



□ 파이썬을 활용한 자동 웹스크래핑

- 웹페이지(뉴스기사, 게시물 등)의 내용을 자동으로 엑셀로 수집하여 정리하고 싶을 때 유용

예시 1 웹페이지 내용 스크랩이 필요할 시, 예제로 스마트도시협회에서 운영하고 있는 ‘스마트시티 솔루션 마켓(<http://smartcitysolutionmarket.com>)’을 스크랩

스마트시티 솔루션마켓 Smartcity SolutionMarket					
로그인					
정부사업	지자체	기업	새싹기업	솔루션 검색	커뮤니티
광주광역시		통합플랫폼 연계 업무용 긴급영상지원서비스	2019	 스마트도시협회	이승준 02-6224-6114
경상남도 김해시		Smart 안심벨시스템 구축사업	2019	 현진아이피티(주)	김성수 02 2097 7400
경상남도 김해시		북거노인 교육센터 안심케어 서비스 구축 사업	2019	 (주)케이티	박상민 010-6733-8265
경상남도 김해시		상수도 원격검침 상수도 원격검침	2019	 (주)하이테크이피씨	최길수 070-5178-9054
전라북도 고창군		통합플랫폼 구축사업	2019	 메타빌드(주)	조성훈 02-598-3327

[절차1] 웹스크랩을 위해서는 먼저 수집하고자 하는 웹주소의 패턴 파악

▶ 스크랩하고 싶은 주소창의 웹 주소를 확인 ⇨ 주소의 패턴을 파악

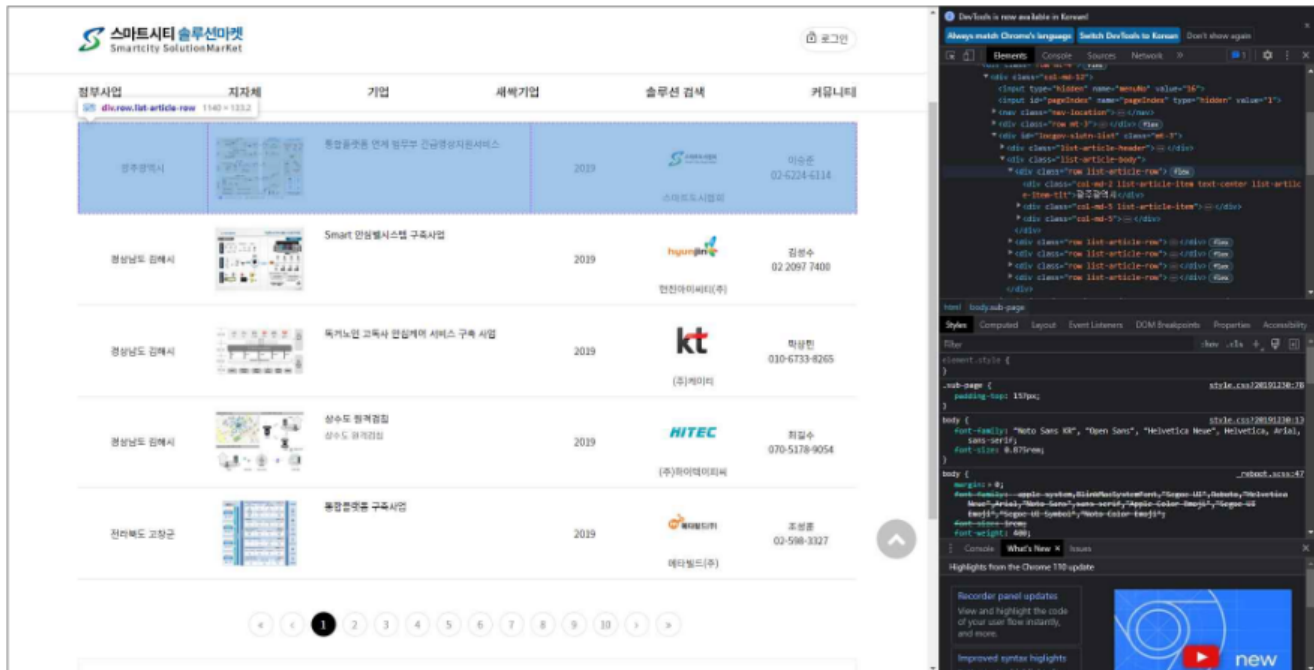
※ 뉴스기사나 게시물 같은 경우 ‘페이지 번호’에 따라서 주소가 변화하거나, ‘검색어’ 별로 링크가 만들어지는 패턴이 있다. 여기서 이러한 패턴을 찾는 것이 필요하다.

