**Pandas trong Python**

**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;

## Đọc file csv

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)

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=',')

## Dataframe trong pandas :

### +Kích thước của dataframe này với thuộc tính shape

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

+ Lấy theo nhiều cột : truyền vào 1 list các tên cột . .head(5) để chỉ in ra 5 bản ghi đầu tiên,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]

Có thể kết hợp lấy theo hàng và cột mong muốn:

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

### + Lấy các bản ghi theo điều kiện:

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

professor.head(5)

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

### **Lấy giá trị trả về numpy arrays**

Để lấy giá trị của một cột trả về dưới dạng numpy array chỉ cần thêm .values

peoples\_df['name'].values

### Thêm, sửa, xóa trong dataframe

#### 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)

### *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)

### *Sắp xếp dataframe*

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

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)

# TÀI LIỆU THAM KHẢO :

<https://viblo.asia/p/huong-dan-su-dung-thu-vien-pandas-trong-python-XL6lAxaDZek>