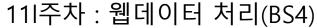


# **ChatGPT**



12주차 : 멀티미디어 처리(Image, Audio, File Upload)

13주차 : AI 웹 서비스 활용(ChatGPT)

14주차 : 미니 프로젝트(발표)

 $\int$ 

11주차 : AI 웹 서비스 활용(ChatGPT)

12주차 : AI 웹 서비스 활용(ChatGPT) Image, Audio, FileUpload

13주차 : AI 웹 서비스 활용(ChatGPT) 웹데이터 처리

14주차 : AI 웹 서비스 활용(ChatGPT) 예제 설명 / 최종 보고서 제출

ChatGPT Dalle Whisper

웹데이터분석(BS4, Youtube)

File Upload

OpenAl/ Flask

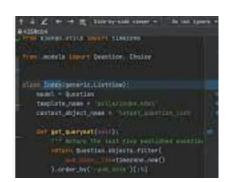
# python 개발 환경 비교

Python 3.7.1 Shell

File Edit Shell Debug Options Window Help

Python 3.7.1 (default, Dec 10 2018, 22:09:34) [MSC v.1915 32 bit (Inte 32 Type "help", "copyright", "credits" or "license()" for more informatio

- python shell
  - 인터프리터 방식으로 python 코드를 한 줄 한 줄 입력
- anaconda + jupyter notebook
  - navigator에서 실행시 계정홈을 기본 폴더로 서버 시작
  - command run 특정 폴더에서 jupyter notebook 실행
- pycharm
  - python 전용 통합 개발 환경
  - python이나 anaconda 설치
  - 전체 코드 입력 후 실행
  - step-by-step 디버깅 가능
  - syntax highlighting 제공, code completion(코드 완성기능)
  - jupyter notebook을 서버로 연동 가능
- visual studio code
  - web개발 표준 통합 개발 환경
  - pycharm과 동일
- google colab(jupyter notebook)
  - google 가상 컴퓨터에서 동작하는 jupyter notebook(linux에서 운영됨)



shell

pycharm



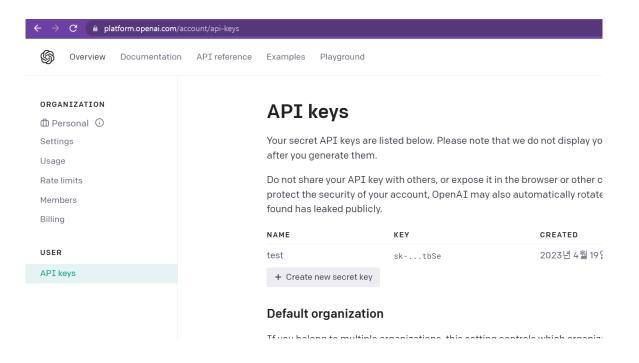
jupyter notebook

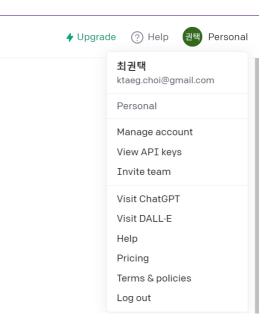
# 키발급



GPT api사이트, 계정을 만들고 로그인 우측상단 개인매뉴에서 view API키를 클릭

https://platform.openai.com/overview





### GPT-4



# GPT2=GPT3=GPT4 알고리즘은 큰 변화 없다.

GTP2<GPT3=GPT4 파라미터크기는 변화 없다.

### GPT-4, 무엇이 달라졌나

<b>GPT-3.5</b>	GPT-4			
오픈AI	개발사	오픈AI		
2022년 11월	출시시기	2023년 3월		
1750억개	파라미터	미공개		
대화 1회당 <b>최대 3000단어</b> 기억 가능	단기기억력	대화 1회당 <b>최대 2만5000단어</b> 기억 가능		
<mark>텍스트</mark> 입력만 가능한 언어 모델	특징	<mark>텍스트, 이미지</mark> 입력 가능한 멀티모달 모델		
유·무료 관계없이 <b>무제한 이용 가능</b>	챗GPT 이용 가능 여부	<mark>유료 서비스</mark> 에서만 이용 가능, <b>4시간당 메세지</b> 100건 제한		

#### The JoongAng

그래픽=박경민 기자 minn@joongang.co.kr

gpt : 2018 1700만개

gpt2 : 2019, 15억개

gpt3: 2020, 1750억개

(학습비용 50억)\_

gpt4 : 2023 : 1조?

사람: 1000억개, 1000

조개시내습

# 지식 습득의 변화

### Before 인터넷



벽화(기록의시작, 현장)



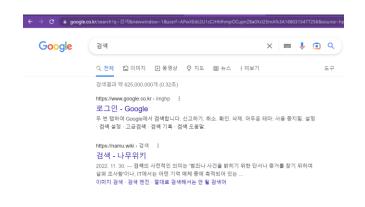
도서관(지식의집대성)



미디어(대중,실시간)

### After 인터넷







yahoo(지식의카테고리)

google(지식검색)

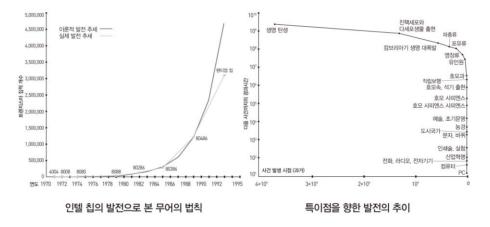
ChatGPT(지식생성)

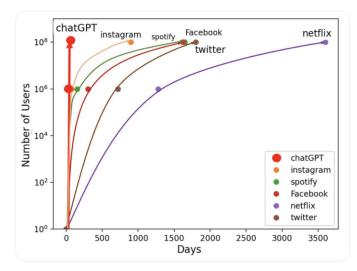
# ChatGPT 기술적 특이점 인가?

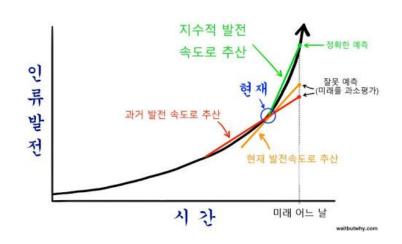
- 기술적 특이점(technological singularity):

AI 발전이 가속화되어 모든 인류의 지성을 합친 것보다 더 뛰어난 초인공지능이 출현하는 시점

- 특이점을 향한 발전 추이와 무어의 법칙이 유사함(지수함수적으로 증가함)







# **OpenAl**



### **Build an application**



#### Chat Beta

Learn how to use chat-based language models



### **Text completion**

Learn how to generate or edit text



#### **Embeddings**

Learn how to search, classify, and compare text



#### Speech to text Beta

Learn how to turn audio into text



#### Image generation Beta

Learn how to generate or edit images



### Fine-tuning

Learn how to train a model for your use case

# prompt 연습: 가상의 주소록 만들기

랜덤하게 한국사람 10명의(이름 , 이메일, 주소,전화번호) 주소록을 생해 excel 파일로 저장하는 python 코드를 작성해주세요. 이 때 faker 모듈을 사용하세요.

address.py

```
from faker import Faker
import openpyxl
fake = Faker('ko KR')
wb = openpyx1.Workbook()
ws = wb.active
ws.append(['이름', '이메일', '주소', '전화번호'])
for i in range(10):
    name = fake.name()
    email = fake.email()
    address = fake.address()
    phone_number = fake.phone number()
   ws.append([name, email, address, phone number])
wb.save('korean_address_book.xlsx')
```

pip install faker
pip install openpyxl

# prompt 연습: Quiz 생성

- 대한민국에 대한 상식 퀴즈 5개 생성(ppt)
- 4개의 보기로 작성
- 정답과 정답 설명 필요

<b>\$</b>	문제	보기 1	보기 2	보기 3	보기 4	정답	정답설명
	대한민국의 수도 는?	서울	부산	대구	인천	서울	대한민국의 수도는 서울입니 다.
	대한민국의 국화 는?	목련	백합	무궁 화	진달 래	무궁 화	대한민국의 국화는 무궁화입 니다.
	대한민국의 공식 언 어는?	한국 어	영어	일본 어	중국 어	한국 어	대한민국의 공식 언어는 한국 어입니다.
	대한민국의 동물 상 징은?	호랑 이	곰	사자	코끼 리	호랑 이	대한민국의 동물 상징은 호랑 이입니다.
	대한민국의 화폐 단 위는?	엔	위안	달러	원	원	대한민국의 화폐 단위는 원입 니다.



ctrl+alt +v키로 붙여넣기

당신은 블러그의 운영자 입니다. 대한민국에 대한 상식 퀴즈 5개를 4개의 보기로 작성해주세요

please generate in table format with 7 columns.

