```
import openpyxl
from openpyxl import Workbook


workbook1 = openpyxl.load_workbook("C:/Users/ahnssi9/Desktop/111.xlsx")
print(workbook1.sheetnames)
Sheet1 = workbook1['Sheet1']
workbook1.remove(Sheet1)
workbook1.save("C:/Users/ahnssi9/Desktop/111.xlsx")
print(workbook1.sheetnames)


workbook1.create_sheet("Sheet1")
workbook1.save("C:/Users/ahnssi9/Desktop/111.xlsx")
print(workbook1.sheetnames)
========================================
C:\Users\ahnssi9\PycharmProjects\pythonProject\venv\Scripts\python.exe
C:/Users/ahnssi9/PycharmProjects/pythonProject/t2.py
['Sheet2', 'Sheet3', 'Sheet5', 'Sheet1']
['Sheet2', 'Sheet3', 'Sheet5']
['Sheet2', 'Sheet3', 'Sheet5', 'Sheet1']

Process finished with exit code 0
========================================================================
import openpyxl
from openpyxl import Workbook
import os
import os.path


graphfile = openpyxl.load_workbook("C:/Users/ahnssi9/Desktop/graphfile22.xlsx")
print(graphfile.sheetnames)
graphSheet = graphfile['Sheet1']
graphfile.remove(graphSheet)
graphfile.save("C:/Users/ahnssi9/Desktop/graphfile22.xlsx")
print(graphfile.sheetnames)


graphfile.create_sheet("Sheet1")
graphfile.save("C:/Users/ahnssi9/Desktop/graphfile22.xlsx")
print(graphfile.sheetnames)
```

https://openpyxl.readthedocs.io/en/stable/api/openpyxl.workbook.workbook.html

del workbook1['Sheet1']

```
import openpyxl as xl
wb = xl.load_workbook()
ws = wb.active
ws.delete_rows(firstrow, numberofrows) #for multiple row deletion
ws.delete_rows(1, ws.max_row + 1) #
for entire sheet
ws.delete_rows(rownum) #for single row
```

==========

remove(*worksheet*) [source]

Remove *worksheet* from this workbook.

remove_named_range(*named_range*)[source]

Remove a named_range from this workbook.

Deprecated: Use del workbook.defined_names[name]

remove_sheet(*worksheet*)[source]

Remove *worksheet* from this workbook.

Deprecated: Use wb.remove(worksheet) or del wb[sheetname]

==========

https://doitnow-man.tistory.com/159

python으로 엑셀 다루기

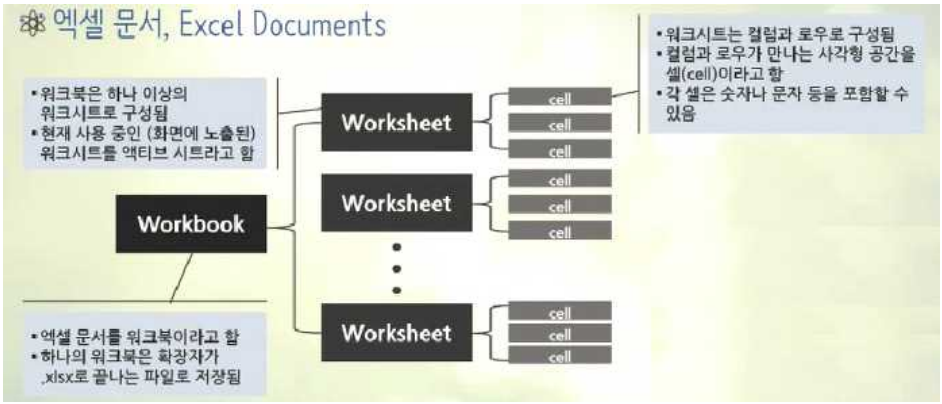
2018. 11. 22. 00:17

python, 엑셀, 엑셀 생성, 엑셀 쓰기

[python] 17. python으로 엑셀 다루기

1. 엑셀의 구조

1) workbook
   - 엑셀 파일을 의미
2) worksheet
   - 엑셀의 sheet를 의미한다.
3) cell
   - 엑셀 sheet의 셀 하나 하나를 의미한다.



