



데이터 처리 프로그래밍

Data Processing Programming

06 입력과 출력

목차

1. 표준 입출력
2. 파일 입출력



1. 표준 입출력

표준 입출력 개념



- `input()` 함수: 콘솔 창을 통해서 사용자 입력을 받는 표준 입력 함수
- `input()` 함수 안에 입력 받기 위한 질문을 텍스트로 넣을 수 있음

```
>>> input = input()
>? 사용자 입력
>>> input
'사용자 입력'
```

```
>>> name = input("이름: ")
이름: >? 이수안
>>> name
'이수안'
>>> age = input("나이: ")
나이: >? 28
>>> age
'28'
```

- print() 함수: 콘솔 창을 통해서 결과를 출력하는 표준 출력 함수

```
>>> print("Hello World")
Hello World
>>> print("안녕! 파이썬")
안녕! 파이썬
>>> i = 123
>>> print(i)
123
>>> f = 1.23
>>> print(f)
1.23
>>> s = "SuanLab"
>>> print(s)
SuanLab
>>> l = ["One", "Two", "Three"]
>>> print(l)
['One', 'Two', 'Three']
```

- print() 함수 내에서 문자열 출력

```
>>> print("SuanLab" ":" "Suan" "Computer" "Laboratory")
SuanLab:SuanComputerLaboratory
>>> print("SuanLab"+" ":"Suan"+"Computer"+"Laboratory")
SuanLab:SuanComputerLaboratory
>>> print("SuanLab", ":", "Suan", "Computer", "Laboratory")
SuanLab : Suan Computer Laboratory
```

```
>>> for i in range(2, 10):
...     for j in range(1, 10):
...         print("{0}x{1}={2}".format(i, j, i*j), end=' ')
...     print()
...
2x1=2 2x2=4 2x3=6 2x4=8 2x5=10 2x6=12 2x7=14 2x8=16 2x9=18
3x1=3 3x2=6 3x3=9 3x4=12 3x5=15 3x6=18 3x7=21 3x8=24 3x9=27
4x1=4 4x2=8 4x3=12 4x4=16 4x5=20 4x6=24 4x7=28 4x8=32 4x9=36
5x1=5 5x2=10 5x3=15 5x4=20 5x5=25 5x6=30 5x7=35 5x8=40 5x9=45
6x1=6 6x2=12 6x3=18 6x4=24 6x5=30 6x6=36 6x7=42 6x8=48 6x9=54
7x1=7 7x2=14 7x3=21 7x4=28 7x5=35 7x6=42 7x7=49 7x8=56 7x9=63
8x1=8 8x2=16 8x3=24 8x4=32 8x5=40 8x6=48 8x7=56 8x8=64 8x9=72
9x1=9 9x2=18 9x3=27 9x4=36 9x5=45 9x6=54 9x7=63 9x8=72 9x9=81
```


Lab. 섭씨 온도를 화씨 온도로 변환



- 섭씨 온도를 화씨 온도로 변환하는 공식
- $F = (C * 1.8) + 32$

```
>>> celsius =   
섭씨 온도: >? 27  
>>> fahrenheit =   
>>> print("화씨 온도: ", fahrenheit)  
화씨 온도: 80.6
```

Lab. 구구단 중의 하나의 단을 입력받아 계산하기



- 섭씨 온도를 화씨 온도로 변환하는 공식
- $F = (C * 1.8) + 32$

```
>>> i =   
출력할 단: >? 8  
>>> for j in range(1, 10):  
...     print()  
...  
8x1=8  
8x2=16  
8x3=24  
8x4=32  
8x5=40  
8x6=48  
8x7=56  
8x8=64  
8x9=72
```