<http://smartcitysolutionmarket.com/scsm/locgov/locgovSlutnList.do?menuNo=16&sortOrderField=&searchCondition=&searchKeyword=&searchWdrLocgovNo=&searchBsisLocgovNo=&pageIndex=1> (위 예제 링크주소의 제일 마지막이 게시판 페이지번호별로 변하는 것을 알 수 있음)



[절차2] 구글 크롬에서 'F12' 키를 누르고 웹사이트의 HTML 구조 파악 및 코드값 정리

- ▶ 오른쪽 개발도구에서 HTML Elements를 선택하고 세분화하다 보면 왼쪽 편에 해당구역에 음영으로 표시됨



- ▶ 수집하고 싶은 정보가 음영으로 표시되면 HTML 코드를 정리

- 첫 번째 값은 지방자치단체로 '광주광역시' 값이다.
광주광역시 텍스트 앞의 `<div class="col-md-2 list-article-item text-center list-article-item-tit">`
코드 값을 복사
- 두 번째 값은 서비스명으로 '통합플랫폼 연계 법무부 긴급영상지원서비스' 앞의 코드
값을 복사 `<div class="col-lg-8">`
- 세 번째 값은 구축연도로 '2019' 년 앞의 `<span class="d-block d-md-none bg-light p-2">`
코드 값을 복사

[절차3] 위 내용이 정리가 되면 ChatGPT에 아래와 같이 명령

프롬프트 명령어 아래 주소의 웹사이트를 크롤링하고 엑셀 파일로 저장하는 파이썬 코드를 생성해줘

'<http://smartcitysolutionmarket.com/scsm/locgov/locgovSlutnList.do?menuNo=16&sortOrderField=&searchCondition=&searchKeyword=&searchWdrLocgovNo=&searchBsisLocgovNo=&pageIndex=>' + 'pagenumber'

위 주소링크는 마지막 pagenumber가 변화한다. 각 페이지의 내용을 수집 한다.
우선, 각 페이지에서 모든 `<div class="row list-article-row">`를 찾는다.



첫번째 열의 명을 '자치단체명'으로 하고, <div class="col-md-2 list-article-item text-center list-artilce-item-tit">의 텍스트를 값으로 한다. 예시는 '광주광역시'이다.
 두번째 열의 명을 '서비스명'으로 하고, <div class="col-lg-8">의 텍스트를 값으로 한다. 예시는 '통합플랫폼 연계 법무부 긴급영상지원서비스' 이다.
 세번째 열의 명을 '구축연도'로 하고, 의 텍스트를 값으로 한다. 예시는 '2019년'이다.

