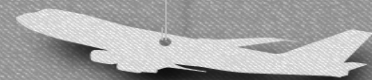


# Chapter - 03



## 엑셀 파일 처리 (openpyxl 모듈)



# 1. openpyxl 모듈



- 다음은 엑셀 파일 조작을 위한 python의 모듈 및 그 특징이다

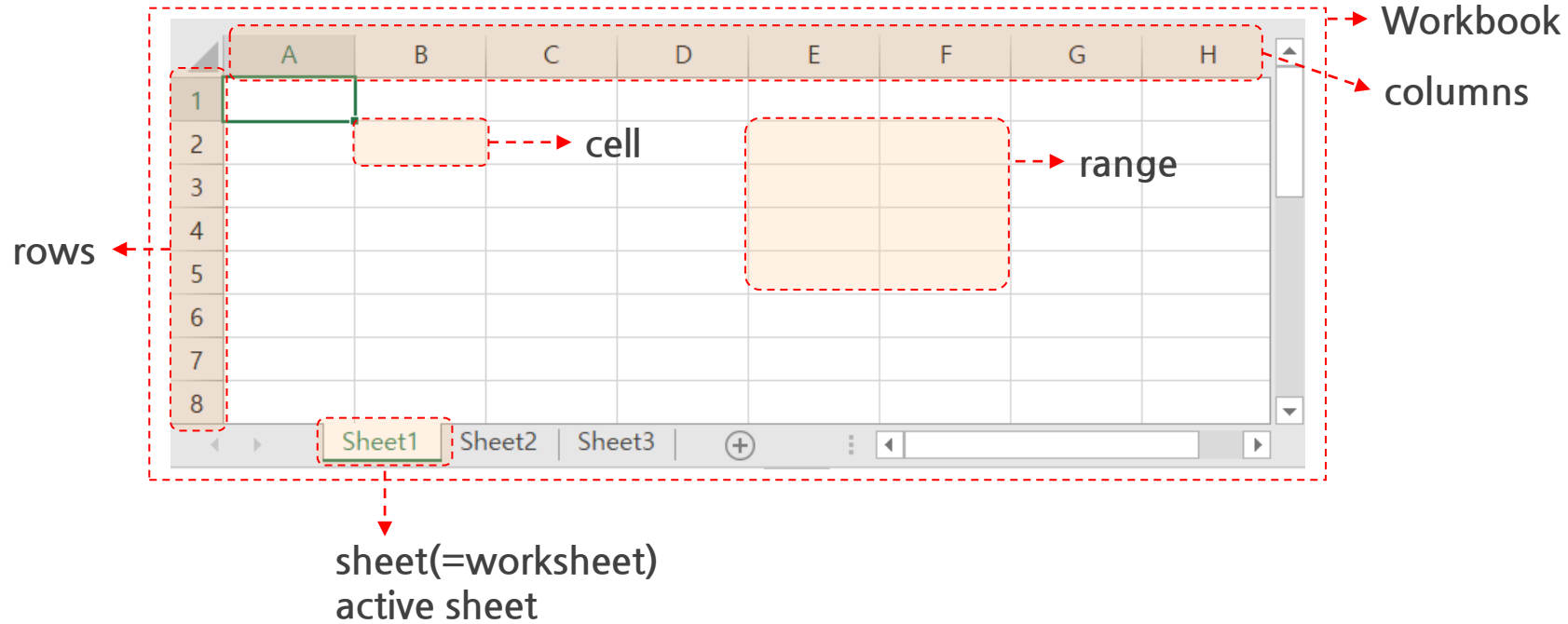
openpyxl (xlsx 지원) XlsxWriter (쓰기만 지원)	xlrd, xlwt (xls 지원)	xlwings (윈도우, MacOS) pywin32 (윈도우)
설치된 엑셀 필요 없음 (윈도우, MacOS, Linux)		설치된 엑셀 필요

- openpyxl 모듈은 엑셀 파일 생성 및 조작이 가능한 모듈이다
  - pip install openpyxl 을 실행하여 설치하고, import openpyxl 하여 사용한다
    - <https://openpyxl.readthedocs.io/en/stable/> 에서 자세한 사용법을 볼 수 있음
  - openpyxl 모듈 사용시 주의할 사항은 다음과 같다
    - 보안이 걸려 있는 파일을 열기 할 수 없고, “모두 새로 고침” 기능 지원 하지 않음
    - 작업 속도가 XlsxWriter(대용량에 유리), PyExceleerate(문서화가 좋지 않음) 보다 느림
  - 따라서, 작업에 따라 pywin32 및 pandas를 함께 사용하는 것이 필요하다
    - <https://openpyxl.readthedocs.io/en/stable/optimized.html> 참조

