

Pandas

시작하기

```
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt
```

Pandas 란

데이터 정제, 분석도구



[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!](#)

Getting started

- [Install pandas](#)
- [Getting started](#)

Documentation

- [User guide](#)
- [API reference](#)
- [Contributing to pandas](#)
- [Release notes](#)

Community

- [About pandas](#)
- [Ask a question](#)
- [Ecosystem](#)

Latest version: 1.0.3

- [What's new in 1.0.3](#)
- Release date:
Mar 18, 2020
- [Documentation \(web\)](#)
- [Documentation \(pdf\)](#)
- [Download source code](#)

Follow us

[Follow @pandas_dev](#)

Get the book



<https://pandas.pydata.org/>

Pandas 란

특징이해하기

쉽고 직관적인 관계형 또는 분류된 데이터로 작업 할 수 있도록 설계
빠르고 유연하며 표현이 풍부한 데이터 구조를 제공

Pandas 데이터 구조

1차원, 2차원

Series			Series			DataFrame		
	apples			oranges			apples	oranges
0	3	+	0	0	=	0	3	0
1	2		1	3		1	2	3
2	0		2	7		2	0	7
3	1		3	2		3	1	2

Pandas 데이터 구조

1차원, 2차원

```
In [2]: import pandas as pd
```

```
In [3]: fruits = pd.DataFrame([[30, 21], [20, 14], [12, 23]], columns=
    ...: ['Apples', 'Bananas'])
```

```
In [4]: fruits
```

```
Out[4]:
```

	Apples	Bananas
0	30	21
1	20	14
2	12	23

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.

Pandas 사용하기

자료읽기

```
In [1]: import pandas as pd
```

```
In [2]: try:
        df = pd.read_csv('400_20200411221745769.csv')
    except OSError as err:
        print("OS error: {0}".format(err))
    df.head()
```

Out[2]:

	format: day	hour	value	location:63_108	Start : 20190401
0	1	0.0			8.1
1	1	100.0			8.8
2	1	200.0			8.3
3	1	300.0			9.8
4	1	400.0			9.5

Pandas 사용하기

자료쓰기

```
In [9]: file_name = "new_temperature.csv"
df.to_csv(file_name, encoding='utf-8')
```

new_temperature

	format: day	hour	temp
0	1	0	8.1
1	1	100	8.8
2	1	200	8.3
3	1	300	9.8
4	1	400	9.5
5	1	500	8.4
6	1	600	8.8
7	1	700	8.3
8	1	800	7.2
9	1	900	4.9
10	1	1000	2.6
11	1	1100	1
12	1	1200	-0.5

Pandas 사용하기

데이터 선택하기

```
In [7]: df_temp = df['temp']  
df_temp.describe()
```

```
Out[7]: count      8784.000000  
mean         12.549966  
std          10.702835  
min          -50.000000  
25%           3.800000  
50%          12.900000  
75%          21.500000  
max           36.700001  
Name: temp, dtype: float64
```

```
In [8]: df_temp.shape
```

```
Out[8]: (8795,)
```

df.A 또는 df['A']

	A	B	C	D
2013-01-01	0.469112	-0.282863	-1.509059	-1.135632
2013-01-02	1.212112	-0.173215	0.119209	-1.044236
2013-01-03	-0.861849	-2.104569	-0.494929	1.071804
2013-01-04	0.721555	-0.706771	-1.039575	0.271860
2013-01-05	-0.424972	0.567020	0.276232	-1.087401
2013-01-06	-0.673690	0.113648	-1.478427	0.524988