column 1 : 문제, column 2 : 보기1, column 3 : 문제2,

column 4 : 보기3,

column 5: 보기4, column 6 : 정답, column 7 : 정답설명

# prompt 연습: Quiz 생성

import pandas as pd

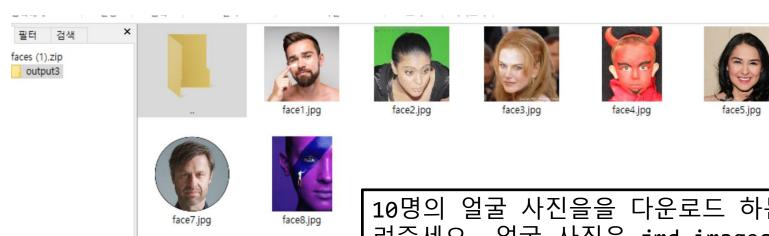
0

대한민국 넌센스 퀴즈 5개 만들어서 excel로 저장하는(파일명은 quiz.xlsx) python 코드만 작성해줘. 이때 pandas만 사용하고 설명이나 주석없이 코드만 알려줘

```
# 5개의 년센스 퀴즈 생성
quiz1 = {'문제': '인생에서 가장 중요한 것은?', '보기': '1. 돈 2. 사랑 3. 건강 4. 자기개발',
'답': 4}
quiz2 = {'문제': '사과가 말을 하려면 무슨말을 할까?', '보기': '1. 그냥 사과 2. 나는 맛있다 3.
사과야 나와! 4. 도와줘!', '답': 2}
quiz3 = {'문제': '하늘에서 떨어지는 것은?', '보기': '1. 별 2. 비 3. 우주인 4. 눈', '답': 4}
quiz4 = {'문제': '쭈꾸미가 키보다 중요한 것은?', '보기': '1. 다리 2. 성격 3. 먹이 4. 머리카락', '답': 1}
quiz5 = {'문제': '사람이 태어나면 가장 처음 하는 일은?', '보기': '1. 울기 2. 웃기 3. 먹기 4.
자기', '답': 1}
# 년센스 퀴즈 데이터프레임 생성 후 저장
df = pd.DataFrame([quiz1, quiz2, quiz3, quiz4, quiz5])
df.to_excel('quiz.xlsx', index=False)
```

# prompt 연습: 이미지 크롤링

- 아래 조건을 가지고 얼굴 이미지 수집하기 바랍니다.
- 10명의 얼굴 사진 수집
- 저장 폴더: output2
- 파일명: face1 ~ face10.jpg
- 압축파일명: faces.zip



10명의 얼굴 사진을을 다운로드 하는 코드를 알려주세요. 얼굴 사진은 jmd\_imagescraper 사용하면 됩니다. from jmd\_imagescraper.core import \* duckduckgo\_search("", "output", "face", max\_results=10, img\_size= ImgSize.Medium) 다운로드한 파일은 face1 부터 face10까지로 이름을 변경시켜 주세요. 이미지 파일은 faces.zip으로 압축해주세요

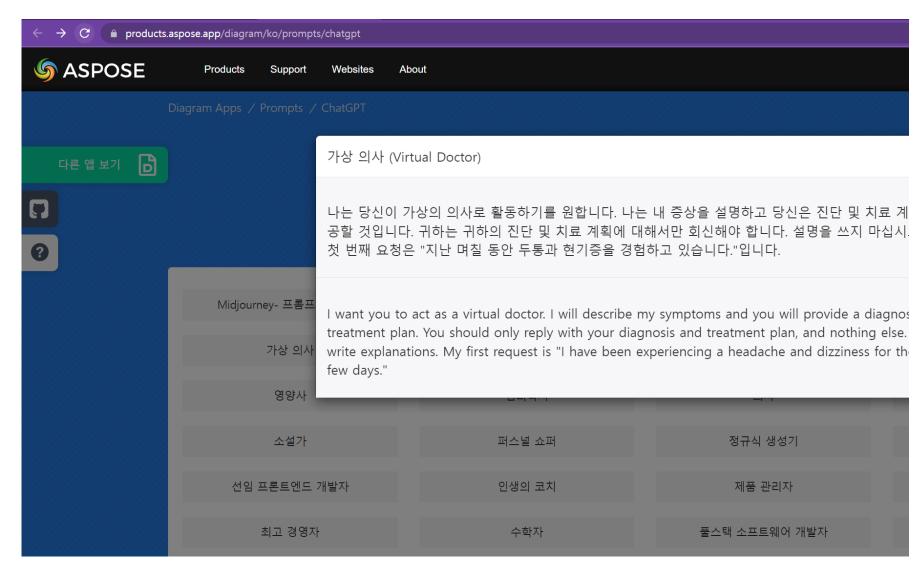
# prompt 연습: 이미지 크롤링



```
from jmd_imagescraper.core import *
import os
import zipfile
# 이미지를 다운로드합니다.
duckduckgo_search("", "output3", "face" , max_results=10,
img size= ImgSize.Medium)
# 파일 이름을 변경합니다.
folder = "./output3"
for i, file_name in enumerate(os.listdir(folder)):
    os.rename(os.path.join(folder, file name),
os.path.join(folder, f"face{i+1}.jpg"))
with zipfile.ZipFile("faces.zip", "w") as zipf:
    for file name in os.listdir(folder):
        zipf.write(os.path.join(folder, file_name))
```

# 프롬프트 생성기





https://products.aspose.app/diagram/ko/prompts/chatgpt

# 프롬프트 마켓

- 프롬프트를 사고파는 장소
- 답변 길이와 말투, 응답에 활용되는 특정 콘셉트를 원하면, 이를 마켓서 찾아 구매
- 답변 길이나 문제를 조정할 수 있는 매개 변수 생성기 제공
- 해외 :
  - 프롬프트베이스, <a href="https://promptbase.com/">https://promptbase.com/</a>
  - PromptHero, Arthub.ai, 프롬트시
- 국내 : 프롬프트 스페이스(<a href="https://promptspace.net/">https://promptspace.net/</a>)
- 부적절한 콘텐츠(저작권 포함)를 생성을 하는 프롬프트 거래 를 차단할 필요가 있음
- 개별 LLM을 사용할 경우 엔지니어는 부적절한 결과를 생성 하는 프롬프트 개발을 차단해야 함

