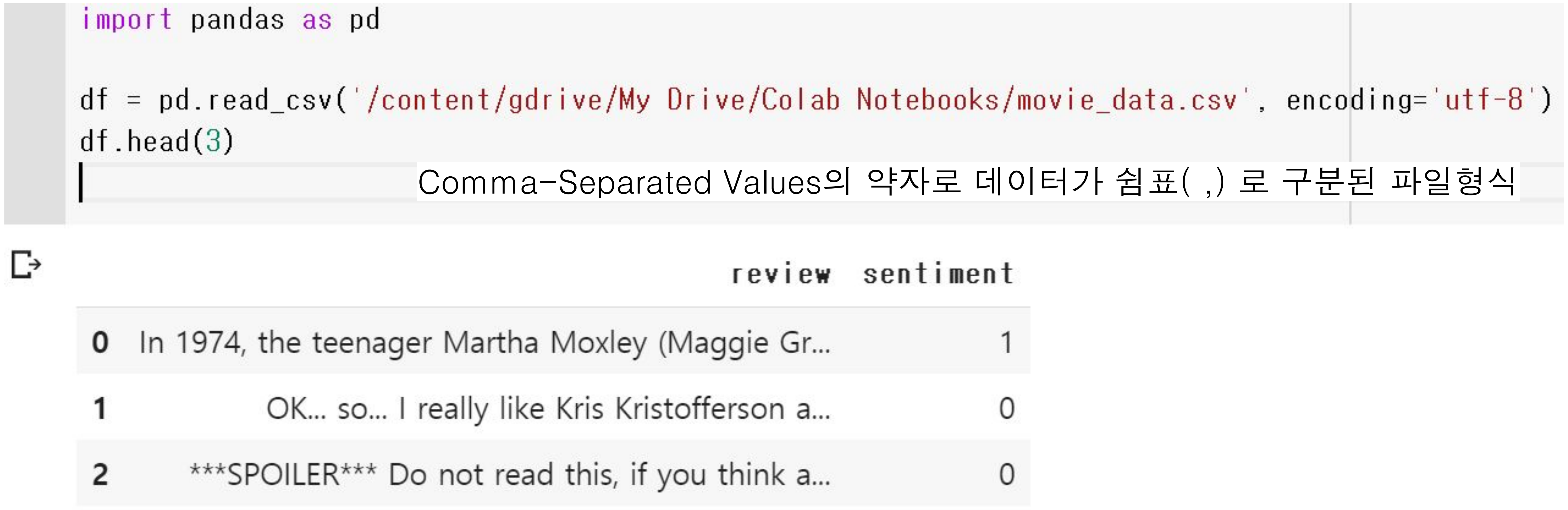
미래 업무 미션:

어떤 사람의 리뷰를 읽고,
그 감정을 맞출 경우
보너스를 받는다면?



```python
import pandas as pd

df = pd.read_csv('/content/gdrive/My Drive/Colab Notebooks/movie_data.csv', encoding='utf-8')
df.head(3)
```

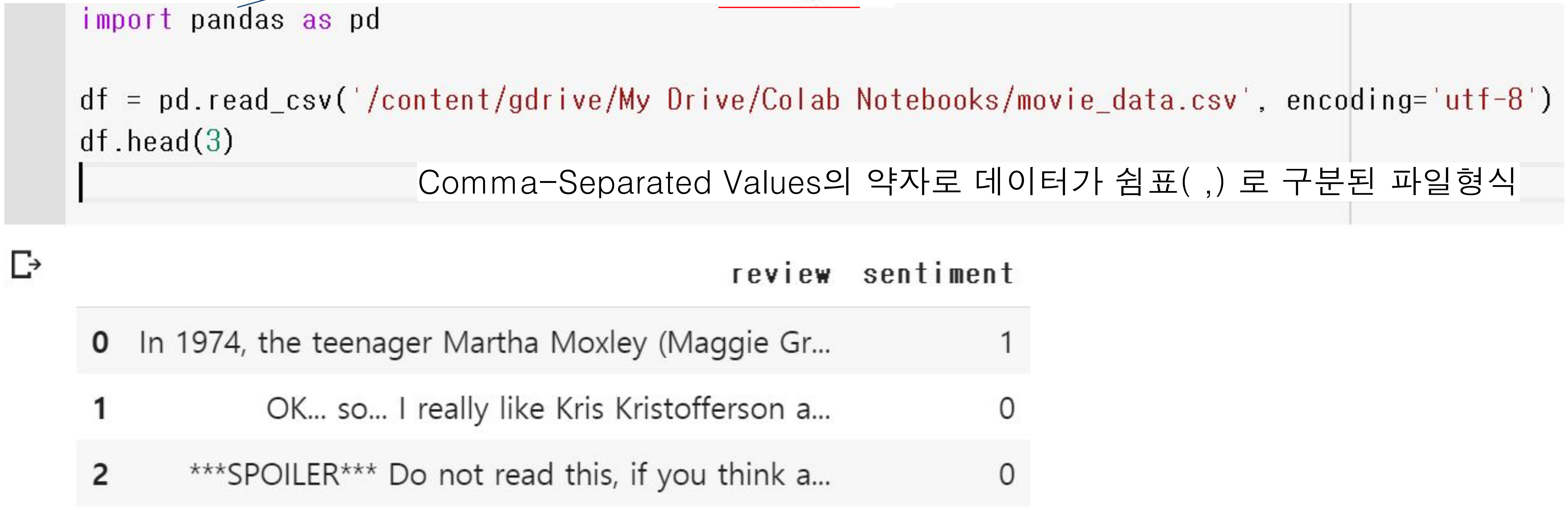
Comma-Separated Values의 약자로 데이터가 쉼표(,) 로 구분된 파일형식

	review	sentiment
0	In 1974, the teenager Martha Moxley (Maggie Gr...	1
1	OK... so... I really like Kris Kristofferson a...	0
2	***SPOILER*** Do not read this, if you think a...	0

The submission paper to PyHPC 2011 says:

> The library's name derives from **pan**el **da**ta, a common term for multidimensional data sets encountered in statistics and econometrics.

AnalySis



```python
import pandas as pd

df = pd.read_csv('/content/gdrive/My Drive/Colab Notebooks/movie_data.csv', encoding='utf-8')
df.head(3)
```

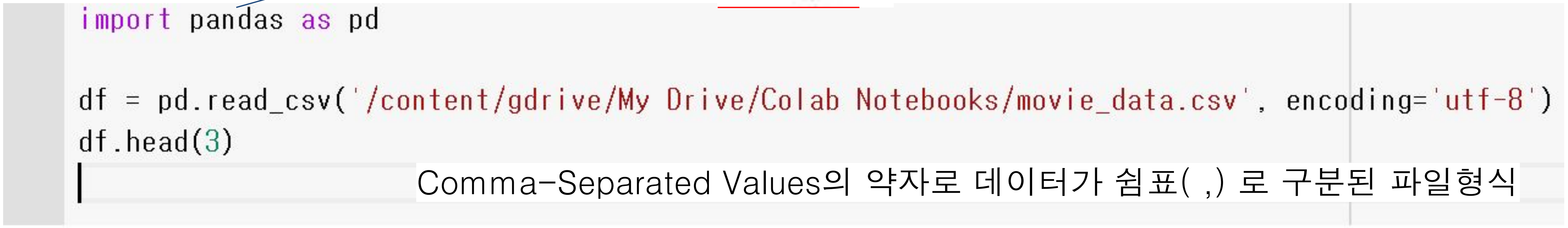
Comma-Separated Values의 약자로 데이터가 쉼표(,) 로 구분된 파일형식

```
                                             review  sentiment
0  In 1974, the teenager Martha Moxley (Maggie Gr...          1
1              OK... so... I really like Kris Kristofferson a...          0
2        ***SPOILER*** Do not read this, if you think a...          0
```

The submission paper to PyHPC 2011 says:

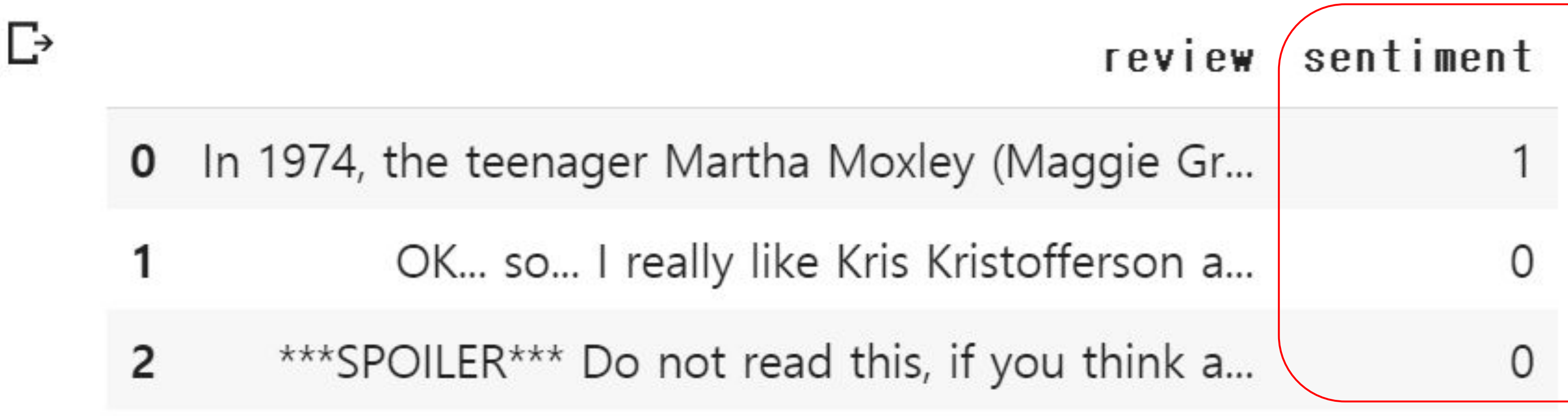
> The library's name derives from **panel data**, a common term for multidimensional data sets encountered in statistics and econometrics.

AnalySis



```
import pandas as pd

df = pd.read_csv('/content/gdrive/My Drive/Colab Notebooks/movie_data.csv', encoding='utf-8')
df.head(3)
```

Comma-Separated Values의 약자로 데이터가 쉼표(,) 로 구분된 파일형식



	review	sentiment
0	In 1974, the teenager Martha Moxley (Maggie Gr...	1
1	OK... so... I really like Kris Kristofferson a...	0
2	***SPOILER*** Do not read this, if you think a...	0

점수≥7점 이상 = 긍정 = 1
점수≤4점 이하 = 부정 = 0

Andrew Y. Ng, and **Christopher Potts**
Stanford University

A negative review has a score ≤ 4 out of 10, and a positive review has a score ≥ 7 out of 10

Panel

old door with oak panels

Pan/D/AS

Pandas

구조화된 데이터 처리용 Python 라이브러리

=Python으로 엑셀, 데이터 묶음 다룰 필요성!

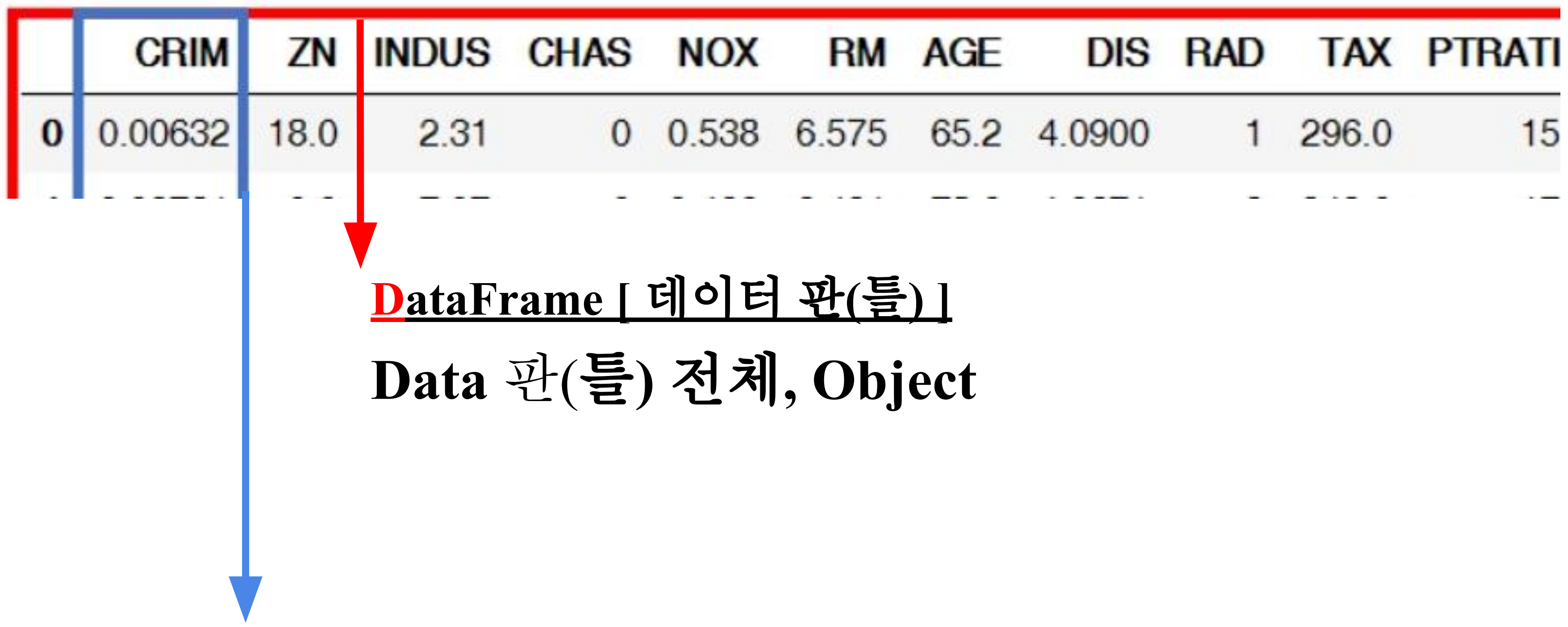
Pandas

- 고성능 Array 계산 라이브러리인 Numpy를 확장하여, 강력한 "스프레드시트" 처리 기능 제공

- 인덱싱, 연산용 함수, 전처리 함수 등을 제공함

DataFrame Overview

Pandas 의 구성



	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATI
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15