# 1-01. 엑셀 구성 요소



다음 엑셀 화면을 통해 엑셀의 기본 용어를 확인하도록 한다



# 1-02. 빈 workbook 파일을 만들고 저장하기



1-02 실습) 빈 workbook 파일을 만들고 sheet의 이름을 수정한 뒤 저장한다

myworkbook.xlsx 파일이 생성되면 더블 클릭하여 열어 확인하도록 한다

```
import openpyxl

dest_filename = "myworkbook.xlsx"
wb = openpyxl.Workbook()
sheet = wb.active
sheet.title = "mysheet1"
sheet['A1'].value = 'name'
sheet['A2'].value = 'yoon so young'
wb.save(dest_filename)
```

	A	B	C
1	name		
2	yoon so young		
3			
4			
5			
6			
7			
8			

mysheet1

myworkbook.xlsx

- sheet['A1'].value와 같은 표현  
→ sheet.cell(row = 1, column = 1).value



# 1-03. 엑셀 파일 열고 sheet 추가/ 객체 얻기



1-03 실습) 엑셀 파일을 열어 sheet를 추가하고, 특정 시트 정보를 얻는다

실행 후, myworkbook\_after.xlsx 파일을 열어 시트 상태를 확인한다

```
import openpyxl

dest_filename = "myworkbook.xlsx"
wb = openpyxl.load_workbook(dest_filename)

print(wb.sheetnames)
wb.create_sheet(index=0, title='mysheet0')
wb.create_sheet(index=2, title='mysheet2')
snames = wb.sheetnames
print(snames)
sheet = wb[snames[1]]
print(sheet['A1'].value)
print(sheet['A2'].value)
wb.save('myworkbook_after.xlsx')
```

	A	B	C	D
1	name			
2	yoon so young			
3				
4				
5				
6				
7				
8				

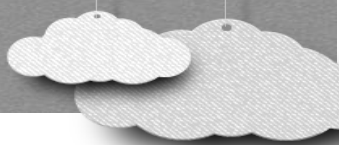
sheet 추가됨

mysheet0   mysheet1   mysheet2

```
['mysheet1']
['mysheet0', 'mysheet1', 'mysheet2']
name
yoon so young
```

▪ sheet 삭제 : `wb.remove(sheetname)` 또는 `del wb[sheetname]`

# 1-04. Range (여러 셀) 사용하기



🍃 [예제3] range에서 cell에 접근하는 방법을 알아 보도록 한다

🍃 range는 여러 셀을 포함하며, 아래와 같이 다양한 방법으로 지정할 수 있다

```
wb = openpyxl.Workbook()
sheet = wb.active
```

```
for x in sheet['A1': 'A9']:
    x[0].value = 2
```

→ 행 단위 Cell들의 tuple을 item으로 하는 tuple (2차원 tuple)

```
for idx, x in enumerate(sheet['B1:B9'], start=1):
    x[0].value = idx
```

```
for x, y in sheet['A1:B9']:
    print(f'{x.value} * {y.value} = {x.value * y.value}')
```

```
for x in sheet['1']:
    print(x, x.value)
```

→ Cell을 item으로 하는 tuple

```
wb.save('gugu.xlsx')
```

	A	B	C
1	2	1	
2	2	2	
3	2	3	
4	2	4	
5	2	5	
6	2	6	
7	2	7	
8	2	8	
9	2	9	
10			

▪ 다른 시트의 셀을 참조할 때는 sheet['SheetName!A1:A5'] 과 같이 사용한다

# 1-05. 수식 작성하기



1-05 실습) 등호로 시작하는 수식을 작성한다

cell 에는 수식이 적혀 있고, 눈에 보일 때는 수식의 결과가 표시된다

```
import openpyxl
```

```
wb = openpyxl.load_workbook('gugu.xlsx')  
sheet = wb['Sheet']
```

```
for idx, x in enumerate(sheet['C1:C9'], start=1):  
    x[0].value = f'=A{idx}*B{idx}'
```

문자열 형식으로 수식을 작성한다

```
for x in sheet['C1:C9']:  
    print(x[0].value)
```

```
wb.save('gugu_2dan.xlsx')
```

	A	B	C
1	2	1	2
2	2	2	4
3	2	3	6
4	2	4	8
5	2	5	10
6	2	6	12
7	2	7	14
8	2	8	16
9	2	9	18

▪ 수식에서 A1 은 셀 값을 참조하는 것이다. 즉, '=A1\*A2' 는 A1값 \* B1값이 된다.

