

19. 파일 다루기

파일의 입 / 출력

- 메모리에 저장 되어 이내 사라지는 데이터 와는 달리 저장 매체(하드디스크, USB) 등에 보관되는 데이터 를 파일 이라고 한다.
- 파이썬 의 파일 입 / 출력 순서



- * 스트림 Open : 파일을 관리 하리 위하여 파일의 입 출력 위치 확인 및 저장 공간에 접근 할 통로를 생성
- Open 된 스트림에 파일을 다루는 방법 설정 (모드)

모드	설명
r	파일을 읽기만 한다. 파일이 없으면 오류 가 발생한다.
w	파일에 기록한다. 파일이 이미 있으면 내용을 덮어 쓴다.
a	파일에 데이터 를 추가한다.
x	파일에 기록하지만 이미 같은 이름의 파일이 있으면 실패 한다.

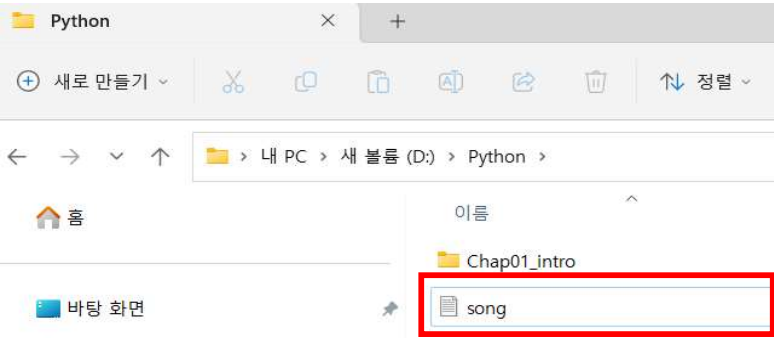
- 파일 쓰기
- . 파일을 생성하고 특정 데이터 추가 하는 예시

```
Chap01_intro > 18.FileManage[파일입출력].py > ...
1 f = open("song.txt","wt")
2 f.write(''Open() 은 파일 스트림을 생성합니다.
3 \"wt\" 에서
4 w 는 파일을 기록 하는것을 의미하고
5 r 은 파일을 읽기 처리 하는 것을 의미합니다.
6 뒤쪽에 있는
7 t 는 텍스트 파일 형태로 만드는것을 의미합니다.
8 '')
9 f.close()
10
```

Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python

> 파일 탐색기 에서 생성 된 txt 파일을 확인 할 수 있다.



- 파일 읽기

. 파일 전체를 한꺼번에 읽어오는 read()

> 위에 서 생성 한 song.txt 파일을 읽어 문자열을 표현한다.

```
Chap01_intro > 18.FileManage[파일입출력].py > ...
1  try:
2      f = open("song.txt","rt")
3      text = f.read()
4      print(text)
5  except FileNotFoundError:
6      print("파일이 없습니다.")
7  finally:
8      f.close()
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Py
Open() 은 파일 스트림을 생성합니다.
"wt" 에서
w 는 파일을 기록 하는것을 의미하고
r 는 파일을 읽기 처리 하는 것을 의미합니다.
뒤쪽에 있는
t 는 텍스트 파일 형태로 만드는데것을 의미합니다.

. Read() 의 인수를 통한 읽기 범위설정

```
Chap01_intro > 18.FileManage[파일입출력].py > ...
1  f = open("song.txt","rt")
2  while True:
3      text = f.read(2)
4      if len(text)== 0:break
5      print(text,end=' ')
6  f.close()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/py
Open() 은 파일 스트림을 생성합니다.
"wt" 에서
w 는 파일을 기록 하는것을 의미하고
r 는 파일을 읽기 처리 하는 것을 의미합니다.
뒤쪽에 있는
t 는 텍스트 파일 형태로 만드는데것을 의미합니다.

- >1행 : song.txt 파일 관리 스트림 생성
- >2행 : 파일의 내용을 모두 읽어 오기 위하여 반복문 시작
- >3행 : read(2) 를 통하여 2자씩 읽어오기
- >4행 : 더 이상 읽어올 행이 없다면 반복문 종료

. 파일을 한줄 씩 만 읽어오기 readline()

```
Chap01_intro > 18.FileManage[파일입출력].py > ...
1 f = open("song.txt", "rt")
2 text = ''
3 line = 1
4 while True:
5     row_text = f.readline()
6     if not row_text: break
7     text += str(line) + " : " + row_text
8     line += 1
9 f.close()
10 print(text)
```

Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:
1 : Open() 은 파일 스트림을 생성합니다.
2 : "wt" 에서
3 : w 는 파일을 기록 하는것을 의미하고
4 : r 은 파일을 읽기 처리 하는 것을 의미합니다.
5 : 뒤쪽에 있는
6 : t 는 텍스트 파일 형태로 만드는데것을 의미합니다.