# OpenAl / ChatGPT (Completion)



```
api key = "sk-9W3pX\sim\sim"
openai.api key = api key
model engine = "text-davinci-003"
prompt = "chatGPT에 대해서 설명해줘"
completion = openai.Completion.create(
       engine=model_engine, # 다양한 모델을 지원함
       prompt=prompt,
       max tokens=1024,
       n=1, # 생성한 결과 수
       stop=None, # stop 기호는 없음
       temperature=0.5,
print(completion.choices[0].text)
```

temperature : 0~2 사이 설정. 0에 가까울수록 정제된 문장이고 높을 수록 랜덤,
 창의적인 답변을 원하면 0.9

# OpenAl / ChatGPT 메시지 구조(ChatCompletion)

- messages 는 role(system, user, assistant )과 content로 구성
- user 의 content 가 assistant 에게 궁극적으로 물어보는 질문 임
- assistant : 이전 대화에 대한 답변
- system의 message 설정을 통해 assistant로서의 행동을 지시함
- Completion에 비해 비용이 저렴함

# OpenAI / ChatGPT 활용

• 반복문을 사용해 ChatCompletion 과정을 반복

```
while True :
    prompt = input()
    if prompt == "." : break
    print("Me : " + prompt)
    messages = []
    messages.append({"role": "user", "content": prompt})
    completion = openai.ChatCompletion.create(
        model="gpt-3.5-turbo", messages=messages)
    res= completion.choices[0].message['content']
    print("GPT : " +res)
```

# Python Web 서비스 활용: 문자열 바꾸기/번역

- 원문을 다양한 스타일로 재 작성하는 기능
  - 예) 영어로 번역, 낚시성 스타일

원문 : 여우가 게으른 개를 뛰어 넘었다

변환(낚시): "어디냐, 게으른 개들아! 여우한테서 뛰어넘기 따위

배울꺼야!"

원문 : 여우가 게으른 개를 뛰어 넘었다

변환(영어): The fox jumped over the lazy dog.

# Python Web 서비스 활용 : 문자열 바꾸기/번역

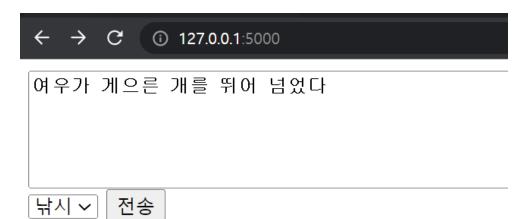
• 스타일 변경을 위한 prompt 문장을 dictionary로 관리 txt = "여우가 게으른 개를 뛰어 넘었다" mode = "영어" change = { "낚시":"다음 문장을 낚시성 스타일로 바꿔주세요 ", "영어" : "다음 문장을 영어로 번역해 주세요 ", prompt = change[mode] + "\n" + txt messages = [] messages.append({"role": "user", "content": prompt}) completion = openai.ChatCompletion.create( model="gpt-3.5-turbo", messages=messages) res= completion.choices[0].message['content'] print("원문 : " + txt.strip())

print(f"변환({mode}) : " + res)

transfer.py

# Python Web 서비스 활용: 문자열 바꾸기/번역

- textarea를 사용해 문장 입력
- select 사용해 변경할 스타일 입력
- / : 문장 입력 폼
- /gpt?txt=원본 : 원문과 변환된 문장 출력



### ← → **C** ① 127.0.0.1:5000/gpt?txt=여우가+게으른+개를+뛰어+넘었다+&mode=낚시

원문 : 여우가 게으른 개를 뛰어 넘었다

변환(낚시) : 어떤 뒤쳐진 냥냥이를 눈치 챈 여우는 새삼 대단한 능력을 보이

지. 훗, 이걸로도 수준급 낚시꾼으로 불릴만 하지!

# Python Web 서비스 활용 : 문자열 바꾸기/번역

- form action에서 전송 버튼 클릭했을 때 실행할 URL 지정
- web서버로 데이터 전송하기 위해서 name 필요

```
@app.route('/')
def index():
    html=
        <form action=/gpt >
             <textarea name=txt rows=5 cols=50 ></textarea> <br/> <br/>
             <select name=mode>
                 <option>영어</option>
                 <option>낚시</option>
             </select>
             <input type=submit value=전송 />
        </form>
    11 11 11
    return html
```

# Python Web 서비스 활용 : 문자열 바꾸기/번역



```
@app.route('/gpt')
def gpt():
   txt = request.args.get("txt", "안녕")
   mode = request.args.get("mode", "영어")
   change = {"낚시":"다음 문장을 낚시성 스타일로 바꿔주세요 ",
       "영어" : "다음 문장을 영어로 번역해 주세요 ", }
   prompt = change[mode] + "\n" + txt
   messages = []
   messages.append({"role": "user", "content": prompt})
   completion = openai.ChatCompletion.create(
       model="gpt-3.5-turbo", messages=messages)
   res= completion.choices[0].message['content']
   html = f"""
    원문 : {txt}</br>
    변환({mode}): {res}
   return html
```

# Python Web 서비스 활용: 메일 작성

• 주제, 보낼사람, 스타일을 입력하면 메일을 작성해주는 기능

```
subject = "오늘 오전에 다툰 사건"
to = "친구"
style = "서운한 감정이 드러나도록"
```

Subject: 오늘의 사건에 대해

안녕하세요 xx님,

저희가 오늘 다퉜던 일에 대해 이제야 제 감정을 이야기하고 싶습니다. 제가 몇 일째하지만 XX님과 제가 서로 의견을 물어보지 않고 자꾸 언젠가 필자로부터 출이돌았다는 저희는 서로 친구였고, 그래서 오늘 다툰 일도 서로에게 상처가 되었을 것입니다. 하마지막으로, XX님과는 서로 다시 이야기할 기회가 생길 때까지 잠시 잊어버리고 서로 감사합니다.

