ESSAYEZ!

PYTHON FACILE POUR GÉRER DES FICHIERS CSV

	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

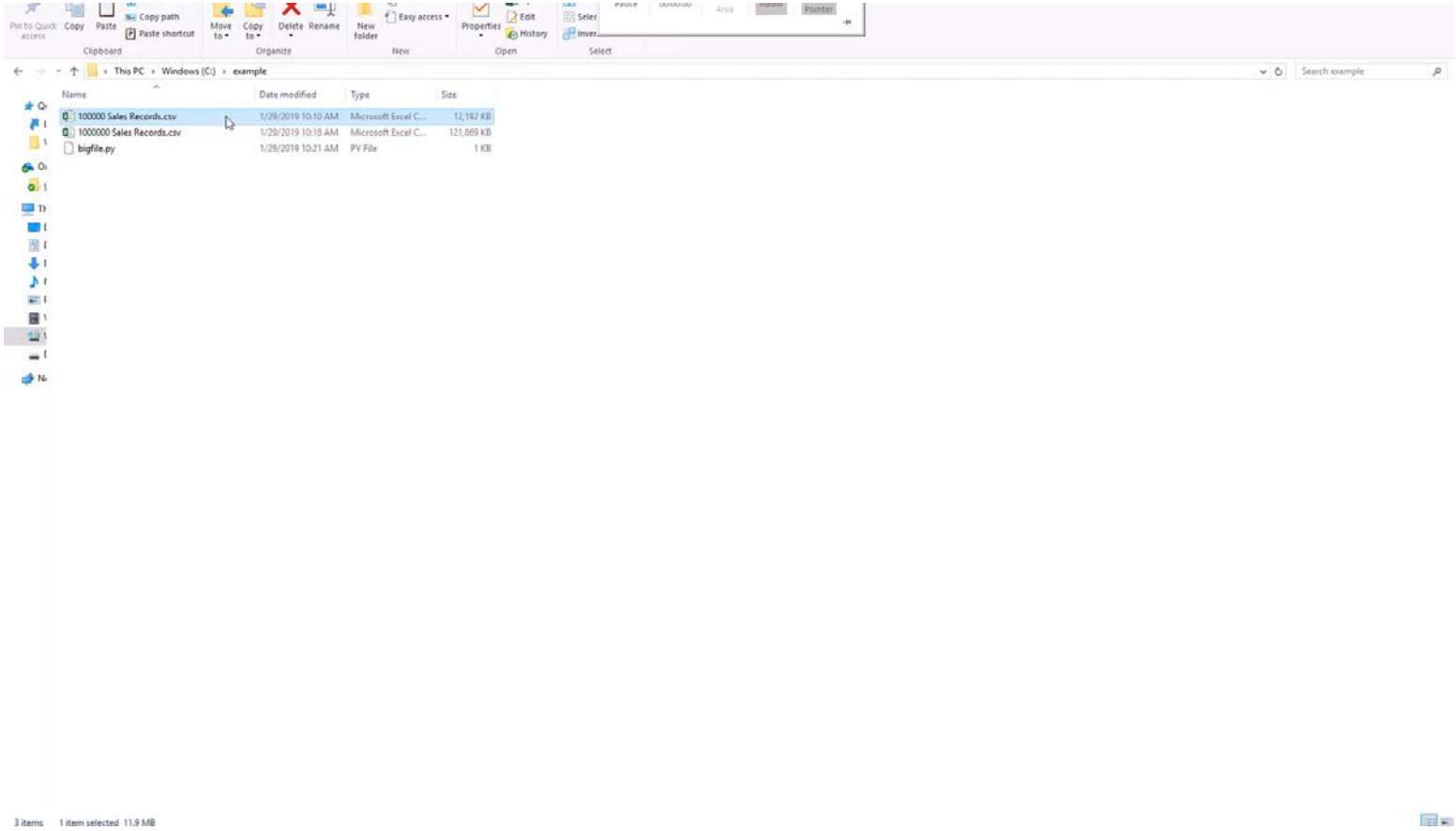
	A	B	C	D	E	F	G	H
1	**Id**	**First**	**Last**	**Email_csv1**	**Company**	**PersonId**	**Email_csv2**	**FavoriteFood**
2	5829	Jimmy	Buffet	jb@example.com	RCA			
3	2894	Shirley	Chisholm	sc@example.com	United States Congress			
4	294	Marilyn	Monroe	mm@example.com	Fox	x934	mm@example.com	Carrots
5	30829	Cesar	Chavez	cc@example.com	United Farm Workers			
6	827	Vandana	Shiva	vs@example.com	Navdanya	02e	vs@example.com	Amaranth
7	9284	Andrea	Smith	as@example.com	University of California	9e84f	as@example.com	Kale
8	724	Albert	Howard	ah@example.com	Imperial College of Science	8xi	ahotherem@example.com	Potatoes
9		Shirley	Temple			983mv	st@example.com	Lollipops
10		Donald	Duck			k28fo	dd@example.com	Pancakes

