

work-10

Create a new report based on columns comparison from 2 excel sheets

Test1 file:

test1 - Microsoft Excel

	A	B	C	D
1	App_Names	quality	secure	
2	App1	good	no	
3	App2	bad	yes	
4	App3	good	yes	
5	App4	good	no	
6	App5	bad	yes	
7	App6	good	yes	
8	App7	bad	yes	
9	App8	good	no	
10	App9	good	no	
11	App10	bad	yes	
12	App11	good	yes	
13	App12	bad	yes	
14	App13	Good	yes	
15				
16				
17				

• Test2 file:

test2 - Microsoft Excel

FILE

HOME

INSERT

PAGE LAYOUT

FORMULAS

Developer

DATA

REVIEW

VIEW

ADD-INS

Foxit PDF

TEAM

xlwir

D3

X

✓

fx

	A	B	C	D	E	F	G
1	App_Names	Name_Of_User	GROUP				
2	App1	Alice	1				
3	App1	John	1				
4	App1	Bob	1				
5	App2	Ben	2				
6	App2	Michael	2				
7	App2	Alice	2				
8	App3	kevin	1				
9	App4	james	2				
10	App4	Lucas	2				
11	App4	Ethan	2				
12	App4	Aiden	2				
13	App5	john	1				
14	App6	Jackson	2				
15	App6	David	2				
16	App6	william	2				
17	App7	christine	2				
18	App8	vagan	1				
19	App8	drake	1				
20	App9	jennifer	2				
21	App10	Henry	1				
22	App11	Joseph	1				
23	App12	Anthony	1				
24	App13	Isaac	1				
25							
26							

←

→

Sheet1

+

READY

Installation

- Download and Install [Anaconda](#)
- NOTE: Don't forget to tick the checkbox corresponding to "Add to path". This will enable using `conda` in the terminal.
- Open the terminal and check for following packages

```
pandas
numpy
```

- If not found, then run these 2 commands in terminal:
 - `pip install pandas`
 - `pip install numpy`

Coding

Modules

- Import packages: Pandas, numpy

```
import pandas as pd
import numpy as np
```

- Define dataframe - df1

```
df1 = pd.ExcelFile("data/test1.xlsx").parse("Sheet1")
```

- Define dataframe - df2

```
df2 = pd.ExcelFile("data/test2.xlsx").parse("Sheet1")
```

- Define dataframe - df3

```
df3 = df2
```

- Insert 2 columns - quality & secure in df3

```
df3.insert(1, column= "quality", value= np.nan)      # position 1: 2nd column
df3.insert(2, column= "secure", value= np.nan)       # position 2: 3rd column
```

- Run a rule in df3 so that the corresponding values of columns - quality & secure is filled.

```
# define 'App_Names' list for the `for` loop
list_app_names = df1['App_Names'].tolist()

"""
Description
=====
for loop introduced to apply the rules in the entire series ['App1', ..... 'App13']

For single `App1`:
-----
df3.loc[df3.App_Names == 'App1', ['quality', 'secure']] = [df1.iloc[0,1], df1.iloc[0,2]]
"""
for i in range(len(list_app_names)):
    df3.loc[df3.App_Names == list_app_names[i], ['quality', 'secure']] = [df1.iloc[i,1], df1.iloc[i,2]]
```

- print the df3 to a NEW output.xlsx

NOTE: mark the index as false

```
df3.to_excel("output.xlsx", index= False)
```

Execution

There are 2 ways to run this:

- **M-1: Unix OS** - run the `run.sh`
- **M-2: Windows OS** - run the `run.bat`

Output

App_Names										
	A	B	C	D	E	F	G	H	I	J
1	App_Names	quality	secure	Name_Of_User	GROUP					
2	App1	good	no	Alice	1					
3	App1	good	no	John	1					
4	App1	good	no	Bob	1					
5	App2	bad	yes	Ben	2					
6	App2	bad	yes	Michael	2					
7	App2	bad	yes	Alice	2					
8	App3	good	yes	kevin	1					
9	App4	good	no	james	2					
10	App4	good	no	Lucas	2					
11	App4	good	no	Ethan	2					
12	App4	good	no	Aiden	2					
13	App5	bad	yes	john	1					
14	App6	good	yes	Jackson	2					
15	App6	good	yes	David	2					
16	App6	good	yes	william	2					
17	App7	bad	yes	christine	2					
18	App8	good	no	vagan	1					
19	App8	good	no	drake	1					
20	App9	good	no	jennifer	2					
21	App10	bad	yes	Henry	1					
22	App11	good	yes	Joseph	1					
23	App12	bad	yes	Anthony	1					
24	App13	Good	yes	Isaac	1					
25										
26										

Sheet1

READY