아래는 요구사항에 맞는 파이썬 코드입니다. 👍 🗨️

결과

```
python
import requests
from bs4 import BeautifulSoup
import pandas as pd

# 크롤링 할 URL의 베이스 주소
base_url = 'http://smartcitysolutionmarket.com/scsm/locgov/locgov$|utnList.do?menuNo=1

# 데이터를 저장할 빈 리스트
data = []

# 1~10 페이지까지 반복
for i in range(1, 11):
    # 현재 페이지 URL
    url = base_url + str(i)

    # URL에서 HTML 데이터 가져오기
    res = requests.get(url)
    soup = BeautifulSoup(res.content, 'html.parser')

    # 모든 자치단체 정보가 들어있는 <div> 태그 찾기
    rows = soup.find_all('div', {'class': 'row list-article-row'})

    # 각 <div> 태그에서 필요한 정보 추출하기
    for row in rows:
        # 첫번째 열 추출 (자치단체명)
        name = row.find('div', {'class': 'col-md-2 list-article-item text-center list-

        # 두번째 열 추출 (서비스명)
        service = row.find('div', {'class': 'col-lg-8'}).get_text().strip()

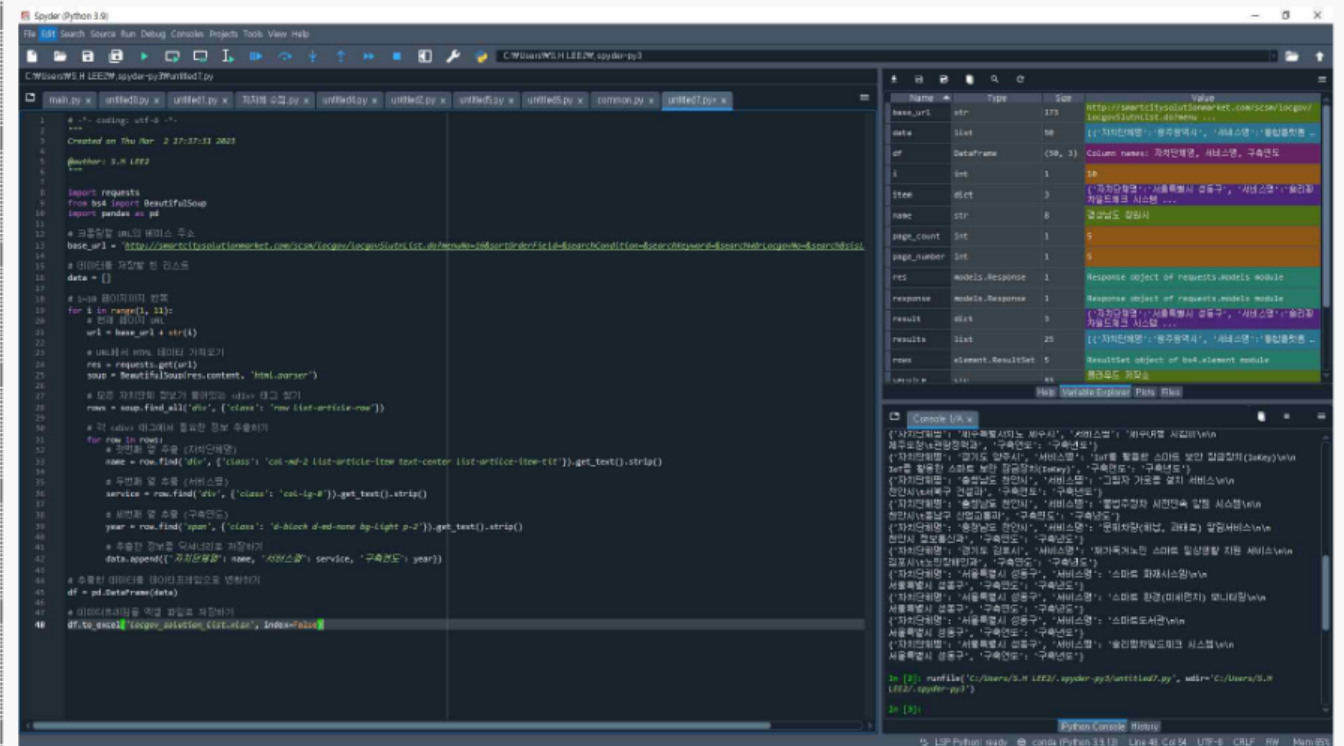
        # 세번째 열 추출 (구축연도)
        year = row.find('span', {'class': 'd-block d-md-none bg-light p-2'}).get_text(

        # 추출한 정보를 딕셔너리로 저장하기
        data.append({'자치단체명': name, '서비스명': service, '구축연도': year})
```