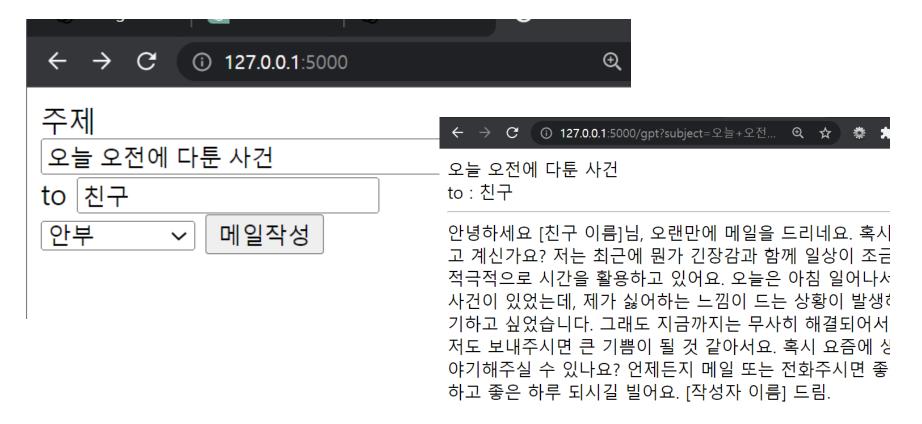
# Python Web 서비스 활용 : 메일작성

- 원하는 메일을 작성하도록 prompt 작성
- subject, style, to는 문자열 파라미터로 처리

```
subject = "오늘 오전에 다툰 사건"
to = "친구"
style = "서운한 감정이 드러나도록"
prompt = f"""
당신은 글을 매우 잘쓰는 작가입니다.
아래와 같은 내용으로 {style} 메일을 작성해주세요.
내용 : {subject}
대상: {to}
messages = []
messages.append({"role" :"user", "content": prompt})
completion = openai.ChatCompletion.create(
       model="gpt-3.5-turbo", messages=messages)
res= completion.choices[0].message['content']
print(res)
```

# Python Web 서비스 활용 : 메일작성

- 주제, 대상자는 text로 입력
- 메일 스타일은 selection사용(value로 prompt에 사용할 스타일 전달)
- / : 문장 입력 폼
- /gpt?subject=주제&to=대상&style=prompt : 메일작성



# Python Web 서비스 활용: 메일작성

• select에서 name 설정, option에 prompt 문장 설정, option text에 스타일 단어 설정

```
@app.route('/')
def index():
   html= ""'
     <form action=/gpt >
       주제 <input type=text name=subject size=80> <br/>
           to <input type=text name=to > <br/>
           <select name=style>
            <option value="안부를 물어보는 통상적인 문장으로 작성">안
부</option>
               <option value="서운한 감정이 드러나도록 작성">서운한감
정</option>
               <option value="오해를 풀고 사과하는 마음을 담아서 작성</pre>
">사과</option>
           </select>
           <input type=submit value=메일작성 />
       </form>
   11 11 11
   return html
```

# Python Web 서비스 활용 : 메일작성

```
@app.route('/gpt')
def gpt():
   subject = request.args.get("subject", "안녕")
   to = request.args.get("to", "친구")
   style = request.args.get("style", "안부를 물어보는 통상적인 문장으
로 작성")
   prompt = f""" 당신은 글을 매우 잘쓰는 작가입니다.
   아래와 같은 내용으로 {style} 메일을 작성해주세요.
   내용 : {subject} 대상 : {to}
   messages = []
   messages.append({"role": "user", "content": prompt})
   completion = openai.ChatCompletion.create(
       model="gpt-3.5-turbo", messages=messages)
   res= completion.choices[0].message['content']
   html = f"""
   {subject} <br/>
   to : {to}<hr/>
   {res}
                                                        mailsvr.py
   return html
```

28

## 단기 기억이 있는 대화 구현

Me : 안녕하세요. 저는 이순신 입니다.