2. 파일 입출력

파일 입출력 개념



파일 입출력 과정

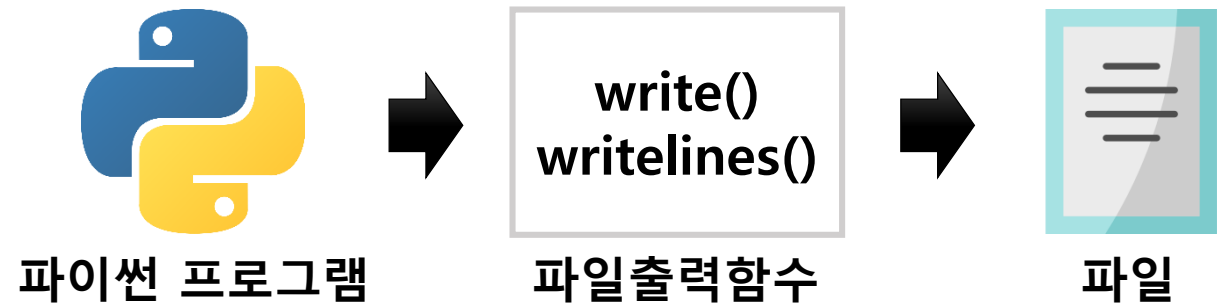


■ 파일.txt 파일을 생성하고 열기

파일 모드	설명
(생략)	r과 동일한 모드
r	읽기 모드 (기본값)
w	쓰기 모드 기존에 파일이 있으면 덮어쓰기
a	쓰기 모드 기존에 파일이 있으면 이어서 쓰기 (append)
+	읽기/쓰기 모드
t	텍스트 모드 (기본값) 텍스트 파일을 처리
b	이진 모드 (이진 파일을 처리)

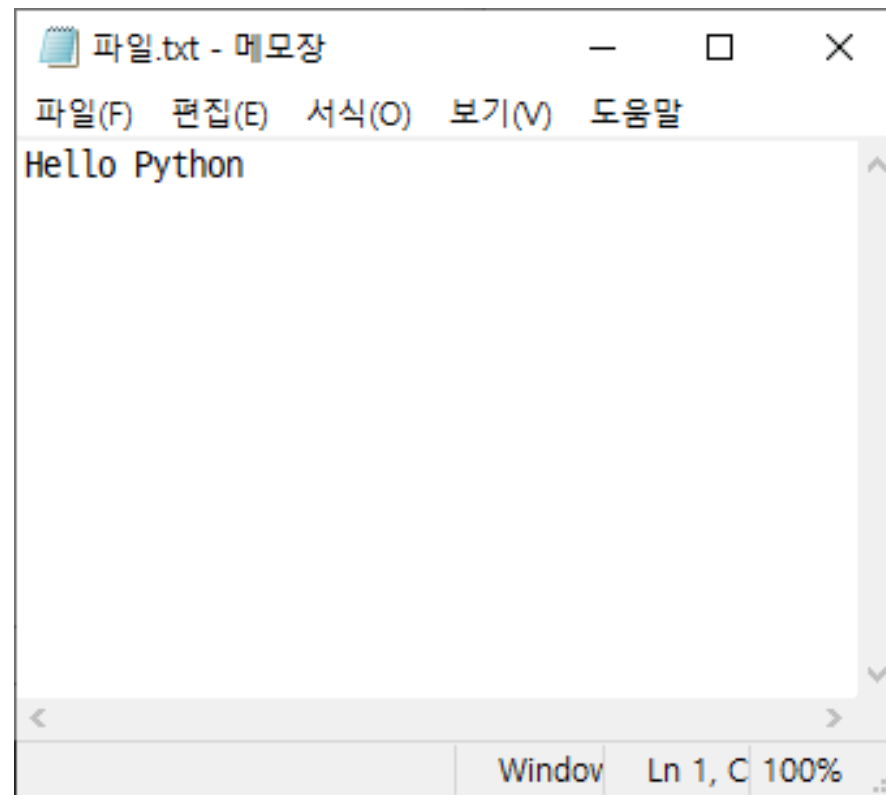
```
>>> f = open("파일.txt", 'w')
>>> f = open("파일.txt", 'r')
>>> f.close()
```