# 1-06. 시트와 셀의 정보 얻기



1-06 실습) 시트와 셀의 정보를 나타내는 속성을 살펴 보자

	A	B	C	D
1	2	1	2	
2	2	2	4	
3	2	3	6	
4	2	4	8	
5	2	5	10	
6	2	6	12	
7	2	7	14	
8	2	8	16	
9	2	9	18	
10				

`sheet.max_column` → 3

`cell = sheet['C5']`

`cell = sheet['C5']` → Cell 타입

`cell.coordinate` → 'C5'

`cell.column` → 'C'

`cell.row` → 5

`cell.col_idx` → 3

`sheet.max_row` → 9

`cell.row`, `cell.column` : read/write 가능  
나머지 속성은 read only 임



# 1-07. 수식 작성연습



1-07 실습) 다음과 같은 결과가 되도록 수식을 작성한다

scores.xlsx 파일을 사용하며, cell 에는 수식이 적혀 있고, 보일 때는 수식의 결과가 표시된다

	A	B	C	D	E	F	G	H	I
1	number	name	kor	eng	mat	total	average	grade	
2	1	yoona	100	100	100	300	100	A	
3	2	kim	90	50	70	210	70	C	
4	3	park	80	95	90	265	88.33333	B	
5	4	choi	100	80	90	270	90	A	
6	5	lee	70	60	100	230	76.66667	C	
7	6	happy	50	60	50	160	53.33333	F	
8	7	jim	65	70	68	203	67.66667	D	

	A	B	C	D	E	F	G	H	I
1	score	0	60	70	80	90			
2	grade	F	D	C	B	A			
3									

total :=SUM(C2:E2), average :=AVERAGE(C2:E2)

grade :=IFERROR(HLOOKUP(G2, table\_data!\$B\$1:\$F\$2, 2, TRUE), "X")

## 2-01. 행/열 삽입, 삭제, 범위 이동, 병합 함수



🍀 행/열 삽입 및 삭제, 범위의 이동, 병합, 병합해제 관련 함수 및 예시이다

함수	예시
sheet.delete_rows(행번호, [연속행의 수]) sheet.delete_cols(열번호, [연속열의 수]) sheet.insert_rows(행번호, [연속행의 수]) sheet.insert_cols(열번호, [연속열의 수])	<ul style="list-style-type: none"><li>▪ sheet.delete_rows(3) # 3 행 삭제</li><li>▪ sheet.delete_rows(6,3) # 6:8 행 삭제</li><li>▪ sheet.delete_cols(3) # C 열 삭제</li><li>▪ sheet.delete_cols(6,3) # F:H 열 삭제</li></ul> ➔ 삽입도 동일한 방식으로 동작한다
sheet.move_range(범위, cols=정수, rows=정수)	<ul style="list-style-type: none"><li>▪ sheet.move_range('C1:F4', cols=-1, rows=3)</li></ul> ➔ [C1:F4]의 내용이 [B4:D7]로 이동된다
sheet.merge_cells(범위)	<ul style="list-style-type: none"><li>▪ sheet.merge_cells('A1:B5')</li></ul> ➔ 가장 왼쪽 상단의 내용만 유지된다
sheet.unmerge_cells(범위)	<ul style="list-style-type: none"><li>▪ sheet.unmerge_cells('A1:B5')</li></ul> ➔ 병합 해지가 불가능하면 Error 발생
sheet.merged_cells.ranges	<ul style="list-style-type: none"><li>▪ 병합된 CellRange의 list</li><li>▪ CellRange의 str()을 구하면 영역문자열이 반환됨 예) 'A1:A5'</li></ul>

## 2-02. 열 삭제 실습



실습 2-02) 다음 시트에서 B, E, F, G 열을 삭제하고 싶다면?

simpleTable.xlsx

	A	B	C	D	E	F	G	H	I
1	1	2	3	4	5	6	7	8	9
2	1	2	2	3	4	5	6	4	5
3	1	2	2	3	4	4	4	4	5

simpleTable\_after.xlsx

	A	B	C	D	E
1	1	3	4	8	9
2	1	2	3	4	5
3	1	2	3	4	5

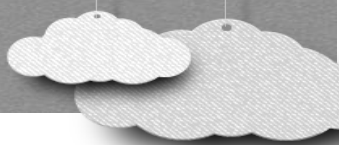
삭제/삽입 수행 후에는 행/열 번호가 변경 되기 때문에 주의 해야 한다

- B 열을 삭제하면, 뒤에 있던 C 열이 B열이 된다

B, E, F, G를 삭제하기 위한 방법은 다음과 같다

- 2번 삭제 후, 4:6을 삭제하거나, 2번 삭제 후, 4번을 세 번 삭제 한다

## 2-03. 모든 병합 해제하기



2-03 실습) 다음 시트에서 모든 병합된 셀을 해제하라

mergedTable.xlsx

	A	B	C
1	City	Name	Phone
2	Seoul	Julie	24371239
3			105426509
4		Sam	27987094
5			102345678
6	Suwon	James	312132345
7			102987123
8	Jeju	Sujan	648971239

