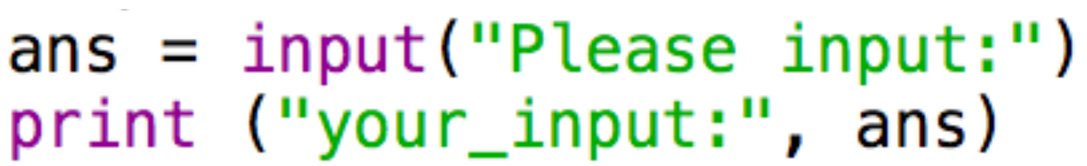
Introduction to Computer Science

File Input and Output

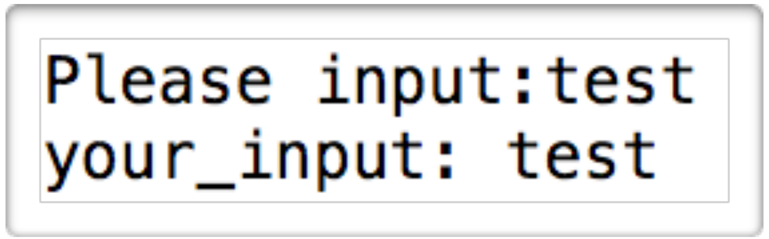
鄒年棣 (Nien-Ti Tsou)
許家維
楊仲齊

Reading keyboard Input

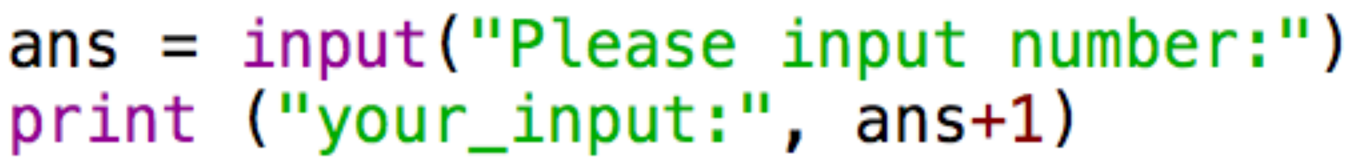
- `input()` is a Python built-in function to read a line of text from standard input.
- It assumes the input is a valid Python string expression and returns the evaluated result to you.



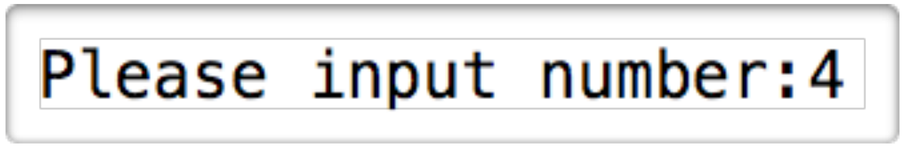
```
ans = input("Please input:")
print ("your_input:", ans)
```



```
Please input:test
your_input: test
```



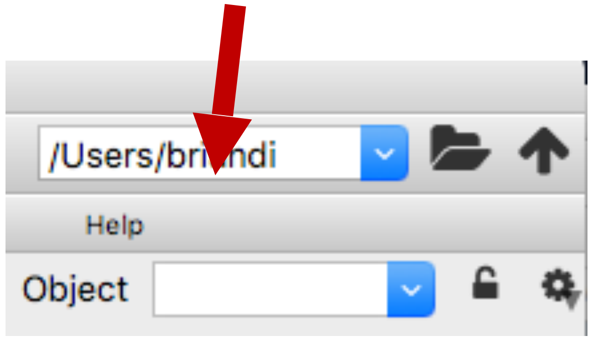
```
ans = input("Please input number:")
print ("your_input:", ans+1)
```



```
Please input number:4
```

Open and close files

File is created in this folder!

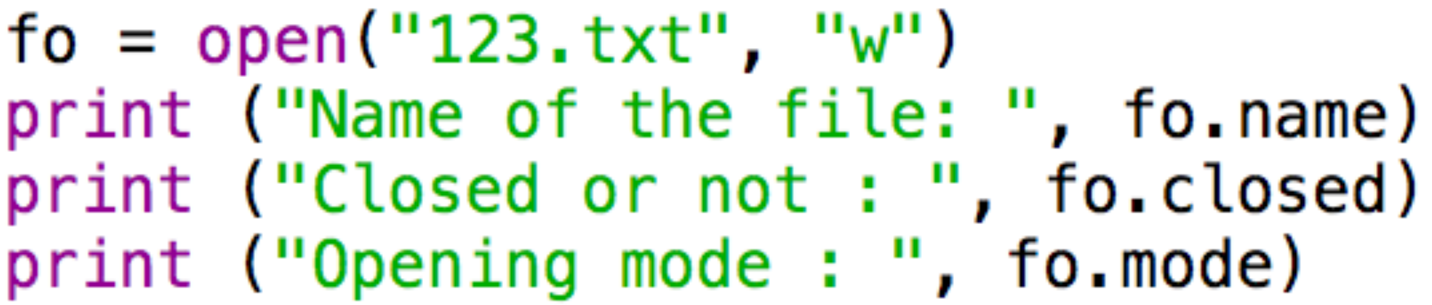


- Before you can read or write a file, you have to open it using Python's built-in `open()` function, which creates a file object.

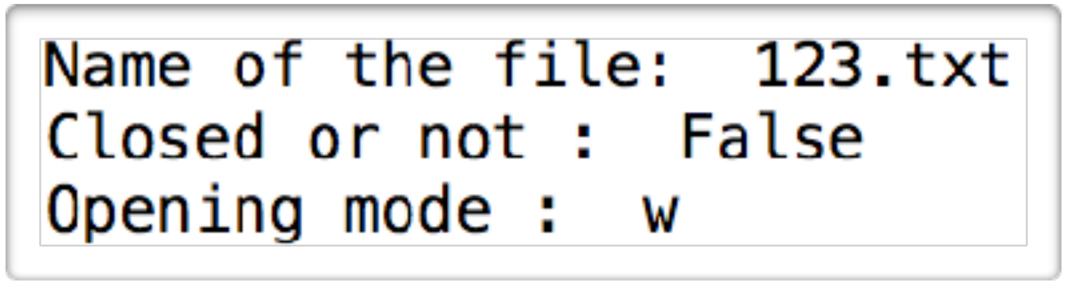
'r' Open file for reading.
'w' Open or create new file for writing.
'a' Open or create new file for writing. Append data to the end of the file.

The name of the file that you want to access.

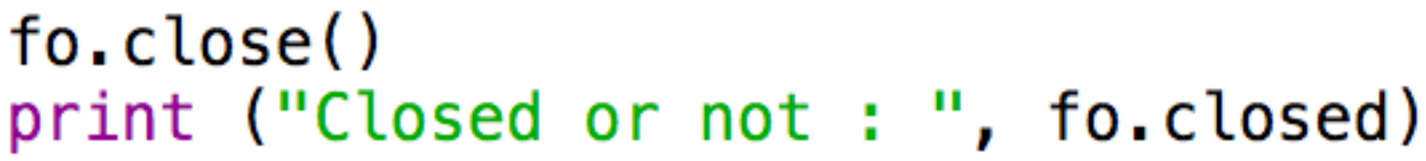
The mode in which the file has to be opened.