텍스트 파일 쓰기



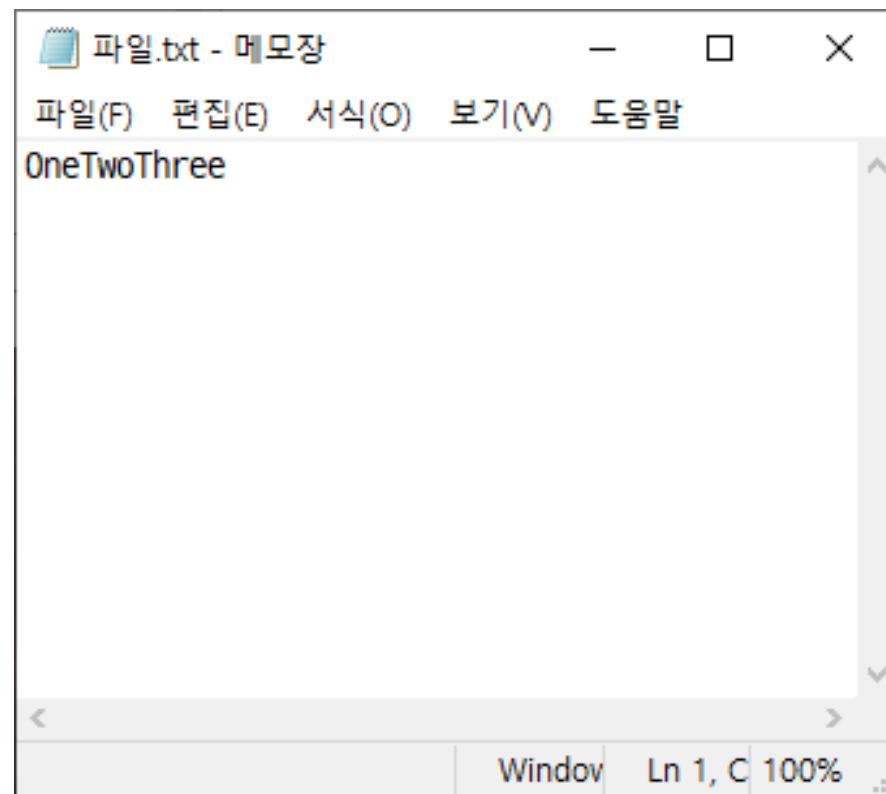
- write() 함수를 이용하여 파일에 쓰기

```
>>> f = open("파일.txt", 'w')  
>>> f.write("Hello Python")  
12  
>>> f.close()
```



- `writelines()` 함수를 이용하여 `list`를 파일에 쓰기

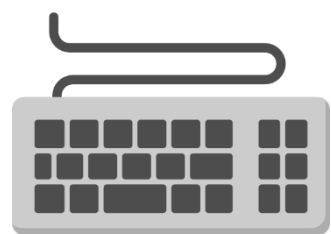
```
>>> f = open("파일.txt", 'w')
>>> list = ["One", "Two", "Three"]
>>> f.writelines(list)
>>> f.close()
```



- 구구단 결과를 파일에 쓰기

```
>>> f = open("파일.txt", 'w')
>>> for i in range(2, 10):
...     for j in range(1, 10):
...         
...     else:
...         f.write("\n")
...
>>> f.close()
```