[절차4] 생성된 파이썬 코드를 파이썬 프로그램에 넣고 실행하면, 웹스크랩 시작

▶ 파이썬 스크립트를 열고 실행



[절차5] 웹스크랩이 끝나면 최종 저장된 파일에 스크랩 내용이 저장된 것을 볼 수 있음

▶ 생성된 엑셀파일을 열어서 웹스크랩이 잘 진행되었는지 확인

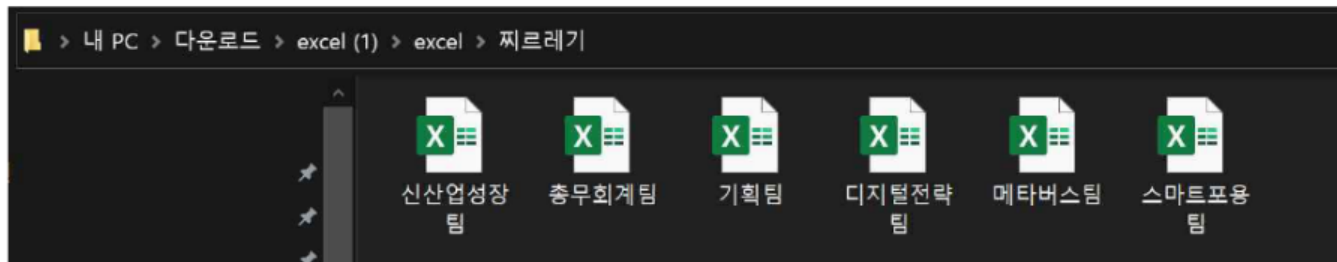
loggov_solution_list - Excel															
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	자치단체명	서비스명	구축연도												
2	광주광역시	통합플랫폼 연계 법무부 긴급영상지원	구축연도												
3	경상남도 김해시	Smart 안심벨시스템 구축사업	구축연도												
4	경상남도 김해시	특거노인 고독사 안심케어 서비스 구축	구축연도												
5	경상남도 김해시	산수도 원격검침	구축연도												
6	전라북도 고창군	통합플랫폼 구축사업	구축연도												
7	전라북도 고창군	112센터 긴급영상 지원 구축 사업	구축연도												
8	충청남도 부여군	특거노인중증장애인응급안전알림서비스	구축연도												
9	경기도 안양시	긴급상황 위치출동 서비스	구축연도												
10	경기도 안양시	공동주택 자량안심 서비스	구축연도												
11	경기도 안양시	여성안심주차 서비스	구축연도												
12	경기도 안양시	고령자 음성 안심 서비스	구축연도												
13	경기도 안양시	여성거주지 안심 서비스	구축연도												
14	경기도 안양시	대중교통운전자 서비스	구축연도												
15	경기도 안양시	안심통학버스 서비스	구축연도												
16	경기도 안양시	안전귀가 서비스	구축연도												
17	제주특별자치도 제주시	제주시여행 지침이	구축연도												
18	경기도 안양시	IoT를 활용한 스마트 보안 잠금장치(IoT)	구축연도												
19	충청남도 천안시	그림자 가로등 설치 서비스	구축연도												
20	충청남도 천안시	불법주정차 사전단속 알림 시스템	구축연도												
21	충청남도 천안시	문제차량(재난, 과태료) 알림서비스	구축연도												
22	경기도 김포시	재가특거노인 스마트 일상생활 지원 서비스	구축연도												
23	서울특별시 성동구	스마트 화재시스템	구축연도												
24	서울특별시 성동구	스마트 환경(미세먼지) 모니터링	구축연도												
25	서울특별시 성동구	스마트도서관	구축연도												



□ 파이썬을 활용한 문서취합기 만들기

- 부서별 요청자료 문서취합 시간을 줄여주는 파이썬 코드 생성

예시 2023년도 예산지출 계획 엑셀 파일을 부서별로 취합해야 할 때, 담당자가 부서별 송부된 파일을 일일이 열고 시트를 취합하기에 번거로운 경우



[절차1] ChatGPT에 '각 개별 파일의 시트를 하나의 파일로 통합하는 코드를 요청'

프롬프트 명령어 매우 많은 다수의 엑셀파일의 시트를 하나의 엑셀 파일에 각각 독립적인 시트를 유지하여 합치려고 한다. 파이썬 코드를 만들어주세요

결과 아래 코드는 하나의 엑셀 파일에 각각의 엑셀 파일의 시트를 독립적인 시트로 유지하며 병합하는 코드입니다.

```
python
import os
import pandas as pd

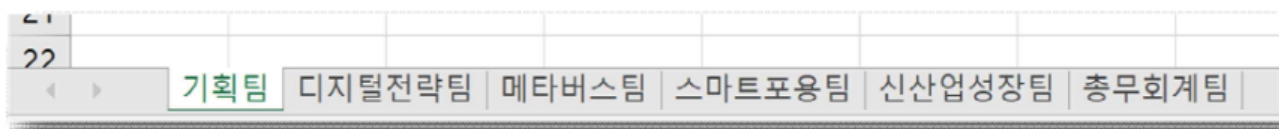
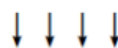
# 입력 폴더와 출력 파일 설정
input_folder = 'input_folder' # 입력 폴더명
output_file = 'output_file.xlsx' # 출력 파일명

# 입력 폴더 내의 모든 엑셀 파일 불러오기
all_files = os.listdir(input_folder)
excel_files = [file for file in all_files if file.endswith('.xlsx')]

# 모든 엑셀 파일의 시트를 하나의 파일로 병합
writer = pd.ExcelWriter(output_file, engine='xlsxwriter')

for file in excel_files:
    file_path = os.path.join(input_folder, file)
    # 각 엑셀 파일의 모든 시트를 불러오기
```

