가 자동으로 만들어 내는 기술
- 모델로 선정된 한 사람의 말소리를 녹음하여 일정한 음성 단위로 분할한 다음, 부호를 붙여 합성기(speech computer)에 입력해 놓고, 지시에 따라 필요한 음성 단위만을 다시 합쳐 말소리를 인위로 만들어내는 기술

파이썬 라이브러리 설치

- 윈도우 명령창  에서 cmd 입력
- 명령프롬프트 창이 뜨면 `pip install` 명령을 이용해서 설치 (예: `pip install PyAudio`)



```
명령 프롬프트
Microsoft Windows [Version 10.0.22631.3527]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ykimh>pip install PyAudio|
```

[참고] `pip upgrade` 오류가 발생하는 경우

```
C:\Users\ykimh> python.exe -m pip install --upgrade pip
```

STT : Speech to Text

- 음성인식 라이브러리 SpeechRecognition 사용
 - 한국어를 포함해 영어, 프랑스어, 중국어 등 다양한 언어의 음성 인식 관련 라이브러리

```
C:\Users\ykimh> pip install SpeechRecognition
```

- 객체화 된 음성인식기를 이용해서 wav 음성파일의 소스에서 텍스트 추출
- 기본 언어는 영어
- 한국어 사용하려면, language = "ko"
- 마이크폰을 사용하기 위해 설치 : PyAudio

```
C:\Users\ykimh> pip install PyAudio
```

```
# stt.py
import speech_recognition as sr

r = sr.Recognizer()

file_name = "audio_files_harvard.wav"
with sr.AudioFile(file_name) as source:
    audio_text = r.listen(source)

try :
    text = r.recognize_google(audio_text)
    print('\nConverting audio transcripts into text..')
    print(text)
except :
    print('try again')

file_name = "샘플_1.wav"
with sr.AudioFile(file_name) as source:
    audio_text = r.listen(source)

try :
    text = r.recognize_google(audio_text, language = "ko")
    print('\nConverting audio transcripts into text..')
    print(text)
except :
    print('try again')
```

파이썬 소리재생 방법 (1/3)

- 표준 라이브러리 winsound
 - 윈도우 플랫폼에서 제공하는 기본 소리 재생
 - wav 형식만 지원

```
import winsound

winsound.Beep(370,2000)

# wav 파일 재생
with open("audio_files_harvard.wav", 'rb') as f :
    winsound.PlaySound(f.read(), winsound.SND_MEMORY)
```

파이썬 소리재생 방법 (2/3)

```
# sound.py
from winsound import Beep
Beep(1000, 1000) # 1000Hz for 1000ms

def freq(o, s):
    if s == '도':      return 524*2**o
    elif s == '도샹':  return 554*2**o
    elif s == '레':    return 587*2**o
    elif s == '레샹':  return 622*2**o
    elif s == '미':    return 659*2**o
    elif s == '파':    return 698*2**o
    elif s == '파샹':  return 740*2**o
    elif s == '솔':    return 784*2**o
    elif s == '솔샹':  return 831*2**o
    elif s == '라':    return 880*2**o
    elif s == '라샹':  return 932*2**o
    elif s == '시':    return 988*2**o

Beep(freq(0, '도'), 500);   Beep(freq(0, '레'), 500);
Beep(freq(0, '미'), 500);   Beep(freq(0, '파'), 500);
Beep(freq(0, '솔'), 500);   Beep(freq(0, '라'), 500);
Beep(freq(0, '시'), 500);   Beep(freq(1, '도'), 500);
```


파이썬 소리재생 방법 (3/3)

- playsound 라이브러리 사용
 - mp3 파일 지원, 간단하게 사용할 수 있는 장점

```
C:\Users\ykimh> pip install playsound
```

- [설치오류] python 3.12 버전부터 distutils 모듈이 제거되어 오류 발생할 수 있음

```
pip install --upgrade setuptools
```

TTS : Text to Speech

- TTS 또는 음성합성

- 인터넷이 연결된 상태에서 합성음성을 제공하는 서비스를 이용
- 구글의 TTS API : gTTS 모듈을 설치해서 사용

```
C:\Users\ykimh> pip install gTTS
```

- 네이버의 Clova Speech Synthesis API : 네이버 개발자 사이트에서 오픈 API 이용 신청을 해야 하며, 하루에 사용할 수 있는 용량은 "10,000글자/일"

```
# tts.py
import playsound
from gtts import gTTS
import os

def speakEng(text, lang="en", speed=False):
    tts = gTTS(text=text, lang=lang , slow=speed)
    tts.save("./tts.mp3")
    playsound.playsound("./tts.mp3")
    os.remove("./tts.mp3")

print("영어 변환")

text = "Your time is limited, so don't waste it living someone
else's life. Don't be trapped by dogma, which is living with the
results of other people's thinking. "

speakEng(text)
```

```
#tts.py
import playsound
from gtts import gTTS
import os

def speakKor(text, lang="ko", speed=False):
    tts = gTTS(text=text, lang=lang , slow=speed)
    tts.save("./tts.mp3")
    playsound.playsound("./tts.mp3")
    os.remove("./tts.mp3")

print("한글 변환")

text2 = "여러분들의 삶은 제한되어 있습니다. 그러니 낭비하지 마십시오.
다른 사람들의 생각에 얽매이지 마십시오."

speakKor(text2)
```

시리만들기 (1/2)

● STT ⇔ TTS

```
#siri.py

import playsound
from gtts import gTTS
import os
import speech_recognition as sr

def speak(text ,lang="ko", speed=False):
    tts = gTTS(text=text, lang=lang , slow=speed)
    filename = "tts.mp3"
    tts.save(filename)
    playsound.playsound(filename)
    os.remove(filename)

Recognizer = sr.Recognizer()
mic = sr.Microphone()
```

시리만들기 (2/2)

```
while True:
    with mic as source:
        speak("말씀하세요")
        audio = Recognizer.listen(source)
    try:
        data = Recognizer.recognize_google(audio , language="ko")
        print(data)
    except:
        speak("이해하지 못하는 말이에요")
        data = ""
    if "시리" in data or "실이" in data :
        speak("네 부르셨어요")
        print("시리 : 네 부르셨어요")
    elif "종료" in data :
        speak("종료합니다")
        exit()
    else:
        speak("다시 불러주세요")
```