표준 입력 → 파일 쓰기



표준입력장치



`input()`

표준입력함수



파이썬 프로그램



`write()`
`writelines()`

파일출력함수



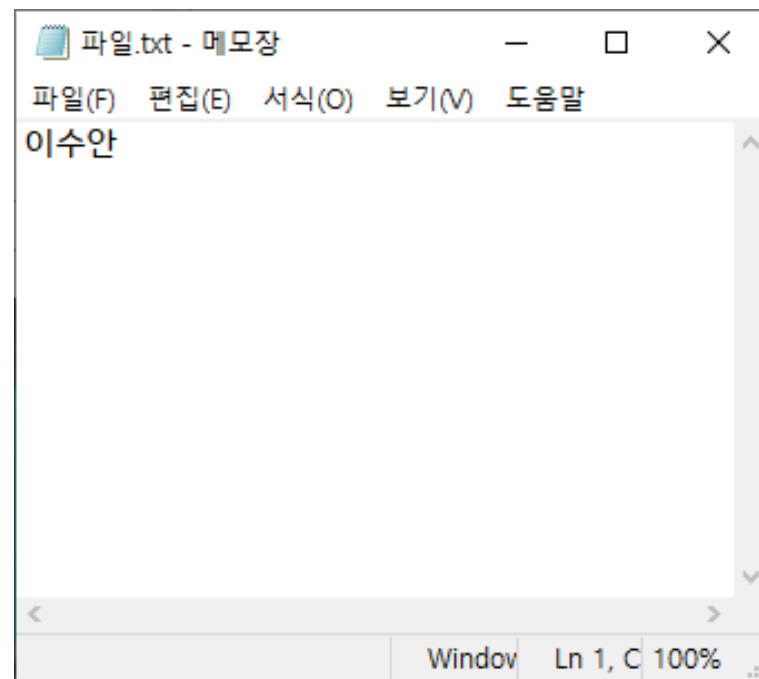
파일

표준 입력 → 파일 쓰기



- 표준 입력을 `input()` 함수를 통해 입력 받고, `write()` 함수를 통해 파일에 쓰기

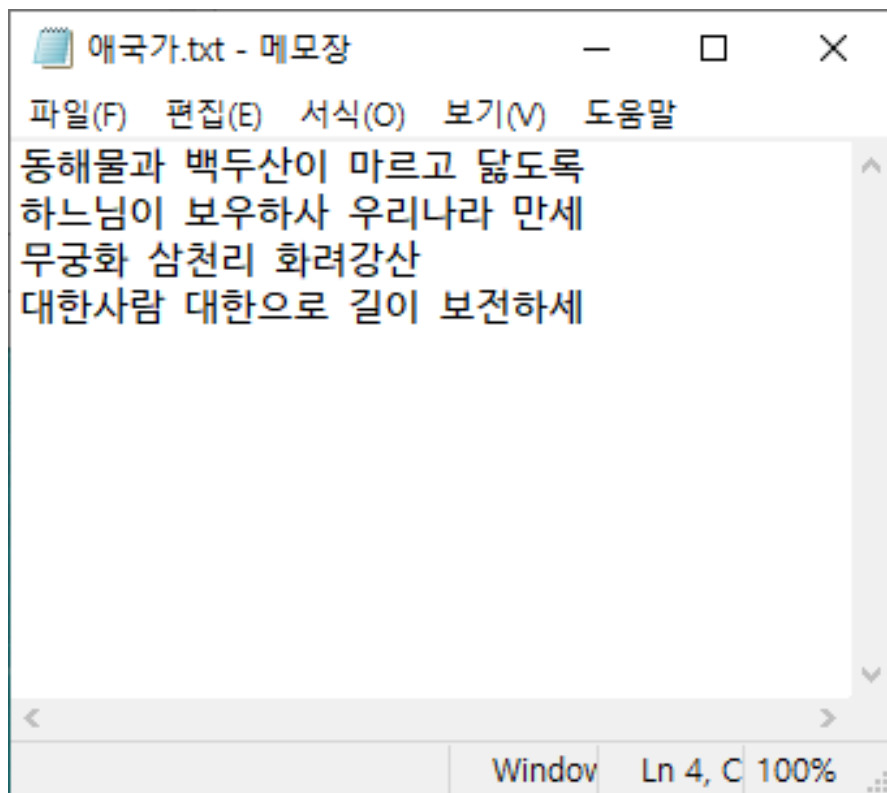
```
>>> f = open("파일.txt", 'w')
>>> input = input("이름: ")
이름: >? 이수안
>>> f.write(input)
3
>>> f.close()
```



