

```
In [1]: a = 1
        b = 3
        c = a + b
        print(c)
```

4

```
In [2]: # 주의할 점: cell을 실행시킨 순서대로 반영 됨
        d = c + 3
        d
```

Out[2]: 7

```
In [3]: c = 100
```

```
In [4]: c
```

Out[4]: 100

## Pandas

- 데이터 분석가들이 많이 사용
- 파이썬에서 엑셀에서 하는 작업들을 하고 싶을 때 사용
- 엑셀보다 훨씬 더 자유롭고 강력한 기능

```
In [5]: # DataFrame 선언하기

import pandas as pd

df = pd.DataFrame([[1, 2, 3], [4, 5, 6]])
df
```

```
Out[5]:
```

	0	1	2
0	1	2	3
1	4	5	6

```
In [6]: # column 설정하기
df.columns = ['a', 'b', 'c']
df
```

```
Out[6]:
```

	a	b	c
0	1	2	3
1	4	5	6

```
In [7]: df.columns
```

Out[7]: Index(['a', 'b', 'c'], dtype='object')

```
In [8]: # dataframe 선언 할 때, column까지 결정하기
df_a = pd.DataFrame([[10, 11, 12], [13, 14, 15]], columns=['aa', 'bb', 'cc'])
df_a
```

```
Out[8]:
```

	aa	bb	cc
0	10	11	12
1	13	14	15

```
In [9]: # 새로운 cell 만들기
df['d'] = df['a'] + df['b']
df
```

```
Out[9]:
```

	a	b	c	d
0	1	2	3	3
1	4	5	6	9

```
In [10]: # sum 만들기
df['sum'] = df.sum(axis=1)
df
```

```
Out[10]:
```

	a	b	c	d	sum
0	1	2	3	3	9
1	4	5	6	9	24

```
In [11]: # axis=0 으로 하면 어떻게 될까?
df_sum = df.sum(axis=0)
df_sum
```

```
Out[11]:
```

a	5
b	7
c	9
d	12
sum	33

dtype: int64

```
In [12]: # df에 df_sum을 행으로 붙여보자
df_sum = pd.DataFrame(df_sum).T
# df_sum.columns = df.columns
display(df_sum)

df = pd.concat([df, df_sum], ignore_index=True)
df
```

	a	b	c	d	sum
0	5	7	9	12	33

```
Out[12]:
```

	a	b	c	d	sum
0	1	2	3	3	9
1	4	5	6	9	24
2	5	7	9	12	33

```
In [13]: df.index
```

```
Out[13]: RangeIndex(start=0, stop=3, step=1)
```

```
In [14]: df.index = [0, 1, 'Total']  
df
```

```
Out[14]:
```

	a	b	c	d	sum
0	1	2	3	3	9
1	4	5	6	9	24
Total	5	7	9	12	33

```
In [15]: df.index
```

```
Out[15]: Index([0, 1, 'Total'], dtype='object')
```

```
In [16]: %pip install seaborn
```

Requirement already satisfied: seaborn in c:\github\melon\_crawler\venv\lib\site-packages (0.13.2)  
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\github\melon\_crawler\venv\lib\site-packages (from seaborn) (2.2.4)  
Requirement already satisfied: pandas>=1.2 in c:\github\melon\_crawler\venv\lib\site-packages (from seaborn) (2.2.3)  
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\github\melon\_crawler\venv\lib\site-packages (from seaborn) (3.10.1)  
Requirement already satisfied: contourpy>=1.0.1 in c:\github\melon\_crawler\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.3.1)  
Requirement already satisfied: cycler>=0.10 in c:\github\melon\_crawler\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)  
Requirement already satisfied: fonttools>=4.22.0 in c:\github\melon\_crawler\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.57.0)  
Requirement already satisfied: kiwisolver>=1.3.1 in c:\github\melon\_crawler\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.8)  
Requirement already satisfied: packaging>=20.0 in c:\github\melon\_crawler\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (24.2)  
Requirement already satisfied: pillow>=8 in c:\github\melon\_crawler\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (11.1.0)  
Requirement already satisfied: pyparsing>=2.3.1 in c:\github\melon\_crawler\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.2.3)  
Requirement already satisfied: python-dateutil>=2.7 in c:\github\melon\_crawler\venv\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)  
Requirement already satisfied: pytz>=2020.1 in c:\github\melon\_crawler\venv\lib\site-packages (from pandas>=1.2->seaborn) (2025.2)  
Requirement already satisfied: tzdata>=2022.7 in c:\github\melon\_crawler\venv\lib\site-packages (from pandas>=1.2->seaborn) (2025.2)  
Requirement already satisfied: six>=1.5 in c:\github\melon\_crawler\venv\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)  
Note: you may need to restart the kernel to use updated packages.