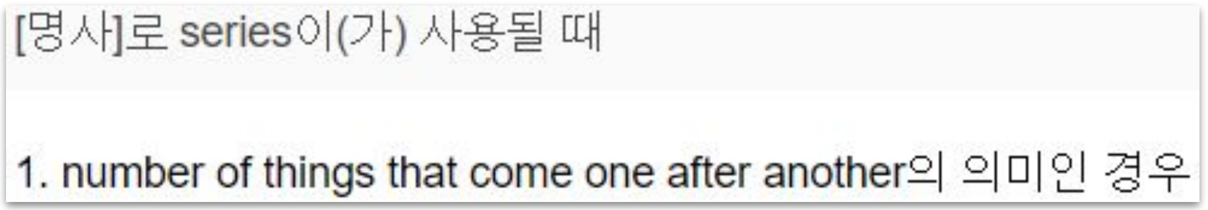
<u>DataFrame [데이터 판(틀)]</u>

Data 판(틀) 전체, Object

<u>Series [연속된 한줄 데이터]</u>

(DataFrame 중) 하나의 Column에

해당하는 Datum의 연속, Object



[명사]로 series이(가) 사용될 때

1. number of things that come one after another의 의미인 경우

DataFrame

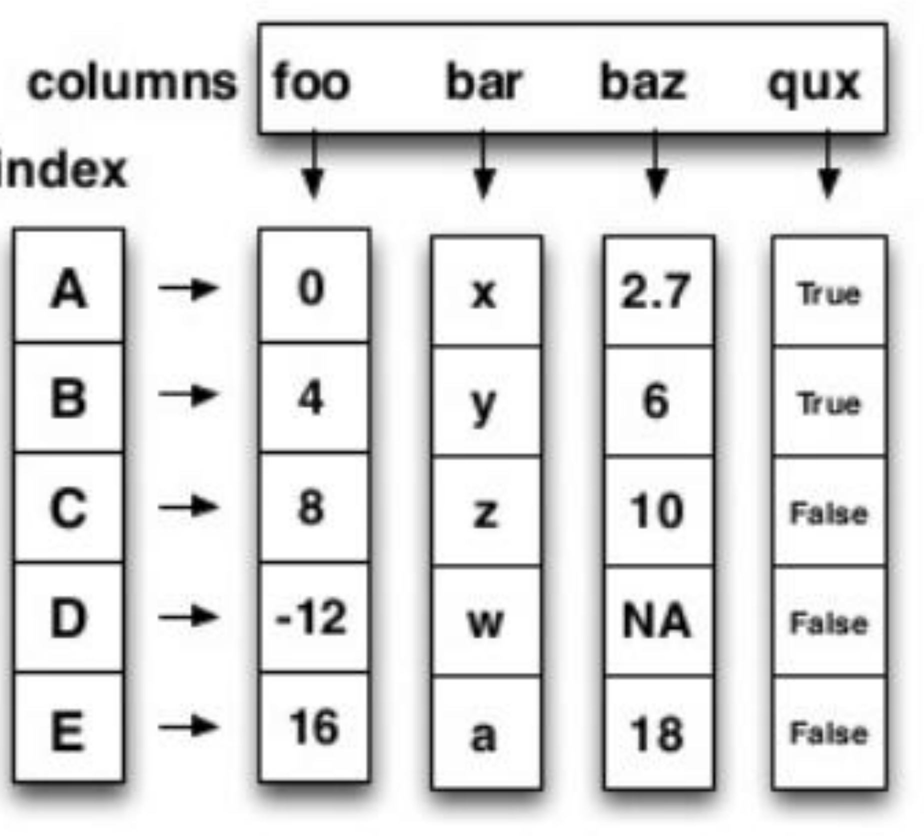
각 특징들

관찰 대상자

Apple

Bear

Cat



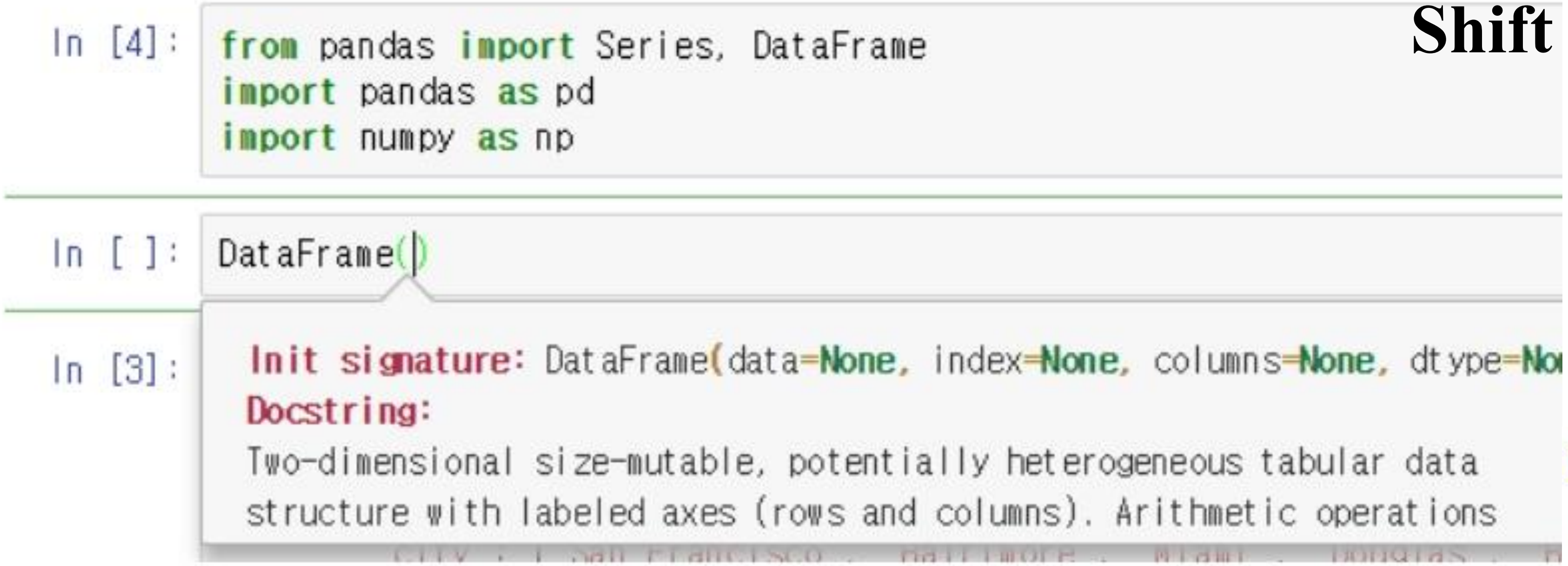
- NumPy array-like
- Each column can have a different type
- Row and column index
- Size mutable: insert and delete columns

https://www.slideshare.net/wesm/pandas-powerful-data-analysis-tools-for-python

DataFrame

Series를 모아서 만든 Data 판(틀) = 기본 2차원

Shift + TAB



```
In [4]: from pandas import Series, DataFrame
        import pandas as pd
        import numpy as np

In [ ]: DataFrame()

In [3]:
```

```
Init signature: DataFrame(data=None, index=None, columns=None, dtype=No
Docstring:
Two-dimensional size-mutable, potentially heterogeneous tabular data
structure with labeled axes (rows and columns). Arithmetic operations
```

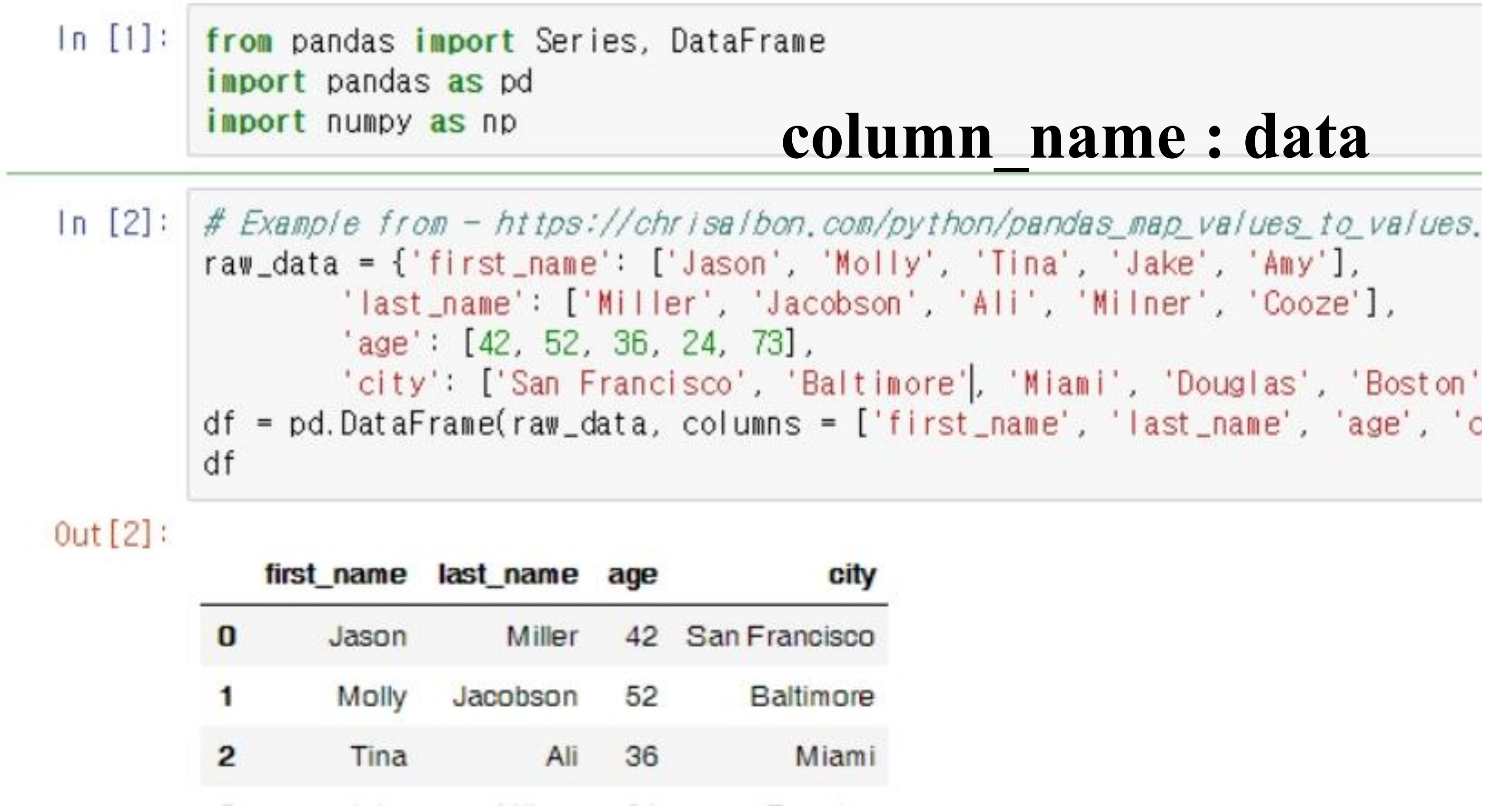
tabular 미국식 ['tæbjələ(r)]

표로 나타낸

table (tabular) 미국·영국 ['teɪbl]

1. 식탁, 테이블, 탁자, (밥)상

DataFrame



column_name : data

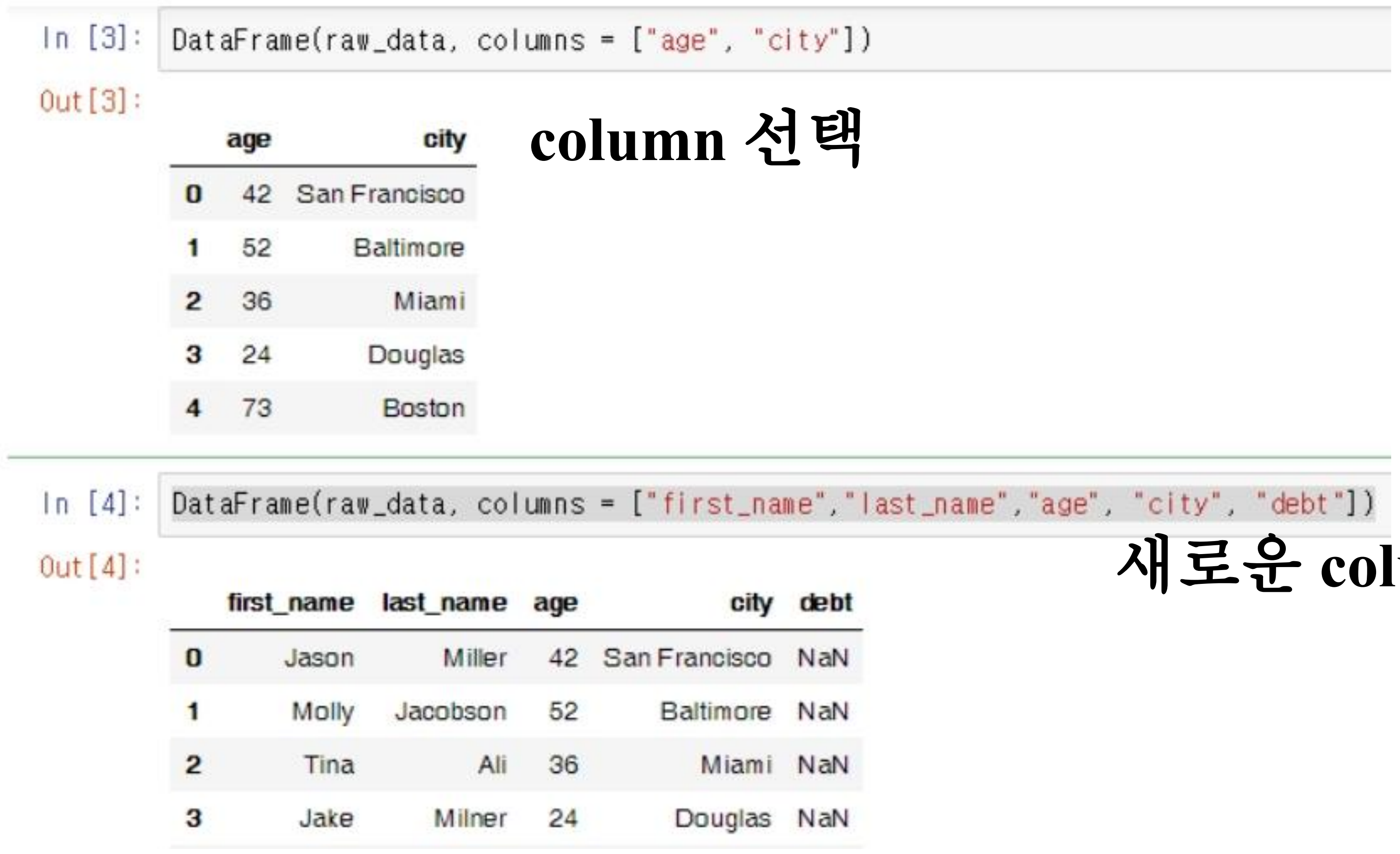
```
In [1]: from pandas import Series, DataFrame
        import pandas as pd
        import numpy as np
```

```
In [2]: # Example from - https://chrisalbon.com/python/pandas_map_values_to_values.
        raw_data = {'first_name': ['Jason', 'Molly', 'Tina', 'Jake', 'Amy'],
                'last_name': ['Miller', 'Jacobson', 'Ali', 'Milner', 'Cooze'],
                'age': [42, 52, 36, 24, 73],
                'city': ['San Francisco', 'Baltimore', 'Miami', 'Douglas', 'Boston'
        df = pd.DataFrame(raw_data, columns = ['first_name', 'last_name', 'age', 'c
        df
```

Out[2]:

	first_name	last_name	age	city
0	Jason	Miller	42	San Francisco
1	Molly	Jacobson	52	Baltimore
2	Tina	Ali	36	Miami

DataFrame



```
In [3]: DataFrame(raw_data, columns = ["age", "city"])
Out[3]:
```

	age	city
0	42	San Francisco
1	52	Baltimore
2	36	Miami
3	24	Douglas
4	73	Boston

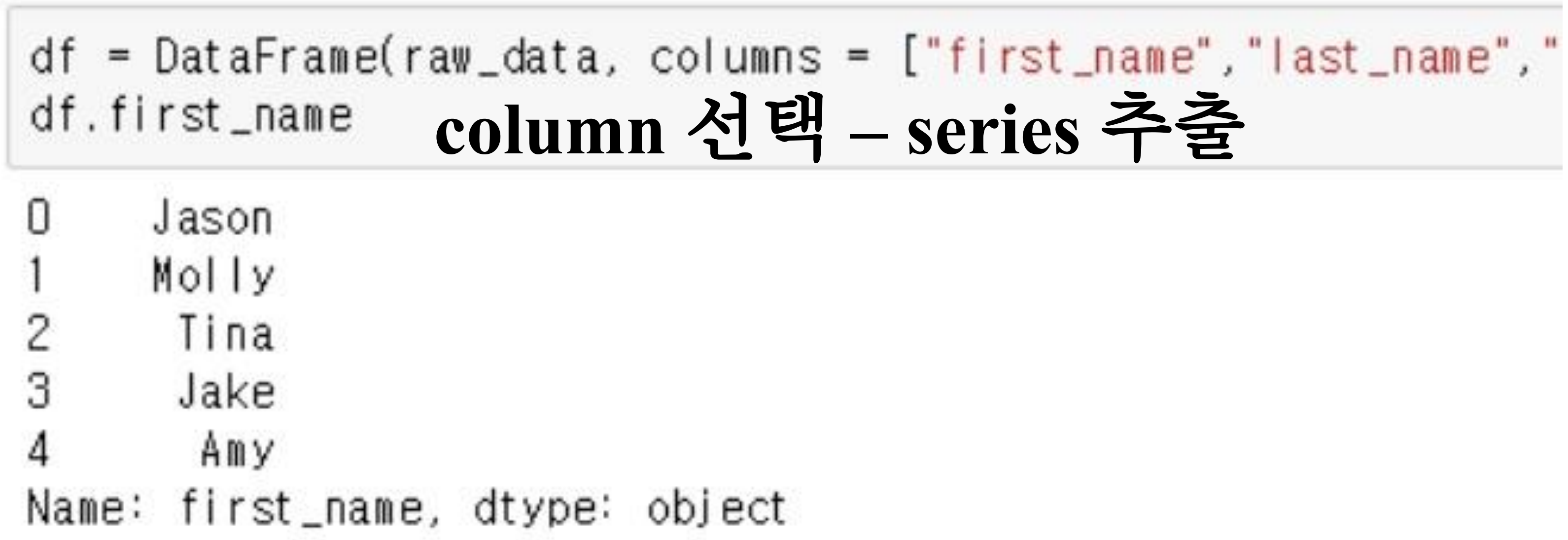
column 선택

```
In [4]: DataFrame(raw_data, columns = ["first_name","last_name","age", "city", "debt"])
Out[4]:
```

새로운 column 추가

	first_name	last_name	age	city	debt
0	Jason	Miller	42	San Francisco	NaN
1	Molly	Jacobson	52	Baltimore	NaN
2	Tina	Ali	36	Miami	NaN
3	Jake	Milner	24	Douglas	NaN