여러개 파일에 분산된 엑셀 시트를 하나로 합치는 코드



※ 파이썬 코드 실행 후 시트가 합쳐진 엑셀 파일 확인하면 모든 시트가 합쳐진 것을 확인



[절차2] 생성된 코드가 잘 작동되는지 확인되었다면, 사용하기 쉽게 그래픽인터페이스*(GUI)를 만들어 달라고 요청

프롬프트 명령어 위에서 만들어준 코드를 그래픽유저인터페이스(GUI) 기반으로 실행할 수 있는 파이썬 코드를 생성해주세요.



결과

GUI 기반으로 실행할 수 있는 파이썬 코드를 생성하기 위해서는 PyQt 또는 Tkinter와 같은 GUI 라이브러리를 사용해야 합니다. 이번 예제에서는 PyQt5를 사용하여 GUI 기반으로 실행할 수 있는 코드를 생성해보겠습니다.

먼저 PyQt5를 설치합니다. Anaconda prompt나 터미널에서 다음 명령어를 입력합니다.

```
pip install PyQt5
```

그리고 다음과 같이 코드를 작성합니다.

```
python
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QFileDialog, QPushButton, QLabel
from PyQt5.QtGui import QIcon
from PyQt5.QtCore import pyqtSlot
import pandas as pd

class App(QWidget):

    def __init__(self):
        super().__init__()
        self.title = 'Excel Merger'
        self.left = 10
        self.top = 10
        self.width = 640
        self.height = 480
        self.file_names = []
        self.initUI()

    def initUI(self):
```

