



TRY IT!

**EASY PYTHON CODE TO
CLEAN UP CSV FILES**

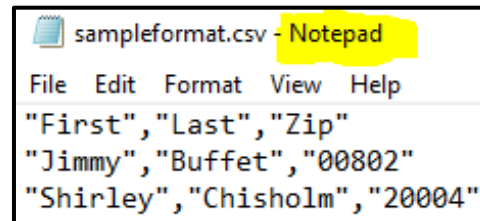
© “PYPANCSV” 2019

TODAY = "MAGIC TRICKS" DEMO



CSV \approx Spreadsheet (XLS)

- Text-editor-friendly
- No formatting
- Database export/import



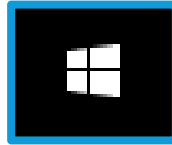
```
sampleformat.csv - Notepad
File Edit Format View Help
"First","Last","Zip"
"Jimmy","Buffet","00802"
"Shirley","Chisholm","20004"
```

- “Table-shaped” data, so Excel often easy
- But sometimes not ...

	A	B	C
1	First	Last	Zip
2	Jimmy	Buffet	802
3	Shirley	Chisholm	20004

How I Found Python

	A	B	C	D
1	Id	First Name	Last Name	Program Registered For
2	29	John	Doe	BasketWeaving
3	29	John	Doe	ScubaDiving
4	872	Jane	Dill	ScubaDiving
5	872	Jane	Dill	Acrobatics
6	872	Jane	Dill	ScubaDiving
7	75	Mick	Jag	ComputerProgramming



```

import numpy
df4['Program Registered For'] = 'Prg_' + df4['Program Registered For']
non_program_columns = list(filter(lambda x: x != 'Program Registered For', df4.keys()))
pivotdf = pandas.pivot_table(df4, index=non_program_columns, columns='Program Registered For', aggfunc=numpy.size)
pivotdf[pandas.notnull(pivotdf)] = 'Registered'
pivotdf.reset_index(inplace=True)
pivotdf.to_csv('C:\\vay\\out_pivoted_program_registrations.csv', index=False, quoting=1)
    
```

	A	B	C	D	E	F	G
1	Id	First Name	Last Name	Prg_Acrobatics	Prg_BasketWeaving	Prg_ComputerProgramming	Prg_ScubaDiving
2	29	John	Doe		Registered		Registered
3	75	Mick	Jag			Registered	
4	872	Jane	Dill	Registered			Registered

DEEP END OF THE POOL



Vocab

- **Python:** programming language
- **Pandas:** module (plugin) for Python
 - Adds CSV-related commands
- You can run Python programs in an IDE
 - **IDE** \approx code-editing software with a run button
 - Online IDEs = repl.it and codebunk.com.
 - ALWAYS USE FAKE DATA!

Today's Data

sample1.csv						sample2.csv					
7 rows, 5 columns (people & employer)						6 rows, 5 columns (people & favorite food)					
	A	B	C	D	E		A	B	C	D	E
1	Id	First	Last	Email	Company	1	PersonId	FirstName	LastName	Em	FavoriteFood
2	5829	Jimmy	Buffet	jb@example.com	RCA	2	983mv	Shirley	Temple	st@example.com	Lollipops
3	2894	Shirley	Chisholm	sc@example.com	United States Congress	3	9e84f	Andrea	Smith	as@example.com	Kale
4	294	Marilyn	Monroe	mm@example.com	Fox	4	k28fo	Donald	Duck	dd@example.com	Pancakes
5	30829	Cesar	Chavez	cc@example.com	United Farm Workers	5	x934	Marilyn	Monroe	mm@example.com	Carrots
6	827	Vandana	Shiva	vs@example.com	Navdanya	6	8xi	Albert	Howard	ahotherem@example.com	Potatoes
7	9284	Andrea	Smith	as@example.com	University of California	7	02e	Vandana	Shiva	vs@example.com	Amaranth
8	724	Albert	Howard	ah@example.com	Imperial College of Science						

Victim #1: Get Us Ready

1. Open a web browser and go to:

<https://repl.it/@rplrpl/40-Minute-Semi-Hands-On-Starter-Code>

2. In the last line of code, type out these 5 lines, using a tab to indent 2 & 3, and hitting backspace if necessary to un-indent line 5:

```
def p(input):  
    print(input)  
    print(' ---D I V I D E R---' )  
  
p(' Hel l o Worl d' )
```