텍스트 파일 출력



텍스트 파일 읽고 화면에 출력 readline()



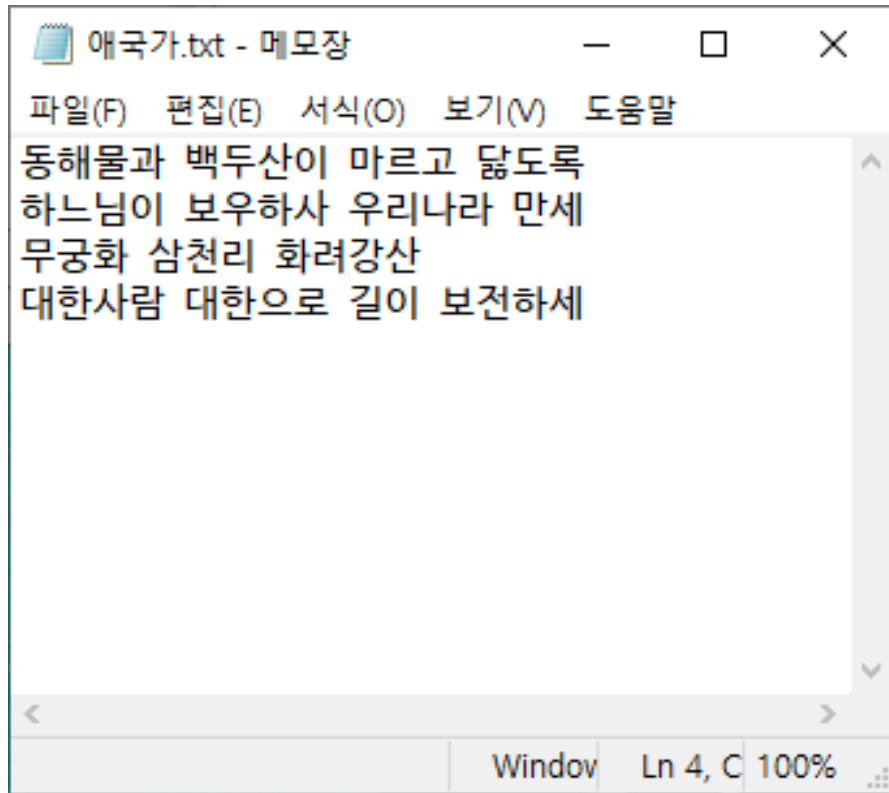
```
>>> f = open("애국가.txt", 'r')
>>> while True:
...     line = f.readline()
...     if not line: break
...     print(line)
...
동해물과 백두산이 마르고 닳도록

하느님이 보우하사 우리나라 만세

무궁화 삼천리 화려강산

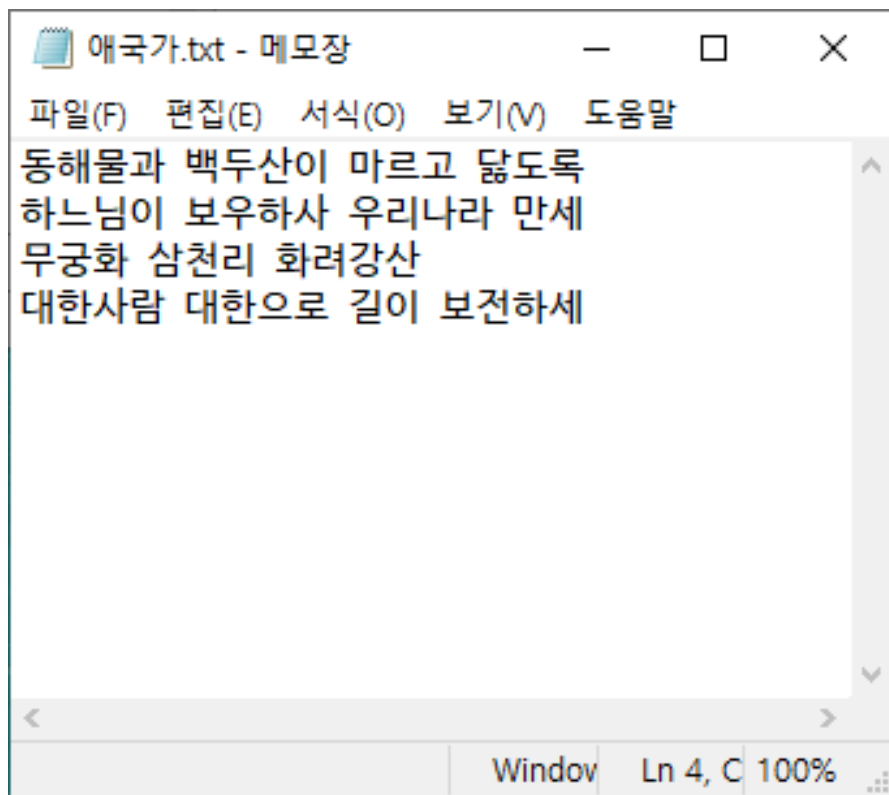
대한사람 대한으로 길이 보전하세
>>> f.close()
```