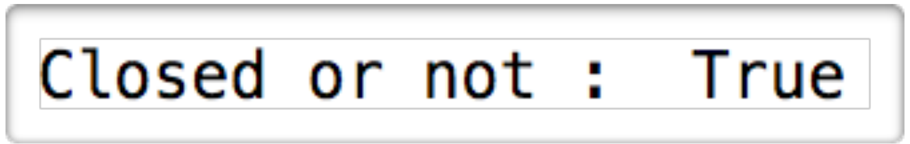
```
fo = open("123.txt", "w")
print ("Name of the file: ", fo.name)
print ("Closed or not : ", fo.closed)
print ("Opening mode : ", fo.mode)
```



```
Name of the file:  123.txt
Closed or not :  False
Opening mode :  w
```



```
fo.close()
print ("Closed or not : ", fo.closed)
```

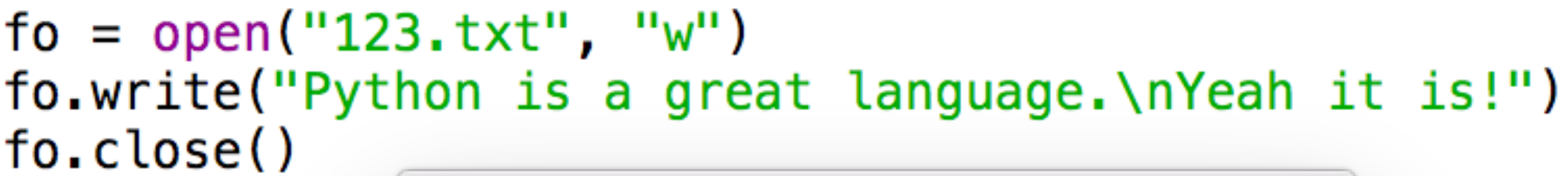


```
Closed or not :  True
```

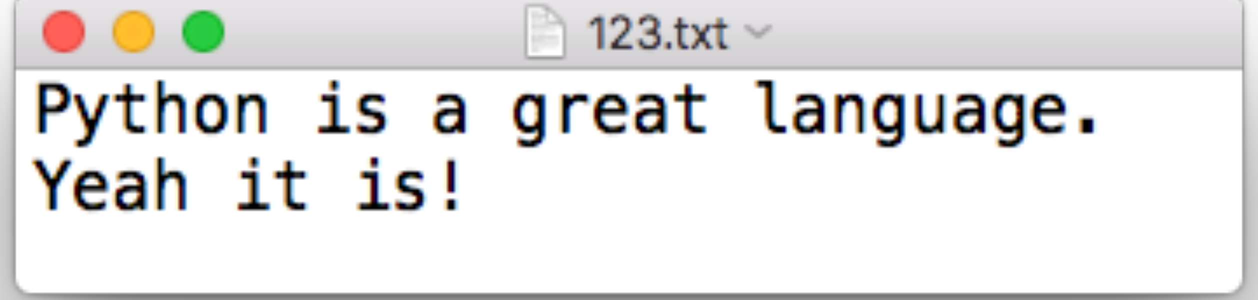
- Note that Python automatically closes a file when the reference object of a file is reassigned to another file.

Writing and Reading files

- The `write()` function writes any string to an open file.

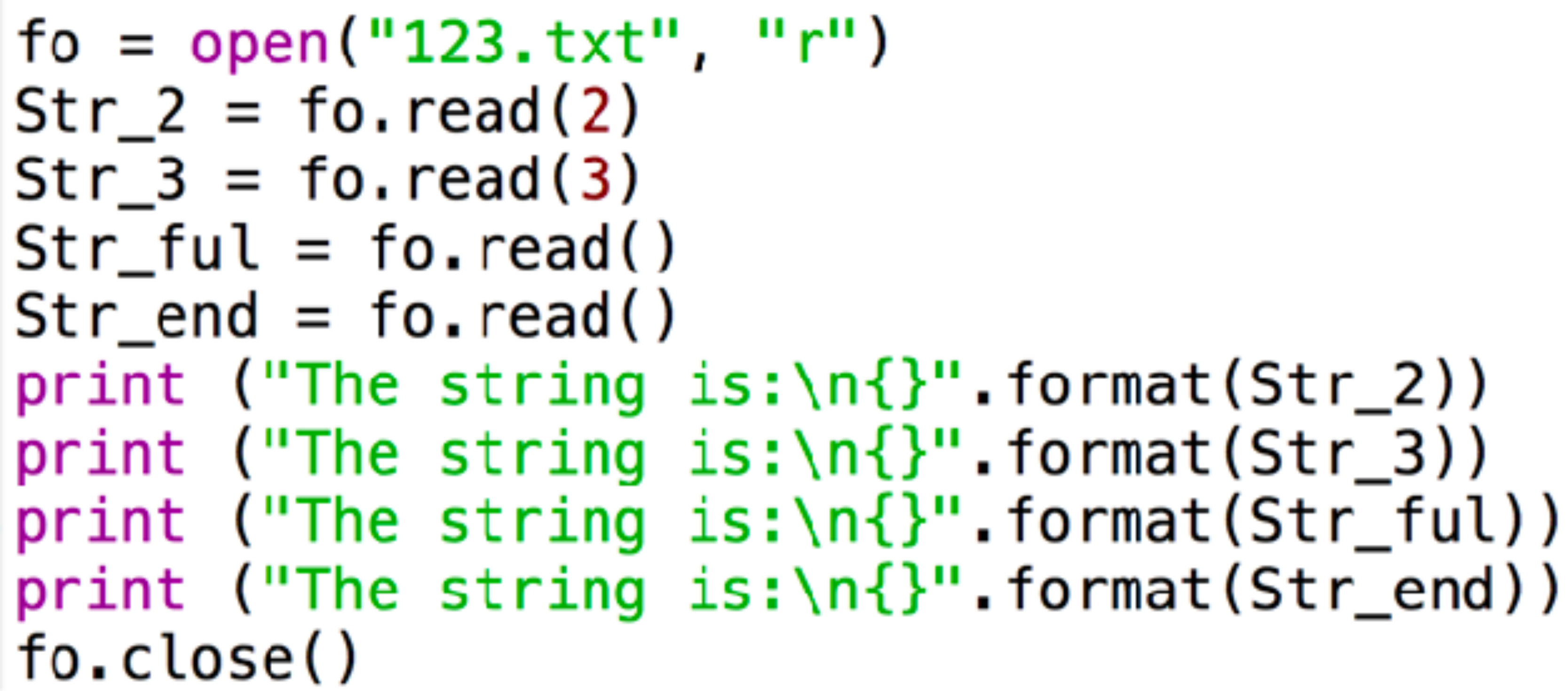


```
fo = open("123.txt", "w")
fo.write("Python is a great language.\nYeah it is!")
fo.close()
```

