mindX Technology School

Computer Science Advanced

BÀI 9. GIỚI THIỆU THƯ VIỆN PANDAS

1. Thư Viện Pandas

Ảnh: Real Python

Pandas là một *thư viện Python* hỗ trợ *xử lý và phân tích trên dữ liệu*.

Cụ thể, ***pandas*** hỗ trợ mạnh về các cấu trúc dữ liệu và hàm xử lý cho:

- *Dữ liệu dạng bảng*
- *Dữ liệu dạng chuỗi thời gian*

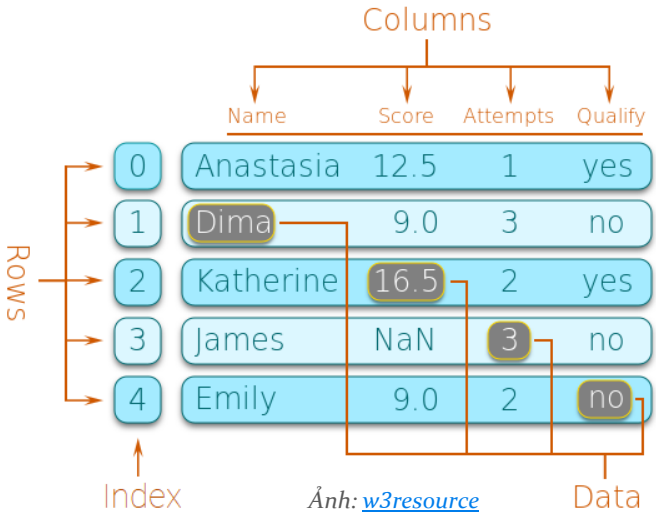
Sau khi cài đặt, ta sử dụng *pandas* trong chương trình bằng lệnh `import pandas as pd`, với `pd` là tên viết tắt thông dụng của thư viện này.

DATAFRAME VÀ SERIES

Pandas tổ chức *dữ liệu dạng bảng* theo *cấu trúc dữ liệu **DataFrame***.

Một ***DataFrame*** chứa các *dòng* và *cột*. Khi truy vấn riêng, mỗi dòng hoặc cột được trả về theo *cấu trúc dữ liệu **Series***.

- Mỗi cột được đánh dấu bằng tên cột.
- Mỗi dòng được mặc định *đánh số bằng cách đếm từ 0*. Danh sách đánh số này gọi là ***index***. Ta có thể tùy chỉnh để *index* theo dữ liệu dạng *string, ngày tháng hoặc các kiểu dữ liệu khác.*



Ảnh: w3resource

Khởi tạo DataFrame:

```
pd.DataFrame([data], [index=…], [columns=…])
```

Ví dụ: Khởi tạo một *DataFrame* chứa 3 dòng và 3 cột đầu tiên trong ảnh minh họa.

Cách 1: *Sử dụng dictionary*

```
df = pd.DataFrame({
  'Name': ['Anastasia', 'Dima', 'Katherine'],
  'Score': [12.5, 9.0, 16.5],
  'Attemps': [1, 3, 2]
})
```

Cách 2: *Sử dụng list chứa list*

```
df = pd.DataFrame(
  [['Anastasia', 12.5, 1],
   ['Dima', 9.0, 3],
   ['Katherine', 16.5, 2]],
  columns=['Name', 'Score', 'Attemps']
)
```

Khởi tạo Series:

```
pd.Series([data], [index=…])
```

Ví dụ: Khởi tạo *Series* tương ứng với cột và dòng đầu tiên trong ảnh.

```
first_col = pd.Series(['Anastasia', 'Dima', 'Katherine'])
first_row = pd.Series(['Anastasia', 12.5, 1],
                      index=['Name', 'Score', 'Attemps'])
```

CÁC THUỘC TÍNH VÀ PHƯƠNG THỨC TRÊN DATAFRAME

Giả sử ta có biến **`df`** chứa ***DataFrame*** đã khởi tạo như trên.

Thuộc tính / Phương thức	Kết quả

`df.dtypes`
Trả về kiểu dữ liệu của từng cột.
	`Name object`
	`Score float64`
	`Attemps int64`
	`Qualify object`
	`dtype: object`

`df.head()`
Trả về 5 dòng đầu tiên.
Ngược lại, **`df.tail()`** trả về 5 dòng cuối cùng.
	`Name Score Attemps Qualify`
	`0 Anastasia 12.5 1 yes`
	`1 Dima 9.0 3 no`
	`2 Katherine 16.5 2 yes`
	`3 James NaN 3 no`
	`4 Emily 9.0 2 no`

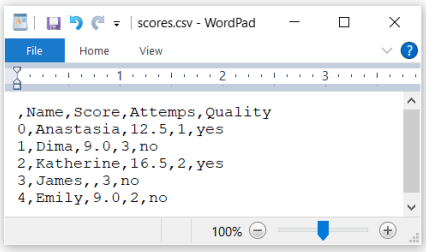
`df.info()`
Tổng hợp thông tin về *DataFrame* và các cột.
	`RangeIndex: 5 entries, 0 to 4`
	`Data columns (total 4 columns):`
	`# Column Non-Null Count Dtype`
	`0 Name 5 non-null object`
	`1 Score 4 non-null float64`
	`2 Attemps 5 non-null int64`
	`3 Qualify 5 non-null object`
	`dtypes: float64(1), int64(1), object(2)`
	`memory usage: 288.0+ bytes`

Đối với ***Series***, ta cũng có thuộc tính **`dtype`** và các phương thức **`head()`**, **`tail()`** với chức năng như trên.

2. Nhập và Xuất Dữ Liệu Dạng Bảng

Pandas hỗ trợ nhập *(import)* và xuất *(export)* dữ liệu theo nhiều định dạng khác nhau. Trong bài học này, ta tập trung vào nhập xuất dữ liệu dạng bảng theo định dạng file *CSV (Comma-Separated Values)* và *Excel*.

CSV (.csv)



Là text file.

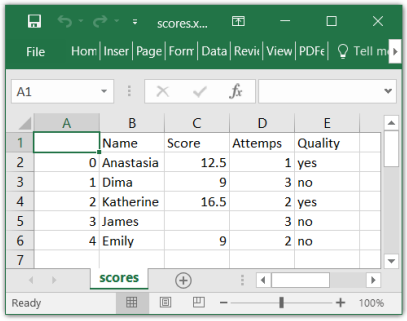
Các giá trị trong cùng một dòng được tách nhau bằng dấu ,

Có thể được mở bằng text editor thông thường hoặc phần mềm đọc bảng tính như Excel.

Phương thức:

```
pd.read_csv(file_path)
df.to_csv(file_path)
```

Excel (.xls hay .xlsx)



Là binary file.

Có thể chứa dữ liệu chữ, định dạng, hình ảnh và biểu đồ.

Có thể chứa nhiều bảng tính *(sheet)* trong cùng một file.

Phương thức:

```
pd.read_excel(file_path, [sheet_name=…])
df.to_excel(file_path, [sheet_name=…])
```

Các phương thức nhập dữ liệu trả về kết quả là một *DataFrame*. ***Ví dụ***:

```
df1 = pd.read_csv('scores.csv')
df2 = pd.read_excel('scores.xls', sheet_name='scores')
```

Các phương thức xuất dữ liệu tạo file với dữ liệu trong *DataFrame*. ***Ví dụ:***

```
df1.to_csv('scores.csv')
df2.to_excel('scores.xls', sheet_name='scores')
```