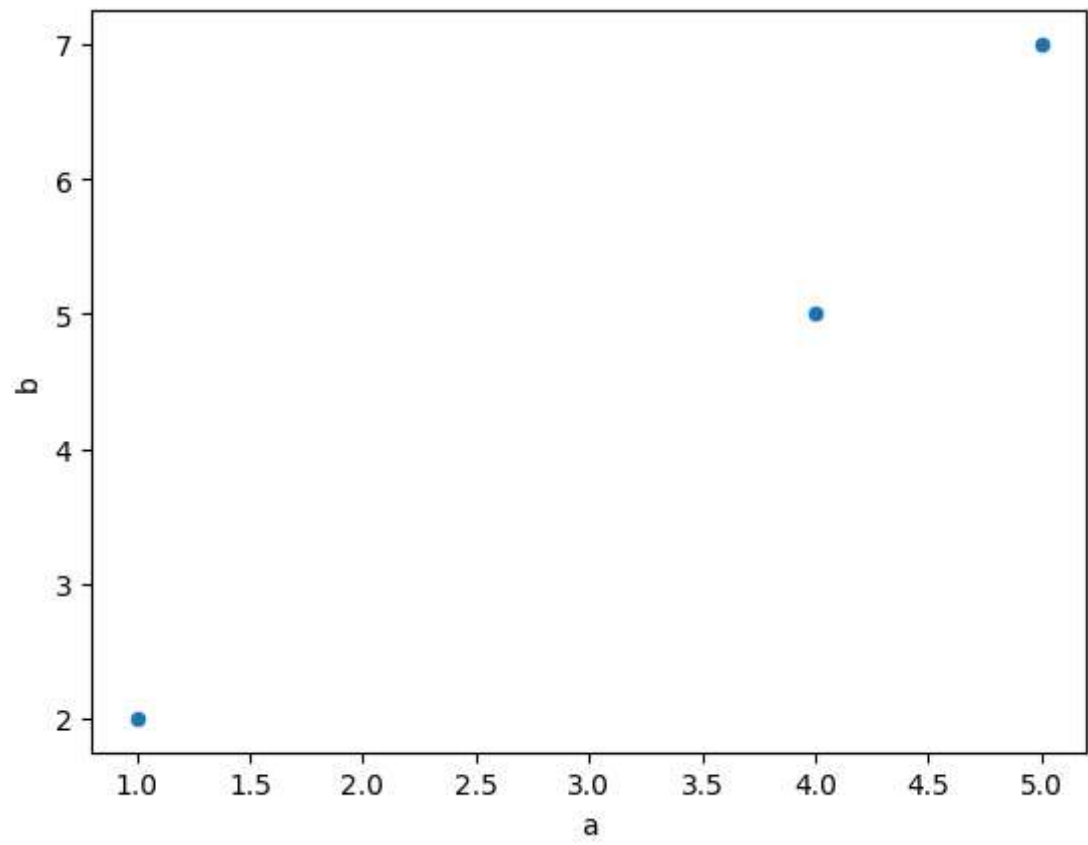
[notice] A new release of pip is available: 24.3.1 -> 25.0.1

[notice] To update, run: python.exe -m pip install --upgrade pip

```
In [17]: # seaborn 맛보기 (그래프가 된다고 했죠?!)
import seaborn as sns

sns.scatterplot(x=df['a'], y=df['b'])
```

Out[17]: <Axes: xlabel='a', ylabel='b'>



To Be Continued.