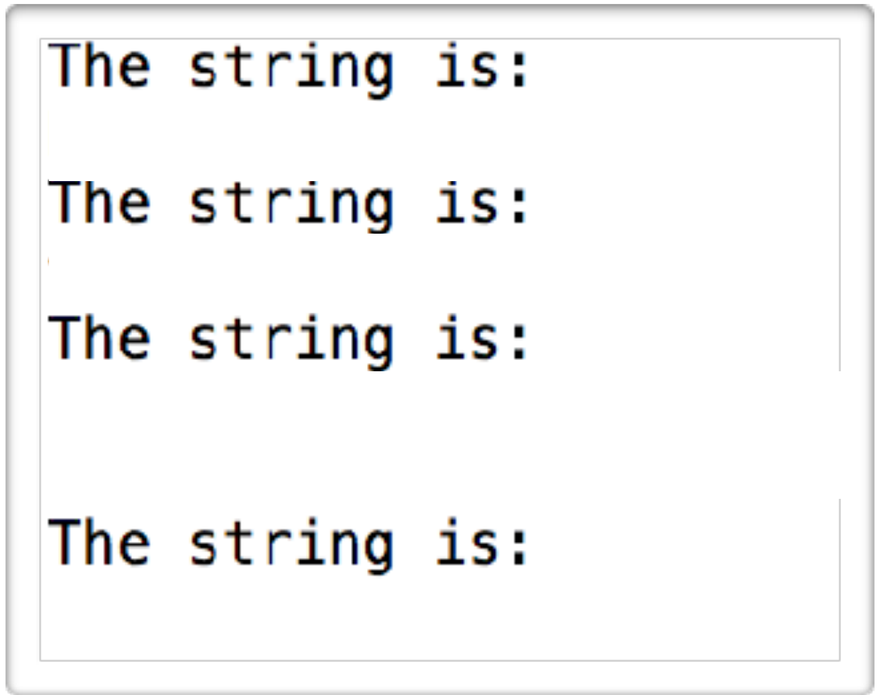


```
Python is a great language.
Yeah it is!
```

- The `read()` function reads a string from an open file.



```
fo = open("123.txt", "r")
Str_2 = fo.read(2)
Str_3 = fo.read(3)
Str_ful = fo.read()
Str_end = fo.read()
print ("The string is:\n{}".format(Str_2))
print ("The string is:\n{}".format(Str_3))
print ("The string is:\n{}".format(Str_ful))
print ("The string is:\n{}".format(Str_end))
fo.close()
```

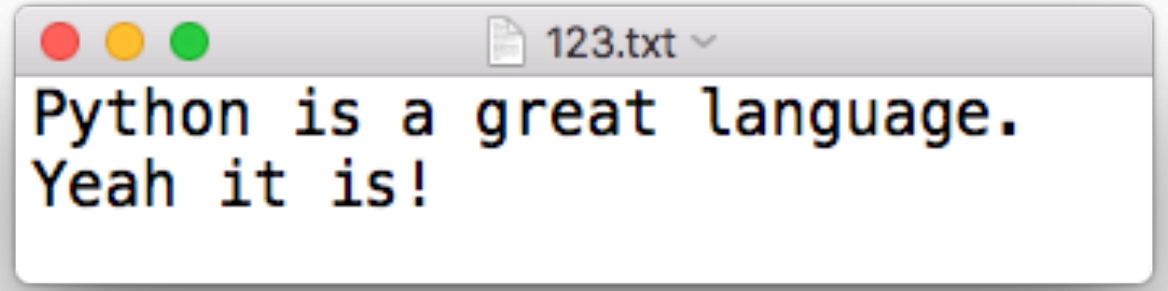


```
The string is:

The string is:

The string is:


The string is:
```

readline()

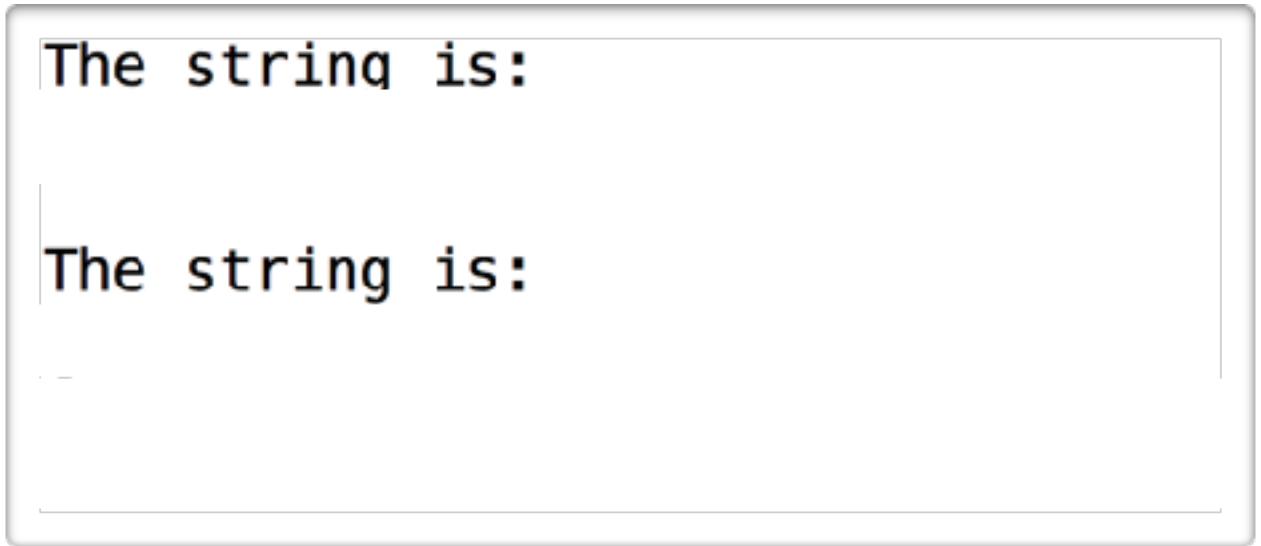
- `readline()` reads a single line from the file.
- a newline character `(\n)` is left at the end of the string.

```
"Python is a great language.\nYeah it is!"
```



```
Python is a great language.
Yeah it is!
```

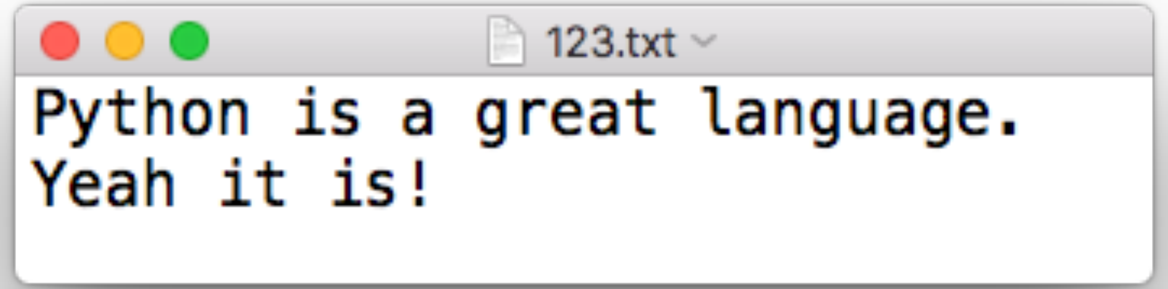
```
fo = open("123.txt", "r")
ListStr = []
for line in fo.readlines():
    print("The string is:\n{}".format(line))
    ListStr.append(line)

print(ListStr)
fo.close()
```



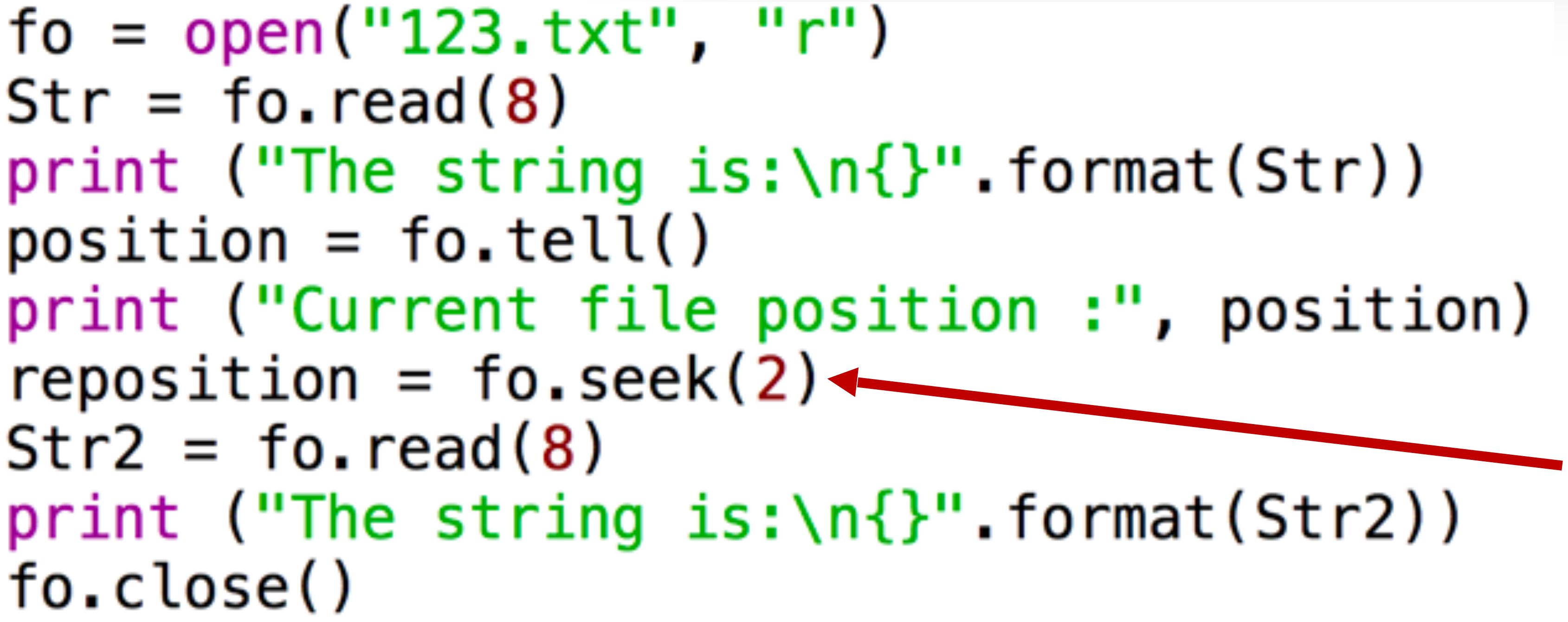
```
The string is:


The string is:
```

File positions

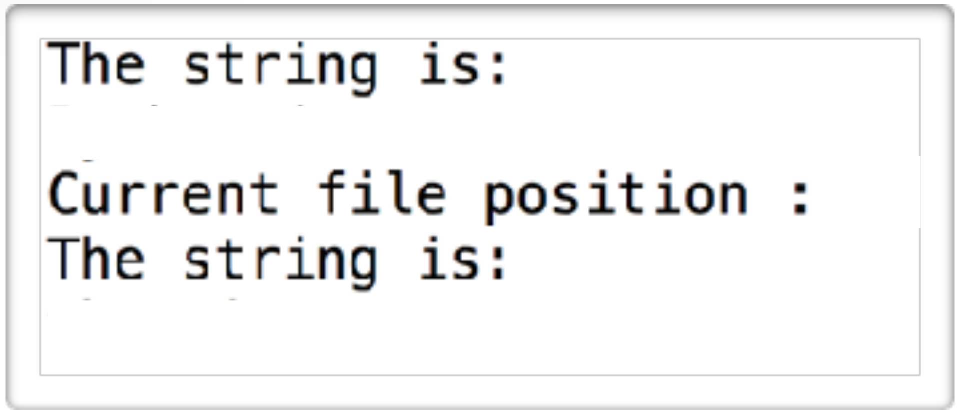
- `tell()` tells you the current position within the file, in other words, the next read or write will occur at there.
- `seek(offset)` changes the current file position to a position offset from the beginning of the file.



```
Python is a great language.
Yeah it is!
```



```
fo = open("123.txt", "r")
Str = fo.read(8)
print ("The string is:\n{}".format(Str))
position = fo.tell()
print ("Current file position :", position)
reposition = fo.seek(2)
Str2 = fo.read(8)
print ("The string is:\n{}".format(Str2))
fo.close()
```



```
The string is:

Current file position :
The string is:
```

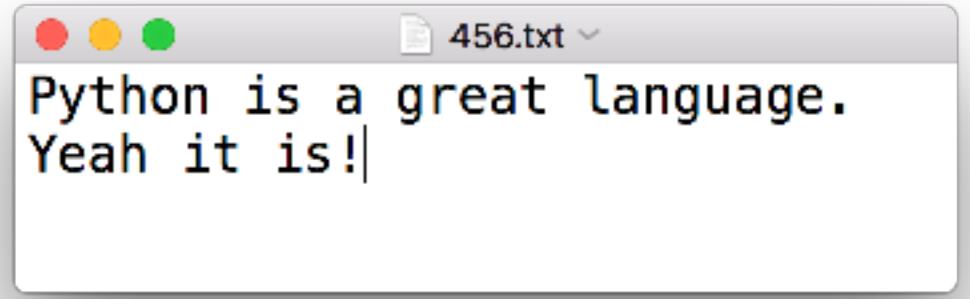
The pointer moved 2 bytes forward from the beginning.

Module os

- Python os module provides methods that help you perform file-processing operations, such as renaming and deleting files.

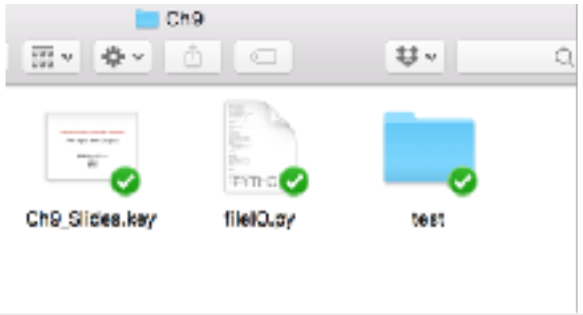
```
import os
os.rename( "123.txt", "456.txt")
```

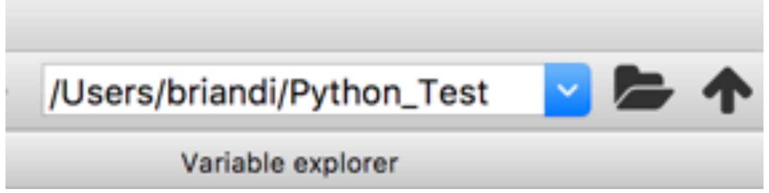
```
os.remove("456.txt")
```



```
Python is a great language.
Yeah it is!
```

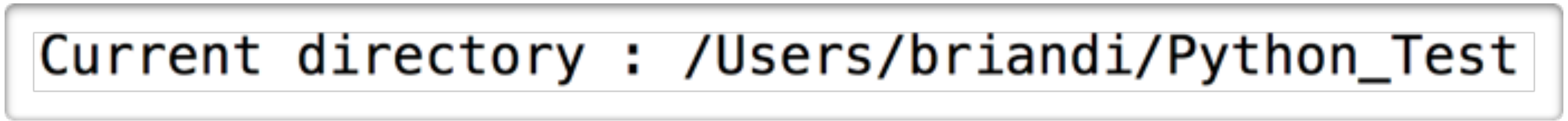
- os module provides methods that can create, remove, and change the directories.





```
os.mkdir("test")
os.mkdir("/Users/briandi/Python_Test")
os.chdir("/Users/briandi/Python_Test")
dir = os.getcwd()
print ("Current directory :", dir)
os.rmdir("/Users/briandi/Python_Test")
```

Please change the name which is suitable for your computer. Ex:
'C:/Users/user/Desktop/test2'

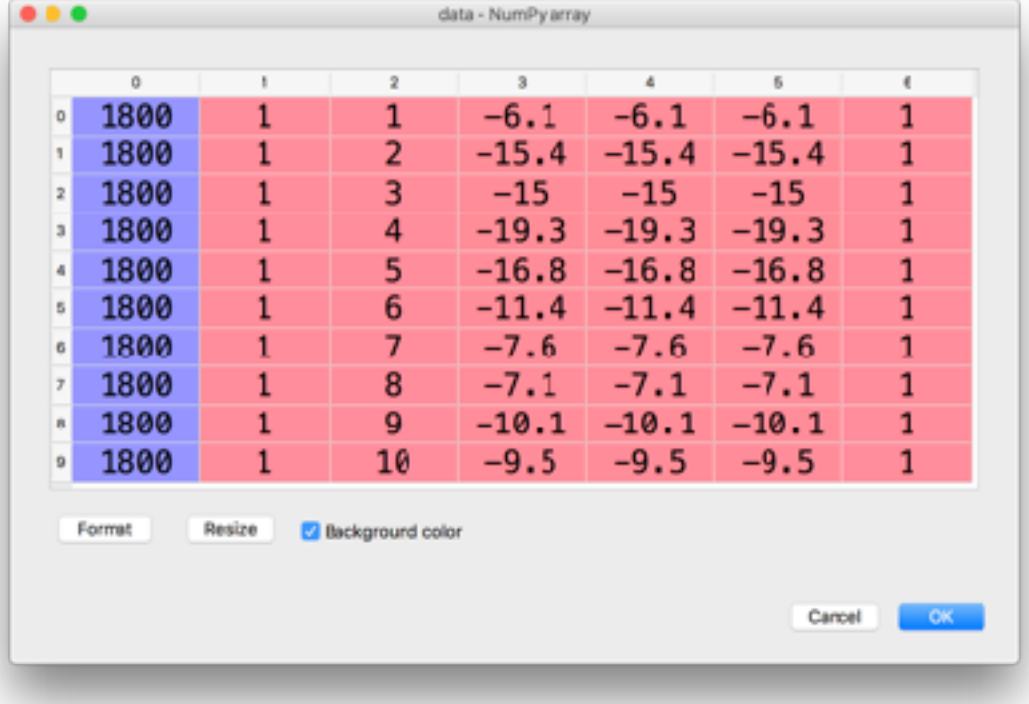


```
Current directory : /Users/briandi/Python_Test
```

Comma-separated values (CSV)

- A very common file format for data files is CSV, or tab-separated values (TSV).
- We can use the `np.genfromtxt` to read data from CSV or TSV into `numpy` arrays.

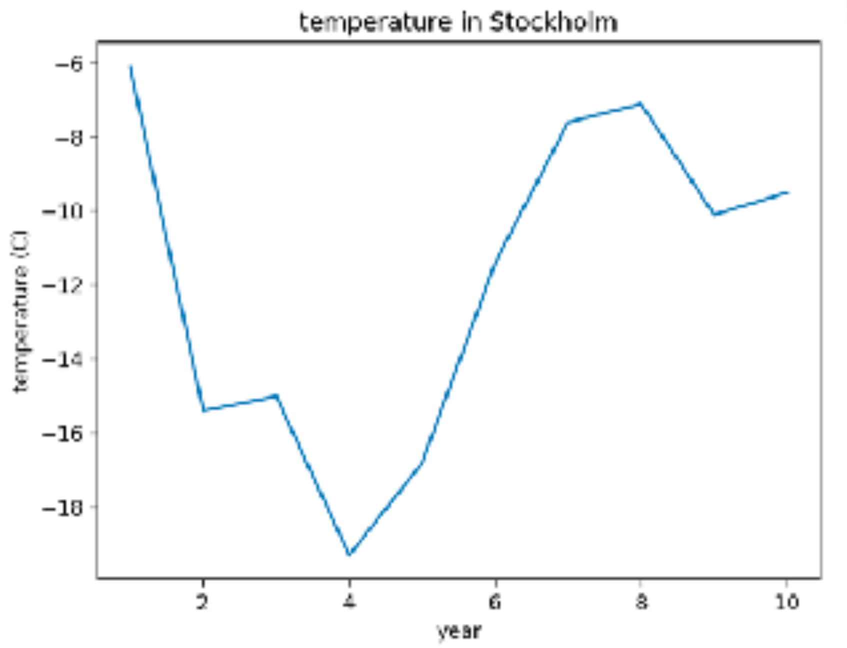
```python
import numpy as np
import matplotlib.pyplot as plt

data = np.genfromtxt('CSV1.txt')
```



data - NumPyarray

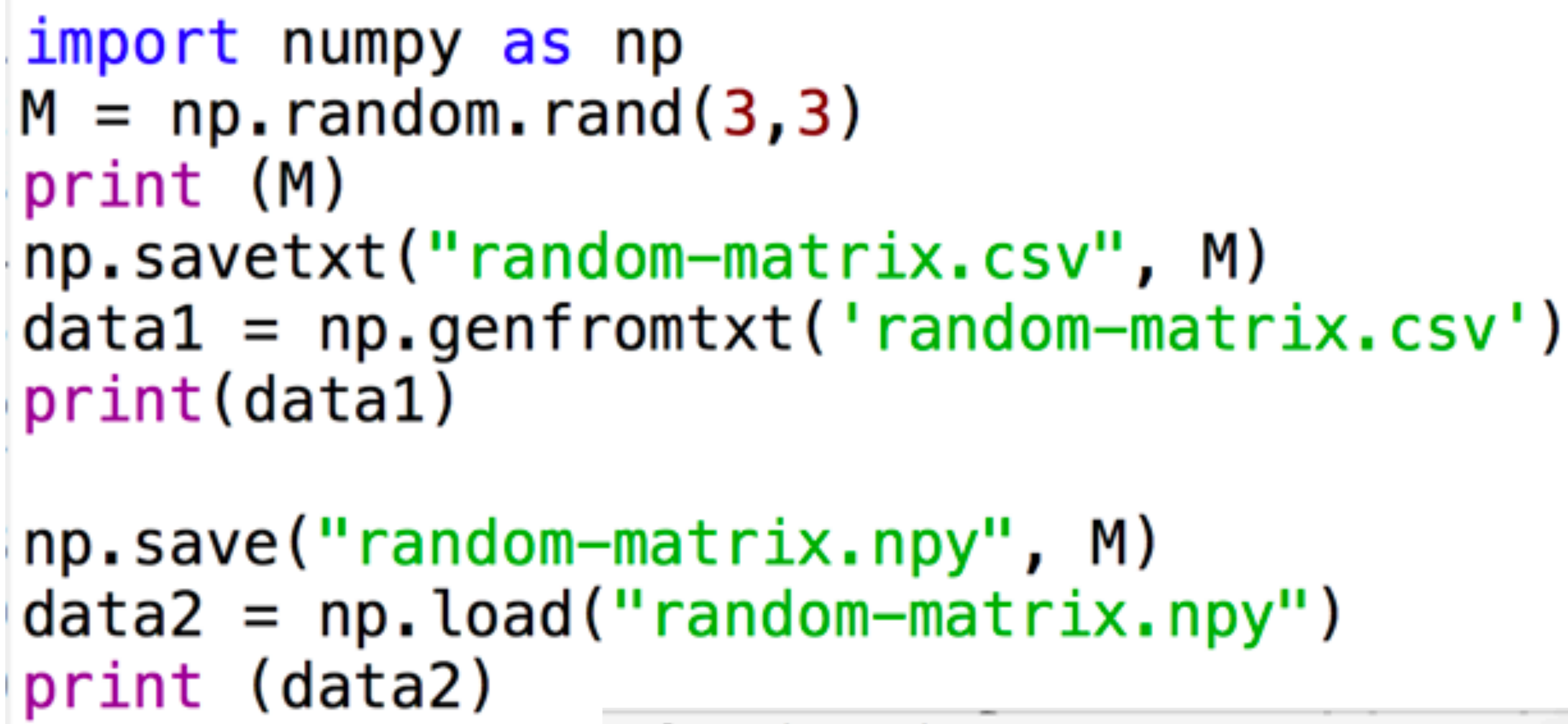
	0	1	2	3	4	5	6
0	1800	1	1	-6.1	-6.1	-6.1	1
1	1800	1	2	-15.4	-15.4	-15.4	1
2	1800	1	3	-15	-15	-15	1
3	1800	1	4	-19.3	-19.3	-19.3	1
4	1800	1	5	-16.8	-16.8	-16.8	1
5	1800	1	6	-11.4	-11.4	-11.4	1
6	1800	1	7	-7.6	-7.6	-7.6	1
7	1800	1	8	-7.1	-7.1	-7.1	1
8	1800	1	9	-10.1	-10.1	-10.1	1
9	1800	1	10	-9.5	-9.5	-9.5	1

```python
fig, ax = plt.subplots()
ax.plot(data[:,2], data[:,3])
ax.set_title("temperature in Stockholm")
ax.set_xlabel('year')
ax.set_ylabel('temperature (C)')
plt.show()
```

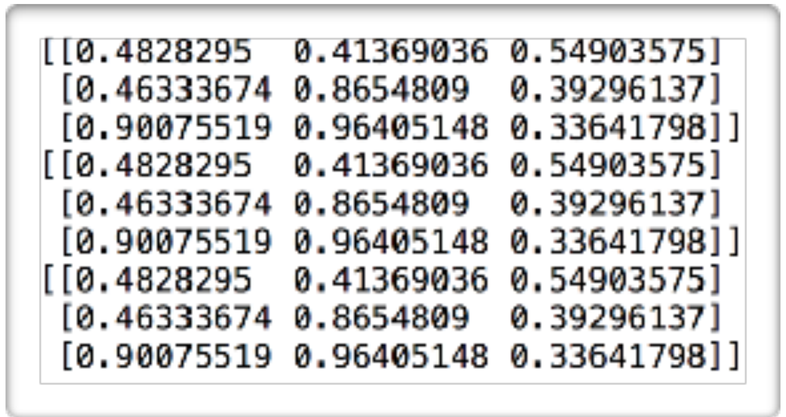


Save and load

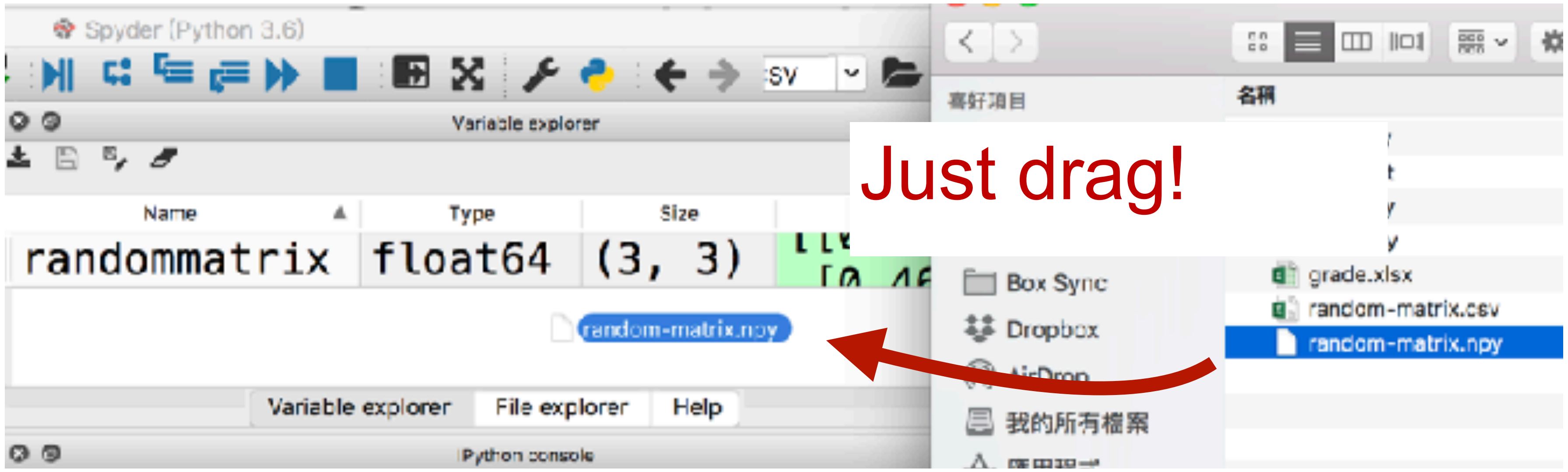
- We can use `np.savetxt` to store an array to a CSV file.
- `np.save` to store an array in numpy’s native format, which can be directly used by `np.load` or just drag into Variable explorer.



```
import numpy as np
M = np.random.rand(3,3)
print (M)
np.savetxt("random-matrix.csv", M)
data1 = np.genfromtxt('random-matrix.csv')
print(data1)

np.save("random-matrix.npy", M)
data2 = np.load("random-matrix.npy")
print (data2)
```



```
[[0.4828295  0.41369036 0.54903575]
 [0.46333674 0.8654809  0.39296137]
 [0.90075519 0.96405148 0.33641798]]
[[0.4828295  0.41369036 0.54903575]
 [0.46333674 0.8654809  0.39296137]
 [0.90075519 0.96405148 0.33641798]]
[[0.4828295  0.41369036 0.54903575]
 [0.46333674 0.8654809  0.39296137]
 [0.90075519 0.96405148 0.33641798]]
```



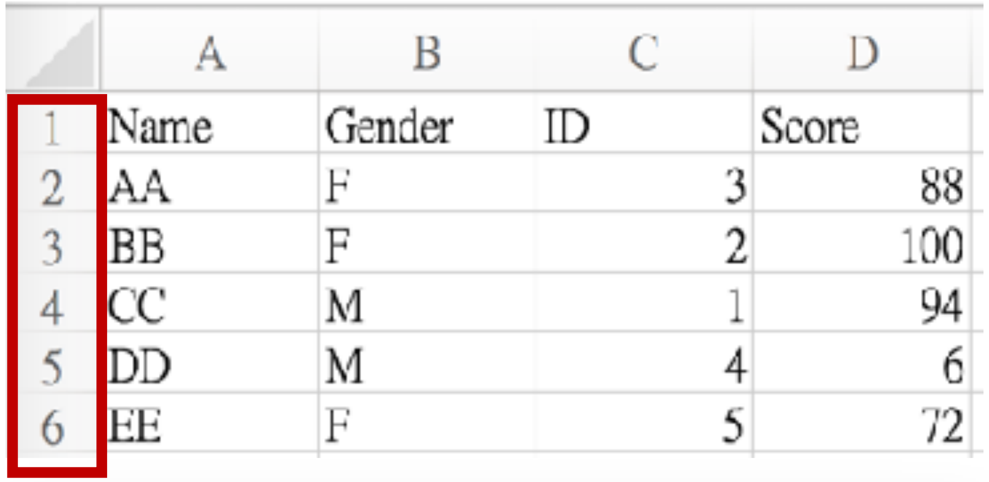
Name	Type	Size
randommatrix	float64	(3, 3)

Just drag!

Data processing in excel

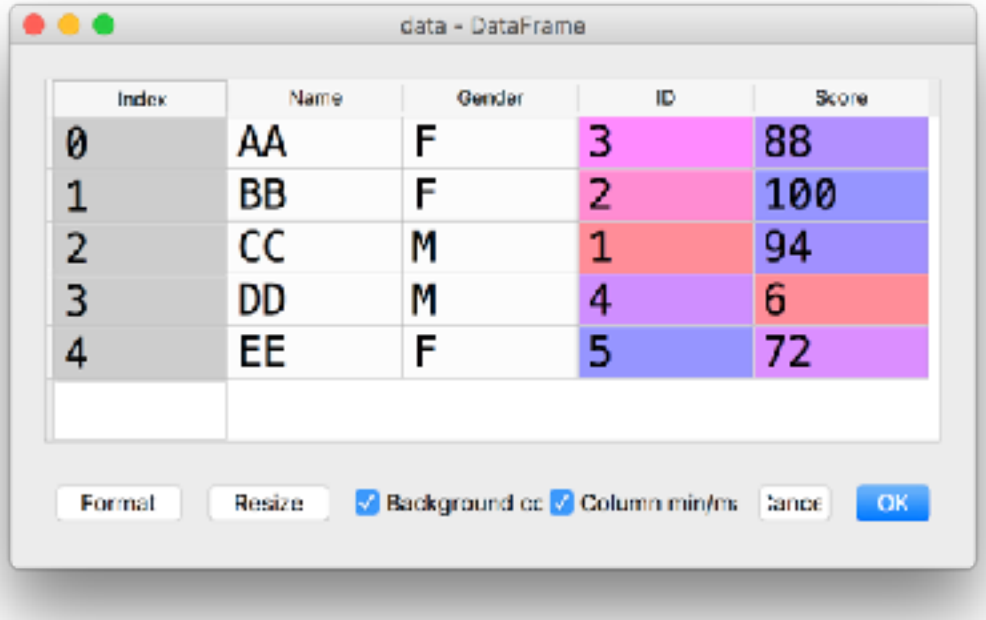
- We want to extract data from an excel file and rearrange the rank by grades.
- `pandas` is a library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming

```
import os
import pandas as pd

path_name = os.getcwd()
file_name = path_name + '/grade.xlsx'
data = pd.read_excel(file_name)
```



Index

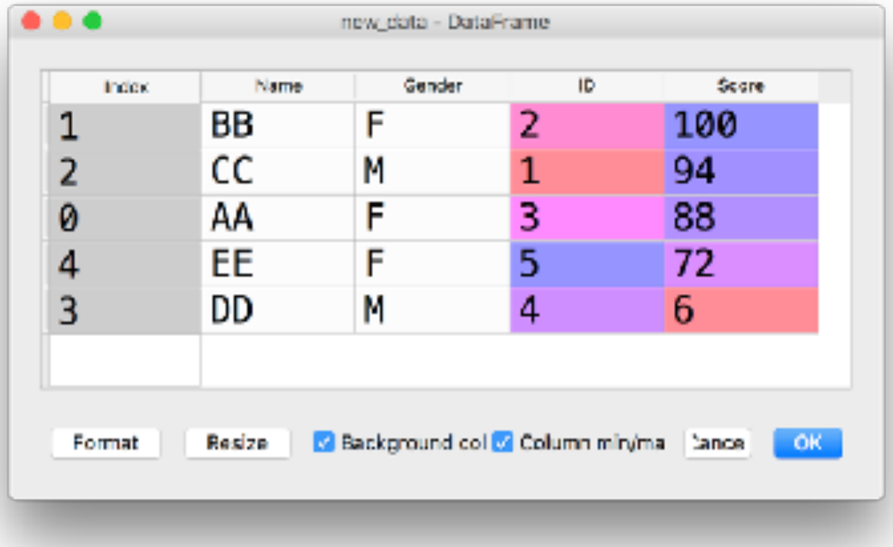
	A	B	C	D
1	Name	Gender	ID	Score
2	AA	F	3	88
3	BB	F	2	100
4	CC	M	1	94
5	DD	M	4	6
6	EE	F	5	72



Index	Name	Gender	ID	Score
0	AA	F	3	88
1	BB	F	2	100
2	CC	M	1	94
3	DD	M	4	6
4	EE	F	5	72

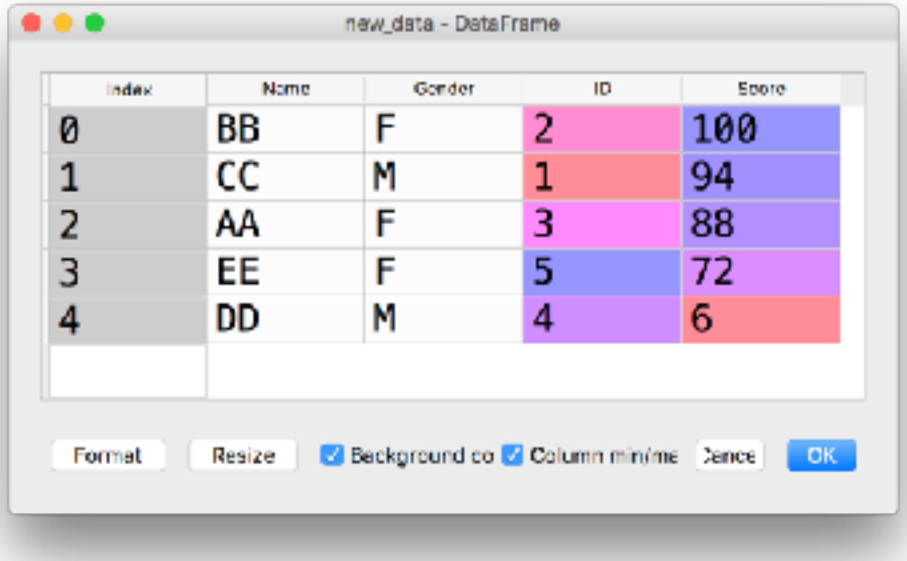
```
new_data = data.sort_values(by = ['Score'],
                            ascending = False)
new_data = new_data.reset_index(drop = True)
```

Replace the old index



Index	Name	Gender	ID	Score
1	BB	F	2	100
2	CC	M	1	94
0	AA	F	3	88
4	EE	F	5	72
3	DD	M	4	6

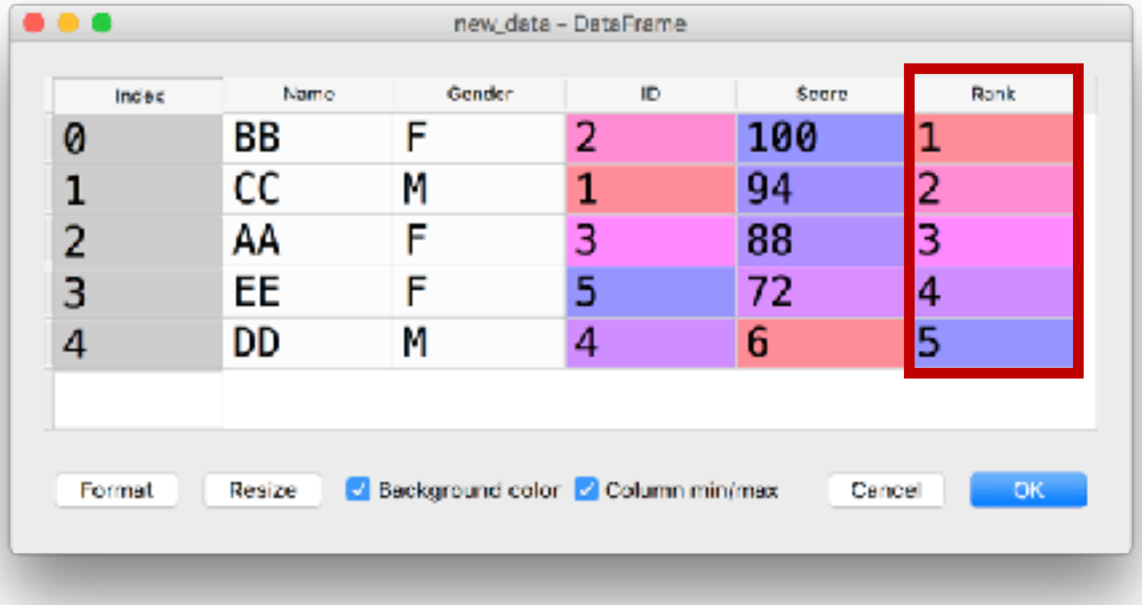
☑ Background col ☑ Column min/ma



Index	Name	Gender	ID	Score
0	BB	F	2	100
1	CC	M	1	94
2	AA	F	3	88
3	EE	F	5	72
4	DD	M	4	6

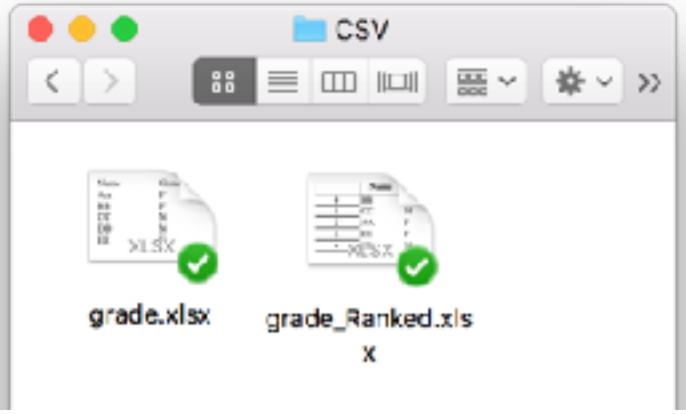
☑ Background co ☑ Column min/ma

```
new_data['Rank'] = new_data.index + 1

Newfile_name = path_name + '/grade_Ranked.xlsx'
new_data.to_excel(Newfile_name)
```



Index	Name	Gender	ID	Score	Rank
0	BB	F	2	100	1
1	CC	M	1	94	2
2	AA	F	3	88	3
3	EE	F	5	72	4
4	DD	M	4	6	5

☑ Background color ☑ Column min/max


CSV
grade.xlsx
grade_Ranked.xlsx