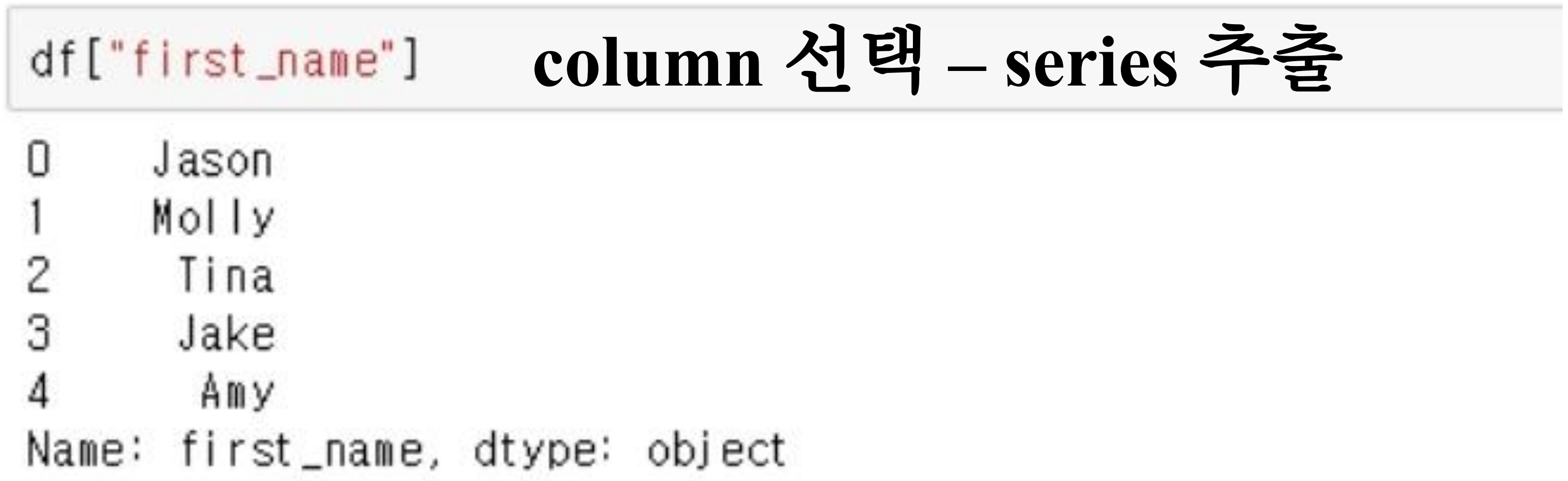
DataFrame



```
df = DataFrame(raw_data, columns = ["first_name","last_name","
df.first_name
```

column 선택 – series 추출

```
0     Jason
1     Molly
2      Tina
3      Jake
4       Amy
Name: first_name, dtype: object
```

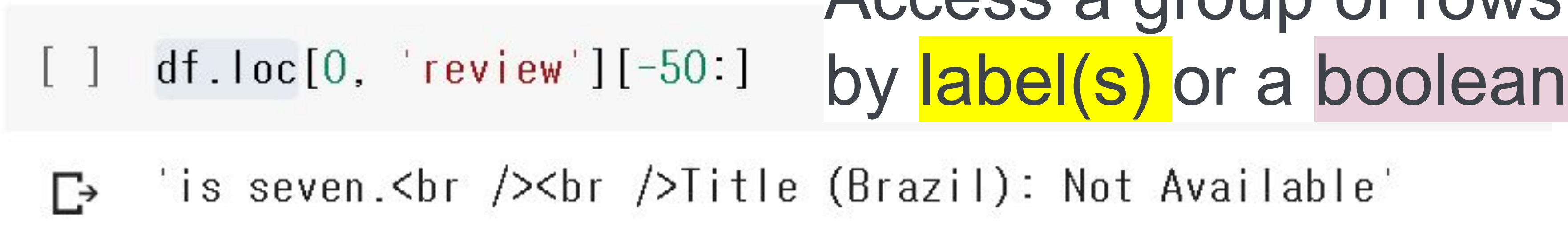


```
df["first_name"]
```

column 선택 – series 추출

```
0     Jason
1     Molly
2      Tina
3      Jake
4       Amy
Name: first_name, dtype: object
```

텍스트 데이터 정제 (리뷰 분석 예제)

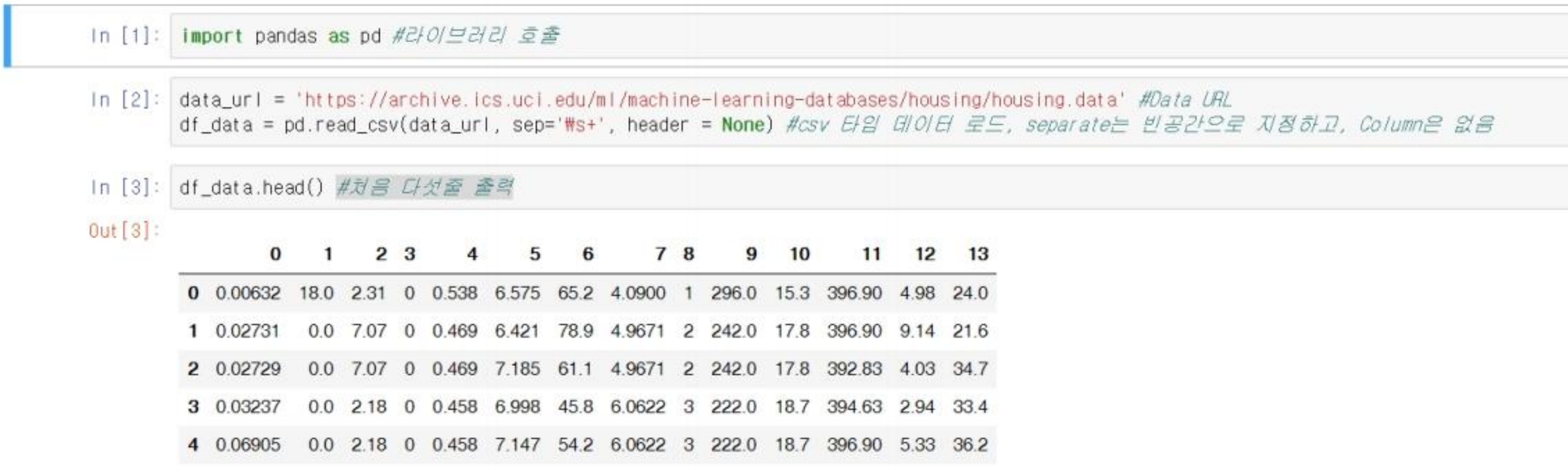


```
[ ] df.loc[0, 'review'][-50:]
```

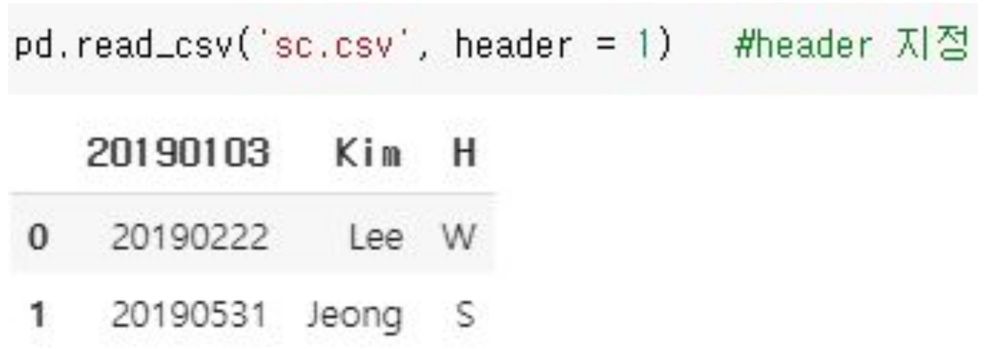
```
'is seven.<br /><br />Title (Brazil): Not Available'
```

Access a group of rows and columns by label(s) or a boolean array.

데이터 read 다른 예

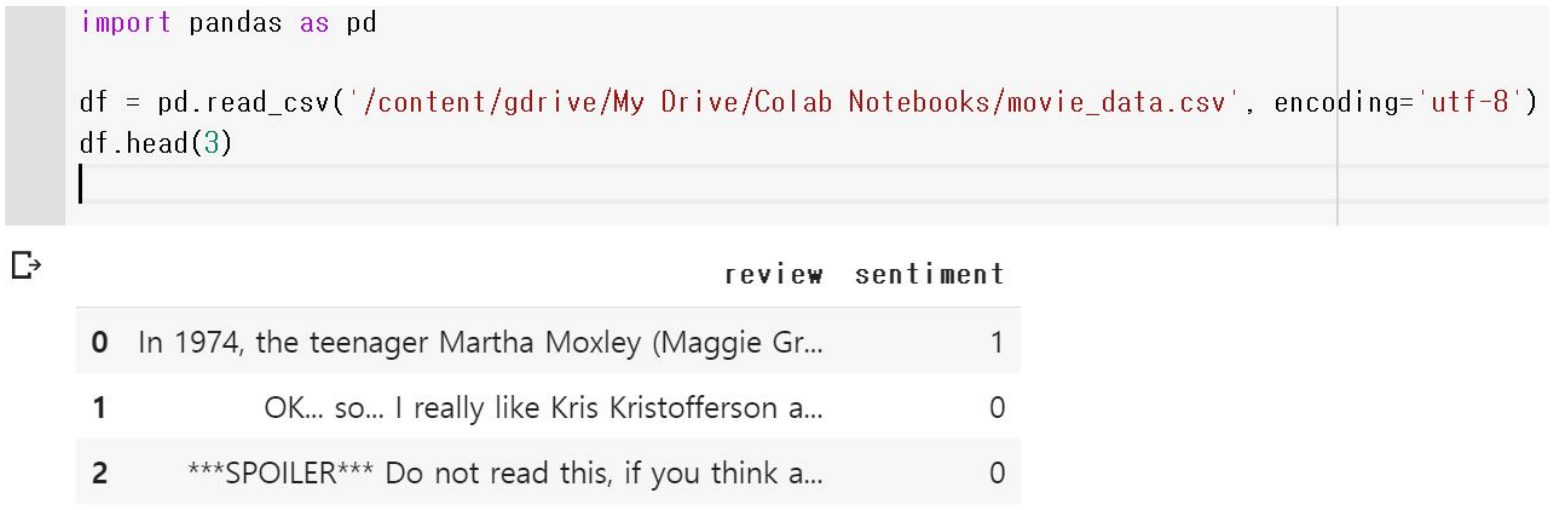


```
In [1]: import pandas as pd #라이브러리 호출

In [2]: data_url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/housing/housing.data' #Data URL
        df_data = pd.read_csv(data_url, sep='₩s+', header = None) #csv 타입 데이터 로드, separate는 빈공간으로 지정하고, Column은 없음

In [3]: df_data.head() #처음 다섯줄 출력

Out[3]:
         0     1     2  3      4      5     6       7  8      9    10      11    12    13
0  0.00632  18.0  2.31  0  0.538  6.575  65.2  4.0900  1  296.0  15.3  396.90  4.98  24.0
1  0.02731   0.0  7.07  0  0.469  6.421  78.9  4.9671  2  242.0  17.8  396.90  9.14  21.6
2  0.02729   0.0  7.07  0  0.469  7.185  61.1  4.9671  2  242.0  17.8  392.83  4.03  34.7
3  0.03237   0.0  2.18  0  0.458  6.998  45.8  6.0622  3  222.0  18.7  394.63  2.94  33.4
4  0.06905   0.0  2.18  0  0.458  7.147  54.2  6.0622  3  222.0  18.7  396.90  5.33  36.2
```



```
pd.read_csv('sc.csv', header = 1)   #header 지정

   20190103    Kim  H
0  20190222    Lee  W
1  20190531  Jeong  S
```

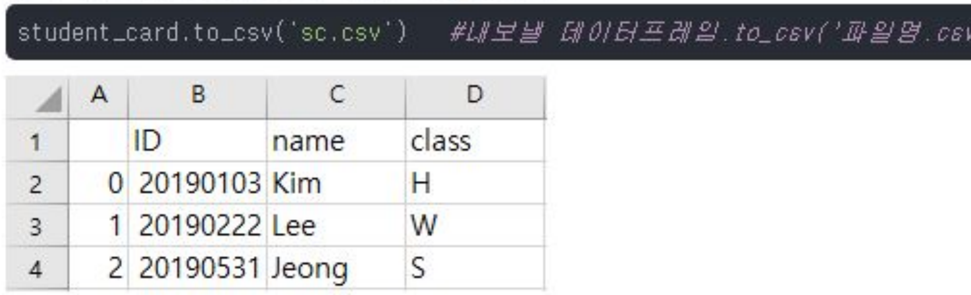
데이터 read 다른 예



```
import pandas as pd

df = pd.read_csv('/content/gdrive/My Drive/Colab Notebooks/movie_data.csv', encoding='utf-8')
df.head(3)
```

	review	sentiment
0	In 1974, the teenager Martha Moxley (Maggie Gr...	1
1	OK... so... I really like Kris Kristofferson a...	0
2	***SPOILER*** Do not read this, if you think a...	0

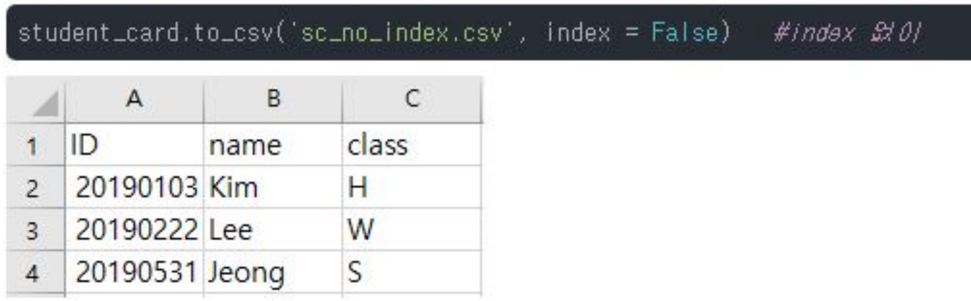
1. 데이터프레임명.to_csv('저장할 파일명.csv')



```
student_card.to_csv('sc.csv')   #내보낼 데이터프레임.to_csv('파일명.csv
```

	A	B	C	D
1		ID	name	class
2	0	20190103	Kim	H
3	1	20190222	Lee	W
4	2	20190531	Jeong	S

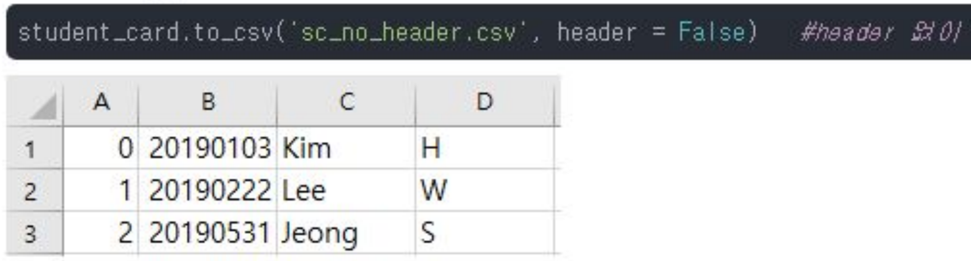
2. index 없이



```
student_card.to_csv('sc_no_index.csv', index = False)   #index 없이
```

	A	B	C
1	ID	name	class
2	20190103	Kim	H
3	20190222	Lee	W
4	20190531	Jeong	S

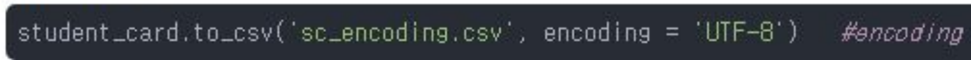
3. header 없이



```
student_card.to_csv('sc_no_header.csv', header = False)   #header 없이
```

