Input Output

- File
- Pickle

File

- 파일을 읽기, 쓰기
  - w : 파일 쓰기 (파일이 존재하면 덮어씀)
  - x : 파일 쓰기 (파일이 존재하지 않을때만 생성)
  - r : 파일 읽기
  - a : 문자열 추가
- 파일 타입
  - t : 텍스트 타입
  - b : 이진 타입

In [1]:
```
!mkdir file
```

In [2]:
```
!ls
```

```
01_juputer_notebook.ipynb 09_function.ipynb         quiz_day_1.ipynb
02_basic_sytax.ipynb      10_class.ipynb            quiz_day_2.ipynb
03_datatype.ipynb         11_module_package.ipynb   quiz_day_3.ipynb
04_change_datatype.ipynb  12_try_except.ipynb       quiz_day_4.ipynb
05_operators.ipynb        13_regex.ipynb            quiz_day_7
06_input_print.ipynb      14_io.ipynb               quiz_day_8.ipynb
07_conditions.ipynb       __pycache__               slack_incoming.ipynb
08_loops.ipynb            file
```

In [10]:
```
# 텍스트 파일 작성하기
s = """datascience
fast campus"""
```

In [11]:
```
try:
    f = open("file/test1.txt", "wt")
    f.write(s)
except Exception as e:
    print(e)
finally:
    f.close()
```

In [12]:
```
# 텍스트 파일을 읽어 오기
```

In [13]:
```
f = open("file/test1.txt", "rt")
s = f.read()
f.close()
print(s)
```

```
datascience
fast campus
```

In [14]:
```
data = bytes(range(5, 10))
data
```

Out[14]:
```
b'\x05\x06\x07\x08\t'
```

In [15]:
```
# 바이너리 파일 쓰기
f = open("file/range.b", "wb")
f.write(data)
f.close()
```

In [16]:
```
%ls file
```

```
range.b    test1.txt
```

In [17]:
```
# 바이너리 파일 읽기
f = open("file/range.b", "rb")
data = f.read()
f.close()
data
```

Out[17]: b'\x05\x06\x07\x08\t'

In [18]:
```
print(list(data))
```

```
[5, 6, 7, 8, 9]
```

In [20]:
```
# file의 open, close를 간단하게 작성시켜주는 with
# with 자동으로 close를 해주는 기능
with open("file/test1.txt", "rt") as f:
    s = f.read()
    print(s)
```

```
datascience
fast campus
```

pickle

- 객체를 파일로 저장할때 직렬화라는 과정을 거쳐서 저장을 합니다.
- 직렬화
  - 객체(데이터 타입), 저장되는 파일(데이터 타입)은 서로 다릅니다.
  - 다른 데이터 타입을 맞춰주는 과정을 직렬화
- 사용하는 이유는 파일을 읽고 쓰는 시간이 더 빠릅니다.

In [21]:
```
import pickle
```

In [22]:
```
class A:

    def __init__(self, data):
        self.data = data

    def disp(self):
        print(self.data)
```

In [23]:
```
obj = A("pickle test")
obj
```

Out[23]: <__main__.A at 0x117d62908>

In [24]:
```
obj.disp()
```

```
pickle test
```

In [25]:
```
# 객체 저장하기
with open("file/obj.pkl", "wb") as f:
    pickle.dump(obj, f)
```

In [26]:
```
%ls file
```

```
obj.pkl    range.b    test1.txt
```

In [27]:
```
# 객체 읽어오기
with open("file/obj.pkl", "rb") as f:
    load_obj = pickle.load(f)
```

In [28]:
```
load_obj.disp()
```

```
pickle test
```