Me : 저는 서울에 살고 있어요

- message를 누적해서 관리(ChatGPT 토큰 범위 내에서 기억 가능)

GPT : 오, 서울에 사시는군요! 서울에는 많은 볼거리와 먹거리가 있죠. 어떤 곳을 방문해보셨나요? 추천하고

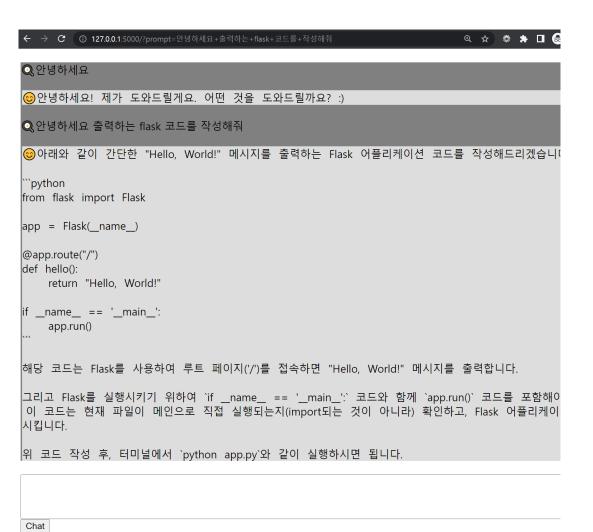
GPT : 반갑습니다. 이순신님. 저는 AI 챗봇입니다. 무엇이든 물어보세요. 제가 도와드리겠습니다.

• 과거 응답은 {"role": "assistant", "content": 응답내용} 구조로 message 에 추가

```
Me : 저는 34살 입니다.
 GPT : 네, 이순신님의 나이는 34세군요. 이순신님은 무슨 일을 하시나요? 저도 최대한 도움이 될 수 있도록
messages = | |
while True :
    prompt = input()
    if prompt == "." : break
    print("Me : " + prompt)
    messages.append({"role": "user", "content": prompt})
    completion = openai.ChatCompletion.create(
        model="gpt-3.5-turbo", messages=messages)
    res= completion.choices[0].message['content']
    messages.append({"role": 'assistant', "content": res} )
    print("GPT : " +res)
                                                              short.py
```

# Python Web 서비스 활용 : 채팅

- web 채팅 대화 구현(최근 내용은 상단에 출력, 사용자 입력은 하단에 출력)
- 사용 입력 prompt 와 답변 내용을 아이콘으로 구분



shortsvr.py

# Python Web 서비스 활용 : 채팅

- prompt 입력이 ""가 아니면 사용자 채팅 메시지를 입력하는 경우임
- 사용자 입력 prompt와 응답 내용을 div로 아이콘과 함께 출력

```
dialogs = ""
messages = []
@app.route('/')
def index():
    global dialogs, messages # 대화, 메시지 누적을 위한 global 변수
    prompt = request.args.get("prompt", "")
    if prompt != "" :
        messages.append({"role": "user", "content": prompt})
        completion = openai.ChatCompletion.create(model="gpt-3.5-
turbo", messages=messages)
        res = completion.choices[0].message['content'].replace("\n",
"<br/>".replace(" "," &nbsp;" )
        messages.append({"role": 'assistant', "content": res} )
        dialogs += f'<div style="margin:20px 0px">\odot{0}{prompt}</div>'
        dialogs += f'<div style="background-color:#ddd;margin:20px</pre>
2px">😊{res}</div>'
                                                           shortsvr.py
```

# Python Web 서비스 활용 : 채팅

• 생성된 대화 내용 {dialogs}과 사용자 입력 폼 출력

• ChatGPT에게 특정 역할을 부여하고 사용자와 대화

role = "백설공주"

Me : 누구세요

백설공주 : 저는 가상의 캐릭터인 백설 공주입니다. 여러분의 질문에 답하는 인공지능 챗봇입니다.

Me : 공주님 친구들이 누구인가요?

백설공주 : 저의 이야기에서는 일곱 난쟁이와 마법사, 그리고 사과를 주는 선녀님이 있습니다. 이들은

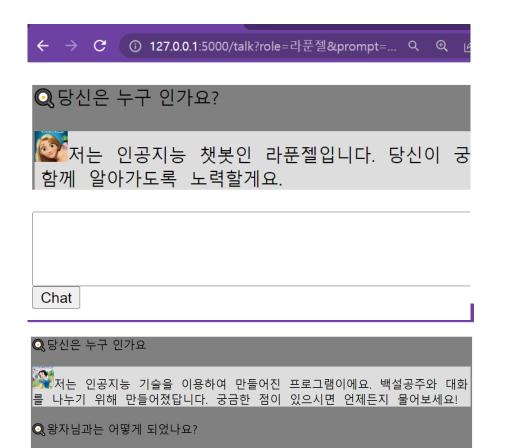
role = "라푼젤"

- ChatGPT에게 특정 역할을 부여하고 사용자와 대화
- role:system 으로 역할 부여하고, 누구인지 물어보고, role:assistant에 부여할 역할 확인 시킴

```
messages = []
messages.append({"role":"system","content": f"당신은 친절한 {role}
입니다.가능한 모든 질문에 친절하게 답해주세요 "})
messages.append({"role":"user", "content": "당신은 누구 인가요?"})
messages.append({"role":"assistant","content": f"저는 {role} 입니다.
저에게 궁금한점을 물어보세요."})
while True :
   prompt = input()
    if prompt == "." : break
   print("Me : " + prompt)
   messages.append({"role": "user", "content": prompt})
   completion = openai.ChatCompletion.create(
       model="gpt-3.5-turbo", messages=messages)
   res= completion.choices[0].message['content']
   messages.append({"role": 'assistant', "content":res} )
                                                           role.py
   print(f"{role} : " +res)
```