Pourquoi coder? 3 raisons.

~~Code:~~

Code:

Pourquoi Coder:
#1 – de très gros fichiers



Editor - C:\example\bigfile-fr.py

bigfile-fr.py

```
import pandas
import datetime

def prfr(x): return '{:,}'.format(x).replace(',','.')
def prfr2(x): return str(x).replace('.',',')

mauvaispays = ['Algeria','Armenia','Australia','Barbados']
df1 = pandas.read_csv('c:\\example\\100000 Sales Records.csv')
df2 = pandas.read_csv('c:\\example\\1000000 Sales Records.csv')

t1 = datetime.datetime.now()
compteavant1 = len(df1)
df1 = df1[~df1['Country'].isin(mauvaispays)]
compteapres1 = len(df1)
difference1 = compteavant1 - compteapres1
t2 = datetime.datetime.now()
tm1 = round((t2 - t1).total_seconds(), 2)

print('Il a fallu  ' + prfr2(tm1) + ' seconds pour suppprimer ' +
      prfr(difference1) + ' des ' + prfr(compteavant1) + ' enregistrements.')

t3 = datetime.datetime.now()
compteavant2 = len(df2)
df2 = df2[~df2['Country'].isin(mauvaispays)]
compteapres2 = len(df2)
difference2 = compteavant2 - compteapres2
t4 = datetime.datetime.now()
tm2 = round((t4 - t3).total_seconds(), 2)

print('Il a fallu  ' + prfr2(tm2) + ' seconds pour suppprimer ' +
      prfr(difference2) + ' des ' + prfr(compteavant2) + ' enregistrements.')
```

```
mauvaispays = ['Algeria','Armenia','Australia','Barbados']
df1 = pandas.read_csv('c:\\example\\100000 Sales Records.csv')
df2 = pandas.read_csv('c:\\example\\1000000 Sales Records.csv')

t1 = datetime.datetime.now()
compteavant1 = len(df1)
df1 = df1[~df1['Country'].isin(mauvaispays)]
compteapres1 = len(df1)
difference1 = compteavant1 - compteapres1
t2 = datetime.datetime.now()
tm1 = round((t2 - t1).total_seconds(), 2)

print('Il a fallu  ' + prfr2(tm1) + ' seconds pour suppprimer ' +
      prfr(difference1) + ' des ' + prfr(compteavant1) + ' enregistrements.')

t3 = datetime.datetime.now()
compteavant2 = len(df2)
df2 = df2[~df2['Country'].isin(mauvaispays)]
```

Variable explorer | File explorer | Help

IPython console

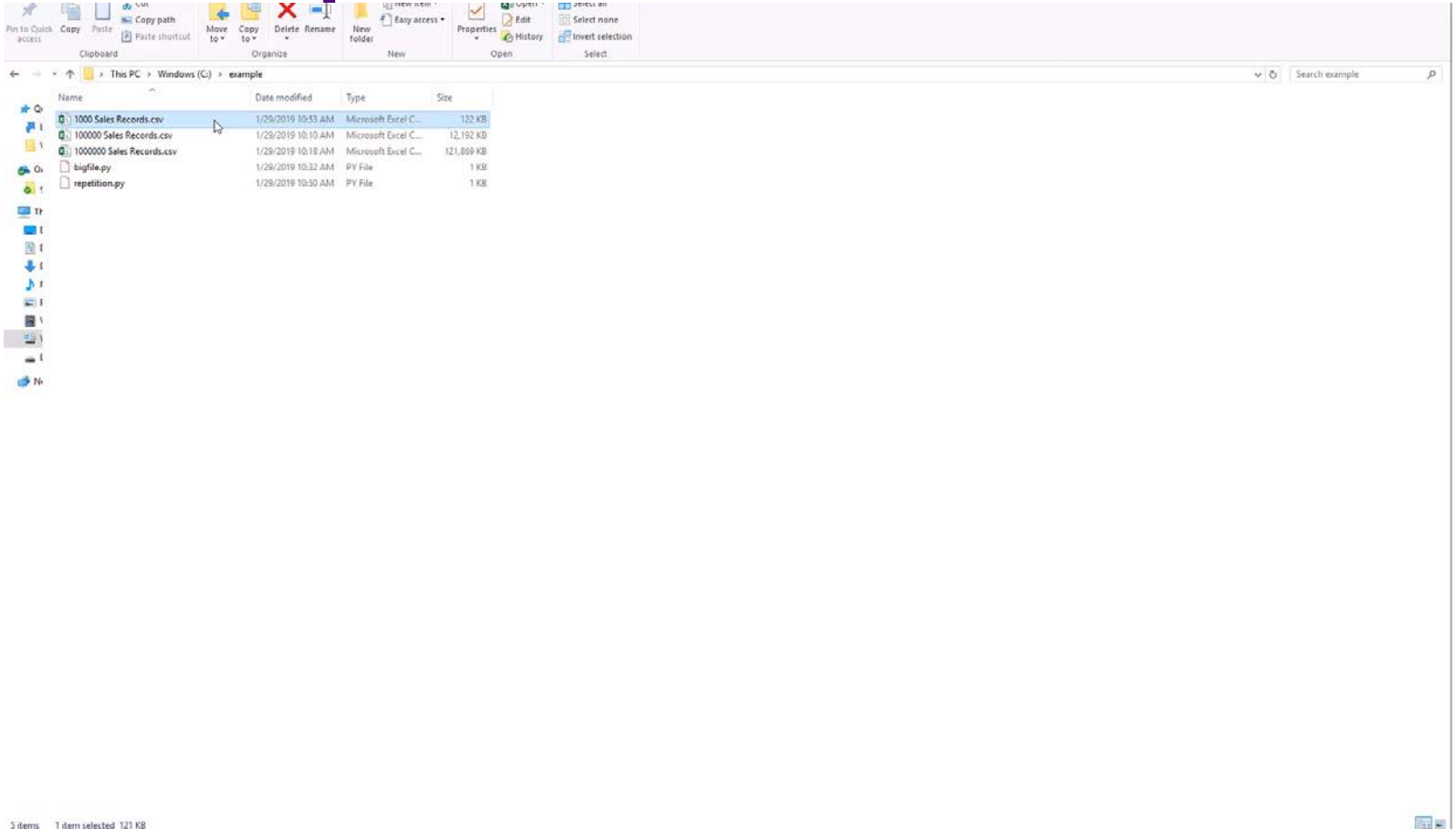
Console 1/A

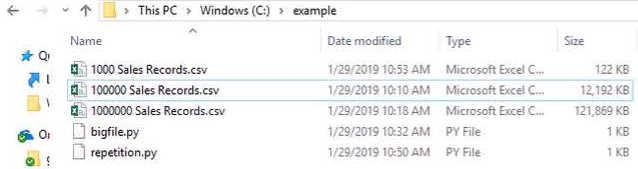
```
In [40]:
```

IPython console | History log

Permissions: RW | End-of-lines: CRLF | Encoding: ASCII | Line: 1 | Column: 1 | Memory: 47 %

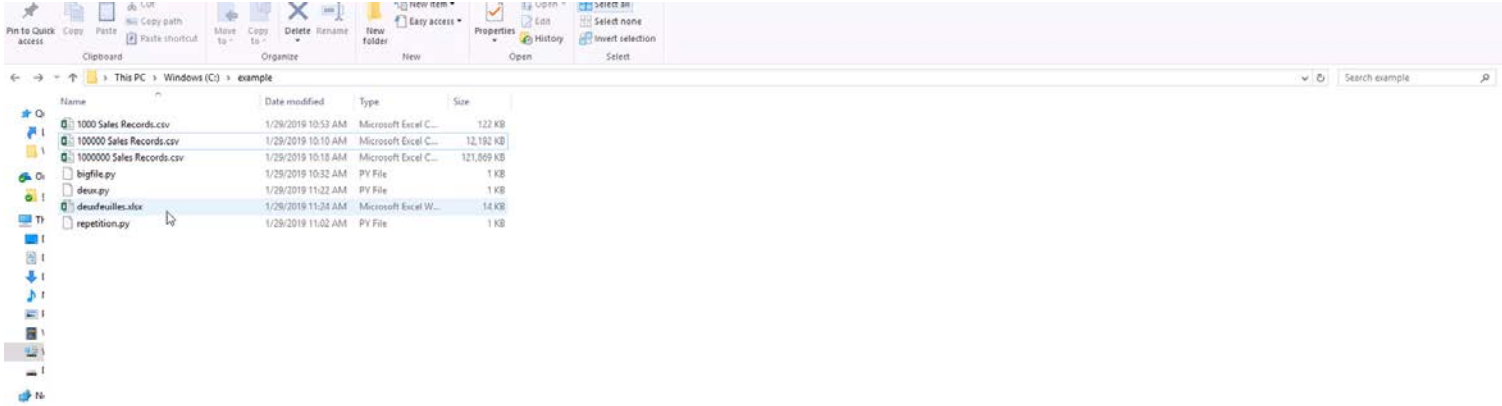
Pourquoi Coder:
#2 – la répétition répétition répétition répétition repetition

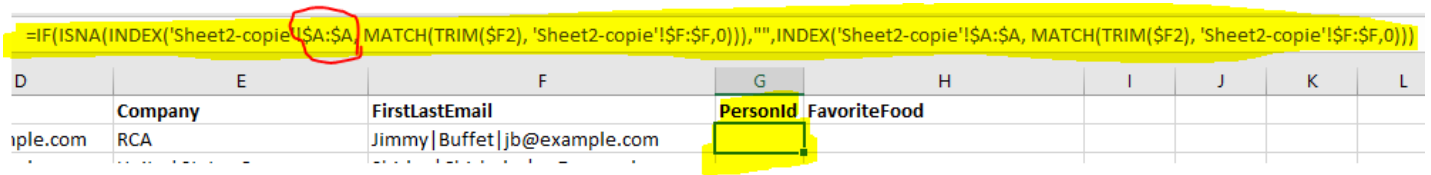


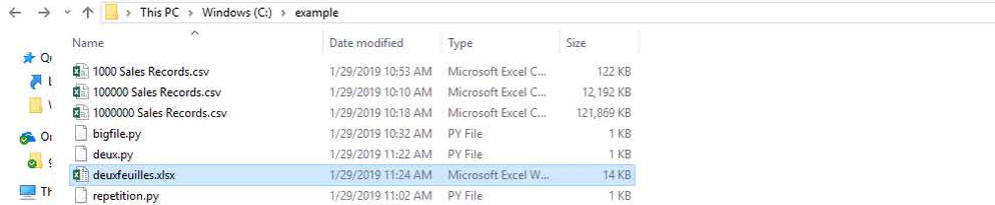



```python
import pandas

df = pandas.read_csv('c:\\example\\1000 Sales Records.csv')

lignes_cool_m = (df['Country'].str.startswith('M')) & (df['Order Priority'] == 'M')
lignes_froid = df['Country'].isin(['Canada','Greenland'])
lignes_gros_commandes = df['Total Revenue'] > 5000000

df['Notes'] = None
df['Notes'][lignes_cool_m] = 'Cool -- deux M !'
df['Notes'][lignes_froid] = 'Il fait froid'
df['Notes'][lignes_gros_commandes] = 'On commande beaucoup'

df.to_csv('c:\\example\\1000-avec-notes.csv', index=False)
```

Pourquoi Coder:
#3 – joindre les feuilles








```python
import pandas

df1 = pandas.read_excel('c:\\example\\deuxfeuilles.xlsx', sheet_name='Sheet1')
df2 = pandas.read_excel('c:\\example\\deuxfeuilles.xlsx', sheet_name='Sheet2')

df2 = df2.rename(columns={'LastName':'Last','FirstName':'First','Em':'Email'})

jointuredf = df1.merge(df2, how='left', on=['Last','First','Email'])
jointure2df = df1.merge(df2, how='outer', on=['Last','First','Email'])

jointuredf.to_excel('c:\\example\\unefeuille_comme_excel.xlsx', index=False)
jointure2df.to_excel('c:\\example\\unefeuille_peutetre_mieux.xlsx', index=False)
```

LANÇONS-NOUS DANS LE GRAND BAIN

Vocabulaire

- **Python:** langage de programmation

- **Pandas:** « module » (extension) du langage Python
  - Ajoute des commandes pour manipuler les fichiers CSV et Excel

- On peut exécuter des logiciels Python dans un « IDE » *(environemment de développement)*
  - **IDE** ≈ un éditeur de texte avec un grand bouton “executer” ☺
  - IDEs en ligne = repl.it & codebunk.com – aujourd’hui, repl.it
    - N’UTILISEZ QUE DES DONNEES INVENTEES EN LIGNE ! JAMAIS DE VRAIES !
    - Installez https://www.anaconda.com/download/ sur votre ordinateur et écrivez votre code dans l’IDE “Spyder” qui y est compris pour faire des calculs avec de vraies données.

Nos données

sample1.csv

7 enregistrements
5 colonnes
(des gens & leur **boîte**)

	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

sample2.csv

6 rows
5 columns
(des gens & leur **cuisine préférée**)

	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

Allez sur https://rebrand.ly/sfpyfrcode

Bénévole n ° 1 :
Écrivez le premier programme !

1. Rappellez-moi d'expliquer le code déjà présent
2. Dans la dernière ligne du programme, sur une nouvelle ligne, tapez ce code.

```
p('Salut !')
```

3. Cliquez sur le gros bouton vert « Run » (exécuter) en haut au centre.
4. À droite, dans la zone noire, vérifiez le résultat:

```
Salut !
---CLOISON---
```

5. **Questions ?** (de n'importe qui)

Étudiant n ° 2 : Chargez et affichez les contenus d'un fichier CSV

1. Tapez un « `#` » avant « `p('Salut !')` » pour que la ligne devienne :

```
#p('Salut !')
```

2. Tapez sur « Entrée » et tapez ces 2 lignes :

```
df1 = pandas.read_csv(chemin1)
p(df1)
```

3. Cliquez sur « Run »

4. On discutera le résultat

Étudiant n ° 3 : Affichez des statistiques sur notre fichier CSV

1. Tapez un « # » avant « `p(df1)` » pour que la ligne devienne :

```
#p(df1)
```

2. Tapez sur « Entrée » et tapez ces 5 lignes :

```
p(len(df1))
p(df1.columns)
p(len(df1.columns))
p(list(df1.columns))
p(sorted(df1.columns))
```

3. Cliquez sur « Run »

Étudiant n ° 4 : Affichez les noms de famille et des statistiques sur ces noms

1. Entourez les 5 lignes de l'étudiant n ° 3 d'une paire de « ''' », chaque « ''' » sur sa proper ligne :

```
'''
p(len(df1))
…
p(sorted(df1.columns))
'''
```

2. À la fin de la programme, Tapez sur « Entrée » et tapez ces 6 lignes :

```
col_nom = df1['Last']
p(col_nom)
p(list(col_nom))
cn_unq = col_nom.unique()
p(cn_unq)
p(len(cn_unq))
```

3. Cliquez sur « Run »

Étudiant n ° 5 : Affichez des noms de famille et les prénoms en deux formats

1. Entourez les 5 dernières lignes (des 6) de l'étudiant n ° 4 d'une paire de «`'''`», chaque «`'''`» sur sa propre ligne, gardant la première ligne «`col_nom = df1['Last']`» hors des «`'''`»:

```
'''
p(col_nom)
…
p(len(cn_unq))
'''
```

2. À la fin de la programme, Tapez sur « Entrée » et tapez ces 6 lignes :

```
interessants = ['First','Last']
cols_nom_prenom = df1[interessants]
p(cols_nom_prenom)
col_prenom = df1['First']
tous_nom = pandas.concat([col_nom, col_prenom])
p(sorted(tous_nom))
```

3. Cliquez sur « Run »

QUESTIONS JUSQU'À ICI?

Étudiant n ° 6 : Rejoindre les 2 fichiers. Qui est unique

1. Entourez les 5 dernières lignes (des 6) de l'étudiant n ° 5 d'une paire de « ''' », chaque « ''' » sur sa propre ligne, gardant la première ligne «interessants = ['First','Last']» hors des « ''' »:

```
'''
cols_nom_prenom = df1[interessants]
…
p(sorted(tous_nom))
'''
```

2. À la fin de la programme, Tapez sur « Entrée » et tapez ces 5 lignes (les lignes en bleu font partie de la ligne précédante; ne tapez pas les "tab") :

```
df2 = pandas.read_csv(chemin2)
df2match = df2.rename(columns={'FirstName':'First','LastName':'Last'})
mergedf = df1.merge(df2match, on=interessants, how='outer', indicator=True)
p(mergedf)
p(mergedf.query('_merge != "both"'))
```

3. Cliquez sur « Run »

Étudiant n ° 7 : Créez un nouveau fichier CSV contenant les resultats précédents

1. Tapez un « # » avant « `p(mergedf)` » et avant «`p(mergedf.query('_merge != "both"'))`» pour avoir:

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

2. Tapez sur « Entrée » et tapez :

```
mergedf.to_csv('resultat.csv', index=False)
```

3. Cliquez sur « Run »

4. À gauche, ouvrez le nouveau fichier «resultat.csv» dans la liste de fichiers.

(Si l'exercise 7 ne marche pas …)

@rplrpl/40-Minute Semi-Hands-On Frenc…
No description

fork | run | share | repl talk ai competition | Sign up

Files

- main.py
- requirements.txt

main.py | history

```
p(len(ch_uniq))
'''

interessants = ['First','Last']
'''
cols_nom_prenom = df1[interessants]
p(cols_nom_prenom)
col_prenom = df1['First']
tous_nom = pandas.concat([col_nom, col_prenom])
p(sorted(tous_nom))
'''

df2 = pandas.read_csv(chemin2)
df2match = df2.rename(columns={'FirstName':'First', 'LastName':'Last'})
mergedf = df1.merge(df2match, on=interessants, how='outer', indicator=True)
#p(mergedf)
#p(mergedf.query('_merge != "both"'))

mergedf.to_csv('resultat.csv', index=False)
p('Fin')
```

```
Python 3.6.1 (default, Dec 2015, 13:05:1
[GCC 4.8.2] on linux
>
```

QUESTIONS?

PRÉVOYEZ-VOUS DES UTILISATIONS DANS VOTRE VIE?

Si on est fini tôt...

- Suggérez-moi des transformations que vous voudriez me voir faire aux données ci-dessous.

	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

RESSOURCES

- **Vidéo, diapositives, notes, et liens pour plus apprendre :**
  https://katiekodes.com/pynotes20190209

- **Etre notifié des formations que j'offre** :
  *https://tinyurl.com/**handson**-pypancsv*