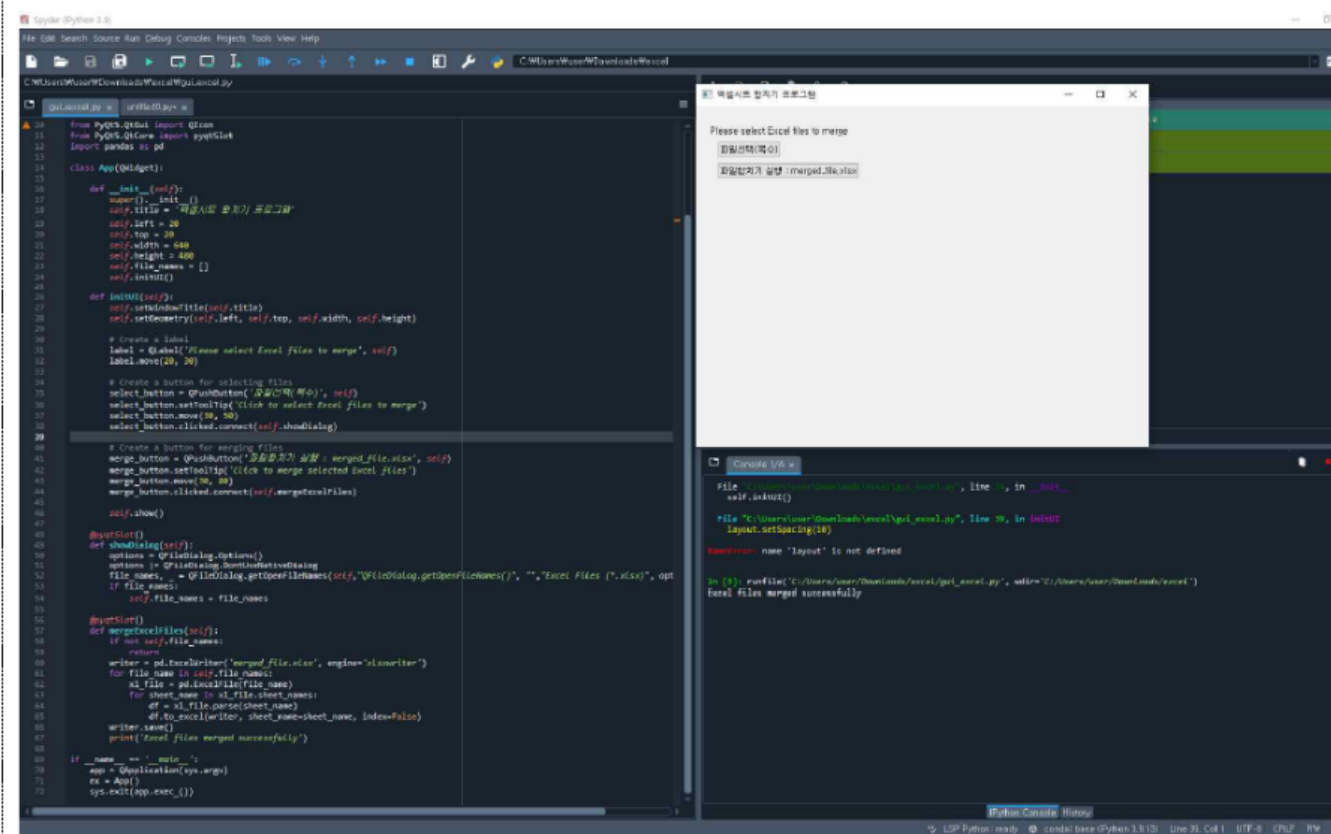
절차1에서 생성된 코드에 그래픽유저인터페이스를 재생산하는 코드



[절차3] 파이썬 프로그램에서 실행하면 그래픽인터페이스(GUI)가 보이는지 확인

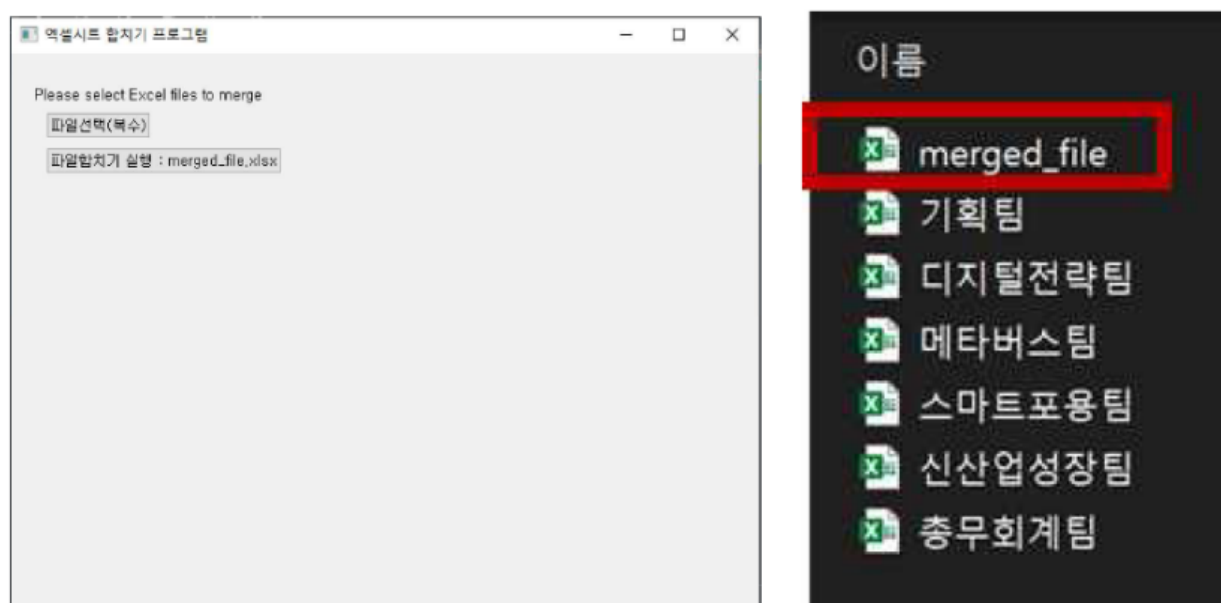
▶ 파이썬을 실행하면 인터페이스 창이 나타남

※ 생성된 GUI는 Pyinstaller를 사용하여 누구에게나 실행파일(EXE)로 배포 가능



[절차4] 프로그램 작동시 엑셀 파일들의 시트들이 하나의 파일(merged_file)로 취합되는지 확인

▶ 인터페이스 프로그램을 통해 실행 후 최종 파일 확인



ChatGPT가 생성해준 프로그래밍 코드를 그래픽인터페이스(GUI)로 변환하여 범용활용 가능