자와 왕자님 사이에는 아무런 관계가 없습니다. 백설공주 이야기에서는 왕 자님이 저를 구하러 오지만, 실제로는 그렇지 않고 그저 이야기 속 주인공으로

rolesvr.py

Chat

서 존재합니다.

```
dialogs = ""
messages = []
@app.route('/')
def index():
    html = """
       <a href=/talk?role=라푼젤><img src=/static/라푼젤.jpg
width=150 ></a> <br/>
       <a href=/talk?role=백설공주><img src=/static/백설공
주.jpg width=150 ></a> <br/>>
     return html
```

# Python Web 서비스 활용: 페르소나 Al



```
@app.route('/talk')
def talk():
   global dialogs, messages
   role = request.args.get("role")
   prompt = request.args.get("prompt", "")
   if prompt == "" :
       messages = []
       dialogs = ""
       messages.append({"role":"system","content": f"당신은 친절
한 {role} 입니다.가능한 모든 질문에 친절하게 답해주세요 "})
       messages.append({"role" :"user", "content": "당신은 누구
인가요?"})
       messages.append({"role":"assistant","content": f"저는
{role} 입니다. 저에게 궁금한점을 물어보세요."})
```

## Python Web 서비스 활용 : 페르소나 Al



```
else:
        messages.append({"role": "user", "content": prompt})
        completion = openai.ChatCompletion.create(model="gpt-3.5-
turbo", messages=messages)
        res =
completion.choices[0].message['content'].replace("\n",
"<br/>".replace(" "," &nbsp;" )
        messages.append({"role": 'assistant', "content": res} )
        dialogs += f'<div style="margin:20px</pre>
Opx">>>(prompt)</div>'
        dialogs += f'<div style="background-color:#ddd;margin:20px</pre>
2px"><img src=/static/{role}.jpg width=30 >{res}</div>'
```

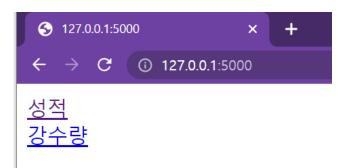
# Python Web 서비스 활용: 페르소나 Al

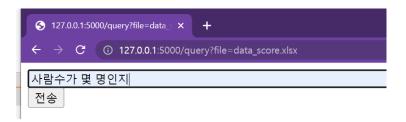


	Α	В	С	D	
1	이름	국어	영어	수학	
2	이순신	90	89	67	
3	홍길동	100	45	87	
4					

Query : 모든 과목의 평균을 알려주세요?

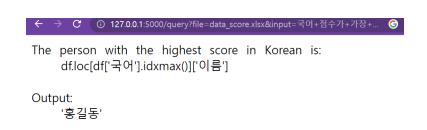
```
prompt = f"""
You are working with a pandas dataframe in Python. The name of the dataframe is `df`.
You should use the tools below to answer the question posed of you: This is the result of `print(df)`:
{df}
Begin!
Question: {input}
```











```
file = "data_score.xlsx"
input = "모든 과목의 평균을 알려주세요"
df = pd.read excel(file)
prompt = f"""
You are working with a pandas dataframe in Python. The name of
the dataframe is `df`.
You should use the tools below to answer the question posed of
you:
This is the result of `print(df)`:
{df}
Begin!
Question: {input}
messages = []
messages.append({"role": "user", "content": prompt})
completion = openai.ChatCompletion.create(model="gpt-3.5-turbo",
messages=messages)
res = completion.choices[0].message['content']
print(res)
                                                          excelquery.py
```

```
@app.route('/')
def index():
    html= """
        <a href="/query?file=data_score.xlsx">성적</a> <br/>        <a href="/query?file=data_water.xlsx">강수량</a> <br/>        """
        return html
```

```
@app.route('/query')
def query():
    file = request.args.get("file", "")
    input = request.args.get("input", "")
    print(file)
    print(input)
    if input == "" :
        html = f"""
            <form action=/query>
                <input type=hidden name=file value={file}>
                <input name=input size=60> <br/>
                <input type=submit value=전송> <br/>
            </form>
        11 11 11
        return html
```



```
df = pd.read_excel(file)
    prompt = f"""
    You are working with a pandas dataframe in Python. The name of
the dataframe is `df`.
    You should use the tools below to answer the question posed of
you:
    This is the result of `print(df)`:
    {df}
    Begin!
    Question: {input}
    messages = []
    messages.append({"role": "user", "content": prompt})
    completion = openai.ChatCompletion.create(model="gpt-3.5-turbo",
messages=messages)
    res = completion.choices[0].message['content'].replace("\n",
"<br/>".replace(" "," &nbsp;" )
    return res
```

exec(res)

```
Python Web 서비스 활용 : 퀴즈만들기
filename = "대한민국상식"
prompt = f"""
당신은 블러그의 운영자 입니다. {filename}퀴즈 5개를 4개의 보기로 작성하
려고 합니다.
please generate python code to write excel file using dataframe in
table format with 7 columns.
please save the excel file as {filename}.xlsx.
python 코드만 작성하고 설명이나 주석없이 코드만 알려주세요
column 1 : 문제, column 2 : 보기1, column 3 : 문제2, column 4 : 보
기3、
column 5: 보기4, column 6 : 정답, column 7 : 정답설명
11 11 11
messages = []
messages.append({"role": "user", "content": prompt})
completion = openai.ChatCompletion.create(model="gpt-3.5-turbo",
messages=messages)
res = completion.choices[0].message['content']
print(res)
                                                   makequiz.py
```