mergedTable\_after.xlsx

	A	B	C
1	City	Name	Phone
2	Seoul	Julie	24371239
3	Seoul	Julie	105426509
4	Seoul	Sam	27987094
5	Seoul	Sam	102345678
6	Suwon	James	312132345
7	Suwon	James	102987123
8	Jeju	Sujan	648971239

해제한 뒤에는 같은 내용으로 병합되었던 셀을 채우기 한다

- sheet.merged\_cells 의 각 항(CellRange object)에 대해서 str()을 취한 것을 x라고 할 때,
- sheet.unmerged\_cells( x ) 하게 되면 병합이 해제 된다
- CellRange 객체는 str()을 취하면 범위가 문자열로 반환된다 (예 'A1:B5')



## 2-04. 활용 예제 - 형식 변환



2-04 실습) 다음 형식의 엑셀 문서를 읽어 예시와 같이 변환하여 저장 하라

sample.xlsx

	A	B	C
1	포유류	발굽포유동물	돼지, 염소, 양, 젖소, 말
2		식충포유동물	고슴도치, 두더지, 땃쥐
3		육식포유동물	북극곰, 호랑이, 사자, 표범, 족제비
4		해양포유동물	바다표범, 바다사자, 돌고래, 범고래
5		설치/토끼류	다람쥐, 청설모, 기니피그, 토끼, 호저
6		유대류	코알라, 주머니쥐, 캥거루
7		영장류	개코원숭이, 침팬지, 여우원숭이, 오랑우탄

sample\_after.xlsx

	A	B	C
1	포유류	발굽포유동물	돼지
2	포유류	발굽포유동물	염소
3	포유류	발굽포유동물	양
4	포유류	발굽포유동물	젖소
5	포유류	발굽포유동물	말
6	포유류	식충포유동물	고슴도치
7	포유류	식충포유동물	두더지
8	포유류	식충포유동물	땃쥐
9	포유류	육식포유동물	북극곰
10	포유류	육식포유동물	호랑이
11	포유류	육식포유동물	사자
12	포유류	육식포유동물	표범
13	포유류	육식포유동물	족제비

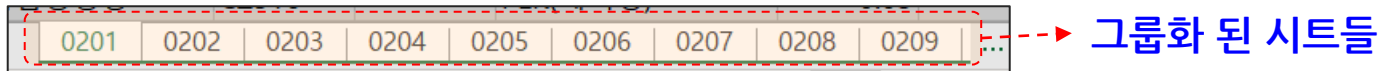
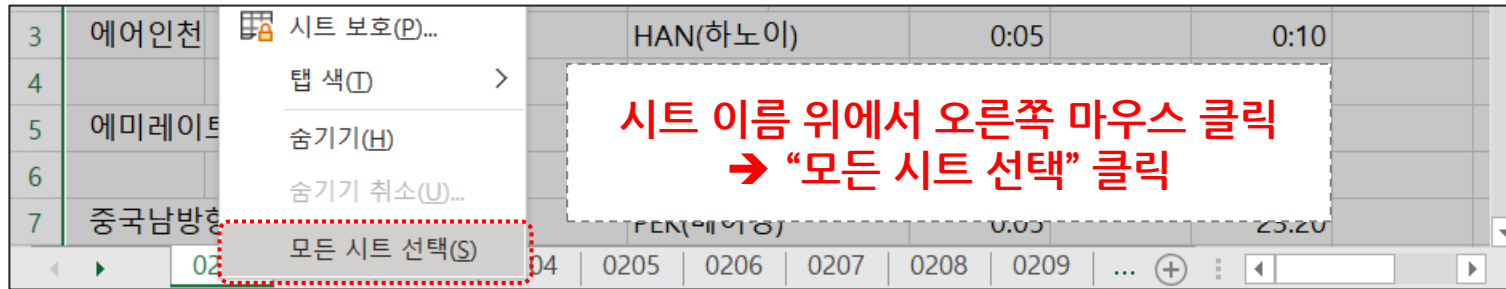
## 2-05. 여러 시트에 동일 작업하기 1/2



