

윤소영

imbgirl@naver.com

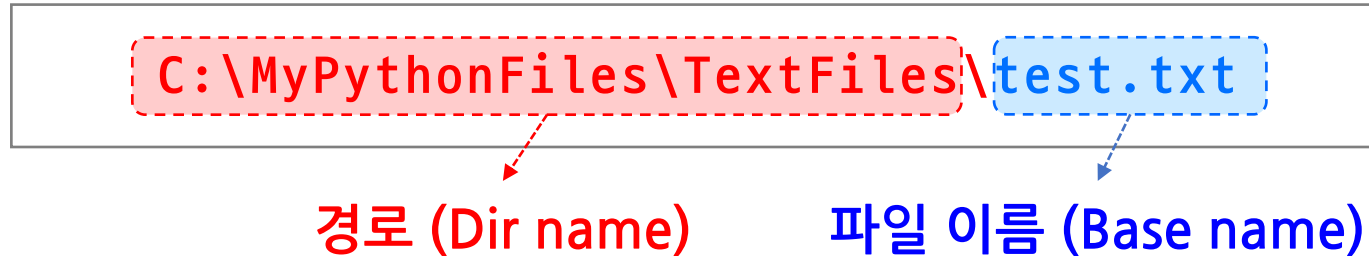
파일 읽기, 쓰기

open, close, shelve, glob

파일



- 프로그램이 종료된 후에도 데이터를 계속 유지하려면 “파일”로 저장해야 한다



\(₩)는 무엇일까요?

Windows 에서 directory 와 directory, base name 구분에 사용함

/ 는 무엇일까요?

OS X, Linux 에서 사용하는 경로 구분자

Python에서는요?

\(₩), / 모두 사용 가능! BUT → ₩는 ₩₩ 와 같이 사용 함

```
file = "C:\\MyPythonFiles\\TextFiles\\test.txt"
```

```
os.path.dirname(file)
```

```
'C:\\MyPythonFiles\\TextFiles'
```

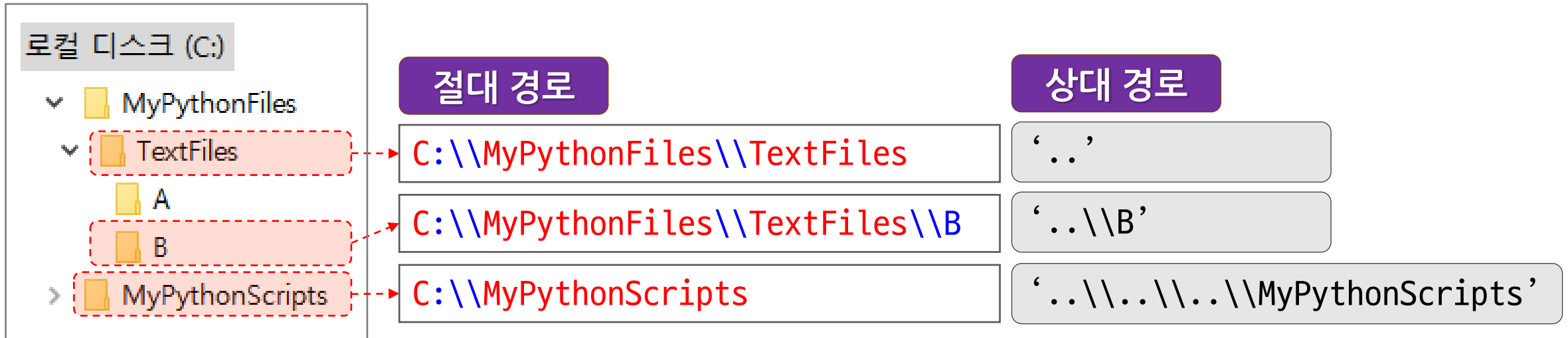
```
os.path.basename(file)
```

```
'test.txt'
```

절대 경로, 상대 경로



- File System에서 경로(Path)는 File, Directory 등의 위치 정보를 의미함
- Path를 표기하는 방법에는 '절대 경로'와 '상대 경로'가 있음
 - 현재 작업 디렉터리가 C:\MyPythonFiles\TextFiles\A' 인 경우



디렉터리 변경 및 알아내기



- 사용자 프로그램은 특정 디렉터리를 사용하게 됨
 - 현재 작업 디렉터리(Current Working Directory)란

사용자 프로그램에서 현재 사용하는 디렉터리

디렉터리 변경(Change Directory)

`os.getcwd()` 로 알 수 있음

`os.chdir(path)` 로 지정함

dirname

file → 'C:\\MyPythonFiles\\source\\one.py'

```
import os
```

```
os.chdir('..')  
print(os.getcwd())
```

상위 디렉터리로 이동

```
C: \MyPythonFiles  
>>>
```

파일의 종류



🍃 텍스트 파일 vs 이진 파일

텍스트 파일

- txt, py, c, h, cpp, sh 등의 확장자인 일반 텍스트 파일
- 텍스트 편집기(메모장)로 열어 내용을 볼 수 있음

이진 파일

- 워드, 엑셀, 파워포인트 문서, PDF, 이미지, 실행 파일 등
- 이진 파일은 형식 마다 고유한 방식으로 처리 됨
→ 일반 메모장 같은 편집기로 확인 할 수 없음

🍃 파이썬에서의 파일 읽기/쓰기의 3 단계는 다음과 같다

1 open() 함수를 호출하여 File 객체 얻기

2 File 객체의 read(), write() 메서드로 읽기 쓰기

3 File 객체의 close() 메서드로 파일 닫기

파일에서 읽기



🍃 파일을 open하고 내용을 읽는 방법을 살펴보자

🍃 1. 파일의 절대/상대 경로를 문자열로 전달하여 open() 함수를 호출한다

File 객체 저장 변수 = open('파일명', mode) # mode 생략 시 읽기 전용 'r'

🍃 2. 파일로부터 읽기 작업은 File 객체의 read() 또는 readlines() 메서드를 사용한다

읽은 내용 저장 변수 = File객체.read() # File객체.readlines()

```
myfile = open('C:\\MypythonFiles\\letter.txt')
mycont = myfile.read()
print(type(mycont))
print(mycont)
myfile.close()
```

```
<class 'str'>
Dear friend,
How are you?
```

하나의 문자열로 읽기

```
myfile = open('C:\\MypythonFiles\\letter.txt')
mycont = myfile.readlines()
print(type(mycont))
print(mycont)
myfile.close()
```

```
<class 'list'>
['Dear friend,\n', 'How are you?\n']
```

줄바꿈

하나의 \n을 하나의 문자열로 하는 list로 읽기

파일에 저장하기



🍃 파일을 쓰기 모드로 open하고 내용을 쓰는 방법을 살펴보자

🍃 1. letter2.txt 파일을 쓰기 모드를 이용해서 열기한다

File 객체 저장 변수 = open('파일명', mode) # 쓰기 모드 'w', 추가 모드 'a'

🍃 2. 파일에 쓰기 작업은 File 객체의 write() 메서드를 사용한다

File 객체 저장 변수.write('내용')

```
myfile = open('C:\\MypythonFiles\\letter2.txt', 'w')
myfile.write('Take care! See u later~\n')
myfile.close()
```

print(): 행 끝에 자동 줄바꿈
write(): 자동 줄바꿈 추가 없음

- 'w' 모드 - 파일이 존재하면 파일의 모든 내용을 삭제하고 다시 쓰기 준비함
- 'a' 모드 - 파일이 존재하면 내용 삭제 없이, 파일의 뒤에 내용을 추가함
- 'b' 모드 - 'r', 'w' 등과 같이 사용하며, 이진 파일에 사용함

with statement



- 파일과 소켓 같이 '리소스'에 접근하는 경우 핸들(handle) 처리가 중요하다
 - 리소스를 제어를 위한 핸들을 얻고, 반환하는 작업이 필요하다
 - 핸들 반환 전에 특정 조건에 의해 예외가 발생하여 리소스 정리를 못하고 종료 될 수 있다
 - with문은 파일, 소켓 같이 열고 닫는 자원 접근 시 사용 된다
 - with 문을 통해 리소스 접근을 특정 블록 내의 동작을 제한한다
 - with문 블록이 끝날 때 리소스의 자동으로 해제 처리 하는 것을 보장한다
 - with문의 용법
 - with문을 콤마를 이용해 나열해 사용하거나 중첩하여 사용할 수 있다

```
with expression1 [as target1[, expression2 as target2]]:  
    실행할 문장들
```

```
with expression1 [as target1] :  
    [with expression2 [as target2] :]  
        실행할 문장들
```


파일 사용에서의 오류 처리



🍃 파일을 사용할 때의 오류 처리의 예를 살펴보고 분석하라

🍃 try ~ except 와 with 문 함께 사용

```
try :  
    with open(file) as fp:  
        msg = fp.read()  
        print(msg)  
except OSError:  
    print("File doesn't exist or No read permission")
```

🍃 파일이 존재하고, 읽기 권한이 있는지 확인 후 사용

```
import os  
if os.access(file, os.F_OK) and os.access(file, os.R_OK):  
    with open(file) as fp:  
        msg = fp.read()  
        print(msg)  
else:  
    print("File doesn't exist or No read permission")
```

shelve 모듈로 객체 저장하고 읽기



shelve 모듈을 이용하여 이진 파일에 객체를 저장하고, 읽을 수 있다

```
import shelve
```

```
mydata = shelve.open('mydata')  
fruits = ['orange', 'banana', 'apple']  
1 mydata['fruits'] = fruits  
2 print(mydata['fruits'])  
mydata.close()
```

- key 를 이용해 객체를 저장하고 불러옴
- key는 문자열을 사용함

mydata.bak
mydata.dat
mydata.dir

3개의 파일이 생성됨
OS X → mydata.db

- mode 부여 필요 없음
- 읽고, 쓰기 모두 가능하게 열림
- 한 번 쓰기 한 내용은 계속 유지됨
- 같은 'key'로 여러 번 저장 시
→ 최근 것으로 저장됨 (갱신)



🍃 glob 모듈은 윈도우의 dir, 리눅스의 ls 명령과 유사한 기능을 제공한다

🍃 glob 모듈의 주요 메서드는 다음과 같다

메서드	기능
glob.glob(path)	<ul style="list-style-type: none">▪ path에 대응되는 모든 파일 및 디렉터리 리스트 반환▪ 와일드 카드 문자('*', '?') 및 [...] 사용 가능
glob.iglob(path)	<ul style="list-style-type: none">▪ glob()와 같은 기능을 수행하지만 결과를 generator로 반환▪ 한 번에 결과를 리스트에 담지 않음▪ 결과가 매우 많은 경우 유용하게 사용될 수 있음

glob 문자	기능	예시
?	1글자 문자를 대신함	a??.txt
*	0글자 이상의 여러 글자 문자를 대신함	a*.xlsx
[...]	[] 내부에 포함된 글자 중 1개 문자	a[bcd].csv