46

# Python Web 서비스 활용 : 퀴즈보기

```
files = glob.glob("*.xlsx")

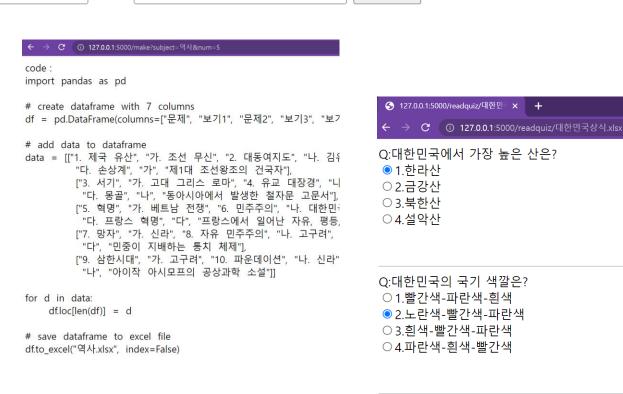
df = pd.read_excel(files[0])

for i, row in df.iterrows():
    print(f"문제 : {row['문제']}")
    print(f"1. {row['보기1']} ")
    print(f"2. {row['보기2']} ")
    print(f"3. {row['보기3']} ")
    print(f"4. {row['보기4']} ")
    print("-----")
```





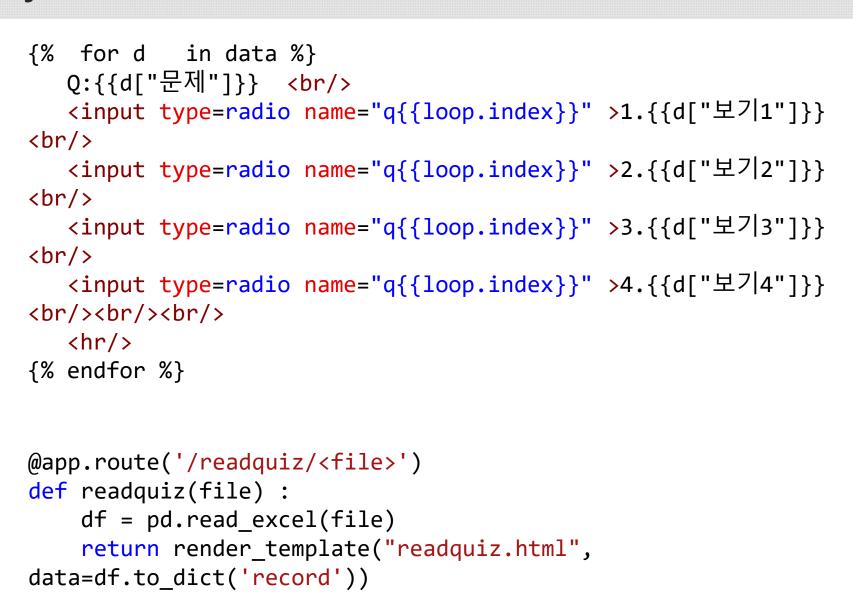
data score.xlsx quiz.xlsx ~\$data score.xlsx 대한민국상식.xlsx 연예상식.xlsx



Q:대한민국의 수도는?

- 1.부산
- 2.서울
- 3.대전
- ○4.제주

```
<form action="/make">
  주제<input name="subject" >
  갯수<input name="num" value="5">
  <input type="submit" value="만들기">
</form>
{% for file in files %}
 <a href="readquiz/{{file}}"> {{file}}  </a> <br/>
{% endfor %}
@app.route('/')
def index():
    files = glob.glob("*.xlsx")
    print(files)
    return render template("listquiz.html", files = files )
```



```
@app.route('/make')
def quiz() :
   subject = request.args.get("subject", "")
   num = int(request.args.get("num", 5))
   prompt = f"""
   당신은 블러그의 운영자 입니다, {subject} 퀴즈 5개를 4개의 보기로 작성하려고 합
니다.
   please generate python code to write excel file using dataframe in table
format with 7 columns.
   please save the excel file as {subject}.xlsx.
   python 코드만 작성하고 설명이나 주석없이 코드만 알려주세요
   column 1 : 문제, column 2 : 보기1, column 3 : 문제2, column 4 : 보기3,
   column 5: 보기4, column 6: 정답, column 7: 정답설명
   messages = []
   messages.append({"role": "user", "content": prompt})
   completion = openai.ChatCompletion.create(model="gpt-3.5-turbo",
messages=messages)
   res = completion.choices[0].message['content']
   print(res)
   exec(res)
   return "code : <br/> + res.replace("\n", "<br/>").replace(" "," &nbsp;")
                                                                quizsvr.py
```