

[3-6. main.py 파일 작성]

1. journal_ledger.py 파일 생성

- 함수에 기존 작성한 코드를 담아서 파이썬 파일에 작성

```

In [ ]: # journal_ledger.py

from journal_file import Journal
from openpyxl import Workbook

def ledger_category(
    journal_name="분개장_샘플.xlsx", file_name="계정별원장.xlsx"
):
    # 분개장 데이터 로딩
    jnl = Journal(journal_name)

    # 워크북 생성
    wb = Workbook()

    # 기본적으로 생성되는 활성 시트 제거
    default_sheet = wb.active
    wb.remove(default_sheet)

    # 각 카테고리에 대해 시트 생성 및 데이터 쓰기
    for category in jnl.df["계정과목"].unique():
        # category 이름으로 신규 시트 생성
        ws = wb.create_sheet(title=category)

        # 시트에 첫번째 셀에 category 이름 출력
        ws.append([category])
        #ws.cell(row=1, column=1, value=category)

        # 한칸 띄우기
        ws.append([])

        # 컬럼명 출력
        ws.append(tuple(jnl.df.columns))

        # category에 해당하는 계정과목 DataFrame에 대하여 한행씩 row 변수에 대입
        for r_idx, row in enumerate(
            jnl.계정과목(category).df.itertuples(index=False),
            start=4
        ):
            # itertuples 메서드 대신 iterrows 메서드 사용 가능
            # (itertuples 메서드가 iterrows 메서드 보다 빠르게 작동)

            # 한 행(row)에 대하여 컬럼번호(c_idx)를 증가시키면서 한 셀씩 출력
            for c_idx, value in enumerate(row, start=1):
                ws.cell(row=r_idx, column=c_idx, value=value)

    # 최종 완성된 파일 엑셀 저장
    wb.save(file_name)

```

```
In [ ]: def ledger_client(
        journal_name="분개장_샘플.xlsx",
        file_name="거래처원장.xlsx"
    ):
        # 분개장 데이터 로딩
        jnl = Journal(journal_name)

        # 워크북 생성
        wb = Workbook()

        # 기본적으로 생성되는 활성 시트 제거
        default_sheet = wb.active
        wb.remove(default_sheet)

        # 각 카테고리에 대해 시트 생성 및 데이터 쓰기
        for client in jnl.df["거래처명"].unique():
            # client 이름으로 신규 시트 생성
            ws = wb.create_sheet(title=client)

            # 시트에 첫번째 셀에 client 이름 출력
            ws.append([client])
            #ws.cell(row=1, column=1, value=client)

            # 한칸 띄우기
            ws.append([])

            # 컬럼명 출력
            ws.append(tuple(jnl.df.columns))

            # client에 해당하는 거래처명 DataFrame에 대하여 한행씩 row 변수에 대입
            for r_idx, row in enumerate(
                jnl.거래처명(client).df.itertuples(index=False),
                start=4
            ):
                # itertuples 메서드 대신 iterrows 메서드 사용 가능
                # (itertuples 메서드가 iterrows 메서드 보다 빠르게 작동)

                # 한 행(row)에 대하여 컬럼번호(c_idx)를 증가시키면서 한 셀씩 출력
                for c_idx, value in enumerate(row, start=1):
                    ws.cell(row=r_idx, column=c_idx, value=value)

            # 최종 완성된 파일 엑셀 저장
            wb.save(file_name)
```

2. journal_ledger.py 파일 테스트

```
In [ ]: from journal_ledger import (
        ledger_category as ldg_category,
        ledger_client as ldg_client
    )
```

```
In [ ]: ldg_category(file_name="계정별원장2.xlsx")
```

```
In [ ]: ldg_client(file_name="거래처원장2.xlsx")
```

3. main.py 파일 작성

```
In [ ]: # main.py

from journal_ledger import ledger_category, ledger_client

def main():
    while True:
        print("실행하고자 하는 항목을 선택하세요: ")
        print("1: 계정별원장 출력")
        print("2: 거래처원장 출력")
        print("exit: 프로그램 종료")
        input_data = input("실행하고자 하는 항목을 선택 >>> ")

        if input_data == "1":
            journal_name = input(
                "분개장 파일명을 입력하세요(default: 분개장_샘플.xlsx): "
            )
            if journal_name == "":
                journal_name = "분개장_샘플.xlsx"
            print(f"분개장 파일명: {journal_name}")

            file_name = input(
                "출력할 파일명을 입력하세요(default: 계정별원장_main.xlsx): "
            )
            if file_name == "":
                file_name = "계정별원장_main.xlsx"
            print(f"출력 파일명: {file_name}")

            ledger_category(journal_name, file_name)
            print(f"{file_name} 파일 생성이 완료되었습니다.\n\n")

        elif input_data == "2":
            journal_name = input(
                "분개장 파일명을 입력하세요(default: 분개장_샘플.xlsx): "
            )
            if journal_name == "":
                journal_name = "분개장_샘플.xlsx"
            print(f"분개장 파일명: {journal_name}")

            file_name = input(
                "출력할 파일명을 입력하세요(default: 거래처원장_main.xlsx): "
            )
            if file_name == "":
                file_name = "거래처원장_main.xlsx"
            print(f"출력 파일명: {file_name}")

            ledger_client(journal_name, file_name)
            print(f"{file_name} 파일 생성이 완료되었습니다.\n\n")

        elif input_data == "":
            continue
```

```
elif input_data == "exit":  
    print("프로그램 종료")  
    return  
  
else:  
    print("프로그램 종료")  
    return  
  
if __name__ == "__main__":  
    main()
```

실행하고자 하는 항목을 선택하세요:

1: 계정별원장 출력

2: 거래처원장 출력

exit: 프로그램 종료

분개장 파일명: 분개장_샘플.xlsx

출력 파일명: 계정별원장_main.xlsx

계정별원장_main.xlsx 파일 생성이 완료되었습니다.

실행하고자 하는 항목을 선택하세요:

1: 계정별원장 출력

2: 거래처원장 출력

exit: 프로그램 종료

분개장 파일명: 분개장_샘플.xlsx

출력 파일명: 거래처원장_main.xlsx

거래처원장_main.xlsx 파일 생성이 완료되었습니다.

실행하고자 하는 항목을 선택하세요:

1: 계정별원장 출력

2: 거래처원장 출력

exit: 프로그램 종료

프로그램 종료

4. main.py 파일 실행

In []: `%run main.py`

실행하고자 하는 항목을 선택하세요:

1: 계정별원장 출력

2: 거래처원장 출력

exit: 프로그램 종료

분개장 파일명: 분개장_샘플.xlsx

출력 파일명: 계정별원장_main2.xlsx

계정별원장_main2.xlsx 파일 생성이 완료되었습니다.

실행하고자 하는 항목을 선택하세요:

1: 계정별원장 출력

2: 거래처원장 출력

exit: 프로그램 종료

분개장 파일명: 분개장_샘플.xlsx

출력 파일명: 거래처원장_main2.xlsx

거래처원장_main2.xlsx 파일 생성이 완료되었습니다.

실행하고자 하는 항목을 선택하세요:

1: 계정별원장 출력

2: 거래처원장 출력

exit: 프로그램 종료

프로그램 종료

In []: