Xử lý file Exel

1. Đầu tiên vào Run gõ pip install xlrd

### 2. Đọc dữ liệu

Ví dụ:

1. import xlrd
2. file\_location = "D:/data\_demo.xlsx"
3. wb = xlrd.open\_workbook(file\_location)
4. sheet = wb.sheet\_by\_index(0)
5. print sheet.nrows
6. print sheet.ncols
7. print sheet.cell\_value(3, 1)

Dòng 1: Giúp import gói xlrd ta vừa cài đặt vào.

Dòng 2: file\_location lưu vị trí của file data\_demo.xlsx tôi vừa tạo ở trên.

Dòng 3: Truy cập đến file excel ở trên thông qua biến wb.

Dòng 4: Dòng này giúp bạn truy cập đến sheet mong muốn trong file. Nếu muốn truy cập đến sheet n của file thì giữa 2 dấu ngoặc tròn là n-1.

Dòng 5: In ra số dòng của bảng.

Dòng 6: In ra số cột của bảng.

Dòng 7: Giá trị tại vị trí dòng 3+1 cột 1+1. Tức vị trí B4 trong bảng tính.

#### **Lấy dữ liệu trên một cột hay một dòng**

Ví dụ 1:

1. for rows in range(sheet.nrows):
2. print sheet.cell\_value(rows, 0)

Ví dụ trên trên giúp in ra tất cả các giá trị của cột đầu tiên.

Ví dụ 2:

1. for col in range(sheet.ncols):
2. print sheet.cell\_value(0, col)

Ví dụ trên giúp in ra tất cả các giá trị của dòng đầu tiên

#### **Lấy toàn bộ dữ liệu của bảng**

1. data = [[sheet.cell\_value(r, c) for c in range(sheet.ncols)] for r in range(sheet.nrows)]

Dòng lên trên giúp lấy toàn bộ dữ liệu trong bảng lưu vào data. Biến data này mang kiểu dữ liệu giống như là một mảng 2 chiều. Vì vậy khi muốn lấy giá trị nào từ bảng ta có thể lấy thông qua biến data này tương tự ví dụ sau:

Ví dụ muốn in ra giá trị của cột B4 tức Mido Lê thì dùng lệnh sau:

1. print data[3][1]

**3. Ghi dữ liệu**

Vào run gõ pip install xlwt để cài đặt thư viện xlwwt

### Ghi dữ liệu

#### **Tạo Workbook**

Để bắt đầu công việc ghi dữ liệu công việc đầu tiên là tiến hành tạo Workbook và các sheet:

1. from xlwt import Workbook
3. wb = Workbook()
4. sheet1 = wb.add\_sheet('sheet 1')
5. sheet2 = wb.add\_sheet('sheet 2')
6. sheet3 = wb.add\_sheet('sheet 3')

#### **Ghi dữ liệu**

Xét ví dụ sau:

1. from xlwt import Workbook
3. wb = Workbook()
4. sheet1 = wb.add\_sheet('sheet 1')
5. sheet2 = wb.add\_sheet('sheet 2')
6. sheet3 = wb.add\_sheet('sheet 3')
8. # Write data in sheet 1
9. sheet1.write(0, 0, 'STT')
10. sheet1.write(0, 1, 'Name')
12. wb.save('demo\_xlwt.xls')

Bạn cũng có thể thay sheet1 bới các biến khác để ghi vào các sheet tương ứng khác của Workbook.

Ta cũng có thể tạo trước một dữ liệu sau đó ghi vào Workbook như sau:

1. from xlwt import Workbook
3. wb = Workbook()
4. sheet1 = wb.add\_sheet('sheet 1')
5. sheet2 = wb.add\_sheet('sheet 2')
6. sheet3 = wb.add\_sheet('sheet 3')
8. # Write data in sheet 1
9. sheet1.write(0, 0, 'STT')
10. sheet1.write(0, 1, 'Name')
12. # data
13. author\_stdio = (
14. [1, 'La Kiến Vinh'],
15. [2, 'Vũ Quang Huy'],
16. [3, 'Ryan Le'],
17. )
19. # Start from the first cell. Rows and columns are zero indexed.
20. row = 1
21. col = 0
23. # Iterate over the data and write it out row by row.
24. for item, cost in (author\_stdio):
25. sheet1.write(row, col, item)
26. sheet1.write(row, col + 1, cost)
27. row += 1

30. wb.save('demo\_xlwt.xls')

Ngoài việc ghi dữ liệu ta còn có thể chiều chỉnh độ rộng của một cột nào đó:

1. sheet1.col(1).width = 7000

Hay nhập công thức cho một ô tính:

1. sheet1.write(3, 1, Formula('SUM(A2:A4)'))

Lưu ý: Nếu muốn nhập công thức cho ô tính nào đó nhớ phải import Fomular vào ở trên như sau:

1. from xlwt import Formula

**Hoặc ta có thể sử dụng thư viện PANDAS**

## Thư viện pandas là gì?

Thư viện pandas trong python là một thư viện mã nguồn mở, hỗ trợ đắc lực trong thao tác dữ liệu. Đây cũng là bộ công cụ phân tích và xử lý dữ liệu mạnh mẽ của ngôn ngữ lập trình python. Thư viện này được sử dụng rộng rãi trong cả nghiên cứu lẫn phát triển các ứng dụng về khoa học dữ liệu. Thư viện này sử dụng một cấu trúc dữ liệu riêng là Dataframe. Pandas cung cấp rất nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này. Chính sự linh hoạt và hiệu quả đã khiến cho pandas được sử dụng rộng rãi.

**Tại sao sử dụng thư viện pandas?**

* DataFrame đem lại sự linh hoạt và hiệu quả trong thao tác dữ liệu và lập chỉ mục;
* Là một công cụ cho phép đọc/ ghi dữ liệu giữa bộ nhớ và nhiều định dạng file: csv, text, excel, sql database, hdf5;
* Liên kết dữ liệu thông minh, xử lý được trường hợp dữ liệu bị thiếu. Tự động đưa dữ liệu lộn xộn về dạng có cấu trúc;
* Dễ dàng thay đổi bố cục của dữ liệu;
* Tích hợp cơ chế trượt, lập chỉ mục, lấy ra tập con từ tập dữ liệu lớn.
* Có thể thêm, xóa các cột dữ liệu;
* Tập hợp hoặc thay đổi dữ liệu với group by cho phép bạn thực hiện các toán tử trên tập dữ liệu;
* Hiệu quả cao trong trộn và kết hợp các tập dữ liệu;
* Lập chỉ mục theo các chiều của dữ liệu giúp thao tác giữa dữ liệu cao chiều và dữ liệu thấp chiều;
* Tối ưu về hiệu năng;
* Pandas được sử dụng rộng rãi trong cả học thuật và thương mại. Bao gồm thống kê, thương mại, phân tích, quảng cáo,…

## Cài đặt Pandas

Để cài đặt thư viện Pandas, bạn có thể làm theo một số cách khác nhau theo [tài liệu hướng dẫn](https://pandas.pydata.org/pandas-docs/stable/install.html):

1. Sử dụng pip: pip install pandas
2. Sử dụng conda: conda install pandas

Bây giờ chúng ta sẽ bắt đầu học cách sử dụng thư viện pandas python. Nhưng trước khi bắt đầu, hãy import thư viện pandas nhé. Chúng ta sẽ dùng cả thư viện matplotlib nữa.

Nếu bạn chưa biết về thư viện matplotlib, hãy đọc [bài viết này](https://nguyenvanhieu.vn/khoa-hoc-lap-trinh-python/#Matplotlib) trước nhé.

# Render our plots inline

%matplotlib inline

import pandas as pd

import matplotlib.pyplot as plt

import random

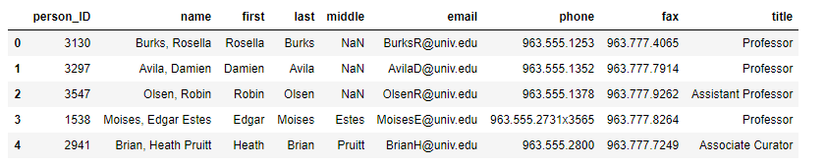
## Đọc file csv sử dụng thư viện pandas

Bạn có thể dễ dàng đọc vào một file .csv bằng cách sử dụng hàm read\_csv và được trả về 1 dataframe. Mặc định, hàm này sẽ phân biệt các trường của file csv theo dấu phẩy. Cách đọc hết sức đơn giản như sau:

peoples\_df = pd.read\_csv('./people.csv')

Bạn có thể in ra n bản ghi đầu tiên của dataframe sử dụng hàm head. Ngược lại của hàm head là hàm tail

peoples\_df.head(5)

Kết quả in ra như sau:

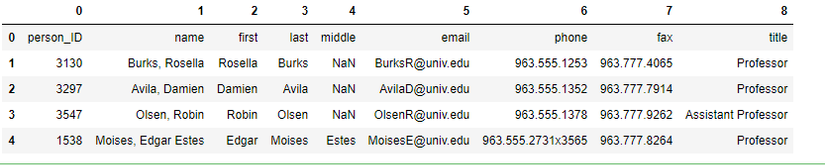
Tuy nhiên, bạn cũng sẽ phải lưu ý một vài tham số của hàm read\_csv như:

* encoding: chỉ định encoding của file đọc vào. Mặc định là utf-8.
* sep: thay đổi dấu ngăn cách giữa các cột. Mặc định là dấu phẩy (‘,’)
* header: chỉ định file đọc vào có header(tiêu đề của các cột) hay không. Mặc định là infer.
* index\_col: chỉ định chỉ số cột nào là cột chỉ số(số thứ tự). Mặc định là None.
* n\_rows: chỉ định số bản ghi sẽ đọc vào. Mặc định là None – đọc toàn bộ.

Ví dụ:

peoples\_df = pd.read\_csv('./people.csv', encoding='utf-8', header=None, sep=',')

peoples\_df.head(5)

Khi tôi chỉ định không có header, dòng header của chúng ta đã biến thành 1 bản ghi dữ liệu:

Bạn đọc có thể xem mô tả đầy đủ từng tham số của hàm read\_csv của thư viện pandas python [tại đây](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_csv.html).

## Thao tác với dataframe trong pandas

### Xem thông tin của dataframe

Bạn có thể xem thông tin của dataframe vừa đọc vào bằng cách sử dụng hàm .info()hoặc xem kích thước của dataframe này với thuộc tính shape. Cụ thể như sau:

# Xem chiều dài của df, tương đương shape[0]

print('Len:', len(peoples\_df))

# Xem thông tin dataframe vừa đọc được

peoples\_df.info()

# Xem kích thước của dataframe

print('Shape:', peoples\_df.shape)

Và kết quả thu được là:

Len: 40

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 40 entries, 0 to 39

Data columns (total 11 columns):

person\_ID 40 non-null int64

name 40 non-null object

first 40 non-null object

last 40 non-null object

middle 15 non-null object

email 40 non-null object

phone 40 non-null object

fax 40 non-null object

title 40 non-null object

age 40 non-null int64

is\_young 40 non-null bool

dtypes: bool(1), int64(2), object(8)

memory usage: 3.2+ KB

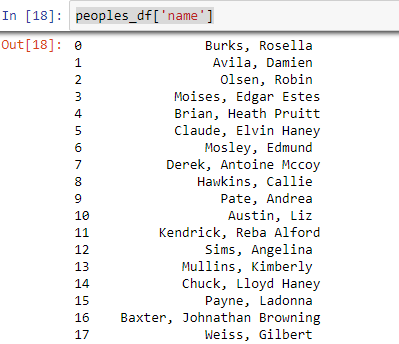
Shape: (40, 11)

### Truy xuất dữ liệu trên dataframe

**Lấy 1 cột theo tên cột**

Để chỉ định cột muốn lấy, bạn chỉ cần truyền vào tên cột như sau:

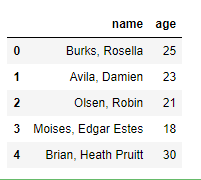
peoples\_df['name']



**Lấy theo nhiều cột**

Thay vì truyền vào 1 string thì hãy truyền vào 1 list các tên cột. Mình thêm .head(5) để chỉ in ra 5 bản ghi đầu tiên cho ngắn, mặc định sẽ lấy hết.

peoples\_df[['name', 'age']].head(5)

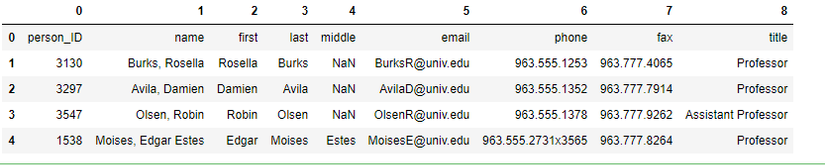


**Lấy bản ghi theo chỉ số**

Để lấy một hoặc nhiều bản ghi liên tiếp trong dataframe, sử dụng cơ chế trượt theo chỉ số giống như trên list trong python. Lấy 5 bản ghi đầu tiên:

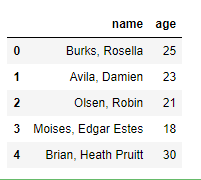
peoples\_df[0:5]

Trong trường hợp này kết quả giống như hàm head phía trên. Đều là lấy 5 bản ghi đầu tiên.



Bạn cũng có thể kết hợp lấy theo hàng và cột mong muốn:

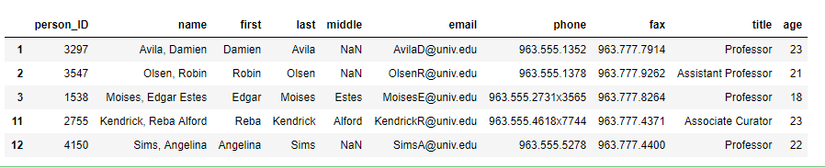
peoples\_df[['name', 'age']][:5]



**Lấy các bản ghi theo điều kiện**

young\_pp = peoples\_df[peoples\_df['age'] < 25]

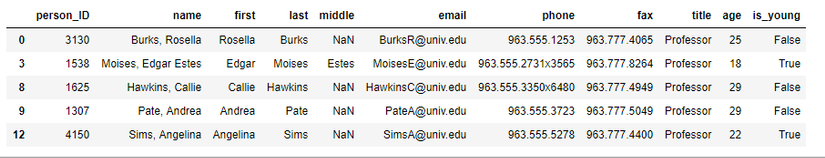
young\_pp[:5]



Một ví dụ khác: Lấy tất cả các bản ghi chứa thông tin của người có chức danh là giáo sư:

professor = peoples\_df[peoples\_df.title == 'Professor']

professor.head(5)



Hoặc 1 ví dụ so sánh chuỗi như sau:

name\_compare = peoples\_df['name'].str.contains('Rosella')

name\_compare.head(5)

Kết quả thu được là một dataframe có 1 cột chưa 2 giá trị True hoặc False

0 True

1 False

2 False

3 False

4 False

Name: name, dtype: bool

**Lấy giá trị trả về numpy arrays**

Để lấy giá trị của một cột trả về dưới dạng numpy array trong thư viện pandas python, bạn chỉ cần thêm .values vào sau, ví dụ:

peoples\_df['name'].values

Output bạn thu được như sau:

array(['Burks, Rosella ', 'Avila, Damien ', 'Olsen, Robin ',

'Moises, Edgar Estes', 'Brian, Heath Pruitt',

'Claude, Elvin Haney', 'Mosley, Edmund ', 'Derek, Antoine Mccoy',

'Hawkins, Callie ', 'Pate, Andrea ', 'Austin, Liz ',

'Kendrick, Reba Alford', 'Sims, Angelina ', 'Mullins, Kimberly ',

'Chuck, Lloyd Haney', 'Payne, Ladonna ',

'Baxter, Johnathan Browning', 'Weiss, Gilbert ',

'Deirdre, Florence Barrera', 'Fernando, Toby Calderon',

'Garrison, Patrica ', 'Effie, Leila Vinson', 'Buckley, Rose ',

'Stanton, Kathie ', 'Banks, Shannon ', 'Barnes, Cleo ',

'Brady, Nellie ', 'Katheryn, Ruben Holt', 'Michael, Dianne ',

'Grant, Adam ', 'Head, Kurtis ', 'Berger, Jami ',

'Earline, Jaime Fitzgerald', 'Evelyn, Summer Frost',

'Quentin, Sam Hyde', 'Dunlap, Ann ', 'Shields, Rich Pena',

'Page, Winnie ', 'Sparks, Ezra ', 'Kaufman, Elba '], dtype=object)

Nếu bạn quan tâm tới numpy array, hãy tìm đọc [bài hướng dẫn về numpy](https://nguyenvanhieu.vn/khoa-hoc-lap-trinh-python/#Numpy)

### Thêm, sửa, xóa trong dataframe

**Thêm cột từ dữ liệu mới**

Để thêm cột vào một dataframe có sẵn. Trước tiên, bạn cần có 1 list dữ liệu tương ứng với cột mà bạn muốn thêm. Tức là chiều dài của list phải tương ứng với số bản ghi của dataframe bạn muốn thêm.

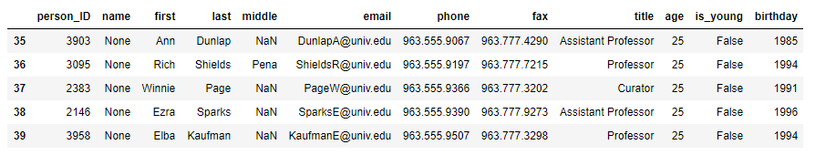
Ở đây, tôi sẽ sử dụng thư viện random để sinh ngẫu nhiên một list năm sinh và thêm vào dataframe như sau:

df\_len = len(peoples\_df)

birthday = [random.randrange(1980, 2000, 1) for i in range(df\_len)]

peoples\_df['birthday'] = birthday

peoples\_df.tail(5)

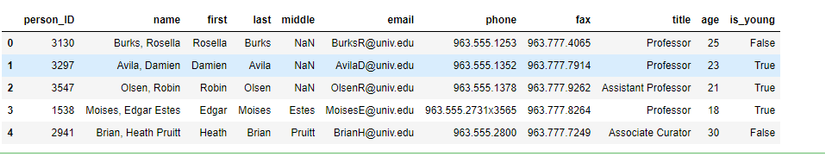


**Thêm cột dựa vào dữ liệu đã có**

Giả sử ở đây mình muốn thêm cột is\_young có giá trị True nếu tuổi < 25 và Falsetrong trường hợp còn lại.

peoples\_df['is\_young'] = peoples\_df['age'] < 25

peoples\_df.head(5)



**Khởi tạo cột mới có giá trị rỗng**

Sử dụng cú pháp đơn giản như dưới đây, bạn sẽ có một trường mới và tất cả các giá trị là None

peoples\_df['new\_column'] = None

**Thêm bản ghi trong dataframe**

Về vấn đề thêm bản ghi, chúng ta thường ít khi sử dụng nên tôi sẽ không trình bày. Bạn đọc quan tâm có thể đọc thêm tại [tài liệu này](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.append.html)

**Sửa giá trị của cột**

Để sửa giá trị của 1 cột, bạn làm tương tự như thêm mới cột. Nhưng khác với thêm ở chỗ là tên cột bạn truyền vào đã có trong dataframe. Còn thêm là 1 tên trường mới hoàn toàn chưa có. Chẳng hạn, bạn muốn thay đổi trường name, bạn chỉ cần làm như sau:

peoples\_df['name'] = xxx #list các tên mới có chiều dài bằng chiều dài của dataframe

// Hoặc reset trường name về None

peoples\_df['name'] = None

**Xóa cột trong dataframe**

Bạn có thể sử dụng một trong các cách sau:

peoples\_df.drop('tên cột cần xóa', axis=1) # Xóa 1 cột

peoples\_df.drop(['cột 1', 'cột 2'], axis=1) # Xóa nhiều cột

df.drop(columns=['B', 'C']) # Xóa các cột có tên là B và C

**Xóa bản ghi theo chỉ số**

peoples\_df.drop([0, 1]) # Xóa bản ghi ở chỉ số 1 và 2

## Hiểu dữ liệu trong dataframe

Thư viện pandas python cung cấp cho bạn một số hàm giúp bạn hiểu về cấu trúc, phân bố của dữ liệu. Dưới đây là cách để bạn khám phá và hiểu dữ liệu của mình.

Tôi đã bổ sung trường age vào file people.csv và tiến hành đọc lại.

peoples\_df['age']

0 25

1 23

2 21

3 18

4 30

5 35

.

.

.

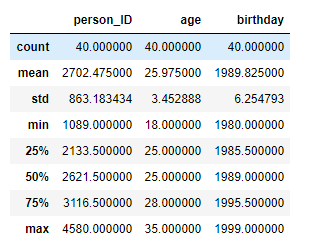
38 25

39 25

Name: age, dtype: int64

Sử dụng hàm describe() cho bạn các thống kê cơ bản về dữ liệu:

peoples\_df.describe()



Xem thống kê cụ thể hơn trên từng cột như sau:

peoples\_df['age'].value\_counts()

25 21

28 6

35 3

29 2

23 2

22 2

30 1

24 1

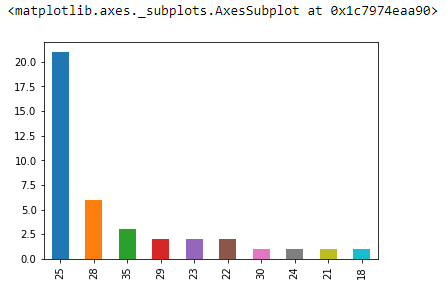
21 1

18 1

Name: age, dtype: int64

Bạn cũng có thể vẽ đồ thị xem phân bố giá trị của một trường trong dataframe như sau:

peoples\_df['age'].value\_counts().plot(kind='bar')



## Tạo mới dataframe

Có một vài cách để tạo ra dataframe trong thư viện pandas python. Bạn có thể dùng cách mà bạn cho là dễ sử dụng, đôi khi cũng phải tùy vào từng trường hợp mà nên chọn cách nào nữa.

### Tạo mới dataframe từ python dictionary

peoples = {'name': ['Nguyễn Văn Hiếu', 'Hiếu Nguyễn Văn'], 'age': [28, 28], 'website': ['https://nguyenvanhieu.vn', None]}

df = pd.DataFrame(peoples)

print(df)

Bạn sẽ có 1 dataframe như sau:

name age website

0 Nguyễn Văn Hiếu 28 https://nguyenvanhieu.vn

1 Hiếu Nguyễn Văn 28 None

### Tạo mới dataframe từ các python list

txts = ['chỗ này ăn cũng khá ngon', 'ngon, nhất định sẽ quay lại', 'thái độ phục vụ quá tệ']

labels = [1, 1, 0]

df = pd.DataFrame()

df['txt'] = txts

df['label'] = labels

print(df)

Và dataframe mà bạn sẽ thu được là:

txt label

0 chỗ này ăn cũng khá ngon 1

1 ngon, nhất định sẽ quay lại 1

2 thái độ phục vụ quá tệ 0

**Lưu ý:** các list này phải có cùng kích thước.

## Một số thao tác khác trên dataframe

### Sắp xếp dataframe

Với thư viện pandas python, bạn có thể sắp xếp dataframe tăng dần, hay giảm dần theo 1 hoặc nhiều cột chỉ định.

# Sắp xếp df tăng dần theo cột nào đó

df = pd.DataFrame({'name': ['Nam', 'Hiếu', 'Mai', 'Hoa'], 'age': [18,18,17,19]})

print('Before sort\n', df)

df = df.sort\_values('age', ascending=True)

print('After sort\n', df)

Kết quả thu được như sau:

Before sort

name age

0 Nam 18

1 Hiếu 18

2 Mai 17

3 Hoa 19

After sort

name age

2 Mai 17

0 Nam 18

1 Hiếu 18

3 Hoa 19

Bạn có thể sắp xếp theo nhiều cột có độ ưu tiên giảm dần, bằng cách truyền vào list tên cột. Ví dụ:

df.sort\_values(['age', 'name'], ascending=True)

### Nối 2 dataframe

Bạn có thể nối 2 dataframe thành 1 dataframe mới bằng cách sử dụng hàm append của thư viện pandas.

# Gộp 2 dataframe

df1 = pd.DataFrame({'name': ['Hiếu'], 'age': [18], 'gender': ['male']})

df2 = pd.DataFrame({'name': ['Nam', 'Mai', 'Hoa'], 'age': [15,17,19]})

df = df1.append(df2, sort=True)

print(df)

Kết quả:

age gender name

0 18 male Hiếu

0 15 NaN Nam

1 17 NaN Mai

2 19 NaN Hoa

### Xáo trộn các bản ghi trong dataframe

Trong xử lý dữ liệu, bạn chắc chắn sẽ cần tới việc xáo trộn dữ liệu. Rất may, thư viện pandas có thể giúp chúng ta làm việc đó.

# Xáo trộn các hàng trong df - shuffle dataframe rows

df = pd.DataFrame({'name': ['Hiếu', 'Nam', 'Mai', 'Hoa'], 'age': [18,15,17,19]})

print('Before shuffle\n', df)

df = df.sample(frac=1).reset\_index(drop=True)

print('After shuffle\n', df)

Kết quả(Có thể khác nhau nhé – xáo trộn mà)

Before shuffle

name age

0 Hiếu 18

1 Nam 15

2 Mai 17

3 Hoa 19

After shuffle

name age

0 Mai 17

1 Nam 15

2 Hiếu 18

3 Hoa 19

Giải thích thêm:

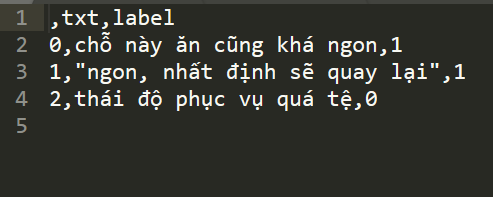
* frac: Chỉ định số bản ghi sẽ trả về ở mỗi lần random. Nếu bằng 1, tức là random ngẫu nhiên tất cả các bản ghi.
* .reset\_index(): Sắp xếp lại cột chỉ số của dataframe.
* drop:với giá trị True, nó sẽ ngăn không cho hàm reset\_index tạo cột mới từ cột chỉ số của dataframe ban đầu.

## Lưu dataframe về file csv

Thư viện pandas python cho phép bạn lưu lại dataframe chỉ với một dòng code. Quá đơn giản phải không nào?

df.to\_csv('comments.csv')

Bạn có thể mở file để xem kết quả lưu:



Các tham số của hàm to\_csv khá tương tự với hàm read\_csv. Bạn đọc có thể xem thêm thông tin đầy đủ của hàm này [tại đây](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.to_csv.html).