

Python에서 pywin32 엑셀 모듈 다루기 Excel - pywin32

2015.05.18. 18:25

<http://blog.naver.com/allenpython/220363265664> sample.csv

파이썬에서 엑셀을 다루는 여러가지 방법 중에는 그래도 Excel에서 제공하는 모든 기능을 사용할 수 있고, 제약이 다른 모듈에 비해서 적은 pywin32 모듈을 사용하기로 마음 먹었다. 그 이유는 pywin32를 사용하면, 기존에 VBA에서 사용하던 Excel에서 제공하는 문법, 기능 등을 그대로 사용할 수 있기 때문이다. 나중에 VBA에서 사용하기에도 훨씬 편할 것이라는 기대감을 포함해서.

아래는 pywin32의 모듈을 이용해서 excel 파일을 하나 열어 cell에서 계산을 하고, 그 결과를 새로운 파일에 저장하는 것이다. 여기서는 일반적인 .xls 파일이나 .xlsx 파일을 사용하지 않고, .csv 파일을 이용했다.

여기서 샘플 코드는 첨부되어 있다. sample.csv.

Column A와 Column B에 각각 Data1과 Data2가 있고, Row 는 총 89개이다.

----- Code -----

```
import win32com.client as win32
excel = win32.gencache.EnsureDispatch("Excel.Application")

wb = excel.Workbooks.Open("C:\\Users\\AllenKim\\workspace\\Lab1\\sample.csv") # point the sample.csv location
ws = wb.Worksheets(1)

excel.Visible = True

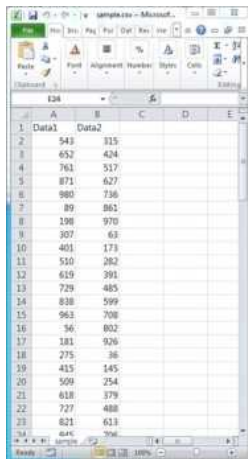
ws.Cells(1, 3).Value = "Plus"

for i in range(2, 100): # row is total 89. So, 100 contains the all row data.
    if ws.Cells(i, 1).Value == None :
        break
    else :
        ws.Cells(i, 3).Value = ws.Cells(i, 1).Value + ws.Cells(i, 2).Value

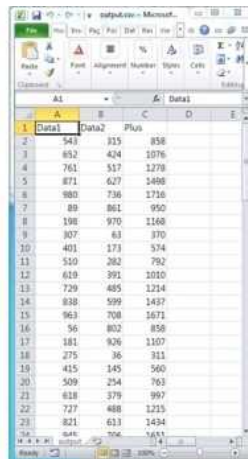
excel.Application.DisplayAlerts = False # save as requires the confirmation.
wb.SaveAs("C:\\Users\\AllenKim\\workspace\\Lab1\\output.csv")
wb.Close()
excel.Application.DisplayAlerts = True
excel.Application.Quit()

-----
```

위의 코드를 실행시키면, output.csv 파일이 생성되고, 아래 사진의 오른쪽 그림처럼 Column C에 Data1과 Data2의 같은 열에 해당하는 두 값의 합이 Column C에 존재하는 것을 확인할 수 있다.



	A	B	C	D	E
1	Data1	Data2			
2	543	315			
3	852	424			
4	761	517			
5	871	627			
6	980	736			
7	89	861			
8	198	970			
9	307	63			
10	401	173			
11	510	282			
12	619	391			
13	729	485			
14	838	599			
15	963	708			
16	56	802			
17	181	926			
18	275	36			
19	415	145			
20	509	254			
21	618	379			
22	727	488			
23	821	613			
24	641	706			

	A	B	C	D	E
1	Data1	Data2	Plus		
2	543	315	858		
3	852	424	1276		
4	761	517	1278		
5	871	627	1498		
6	980	736	1716		
7	89	861	950		
8	198	970	1168		
9	307	63	370		
10	401	173	574		
11	510	282	792		
12	619	391	1010		
13	729	485	1214		
14	838	599	1437		
15	963	708	1671		
16	56	802	858		
17	181	926	1107		
18	275	36	311		
19	415	145	560		
20	509	254	763		
21	618	379	997		
22	727	488	1215		
23	821	613	1434		
24	641	706	1347		