python pandas tutorial

민 정 원
jeongwon8694@gmail.com

contents

- What is pandas?
- Series and DataFrame
- Tutorial1 : Series
- Tutorial2 : DataFrame
- Tutorial3 : hepatitis, hepatoma data analysis

pandas

About us Getting started Documentation Community Contribute

pandas

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

Install pandas now!

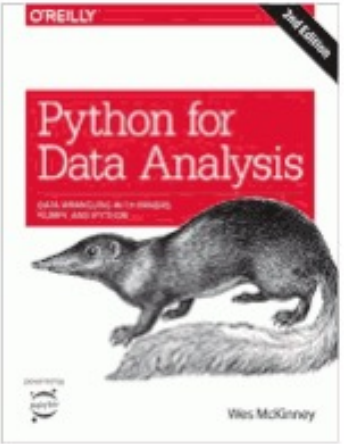
Latest version: 1.2.3

- What's new in 1.2.3
- Release date: Mar 02, 2021
- Documentation (web)
- Documentation (pdf)
- Download source code

Follow us

Follow @pandas_dev

Get the book



Getting started

- Install pandas
- Getting started

Documentation

- User guide
- API reference
- Contributing to pandas
- Release notes

Community

- About pandas
- Ask a question
- Ecosystem

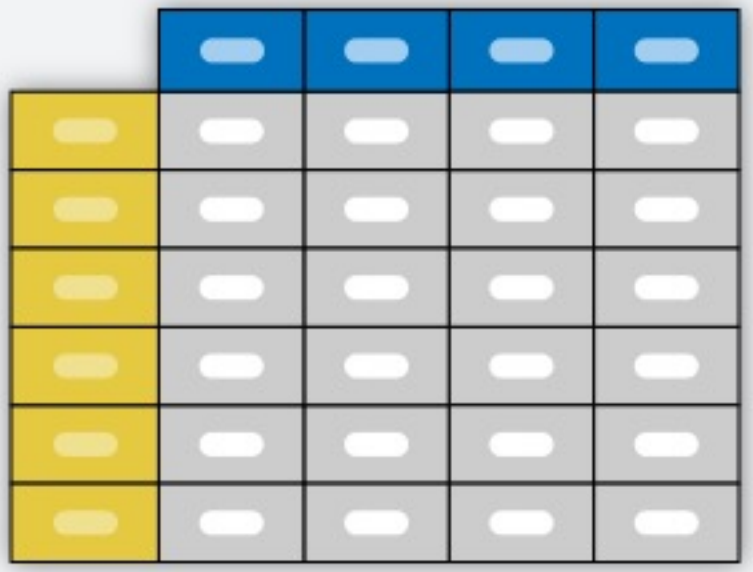
With the support of:

Pandas 에서 사용되는 대표적인 데이터 오브젝트

시리즈 (Series)

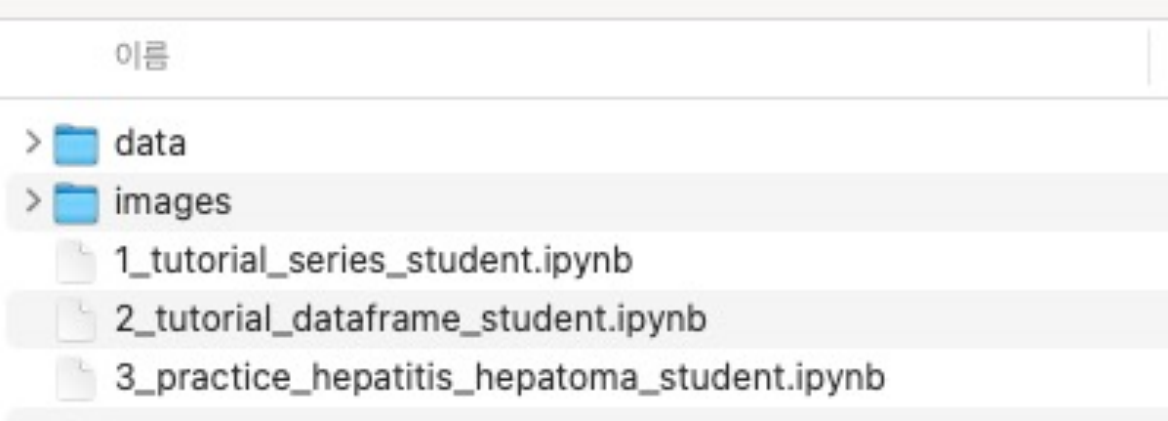
Series 는 1차원 배열의 형태를 갖는다.
인덱스(노란색)라는 한 가지 기준에
의하여 데이터가 저장된다.

데이터프레임 (DataFrame)



DataFrame 은 2차원 배열의 형태를 갖는다.
인덱스(노란색)와 컬럼(파란색)이라는 두 가지
기준에 의하여 표 형태처럼 데이터가 저장된다.

dandyrilla.github.io


이름
data
images
1_tutorial_series_student.ipynb
2_tutorial_dataframe_student.ipynb
3_practice_hepatitis_hepatoma_student.ipynb

3. values and indices

```
df.values

df.columns

df.columns = ['번호', '기업', '국가', '순위', '수익']

df.index

df.index = ['가', '나', '다', '라', '마']

df = pd.DataFrame(data, columns=columns, index=index)
```

4. data overview

```
df.head(n=3)

df.tail(n=2)

df.info()

df.describe()

df.size

df.shape
```

```
df.sort_values(by='Rank', ascending=False)

df['Headquarters'].unique()

df['Headquarters'].nunique()
```

5. indexing and selecting data

```
df.['ID'] == df.ID
: selecting a column

df[['ID', 'Score']]
: selecting multiple columns

df[0:1]
: selecting the row with index number 0

df[2:4]
: selecting the rows with index number 2~3
```

```
df.loc[rows, columns]
: selecting by labels(names)

df.iloc[rows, columns]
: selecting by index numbers

df[df['Name']=='Gildong']
: selecting by conditions
```

6. grouping data

```
df.groupby(['Location']).mean()

df.groupby(['Location']).min()

df.groupby(['Location']).count()

df.groupby(['Location']).sum()

df.groupby(['Location']).size()

df.groupby(['Location', 'Fruit']).count()

df.groupby(['Location', 'Fruit']).agg(['min', 'max', 'mean'])
```

7. missing values

```
df.isnull()
df['A'].isnull()
df.isnull().sum()
df['A'].isnull().sum()
df.notnull().sum()
```

```
df.dropna()
df.dropna(subset=['B', 'D'])
df.fillna(value=-1)
```