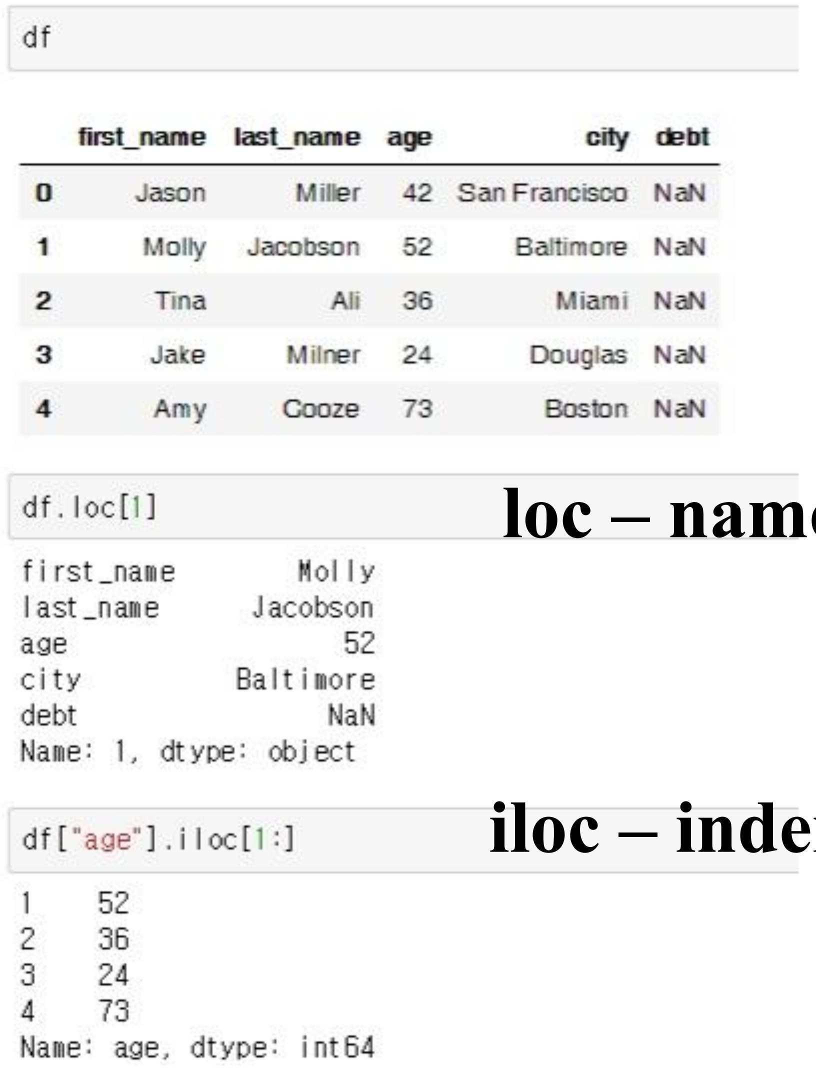
	A	B	C	D
1	0	20190103	Kim	H
2	1	20190222	Lee	W
3	2	20190531	Jeong	S

4. 인코딩 사용



```
student_card.to_csv('sc_encoding.csv', encoding = 'UTF-8')   #encoding
```

DataFrame



```
df
```

	first_name	last_name	age	city	debt
0	Jason	Miller	42	San Francisco	NaN
1	Molly	Jacobson	52	Baltimore	NaN
2	Tina	Ali	36	Miami	NaN
3	Jake	Milner	24	Douglas	NaN
4	Amy	Cooze	73	Boston	NaN

```
df.loc[1]
```

```
first_name        Molly
last_name      Jacobson
age                  52
city          Baltimore
debt                NaN
Name: 1, dtype: object
```

loc – name loc

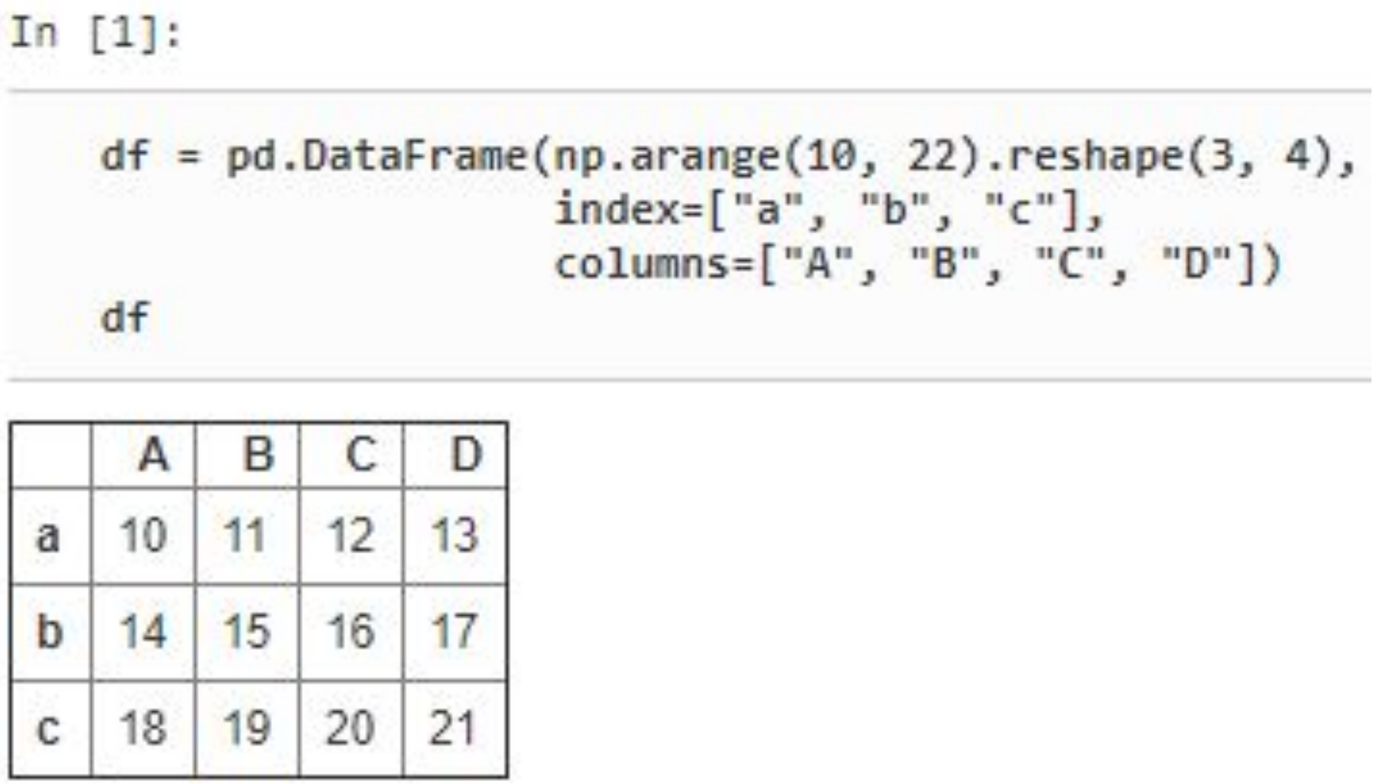
```
df["age"].iloc[1:]
```

```
1    52
2    36
3    24
4    73
Name: age, dtype: int64
```

iloc – index(number) loc

iloc

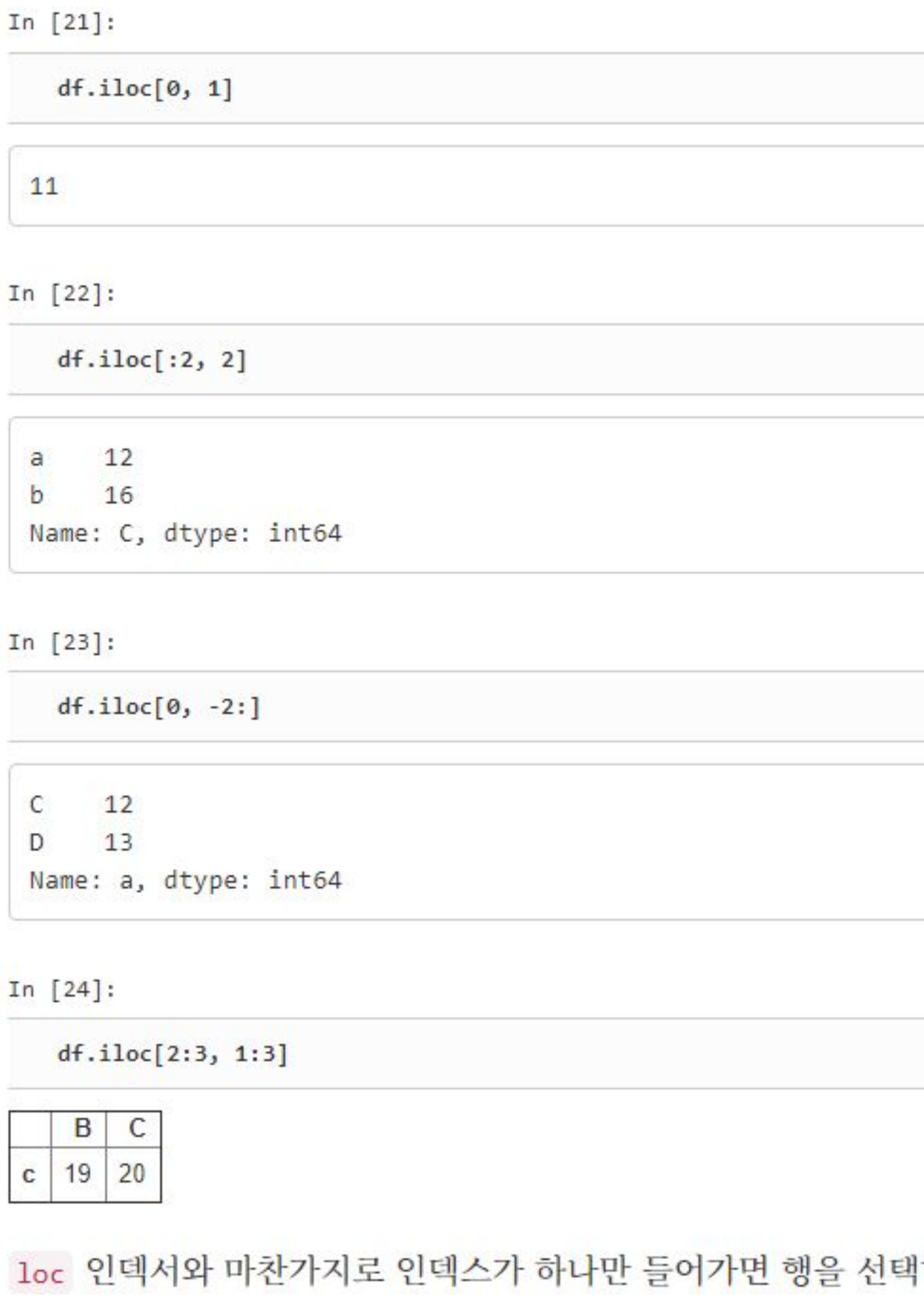
~ numpy 익숙한 분들을 위해서!



In [1]:

```
df = pd.DataFrame(np.arange(10, 22).reshape(3, 4),
                  index=["a", "b", "c"],
                  columns=["A", "B", "C", "D"])
df
```

	A	B	C	D
a	10	11	12	13
b	14	15	16	17
c	18	19	20	21



In [21]:

```
df.iloc[0, 1]
```

```
11
```

In [22]:

```
df.iloc[:2, 2]
```

```
a    12
b    16
Name: C, dtype: int64
```

In [23]:

```
df.iloc[0, -2:]
```

```
C    12
D    13
Name: a, dtype: int64
```

In [24]:

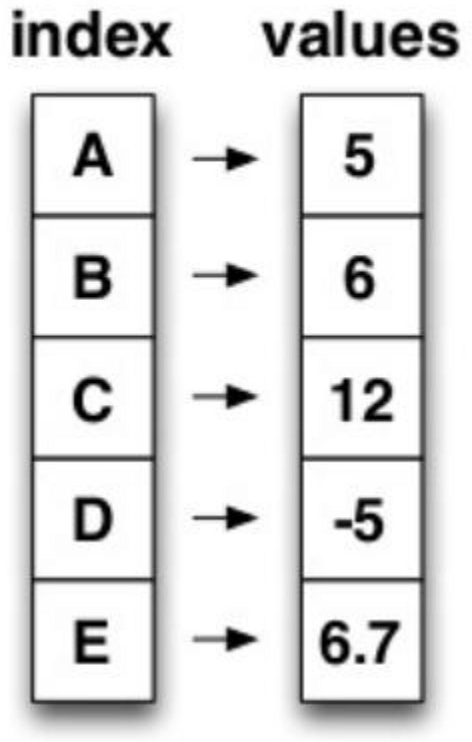
```
df.iloc[2:3, 1:3]
```

	B	C
c	19	20

`loc` 인덱서와 마찬가지로 인덱스가 하나만 들어가면 행을 선택

Series

Series

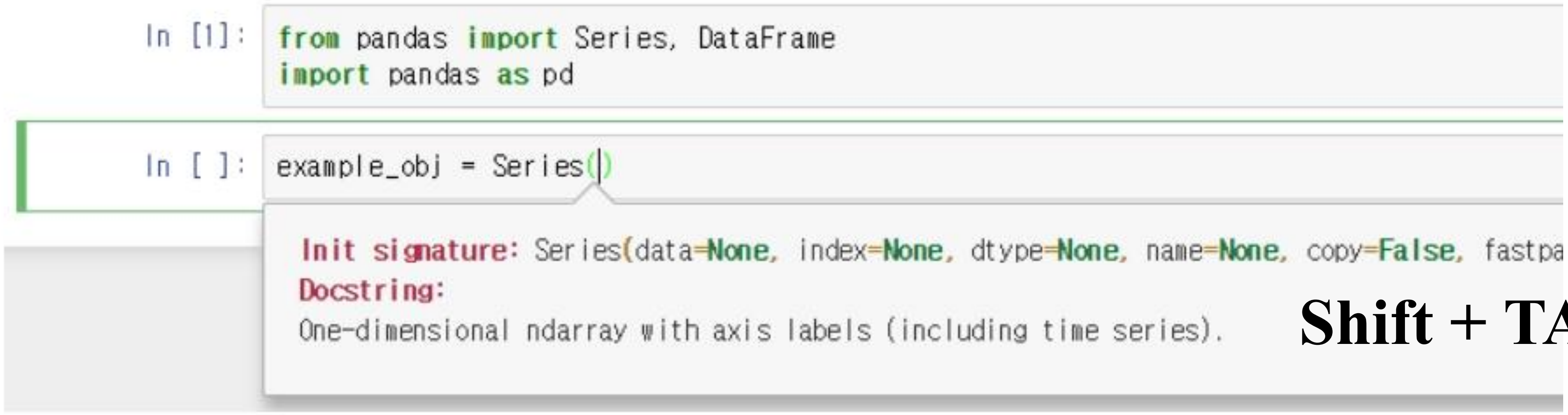


- Subclass of `numpy.ndarray`
- Data: any type
- Index labels need not be ordered
- Duplicates are possible (but result in reduced functionality)

https://www.slideshare.net/wesm/pandas-powerful-data-analysis-tools-for-python

Series

Column Vector를 표현하는 object

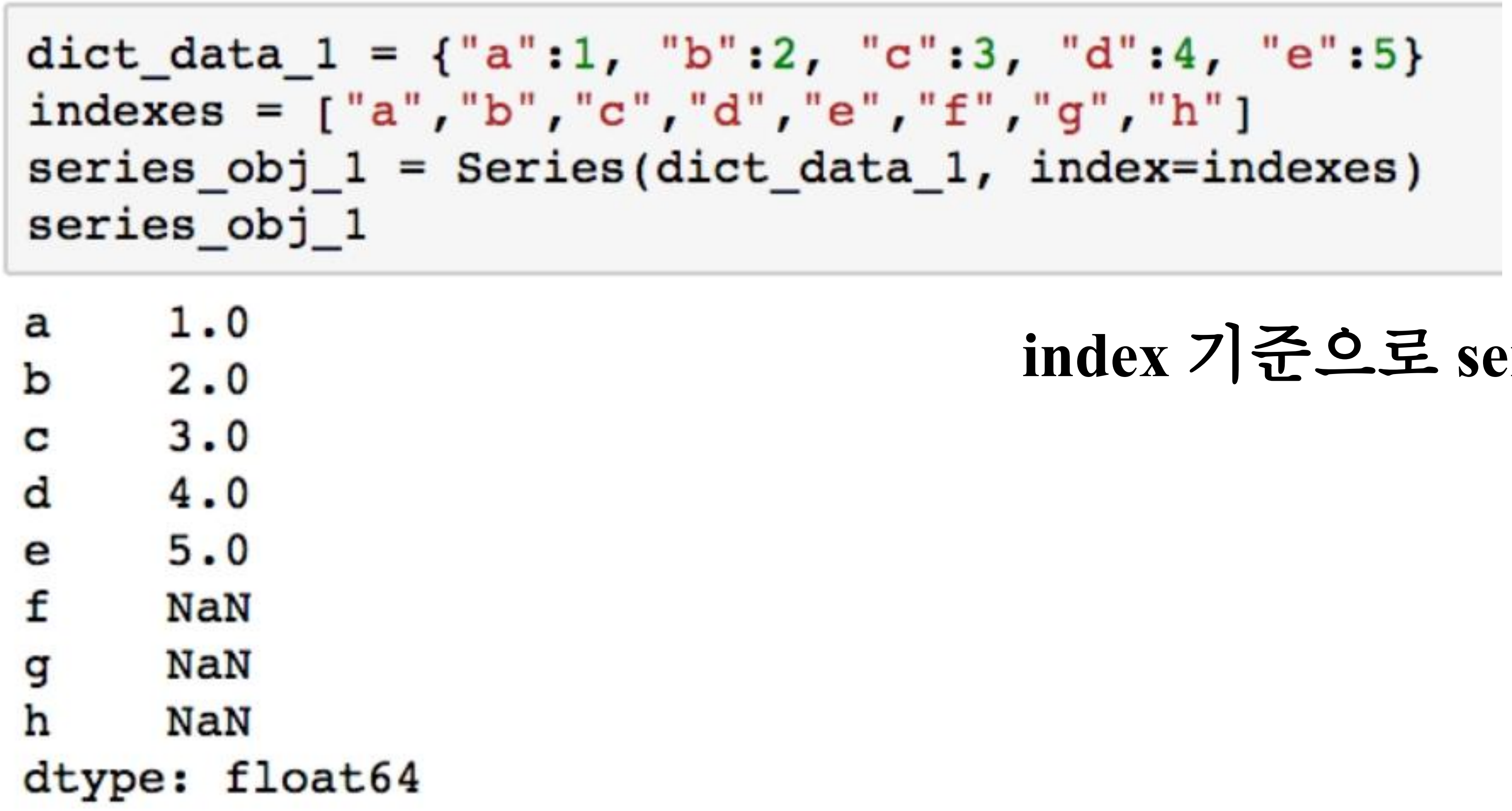


```
In [1]: from pandas import Series, DataFrame
        import pandas as pd

In [ ]: example_obj = Series()

Init signature: Series(data=None, index=None, dtype=None, name=None, copy=False, fastpa
Docstring:
One-dimensional ndarray with axis labels (including time series).
```

Shift + TAB

Series from dict

“key”: value
dict와 굉장히 유사하네요?