텍스트 파일 읽고 화면에 출력 readlines()



```
>>> f = open("애국가.txt", 'r')
>>> lines = f.readlines()
>>> for line in lines:
...     print(line)
...
동해물과 백두산이 마르고 닳도록
하느님이 보우하사 우리나라 만세
무궁화 삼천리 화려강산
대한사람 대한으로 길이 보전하세
>>> f.close()
```

텍스트 파일 읽고 화면에 출력 read()



```
>>> f = open("애국가.txt", 'r')
>>> data = f.read()
>>> print(data)
동해물과 백두산이 마르고 닳도록
하느님이 보우하사 우리나라 만세
무궁화 삼천리 화려강산
대한사람 대한으로 길이 보전하세
>>> f.close()
```


with문으로 파일 열고 닫기



- 항상 파일을 open() 함수로 열고, close() 함수로 닫아야 하는 일을 자동으로 처리
- with문을 이용하면 with 블록 내에서 파일을 열고 벗어나면 파일을 닫음

```
>>> f = open("파일.txt", 'w')
>>> f.write("Hello Python")
12
>>> f.close()
```



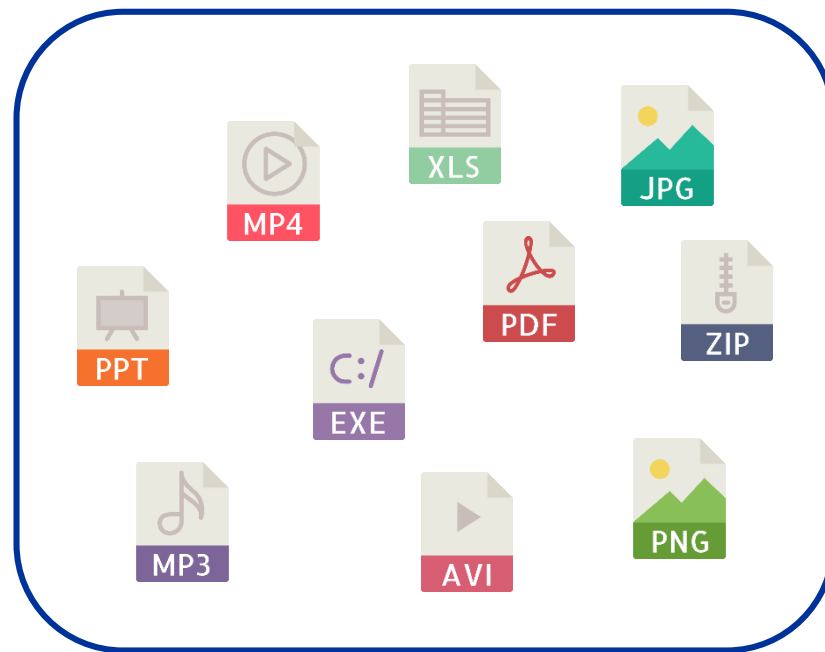
```
>>> with open("파일.txt", 'w') as f:
...     f.write("Hello Python")
...
```