3. Hit “run” (top center; green button)

Student #2: Load & Display a CSV

1. Put a “#” before “`p('Hello World')`” so that instead, it looks like:
`#p('Hello World')`

2. Hit “enter” for a new line and type out these 3 lines:

```
df1 = pandas.read_csv(filepath1)
p(df1)
```

3. Hit “run” (top center; green button)

Student #3: Display CSV Stats

1. Put a “#” before “`p(df1)`” so that instead, it looks like this:
`#p(df1)`

2. Hit “enter” for a new line and type out these 5 lines:

```
p(len(df1))  
p(df1.columns)  
p(len(df1.columns))  
p(list(df1.columns))  
p(sorted(df1.columns, key=str.lower))
```

3. Hit “run” (top center; green button)

Student #4: Display Last Names

1. Put a pair of “'''”s on their own lines before & after stu. #3’s code, like:

```
'''  
#p(len(df1))  
p(sorted(df1.columns, key=str.lower))
```

2. Hit “enter” for a new line and type out these 6 lines:

```
lcol = df1['Last']  
p(lcol)  
p(list(lcol))  
lcol_unq = lcol.unique()  
p(lcol_unq)  
p(len(lcol_unq))
```

3. Hit “run” (top center; green button)

Student #5: Display F&L Names

1. Put a pair of “' ’”s on their own lines before & after all but the first line, “l col = df1[' Last']” (*we like that line*), in student #4’s code, like:

```
'''  
l col = df1[' Last' ]  
'''  
p(l col )  
p(len(l col unq))
```

2. Hit “enter” for a new line and type out these 6 lines:

```
fl col names = [' First' , ' Last' ]  
fl col s = df1[fl col names]  
p(fl col s)  
fcol = df1[' First' ]  
al l names = pandas.concat([l col , fcol ])  
p(sorted(al l names))
```

3. Hit “run” (top center; green button)

QUESTIONS SO FAR?

Student #6: Combine 2 Tables

1. Put a pair of “'''”s on their own lines before & after all but the first line, “lcol = df1['Last']” (*we like that line*), in student #5’s code, like:

```
'''  
lcol names = ['First', 'Last']  
p(sorted(all names))
```

2. Hit “enter” for a new line and type out these 5 lines (here, indented = part of prev. line):

```
df2 = pandas.read_csv(file path2)  
df2match =  
    df2.rename(columns=  
        {'FirstName': 'First', 'LastName': 'Last'})  
mergedf = df1.merge(df2match,  
    on=lcol names, how='outer', indicator=True)  
p(mergedf)  
p(mergedf.query('_merge != "both"))
```

3. Hit “run” (top center; green button)

Student #7: Create a CSV

1. Put a “#” before “#p(mergedf)” and “p(mergedf.query('_merge != "both"'))” so that instead, they look like:

```
#p(mergedf)
#p(mergedf.query('_merge != "both"'))
```

2. Hit “enter” for a new line and type out this 1 line:

```
mergedf.to_csv('mergeoutput.csv', index=0)
```

3. Hit “run” (top center; green button)
4. Open up the new “mergeoutput.csv” in the file list at left.

QUESTIONS?

ANY POTENTIAL FOR YOU?

If We Finish Early...

- What would you like to do, business-wise, to our sample spreadsheets?
- Describe/Doodle what the sample output would look like
- We'll live-code together.

	A	B	C	D	E
1	Id	First	Last	Email	Company
2	5829	Jimmy	Buffet	jb@example.com	RCA
3	2894	Shirley	Chisholm	sc@example.com	United States Congress
4	294	Marilyn	Monroe	mm@example.com	Fox
5	30829	Cesar	Chavez	cc@example.com	United Farm Workers
6	827	Vandana	Shiva	vs@example.com	Navdanya
7	9284	Andrea	Smith	as@example.com	University of California
8	724	Albert	Howard	ah@example.com	Imperial College of Science

	A	B	C	D	E
1	PersonId	FirstName	LastName	Em	FavoriteFood
2	983mv	Shirley	Temple	st@example.com	Lollipops
3	9e84f	Andrea	Smith	as@example.com	Kale
4	k28fo	Donald	Duck	dd@example.com	Pancakes
5	x934	Marilyn	Monroe	mm@example.com	Carrots
6	8xi	Albert	Howard	ahotherem@example.com	Potatoes
7	02e	Vandana	Shiva	vs@example.com	Amaranth

RESOURCES

- **Today's slides with code (editable/runnable online) & quizzes!**
+ “common operations & how to use them” list:

<https://tinyurl.com/pypancsv>