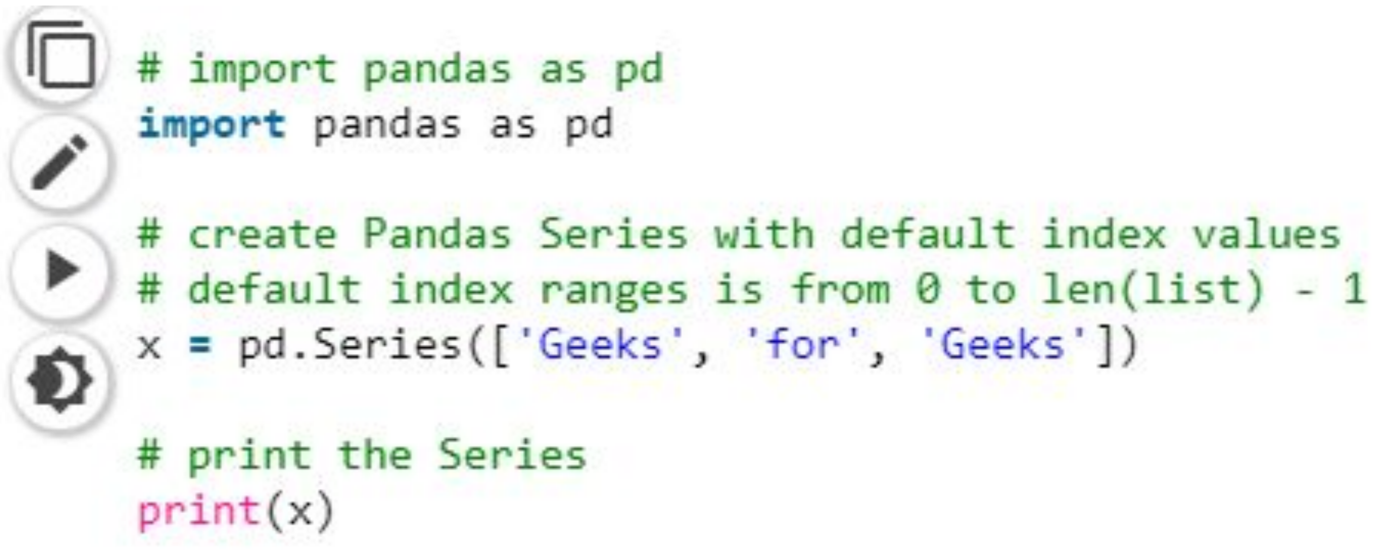
```
dict_data_1 = {"a":1, "b":2, "c":3, "d":4, "e":5}
indexes = ["a","b","c","d","e","f","g","h"]
series_obj_1 = Series(dict_data_1, index=indexes)
series_obj_1
```

```
a    1.0
b    2.0
c    3.0
d    4.0
e    5.0
f    NaN
g    NaN
h    NaN
dtype: float64
```

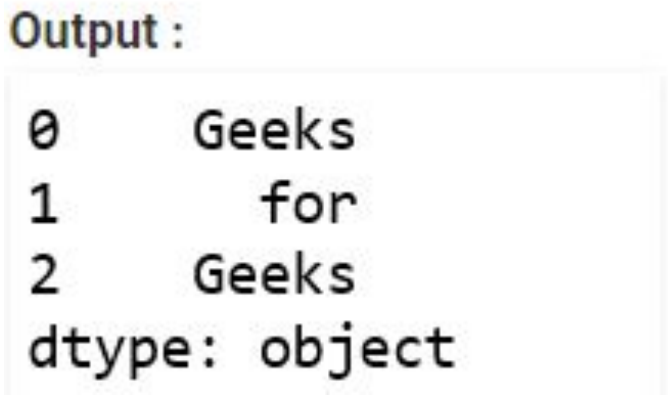
index 기준으로 series 생성

Series from list

list를 그냥 넣어도
기본 숫자 index 부여됨



```
# import pandas as pd
import pandas as pd

# create Pandas Series with default index values
# default index ranges is from 0 to len(list) - 1
x = pd.Series(['Geeks', 'for', 'Geeks'])

# print the Series
print(x)
```

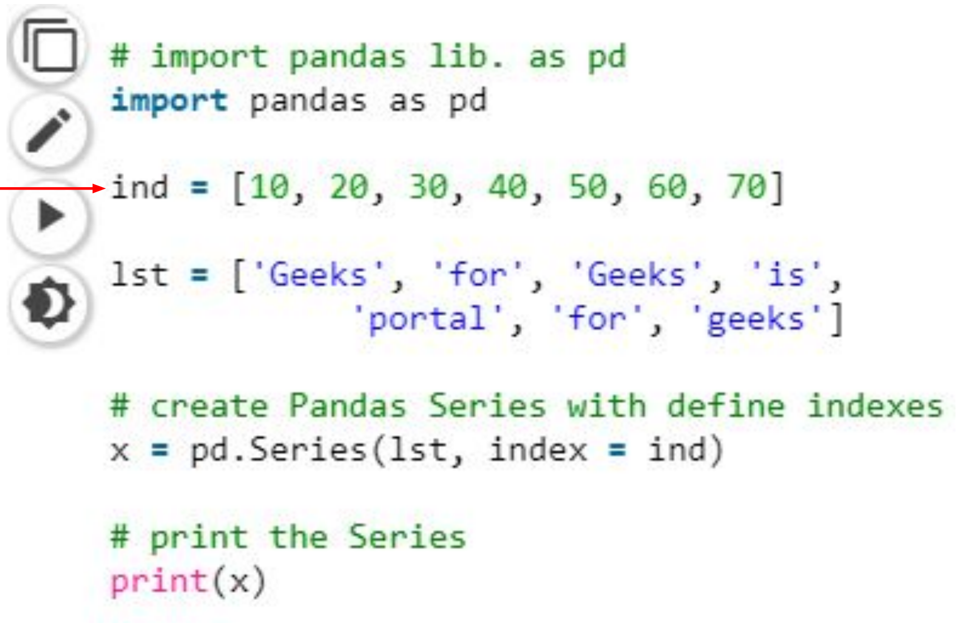


Output :

```
0    Geeks
1      for
2    Geeks
dtype: object
```

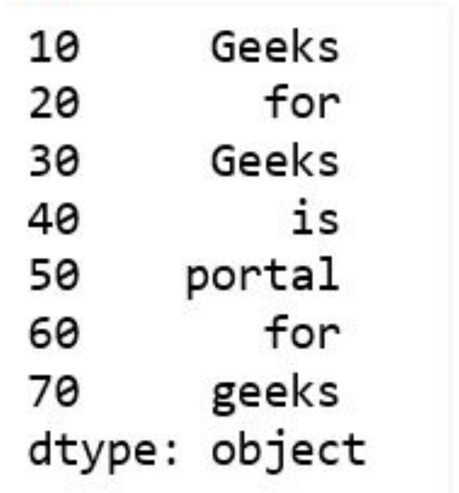
Series from list

일부러 index 네임 지정하기도 가능



```python
# import pandas lib. as pd
import pandas as pd

ind = [10, 20, 30, 40, 50, 60, 70]

lst = ['Geeks', 'for', 'Geeks', 'is',
            'portal', 'for', 'geeks']

# create Pandas Series with define indexes
x = pd.Series(lst, index = ind)

# print the Series
print(x)
```

Output:



```
10      Geeks
20        for
30      Geeks
40         is
50     portal
60        for
70      geeks
dtype: object
```

Series를 쓰는 이유 1

```
example_obj.values
```

값 리스트만

```
array([ 3.20000005,  2.        ,  3.        ,  4.        ,  5.        ],
 dtype=float32)
```

```
example_obj.index
```

Index 리스트만

```
Index(['a', 'b', 'c', 'd', 'e'], dtype='object')
```

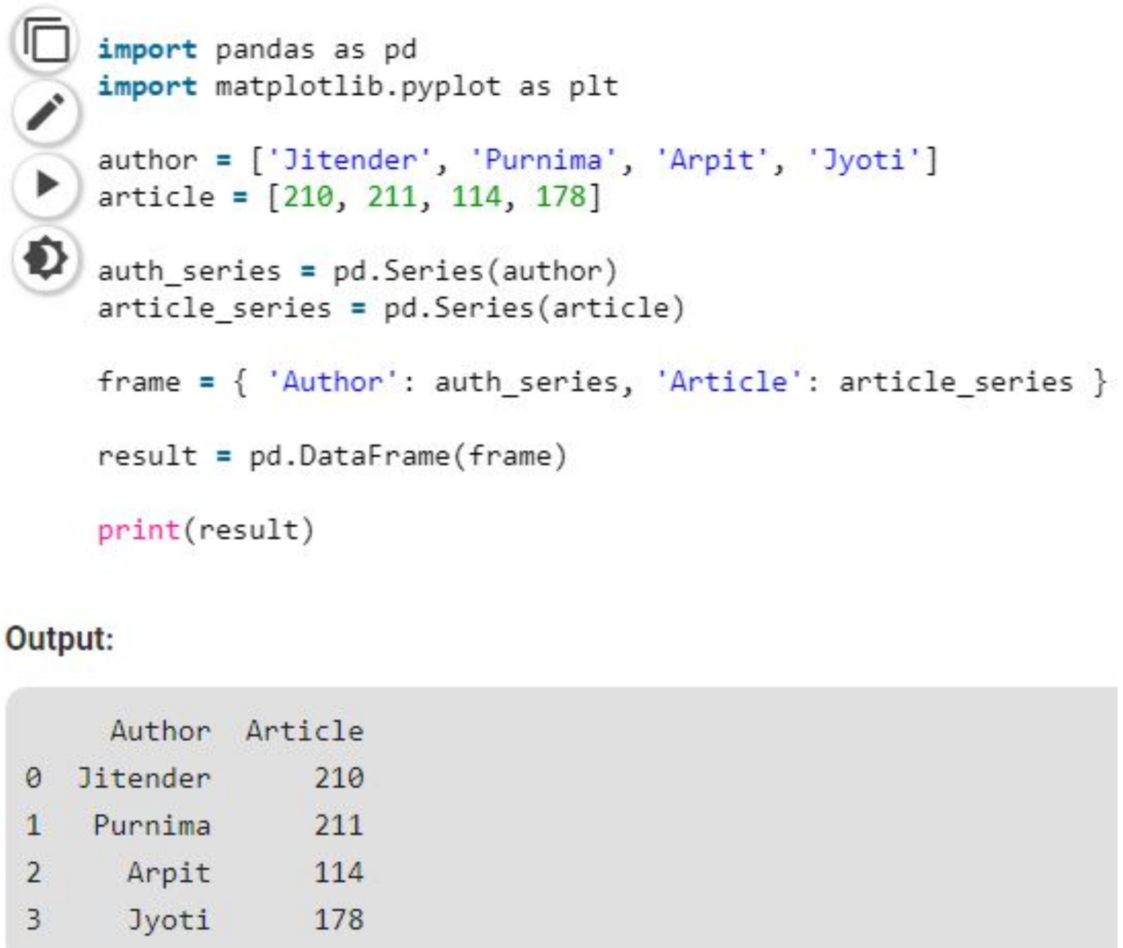
```
example_obj.name = "number"
example_obj.index.name = "alphabet"
example_obj
```

Data에 대한 정보를 저장

```
alphabet
a     3.2
b     2.0
c     3.0
d     4.0
e     5.0
Name: number, dtype: float32
```

Series를 쓰는 이유 2

Series로부터 DataFrame 만들기와 추가 쉬움



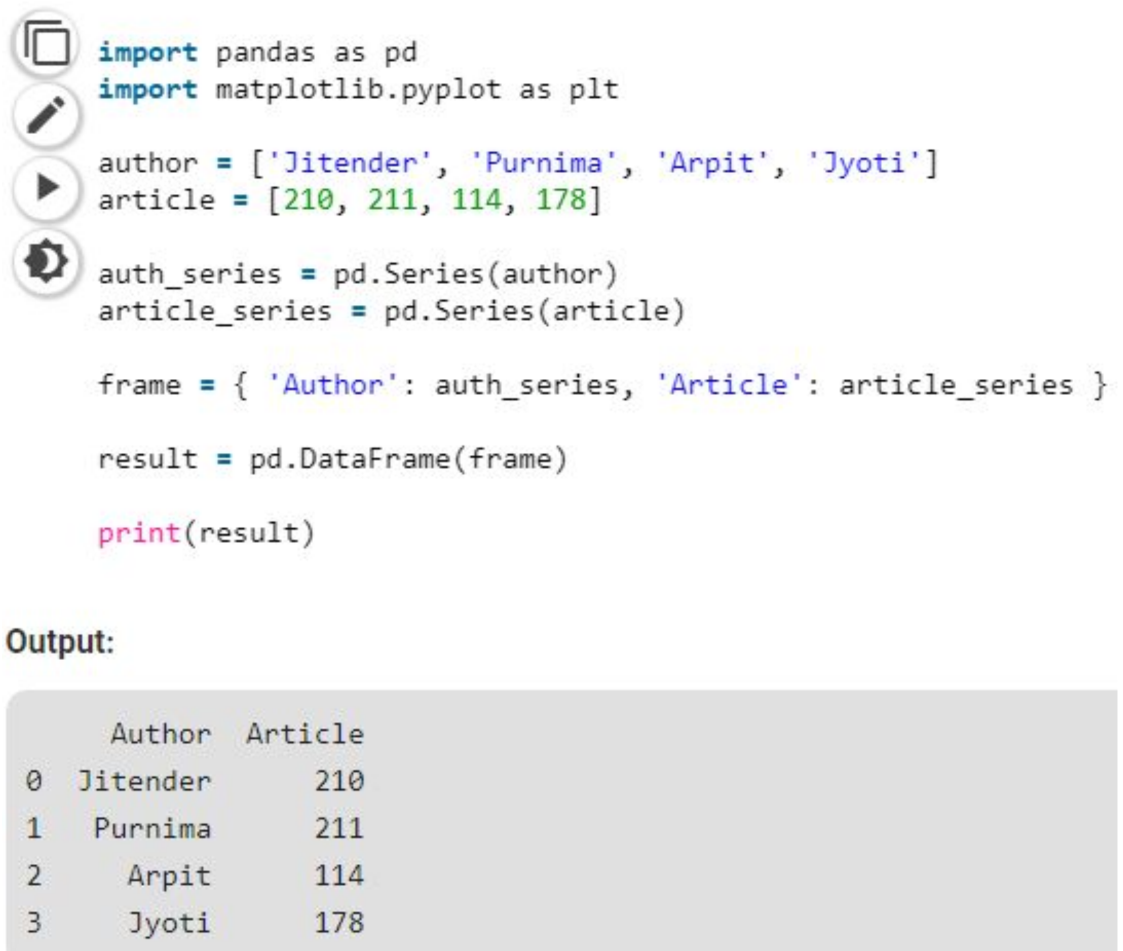
```
import pandas as pd
import matplotlib.pyplot as plt

author = ['Jitender', 'Purnima', 'Arpit', 'Jyoti']
article = [210, 211, 114, 178]

auth_series = pd.Series(author)
article_series = pd.Series(article)

frame = { 'Author': auth_series, 'Article': article_series }

result = pd.DataFrame(frame)

print(result)
```

