

OS

: 운영 체제 (Operating System)를 제어하는 모듈

Ex)

C:\users\user_name\Onedrive\문서\python

.vscode

launch.json

csv

csv_read_1.py

csv_write_2.py

csv_write_2-1.py

...

os

os_1

os_2

pickle

pickle_1

pickle_2

기본적인 기능들

- `os.getcwd()`

현재 작업 중인 디렉터리 위치를 가져옵니다.

```
import os

cwd = os.getcwd()
print(cwd)

# 실행 결과
# C:\Users\jky07\OneDrive\문서\python
```

- `os.listdir(path)`

해당 경로에 있는 디렉터리와 파일들의 리스트를 반환합니다.

```
import os

listdir = os.listdir()
print(listdir)

# 실행 결과
# ['.vscode', 'csv', 'os', 'pickle']
```

- `os.chdir(path)`

현재 작업 디렉터리 위치를 변경합니다.

`..` 은 상위 디렉터리를 나타냅니다.

```
import os

os.chdir('pickle') # -- pickle 디렉터리로 위치 변경

chdir = os.getcwd()
print(chdir) # -- 현재 작업 디렉터리 위치 출력

# 실행 결과
# C:\Users\jky07\OneDrive\문서\python\pickle

listdir = os.listdir() # -- 현재 작업 디렉터리에 있는
print(listdir) # -- 파일, 디렉터리 반환

# 실행 결과
# ['pickle_1.py', 'pickle_2.py']

os.chdir('..') # -- 상위 디렉터리 출력
chdir2 = os.getcwd()
print(chdir2)

# 실행 결과
# C:\Users\jky07\OneDrive\문서\python
```

- `os.access(path, mode)`

#F_OK : 해당 path의 존재 여부 확인

W_OK : path의 쓰기 가능 여부 확인

X_OK : path의 실행 가능 여부 확인

R_OK : path의 읽기 가능 여부 확인

```
import os

access_F = os.access('pickle', os.F_OK)
print(access_F)

# 실행 결과
# True
```

```
access2 = os.access('pickle' , os.W_OK | os.X_OK | os.R_OK)
print(access2)

# 실행 결과
# True
```

- `os.mkdir(path[, mode])`

path에 해당하는 디렉토리를 생성합니다.

- `os.makedirs(path[, mode])`

path에 해당하는 재귀적 디렉토리를 생성합니다.

이미 디렉토리가 있거나 권한이 없는 경우 다음과 같은 에러가 발생합니다.

```
import os

os.mkdir('test') # -- test 디렉토리 생성

os.makedirs('test1/ex1/ex2/empty') # 실행 후 # -- test1\ex1\ex2\empty 디렉토리 생성

os.makedirs('test1/ex1/ex2/empty') # 실행 # -- FileExistsError 발생

# FileExistsError
# Traceback (most recent call last):
# File "<stdin>", line 1, in <module>
# File "C:\Users\jky07\AppData\Local\Programs\Python\Python39\lib\os.py", line 225, in makedirs
# mkdir(name, mode)
# FileExistsError: [WinError 183] 파일이 이미 있으므로 만들 수 없습니다: 'test1/ex1/ex2/empty'
```

- `os.remove(path) / os.unlink(path)`

파일을 삭제합니다.

```
import os

os.remove('test.txt')

os.unlink('empty.txt')
```

- `os.rmdir(path)`

path에 해당하는 디렉토리를 삭제합니다.

디렉토리는 비어 있어야 합니다. 비어 있지 않을 경우 다음과 같은 에러가 발생합니다.

- `os.removedirs(path)`

path에 해당하는 재귀적 디렉토리를 삭제합니다.

```
import os

os.rmdir('test') # -- 비어 있지 않을 경우 OSError 에러 발생

os.removedirs('test1/ex1/ex2/empty')

# OSError
# Traceback (most recent call last):
# File "<stdin>", line 1, in <module>
# OSError: [WinError 145] 디렉터리가 비어 있지 않습니다: 'test'
```

- **os.rename(src, dst)**

src → dst 로 이름을 변경하거나 이동합니다. 파일, 디렉토리 모두 적용 가능합니다.

- **os.rename(src, dst)**

src → dst 로 이름을 변경하거나 이동합니다. 파일, 디렉토리 모두 적용 가능합니다.

renames는 이동 시 디렉토리들을 자동으로 생성합니다.

```
import os

os.rename('empty.txt', 'ex.txt')

os.renames('ex.txt', 'test/test2.txt')
```

```
import os

# 디렉토리 이동
os.rename('test\\image', 'test\\image_move') # -- 이동하려는 목적지에 상위 디렉터리가
# -- 존재하지 않으면 에러 발생

os.renames('test\\image', 'test2\\image') # -- 이동하려는 목적지에 상위 디렉토리를
# -- 만들려면 os.renames를 사용
```

- **os.stat(path)**

아래와 같은 순서로 경로에 대한 정보를 출력합니다.

```
import os

os.stat('empty.txt')

# 실행 결과
# os.stat_result
# (st_mode=33206, st_ino=1125899907407435, st_dev=3837836524,
# st_nlink=1, st_uid=0, st_gid=0, st_size=0, st_atime=1627695275,
# st_mtime=1627697740, st_ctime=1627695275)
```