2. 엑셀관련 모듈
   - 엑셀 관련 모듈은 여러개가 있다.
   1) xlwt
   2) OpenPyXL(이게 가장좋음)
      - http://openpyxl.readthedocs.org
      - 문서화가 제일 잘되어 있다
        (1) 지원 기능
          - 로컬에 엑셀 프로그램이 설치되어 있지 않아도 엑셀 파일 생성과 일기가 가능
          - 대요량 지원, 이미지원 등
   3) XlsxWriter
   4) PyExcelerate

3. OpenPyXL 설치
   1) 설치
      - pip install openpyxl

4. OpenPyXL sheet 열기 사용법
   1) import 모듈명
      - openpyxl

   2) 파일 열기
      - openpyxl.load_workbook('파일명')

Ex) wb = openpyxl.load_workbook('./test.xlsx')

3) 파일 닫기
- close()
Ex) wb.close()

3) sheet 열기
(1) index를 sheet이름으로하여 찾기(이게 더 좋다)
- wb['sheet1']

(2) get_sheet_by_name('Sheet1') 함수 이용(비 추천 pychame에서 경고 발생)
- sheet = wb.get_sheet_by_name('Sheet1')

4) 현재 열린 sheet를 열기
- wb.actvie
Ex) sheet = wb.actvie

5. OpenPyXL cell 조작 사용법
1) cell 접근 방법
- sheet = wb['Sheet1'] 를 통해 우선 sheet 정보를 얻는다.
(1) cell의 index 를 이용한 방법
>> A1 = sheet['A1']
>> A1.value
Sheet1 A1 value
(2) cell()함수를 이용
- 파라미터로 row와 column을 사용한다.
- 마지막 값이 존재하는 셀(row) : sheet.max_row
- 마지막 값이 존재하는 셀(column) : sheet.max_column
Ex)
>> A1 = sheet.cell(row = 1, column = 1)
Sheet1 A1 value
(3) 전체 예제 코드

```
import openpyxl

wb = openpyxl.load_workbook('./test.xlsx')
sheet = wb['Sheet1']
print(sheet.cell(row=1, column=1).value)
print("%s" % sheet.max_column)
```

2) 여러 cell 접근 하기
(1) 특정 범위
- cell_range = sheet['A1':'C2']

(2) 특정 row
- row10 = sheet[10]
(3) 특정 row 범위
- row_range = sheet[5:10]
(4) 특정 Column
- colC = sheet['C']
(5) 특정 Column 범위
- col_range = sheet['C:D']

Ex) 예제 코드
- 전체 컬럼과 로우에 접근하느 코드

6. OpenPyXL 엑셀파일 생성
1) workbook 생성
- wb = openpyxl.Workbook() # workbook 함수을 통해 workbook 객체 생성
생성시 기본적으로 sheet1이 생성된다.
2) sheet 생성
- sheet2 = wb.create_sheet('sheet2') #마지막에 추가
- sheet3 = wb.create_sheet('sheet3', 1) #sheet1 자리에 삽입 하여 추가

3) sheet 이름 변경
- sheet2.title = '업무자동화'

4) workbook 저장
- wb.save('저장할 파일이름.xlsx') #워크북에 저장
- wb.close() #파일 닫기

Ex) 전체 코드

```
import openpyxl

wb = openpyxl.Workbook()
sheet2 = wb.create_sheet('sheet2')
sheet3 = wb.create_sheet('sheet3', 1)
sheet2.title = '업무자동화'
wb.save('./new_test_file.xlsx')
wb.close()
```

7. OpenPyXL 엑셀파일에 쓰기

1) index를 이용하여 cell에 데이터 쓰기
- sheet['A1'] = 'hello' #A1에 쓰기

2) cell()함수를 이용하여 데이터 쓰기
- sheet.cell(row = 1, column = 1, value='wolrd') # A1에 값쓰기

* wb.save('./new_test_file.xlsx')를 사용하여 저장을 해야 완전히 반영된다.

8. OpenPyXL에서 지원하는 기능?
- 거의 다 지원함으로 메뉴얼을 보시기 바랍니다. (https://openpyxl.readthedocs.io/en/stable/)