Output:

```
     Author  Article
0  Jitender      210
1   Purnima      211
2     Arpit      114
3     Jyoti      178
```

Series를 쓰는 이유 2

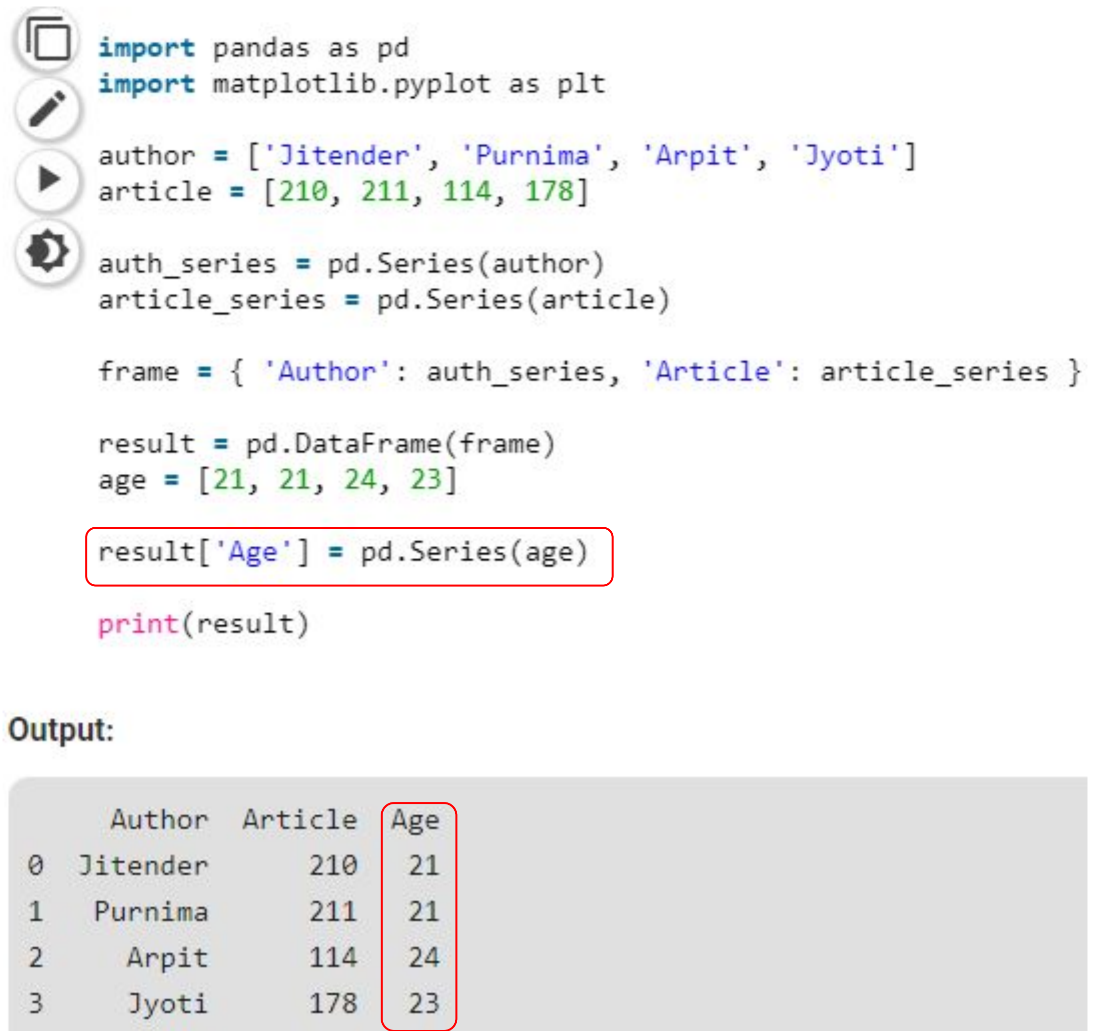
Series로부터 DataFrame 만들기와 추가 쉬움



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

author = ['Jitender', 'Purnima', 'Arpit', 'Jyoti']
article = [210, 211, 114, 178]

auth_series = pd.Series(author)
article_series = pd.Series(article)

frame = { 'Author': auth_series, 'Article': article_series }

result = pd.DataFrame(frame)

print(result)
```

Output:

```
     Author  Article
0  Jitender      210
1   Purnima      211
2     Arpit      114
3     Jyoti      178
```



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

author = ['Jitender', 'Purnima', 'Arpit', 'Jyoti']
article = [210, 211, 114, 178]

auth_series = pd.Series(author)
article_series = pd.Series(article)

frame = { 'Author': auth_series, 'Article': article_series }

result = pd.DataFrame(frame)
age = [21, 21, 24, 23]

result['Age'] = pd.Series(age)

print(result)
```

Output:

```
     Author  Article  Age
0  Jitender      210   21
1   Purnima      211   21
2     Arpit      114   24
3     Jyoti      178   23
```

Selection & Drop

Selection with column names

```
df["account"].head(3)
```

1개의 column 선택시

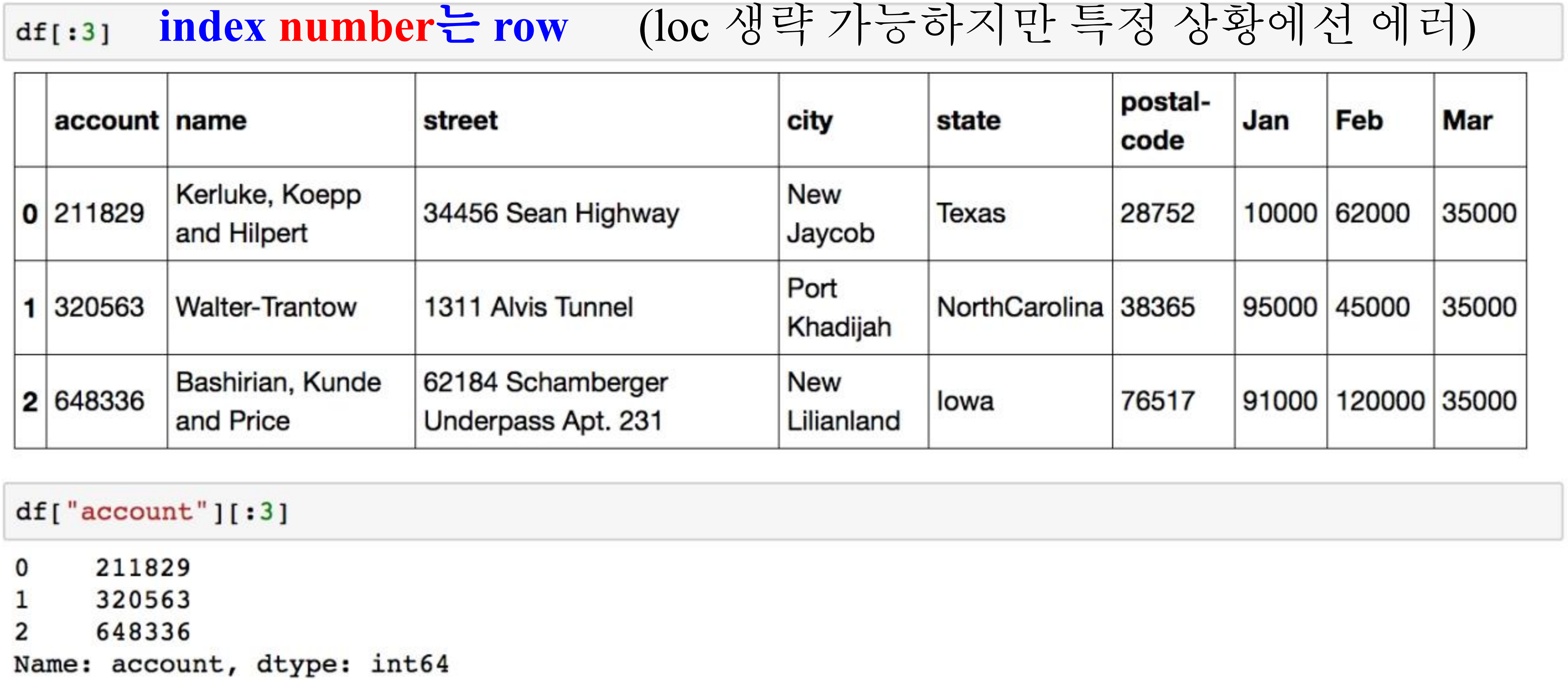
```
0     211829
1     320563
2     648336
Name: account, dtype: int64
```

다수의 column 선택시

```
df[["account", "street", "state"]].head(3)
```

	account	street	state
0	211829	34456 Sean Highway	Texas
1	320563	1311 Alvis Tunnel	NorthCarolina
2	648336	62184 Schamberger Underpass Apt. 231	Iowa

Selection with index number



```
df[:3]
```

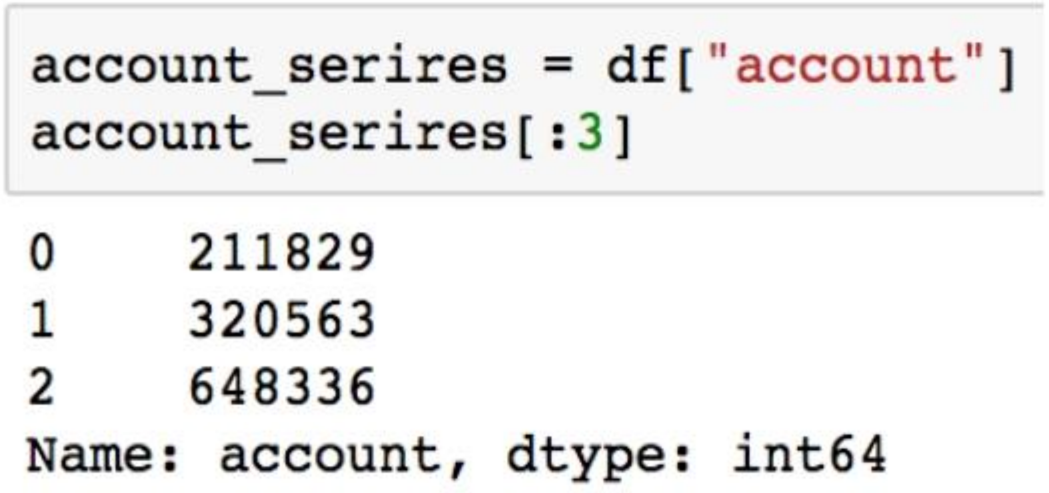
index number는 row (loc 생략 가능하지만 특정 상황에선 에러)

	account	name	street	city	state	postal-code	Jan	Feb	Mar
0	211829	Kerluke, Koepp and Hilpert	34456 Sean Highway	New Jaycob	Texas	28752	10000	62000	35000
1	320563	Walter-Trantow	1311 Alvis Tunnel	Port Khadijah	NorthCarolina	38365	95000	45000	35000
2	648336	Bashirian, Kunde and Price	62184 Schamberger Underpass Apt. 231	New Lilianland	Iowa	76517	91000	120000	35000

```
df["account"][:3]
```

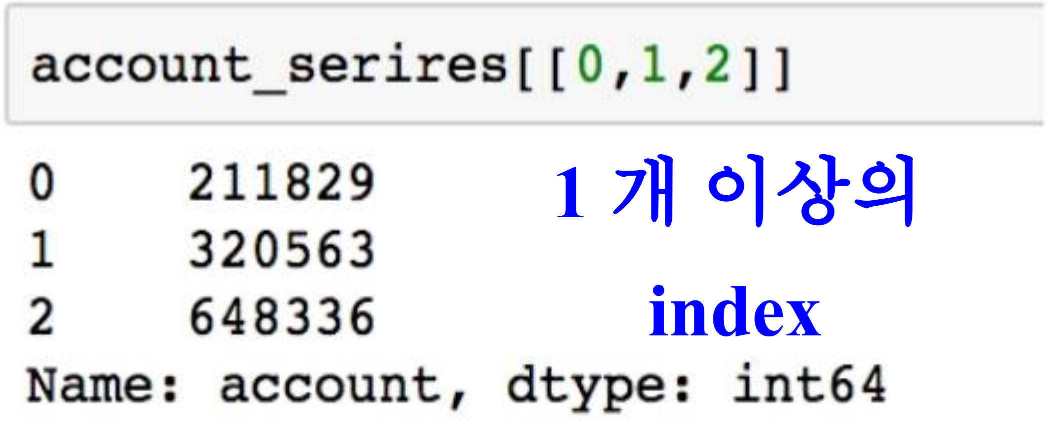
```
0    211829
1    320563
2    648336
Name: account, dtype: int64
```

Series selection



```
account_serires = df["account"]
account_serires[:3]
```

```
0      211829
1      320563
2      648336
Name: account, dtype: int64
```

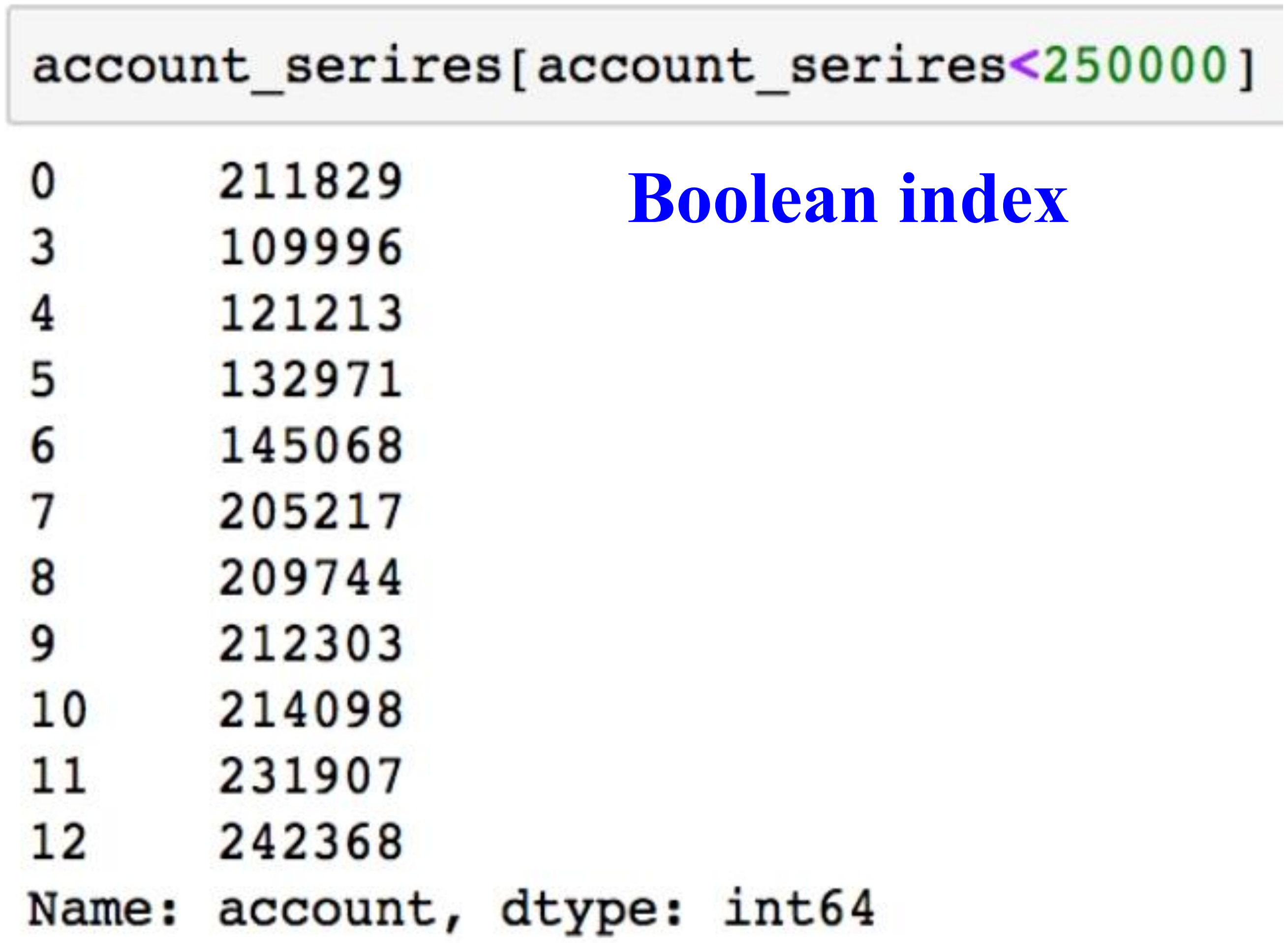


```
account_serires[[0,1,2]]
```

```
0      211829
1      320563
2      648336
Name: account, dtype: int64
```