이진 파일 개념

- 이진 파일^{binary file}: 텍스트가 아닌 비트 단위의 파일
- 이진 파일의 종류: 그림, 음악, 영상, 프로그램 파일 등



텍스트 파일



이진 파일

- 하나의 이진 파일을 읽어서 다른 이진 파일로 쓰기 (복사)



image.png



image2.png

```
>>> input = open("image.png", "rb")
>>> output = open("image2.png", "wb")
>>> while True:
...     inFp = input.read(1)
...     if not inFp: break
...     output.write(inFp)
...
>>> input.close()
>>> output.close()
```

- 디렉토리나 파일을 다루는 다양한 함수를 제공하는 `shutil`와 `os` 라이브러리
- `mkdir()`: 디렉토리 생성

```
>>> import os
>>> import shutil
>>> os.mkdir('test')
>>> os.mkdir('test/1')
>>> os.mkdir('test/2')
>>> os.mkdir('test/3')
```

- `shutil.copy()`: 파일 복사
- `shutil.copytree()`: 디렉토리 전체 복사

```
>>> import shutil
>>> shutil.copy('image.png', 'image_copy.png')
>>> shutil.copytree('C:/Windows/debug/', 'debug')
```

- `isdir()`: 디렉토리 존재 확인
- `isfile()`: 파일 존재 확인
- `exists()`: 디렉토리/파일 존재 확인

```
>>> import os.path
>>> os.path.isdir('C:\\Windows\\Fonts')
True
>>> os.path.isfile('C:\\Windows\\Fonts\\times.ttf')
True
>>> os.path.exists('C:\\Windows\\Fonts')
True
>>> os.path.exists('C:\\Windows\\Fonts\\times.ttf')
True
```

- `os.walk()`:
디렉토리 목록
보기

```
>>> import os
>>> for dir_name, dir_list, file_names in os.walk('C:\\Windows\\Fonts'):
...     for file_name in file_names:
...         print(os.path.join(dir_name, file_name))
...
C:\Windows\Fonts\8514fix.fon
C:\Windows\Fonts\8514fixe.fon
C:\Windows\Fonts\8514fixg.fon
C:\Windows\Fonts\8514fixr.fon
C:\Windows\Fonts\8514fixt.fon
C:\Windows\Fonts\8514oem.fon
C:\Windows\Fonts\8514oeme.fon
C:\Windows\Fonts\8514oemg.fon
C:\Windows\Fonts\8514oemr.fon
C:\Windows\Fonts\8514oemt.fon
C:\Windows\Fonts\8514sys.fon
```

디렉토리 및 파일 삭제

- `os.remove()`: 파일 삭제
- `shutil.rmtree()`: 디렉토리 안에 모든 파일/
디렉토리 삭제

```
>>> import os
>>> os.remove('image_copy.png')
>>> os.remove('image2.png')
```

```
>>> shutil.rmtree('test')
```


파일 크기

- `os.path.getsize()`: 파일의 크기를 바이트 단위로 출력

```
>>> import os.path
>>> os.path.getsize('image.png')
7729
```

파일 압축

- zipfile: 압축 관련 라이브러리
- ZipFile(): 압축 파일 열기
- write(): 압축 파일에 쓰기
- extractall(): 전체 압축 해제
- close(): 압축 파일 닫기

```
>>> import zipfile
>>> comp = zipfile.ZipFile('new.zip', 'w')
>>> comp.write('image.png')
>>> comp.close()
```

```
>>> decomp = zipfile.ZipFile('new.zip', 'r')
>>> decomp.extractall('new')
>>> decomp.close()
```

Q & A