>6행 : 불러온 결과가 아무 값도 없다면 break

* line 이 1 씩 증가 하며 한줄 씩 처리 되는 결과를 확인 할 수 있다

. 리스트 형식으로 읽어온 데이터 를 등록 해 두는 readlines()

```
Chap01_intro > 18.FileManage[파일입출력].py > ...
1 f = open("song.txt", "rt")
2 rows = f.readlines()
3 for row in rows :
4     print(row, end='')
5 f.close()
```

Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:
Open() 은 파일 스트림을 생성합니다.
"wt" 에서
w 는 파일을 기록 하는것을 의미하고
r 은 파일을 읽기 처리 하는 것을 의미합니다.
t 는 텍스트 파일 형태로 만드는데것을 의미합니다.

>2행 : readlines() 기능으로 rows 에 읽은 행을 리스트 형태로 할당한다.

* readlines() 에 마우스를 위치하면 list[str] 을 반환하는 메서드임을 확인 할 수 있다.

```
.readlines()
i (method) def readlines(
t     __hint: int = -1,
)     /
) -> list[str]
```

```
for row in rows :
    print(row, end='')
f.close()
```

파
f = open(
for row i
len(): 9

['Open() 은 파일 스트림을 생성합니다.\n', 'wt' 에서 \n', 'w 는 파일을 기록 하는것을 의미하...
> special variables
> function variables
0: 'Open() 은 파일 스트림을 생성합니다.\n'
1: 'wt' 에서 \n'
2: 'w 는 파일을 기록 하는것을 의미하고\n'
3: 'r 은 파일을 읽기 처리 하는 것을 의미합니다.\n'
4: '뒤쪽에 있는 \n'
5: 't 는 텍스트 파일 형태로 만드는데것을 의미합니다. \n'
6: '\n'
7: '추가로 a는 파일의 내용을 추가로 이어 쓰는 기능입니다.\n'
8: '추가로 a는 파일의 내용을 추가로 이어 쓰는 기능입니다.'
len(): 9

. 파일 스트림 자체 를 반복하여 표현하기

> 텍스트 파일 도 행의 집합 이므로 곧바로 반복문을 사용 할 수 있다.

```
Chap01_intro > 18.FileManage[파일입출력].py > ...
1 f = open("song.txt", "rt")
2 for row in f :
3     print(row, end='')
4 f.close()
```

Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:
Open() 은 파일 스트림을 생성합니다.
"wt" 에서
w 는 파일을 기록 하는것을 의미하고
r 은 파일을 읽기 처리 하는 것을 의미합니다.
t 는 텍스트 파일 형태로 만드는데것을 의미합니다.

>2행 : 파일스트림 f 로 읽어온 내용을 곧바로 추출하여 표현한다.

내용추가

- 파일의 내용을 이어쓰기 a

```
Chap01_intro > 18.FileManage[파일입출력].py > ...
1 f = open("song.txt","at")
2 f.write('\n추가로 ' + 'a는 파일의 내용을 추가로 이어 쓰는 기능입니다.')
```

3

```
4 f = open("song.txt","rt")
5 for row in f :
6     print(row,end='')
7 f.close()
```

오

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_intro/18.FileManage.py
Open() 은 파일 스트림을 생성합니다.
"wt" 에서
w 는 파일을 기록 하는것을 의미하고
r 은 파일을 읽기 처리 하는 것을 의미합니다.
뒤쪽에 있는
t 는 텍스트 파일 형태로 만드는것을 의미합니다.

추가로 a는 파일의 내용을 추가로 이어 쓰는 기능입니다.

- > 1행 : 파일의 내용을 추가로 작성 할 at 파일 스트림 생성
- > 2행 : 추가 로 등록 할 파일의 내용 작성 및 등록 write()
- > 4행 : 등록을 완료 한 텍스트 파일 읽어오기 파일 스트림 생성
- > 5행 : 읽어온 text 파일 출력

- 파일 스트림 자동 close()

- . 파일 스트림은 한번 사용하고 나면 로직 내에서 반드시 close 를 해 주어야 한다.
- . 파일 스트림이 선점 한 파일은 외부에서 접근 할 수 가 없게 된다.
 - * 프로그램이 종료 되면 파일 스트림은 소멸 된다.
- . 간단한 파일 입/출력 의 구문일 경우 with / as 를 사용하여 파일 입/출력 을 구현 할 수 있다.

```
Chap01_intro > 18.FileManage[파일입출력].py > ...
1 with open('song.txt','rt') as f:
2     text = f.read()
3 print(text)
```

아

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_intro/18.FileManage.py
Open() 은 파일 스트림을 생성합니다.
"wt" 에서
w 는 파일을 기록 하는것을 의미하고
r 은 파일을 읽기 처리 하는 것을 의미합니다.
뒤쪽에 있는
t 는 텍스트 파일 형태로 만드는것을 의미합니다.

추가로 a는 파일의 내용을 추가로 이어 쓰는 기능입니다.
추가로 a는 파일의 내용을 추가로 이어 쓰는 기능입니다.

- > with 구문 블록 내에서 파일 스트림을 사용하며 블록이 끝나는 지점에서 파일스트림은 자동으로 close() 된다.