1 개 이상의

index

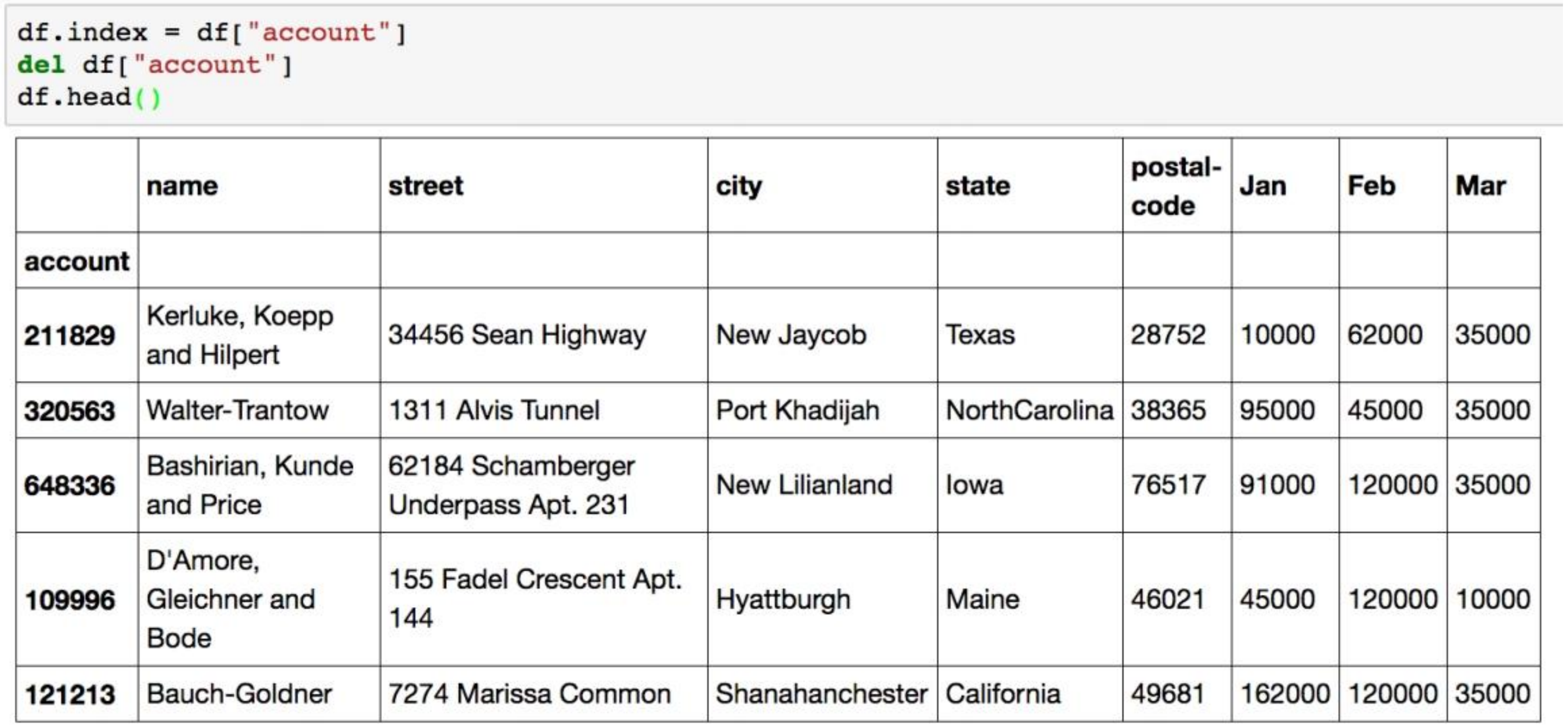


```
account_serires[account_serires<250000]
```

```
0       211829
3       109996
4       121213
5       132971
6       145068
7       205217
8       209744
9       212303
10      214098
11      231907
12      242368
Name: account, dtype: int64
```

Boolean index

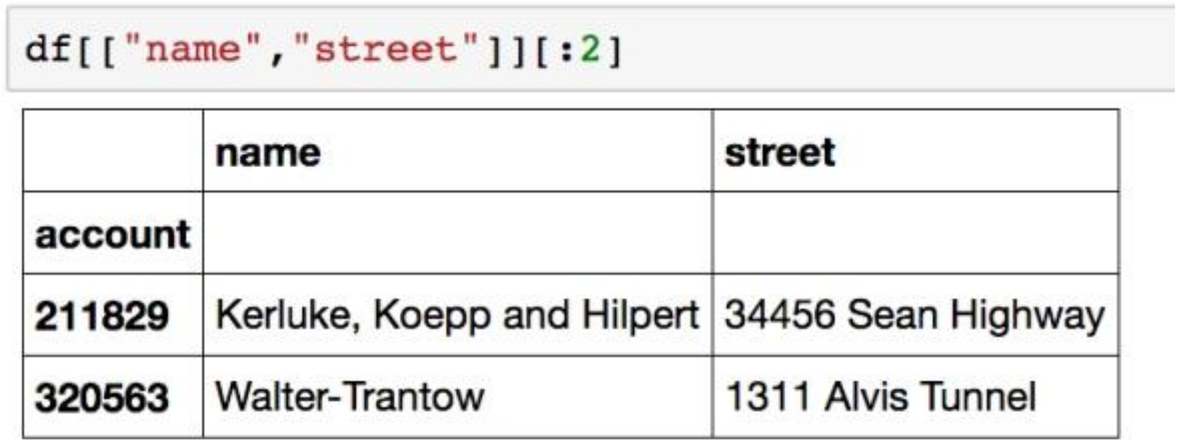
Index 변경



```python
df.index = df["account"]
del df["account"]
df.head()
```

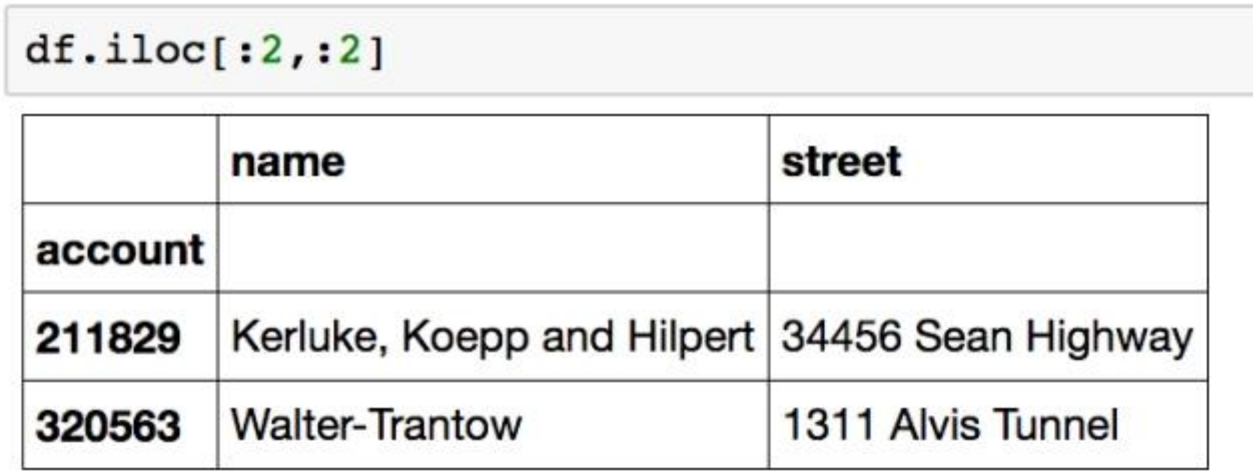
	name	street	city	state	postal-code	Jan	Feb	Mar
account								
211829	Kerluke, Koepp and Hilpert	34456 Sean Highway	New Jaycob	Texas	28752	10000	62000	35000
320563	Walter-Trantow	1311 Alvis Tunnel	Port Khadijah	NorthCarolina	38365	95000	45000	35000
648336	Bashirian, Kunde and Price	62184 Schamberger Underpass Apt. 231	New Lilianland	Iowa	76517	91000	120000	35000
109996	D'Amore, Gleichner and Bode	155 Fadel Crescent Apt. 144	Hyattburgh	Maine	46021	45000	120000	10000
121213	Bauch-Goldner	7274 Marissa Common	Shanahanchester	California	49681	162000	120000	35000

Basic, loc, iloc selection



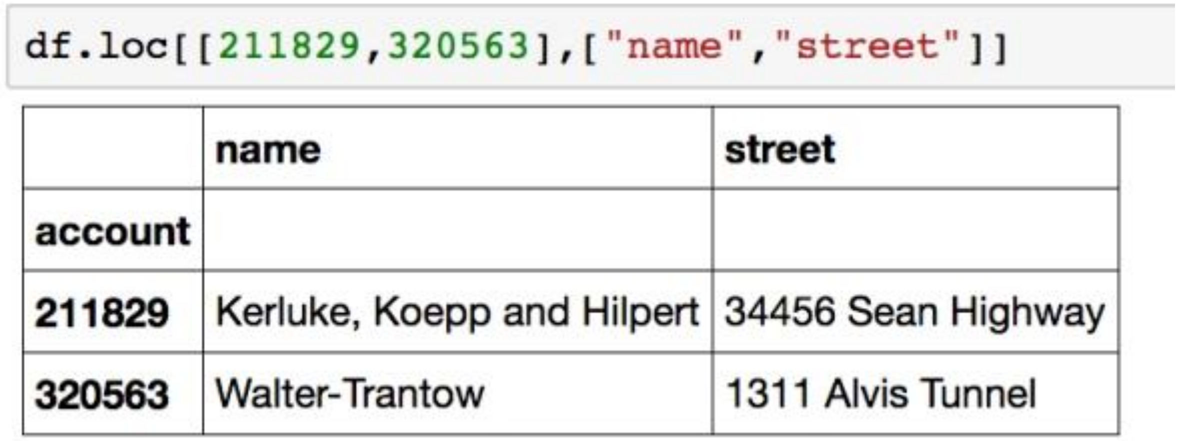
```
df[["name","street"]][:2]
```

	name	street
account		
211829	Kerluke, Koepp and Hilpert	34456 Sean Highway
320563	Walter-Trantow	1311 Alvis Tunnel



```
df.iloc[:2,:2]
```

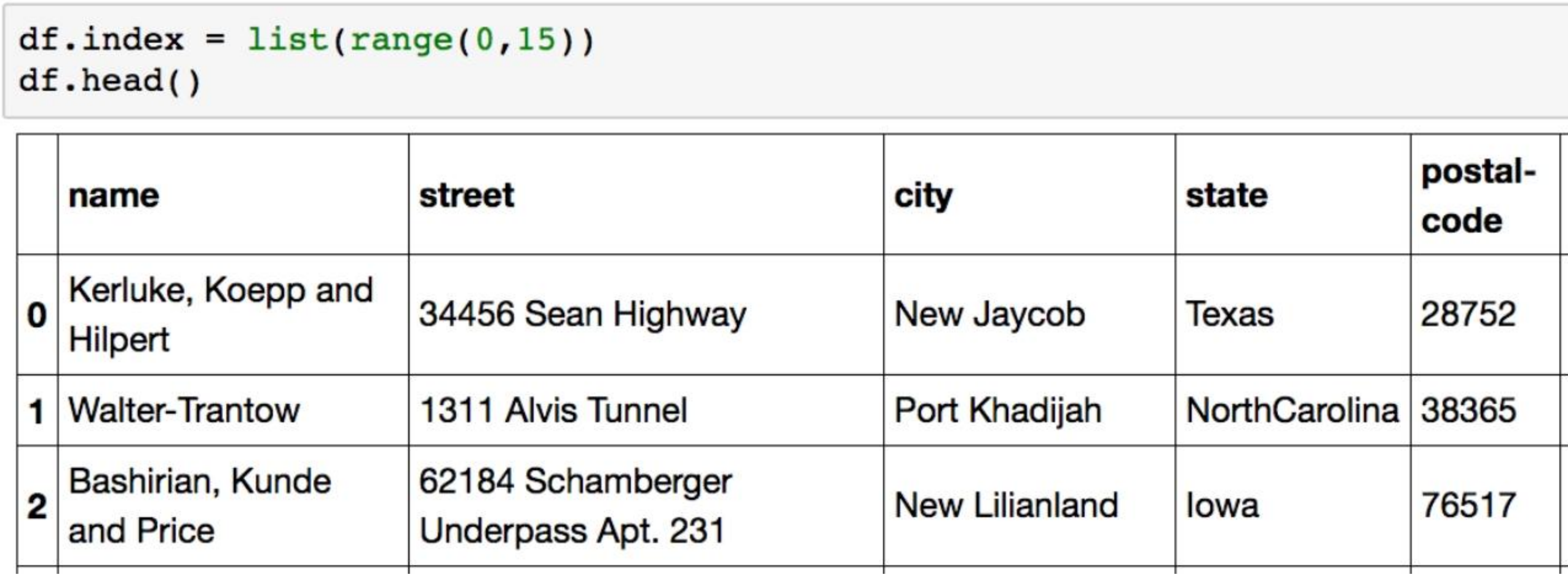
	name	street
account		
211829	Kerluke, Koepp and Hilpert	34456 Sean Highway
320563	Walter-Trantow	1311 Alvis Tunnel



```
df.loc[[211829,320563],["name","street"]]
```

	name	street
account		
211829	Kerluke, Koepp and Hilpert	34456 Sean Highway
320563	Walter-Trantow	1311 Alvis Tunnel

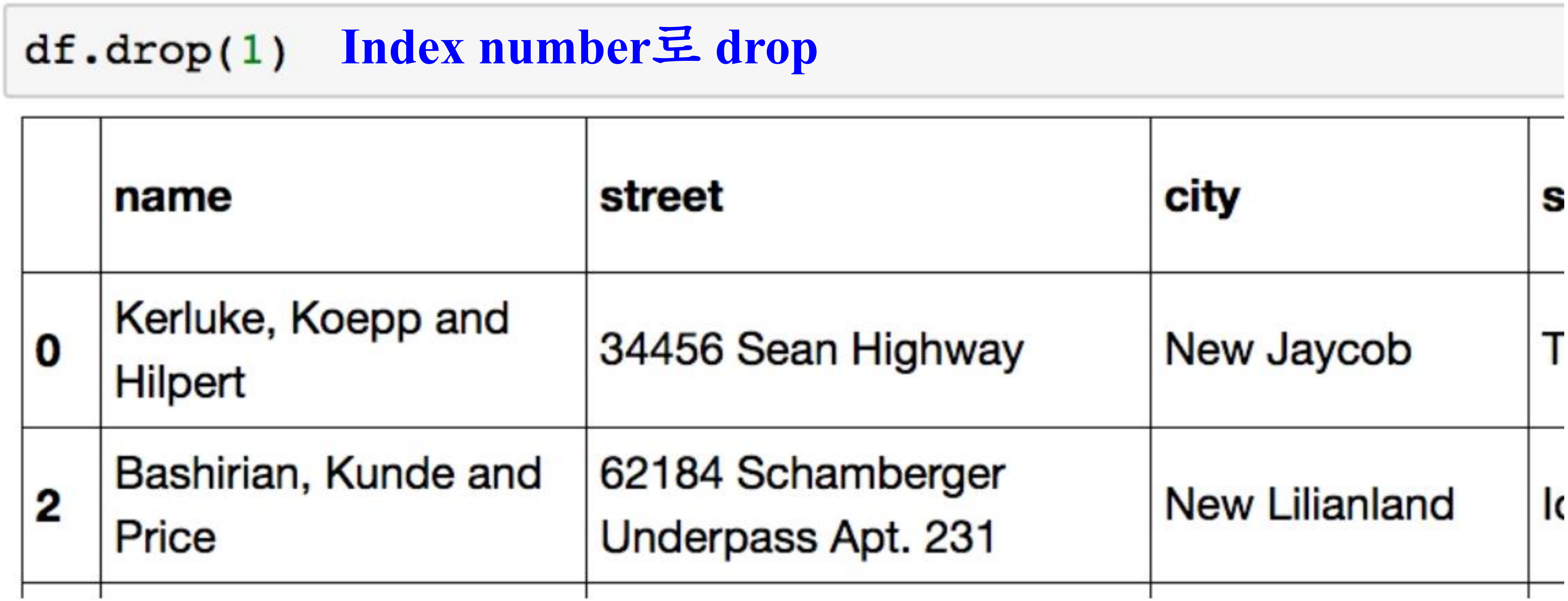
index 재설정



```
df.index = list(range(0,15))
df.head()
```

	name	street	city	state	postal-code
0	Kerluke, Koepp and Hilpert	34456 Sean Highway	New Jaycob	Texas	28752
1	Walter-Trantow	1311 Alvis Tunnel	Port Khadijah	NorthCarolina	38365
2	Bashirian, Kunde and Price	62184 Schamberger Underpass Apt. 231	New Lilianland	Iowa	76517