여러 시트에 동일 작업을 하기 위해서는 '시트 그룹'을 사용한다

./flight 폴더에 있는 파일의 모든 시트 첫 행에 필드 명을 추가하는 작업을 수행한다

1) 엑셀 문서를 열어 모든 시트에 대해 '시트 그룹'을 만든다



2) 첫 행을 삽입하고, '항공사', '편명', '출발지', '계획', '예상', '도착', '구분', '현황'을 입력한다

	A	B	C	D	E	F	G	H	I	J
1	항공사		편명		출발지		계획		예상	
2	산동항공		SC4722		TAO(청도)		0:05		23:50	

## 2-05. 여러 시트에 동일 작업하기 2/2



여러 시트에 동일 작업을 하기 위해서는 '시트 그룹'을 사용한다

3) [Ctrl]키를 누른 상태에서 'B', 'D', 'F', 'H', 'J', 'L', 'M' 열 이름을 클릭한다 (삭제할 열 선택)

	A	B	C	D	E	F	G	H	I	J	K
1	항공사		편명		출발지		계획		예상		도착
2	산동항공		SC4722		TAO(청도)		0:05		23:50		0:08
3											
4	에어인천		KJ511		HAN(하노이)		0:05		0:10		0:35

4) 열 이름 위에서 오른쪽 마우스 클릭 후, '삭제'를 클릭한다

5) 시트 이름 위에서 오른쪽 마우스 클릭 후, '시트 그룹 해제'를 클릭하여 그룹을 해제한다

6) Ctrl + S를 눌러 저장한다

- 간단한 작업으로 보이지만, 이런 작업을 할 분량이 많다면 어떨까?
- 불필요한 행을 삭제하기 위해 실수 없이 모든 행을 선택할 수 있을까?  
(선택해야 하는 행의 수는 시트당 500 개 이상이다)

## 2-05. openpyxl을 이용한 작업 자동화



### 2-05 실습1) 불필요한 열 제거 및 첫 행을 삽입하는 프로그램을 작성한다

- ../flight/인천도착\*.xlsx 의 모든 시트에 대해서 작업하도록 한다

- removeCols(시트명, 삭제할 열의 수)

../flight/인천도착XX.xlsx

상위

	A	B	C	D	E	F	G	H	I
1	칼리타화물항공	K42677		JFK(뉴욕)		0:05		0:29	
2									
3	비엠텔트	VJ874		DAD(다낭)		0:20		0:05	
4									
5	비엠텔트	VJ838		CXR(나트랑)		0:40		0:24	

../flight/step1/인천도착XX.xlsx

	A	B	C	D	E	F	G	H	I
1	항공사	편명	출발지	계획	예상	도착	구분	현황	첫 행 삽입
2	칼리타화	K42677	JFK(뉴욕)	0:05	0:29	0:29	화물	도착	
3									
4	비엠텔트	VJ874	DAD(다낭)	0:20	0:05	0:05	여객	도착	
5									



## 2-05. openpyxl을 이용한 작업 자동화



### 2-05 실습2) 불필요한 열 제거 및 첫 행 삽입 프로그램 성능을 분석해보자

```
rdpath = '../flight/'
wtpath = '../flight/step1/'
stime = datetime.datetime.now()
for filename in glob.glob(rdpath+'인천도착*.xlsx'):
    wb = openpyxl.load_workbook(filename)
    for sname in wb.sheetnames:
        sheet = wb[sname]
        removeCols(sheet, sheet.max_column//2 + 1)
        #removeRows(sheet, sheet.max_row//2 + 1)
        insertTitle(sheet, title)
    name = os.path.basename(filename)
    print(name)
    wb.save(wtpath+name)
etime = datetime.datetime.now()
print(etime - stime)
```

인천도착1월.xlsx  
인천도착2월.xlsx  
인천도착3월.xlsx  
인천도착4월.xlsx  
인천도착5월.xlsx  
인천도착6월.xlsx  
인천도착7월.xlsx  
인천도착8월.xlsx  
0:01:59.732514

- removeRows() 작업을 함께 수행하면 시간이 얼마나 걸리는지 1월만 수행해 검토 해 보자

인천도착1월.xlsx  
0:03:38.160477

- 1개 파일에 대해 작업하는데 3분 38초 정도의 시간이 걸렸다(openpyxl은 대용량에 부적합)  
→ 불필요한 행의 제거는 pandas를 사용하여 작업하는 것이 효율적이다