Data drop

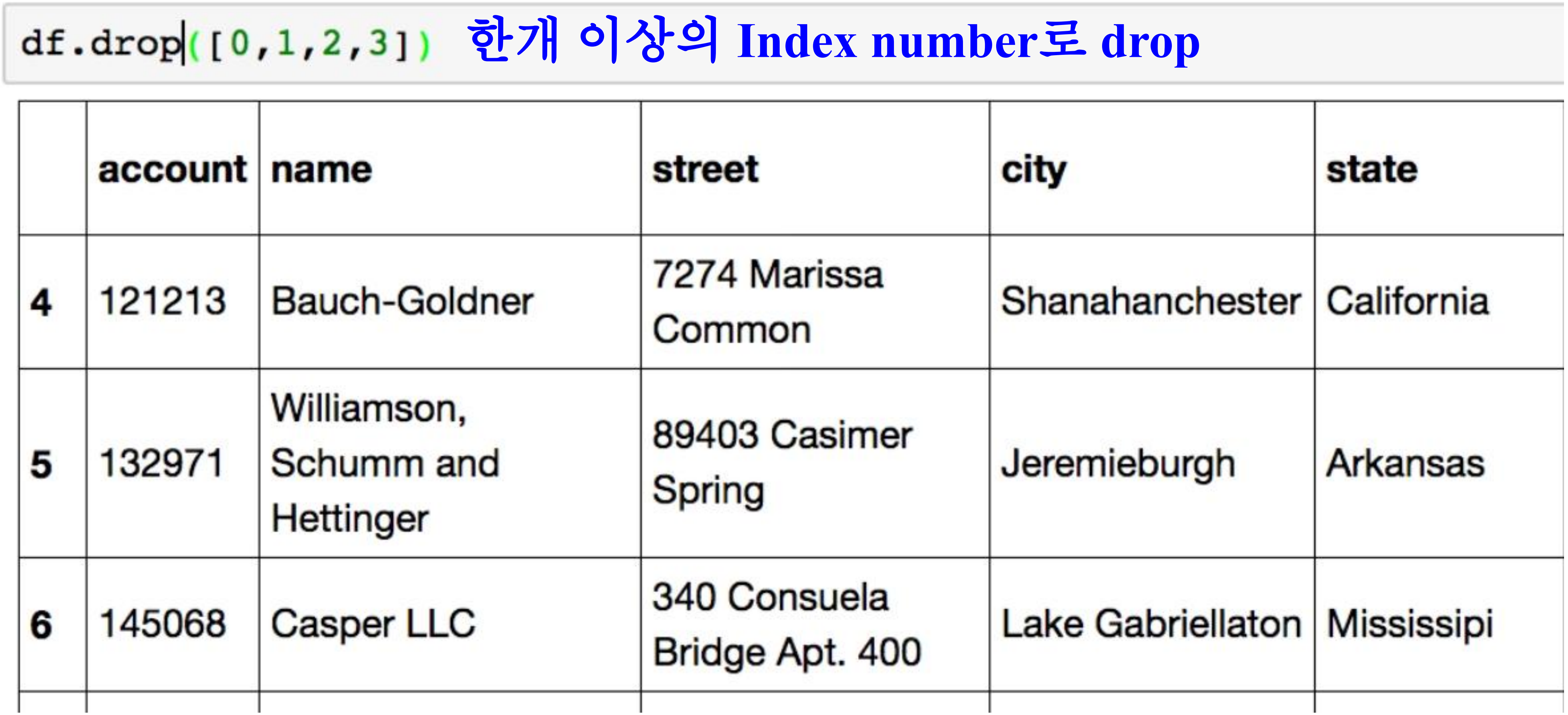


```
df.drop(1)
```

Index number로 drop

	name	street	city	s
0	Kerluke, Koepp and Hilpert	34456 Sean Highway	New Jaycob	T
2	Bashirian, Kunde and Price	62184 Schamberger Underpass Apt. 231	New Lilianland	I

Data drop



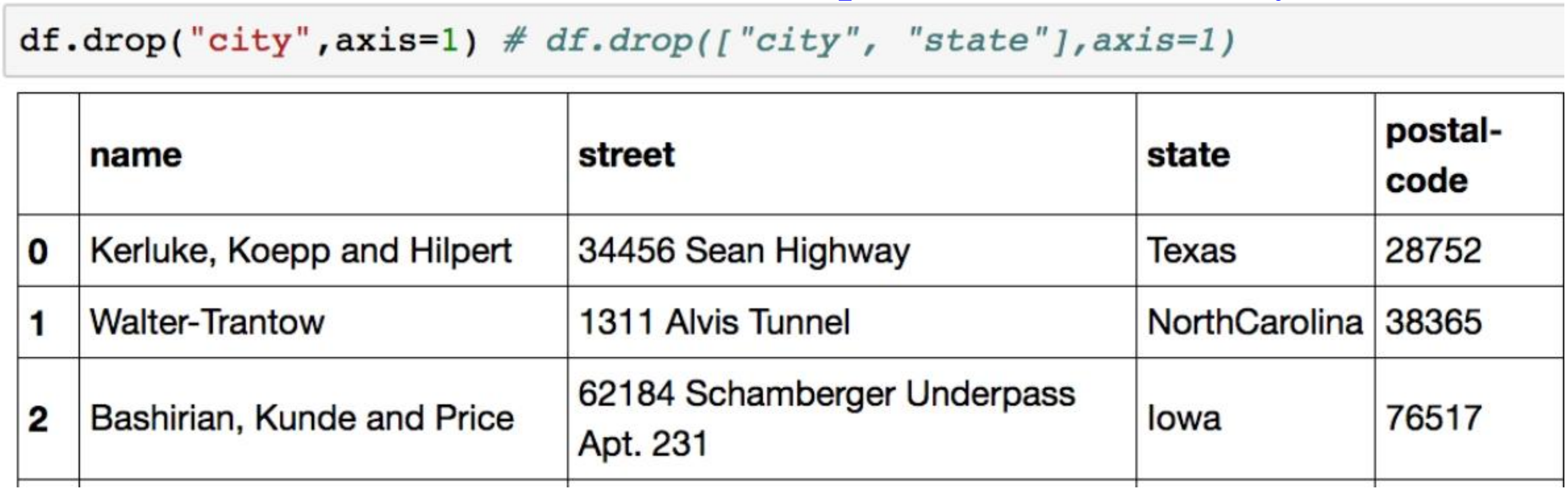
```
df.drop([0,1,2,3])
```

한개 이상의 Index number로 drop

	account	name	street	city	state
4	121213	Bauch-Goldner	7274 Marissa Common	Shanahanchester	California
5	132971	Williamson, Schumm and Hettinger	89403 Casimer Spring	Jeremieburgh	Arkansas
6	145068	Casper LLC	340 Consuela Bridge Apt. 400	Lake Gabriellaton	Mississipi

Data drop

axis 지정으로 축을 기준으로 drop column 중에 “city”

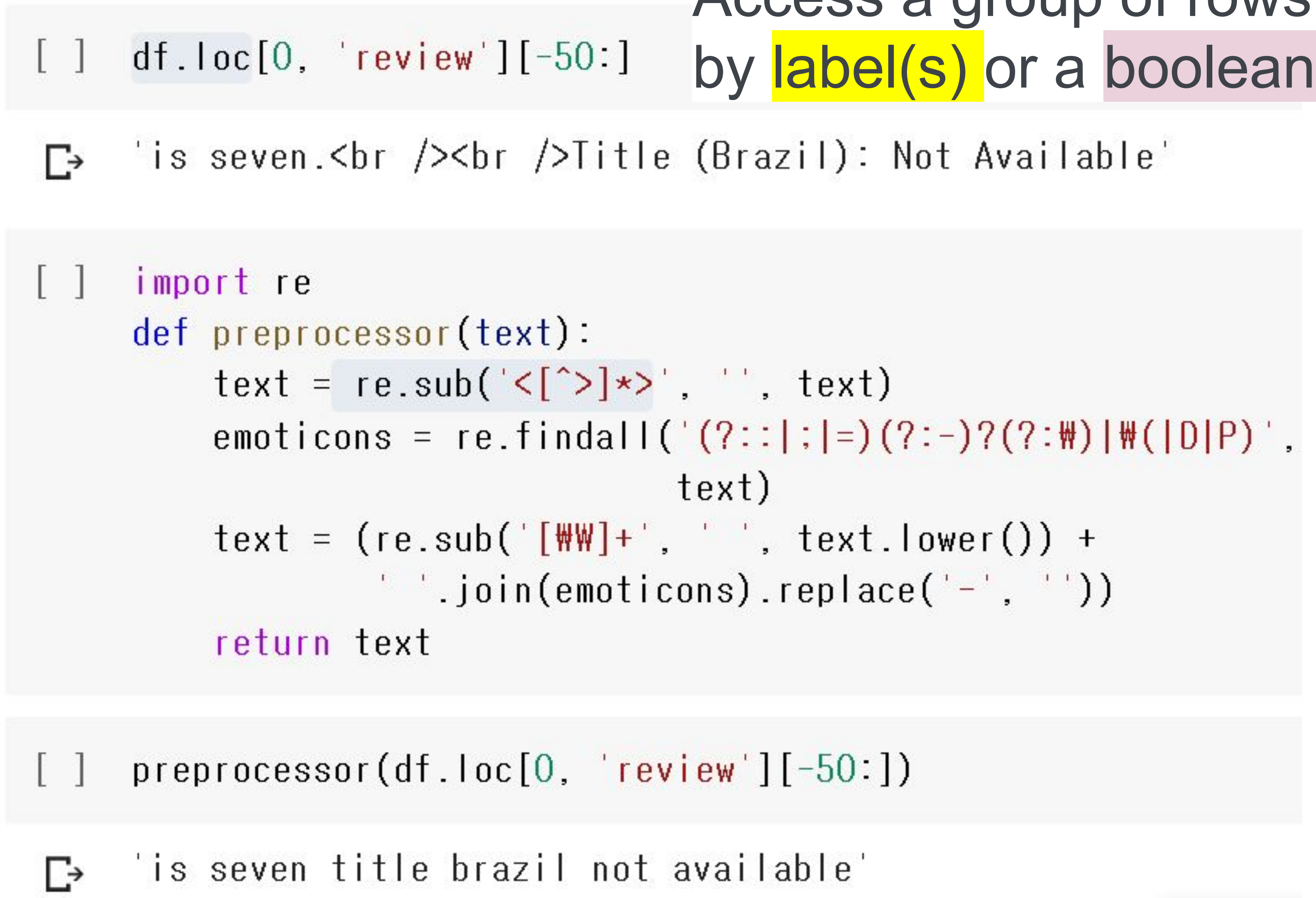


```
df.drop("city",axis=1) # df.drop(["city", "state"],axis=1)
```

	name	street	state	postal-code
0	Kerluke, Koepp and Hilpert	34456 Sean Highway	Texas	28752
1	Walter-Trantow	1311 Alvis Tunnel	NorthCarolina	38365
2	Bashirian, Kunde and Price	62184 Schamberger Underpass Apt. 231	Iowa	76517

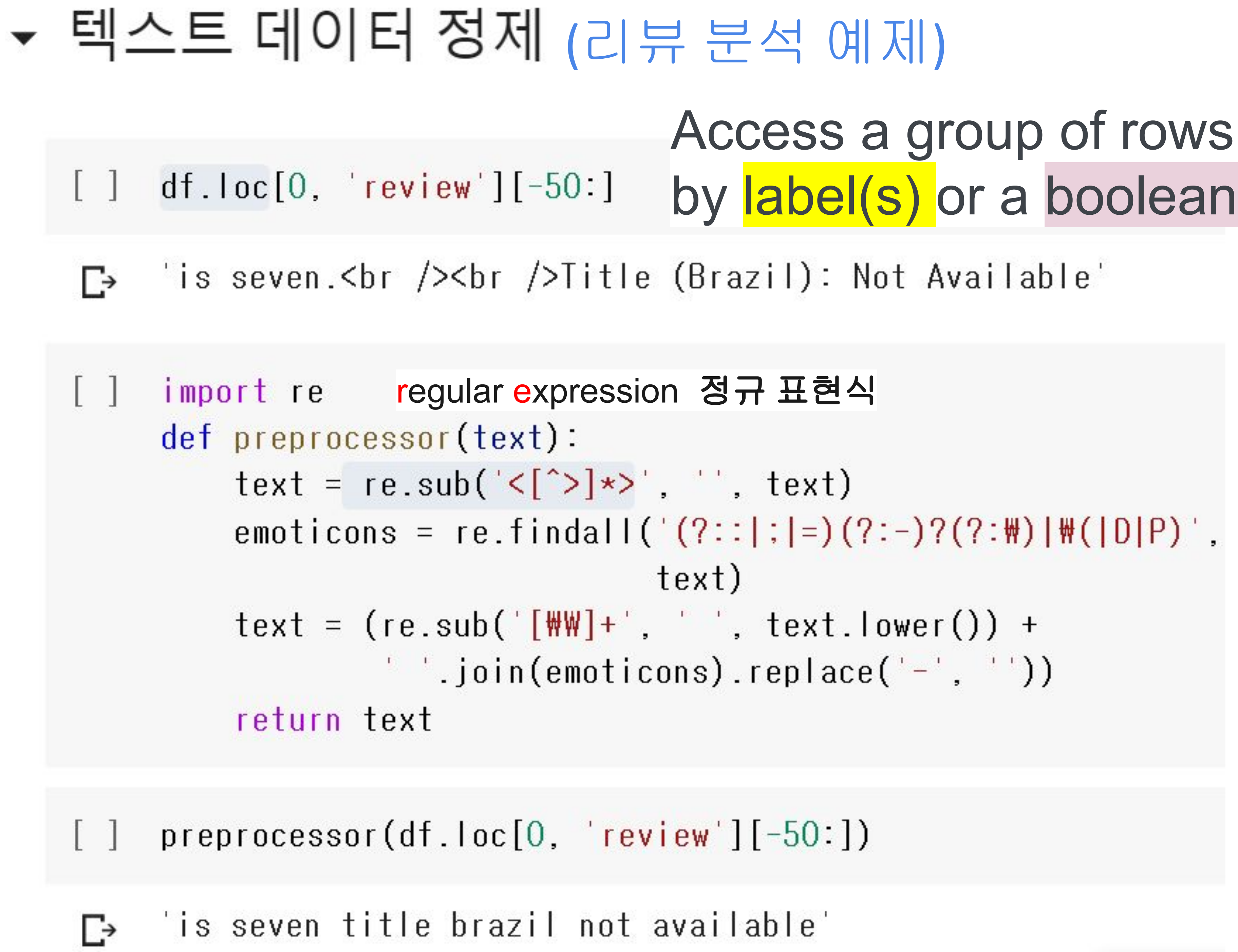
텍스트 데이터 정제 (리뷰 분석 예제)

Access a group of rows and columns by label(s) or a boolean array.



```
[ ]  df.loc[0, 'review'][-50:]

     'is seven.<br /><br />Title (Brazil): Not Available'

[ ]  import re
     def preprocessor(text):
         text = re.sub('<[^>]*>', '', text)
         emoticons = re.findall('(?::|;|=)(?:-)?(?:₩)|₩(|D|P)',
                                text)
         text = (re.sub('[₩W]+', ' ', text.lower()) +
                 ' '.join(emoticons).replace('-', ''))
         return text

[ ]  preprocessor(df.loc[0, 'review'][-50:])

     'is seven title brazil not available'
```

텍스트 데이터 정제 (리뷰 분석 예제)



Access a group of rows and columns by label(s) or a boolean array.

```
[ ]  df.loc[0, 'review'][-50:]
```

```
'is seven.<br /><br />Title (Brazil): Not Available'
```

regular expression 정규 표현식

```
[ ]  import re
     def preprocessor(text):
         text = re.sub('<[^>]*>', '', text)
         emoticons = re.findall('(?::|;|=)(?:-)?(?:₩)|₩(|D|P)',
                                text)
         text = (re.sub('[₩W]+', ' ', text.lower()) +
                 ' '.join(emoticons).replace('-', ''))
         return text
```

```
[ ]  preprocessor(df.loc[0, 'review'][-50:])
```

```
'is seven title brazil not available'
```