

파이썬

29강. 파일 시스템

1. 파일 시스템

- 프로그램을 이용해서 파일로부터 자료를 읽고, 쓰기 위해서는 운영체제에서 지원하는 파일 시스템의 도움을 받아야 한다. 예를 들면 특정 파일의 자료를 읽기 위해서는 파일이 저장된 디렉터리(폴더) 위치와 경로를 지정해야 한다. 또한 자료를 파일에 쓰기 위해서는 해당 파일이 만들어질 디렉터리를 만들고, 이동하는 작업 등이 필요하다. 이절에서는 파일과 디렉터리를 조작하고 경로를 설정할 수 있는 `os` 와 `glob` 모듈에 대해서 알아본다.

2. 파일과 디렉터리 관련 함수

- os 모듈에서 제공하는 파일과 디렉터리를 조작하고 처리하는 관련 함수들에 대해서 알아본다.

함수(파라미터) ↵	기능(실습 예문) ↵
↵ os.getcwd() ↵	현재 작업 디렉터리의 위치를 반환한다. ↵ >>>os.getcwd() ↵
↵ os.chdir(path) ↵	현재 작업 디렉터리의 위치를 변경한다. ↵ >>>chdir('test') ↵
↵ os.listdir(path) ↵	해당 경로에 존재하는 파일과 디렉터리의 목록을 반환한다. ↵ >>>os.listdir('.') ↵
↵ os.remove(path) ↵	해당 파일을 삭제한다. ↵ >>>os.remove('test.txt') ↵
↵ os.mkdir(path) ↵	해당 디렉터를 생성한다. ↵ >>>os.mkdir('test2') ↵

2. 파일과 디렉터리 관련 함수

↵ os.rmdir(path) ↵	해당 디렉터를 삭제한다.(내용이 있는 디렉터리 삭제 불가) ↵
	>>>os.rmdir('test2') ↵
↵ os.makedirs(path) ↵	여러 개의 디렉터를 순차적으로 생성한다. 만약 이미 디렉터리가 있으면 예외가발생한다. ↵
	>>>os.makedirs('test3/test4/test5') ↵
↵ os.removedirs(path)) ↵	여러 개의 디렉터를 역순으로 삭제한다. ↵
	>>>os.removedirs('test3/test4/test5') ↵
↵ os.rename(src, dst)	파일이나 디렉터리의 이름(src)을 dst로 변경한다. ↵
	>>>os.rename('test.txt', 'example.txt') ↵

3. os 모듈의 파일과 디렉터리 관련 함수

- 다음은 os 모듈에서 제공하는 파일과 디렉터리 관련 함수의 실행 결과를 나타내는 예문이다.

3. os 모듈의 파일과 디렉터리 관련 함수

```
chapter08.lecture.step03_file_system.py↵
```

```
import os.path # windows 파일 경로를 조작하는 모듈↵

# 현재 경로 확인↵
os.getcwd() # 'D:\\Pywork\\workspace'
# 경로 변경↵
os.chdir('chapter08')↵
os.getcwd() # 'D:\\Pywork\\workspace\\chapter08'↵

# lecture 디렉터리의 step01_try_except.py 파일 절대경로
os.path.abspath('lecture/step01_try_except.py')
#'D:\\Pywork\\workspace\\chapter08\\lecture\\step01_try_except.py'↵

# step01_try_except.py 파일의 디렉터리 이름↵
os.path.dirname('lecture/step01 try except.py') #'lecture'↵

# workspace 디렉터리 유무 확인↵
os.path.exists('D:\\Pywork\\workspace') # True↵

# step01_try_except.py 파일 유무 확인↵
os.path.isfile('lecture/step01 try except.py') # True↵

# lecture 디렉터리 유무 확인↵
os.path.isdir('lecture') # True↵

# 디렉터리와 파일 분리↵
os.path.split("c:\\test\\test1.txt") # ('c:\\test', 'test1.txt')
# 디렉터리와 파일 결합↵
os.path.join('c:\\test', 'test1.txt') # 'c:\\test\\test1.txt'↵

# step01_try_except.py 파일 크기↵
os.path.getsize('lecture/step01 try except.py') # 1883 byte↵
```

4. glob 모듈

- 유닉스(Unix) 셸이 사용하는 규칙에 의해서 지정된 패턴과 일치하는 모든 파일과 디렉터리의 목록을 반환하는 관련 함수들을 제공하는 모듈이다. 목록을 검색할 때 특수문자(*, ?, [])를 이용하여 패턴을 지정할 수 있고, 검색된 목록은 순서가 없다.

4. glob 모듈

함수(파라미터) ↴	기능(실습 예문) ↴
↴ glob.glob(pathname, *, recursive=False) ↴	pathname에 해당하는 경로를 기준으로 파일과 디렉터리의 목록을 순서 없이 반환한다.(목록이 없으면 빈 리스트 반환) ↴
↴ glob.iglob(pathname, *, recursive=False) ↴	glob() 함수와 같은 값을 산출하지만, 반복 가능한 이터레이터를 반환한다. ↴
↴ glob.glob(pathname, *, recursive=False) ↴	특수문자(*, ?, [])를 이스케이프 처리한다. 특수문자가 들어있을 수 있는 임의의 리터럴 문자열을 일치시키려는 경우 유용하다. ↴

4. glob 모듈

- 다음은 프롬프트에서 glob() 함수와 특수문자를 패턴으로 지정하여 파일의 목록을 반환하는 예문이다.

4. glob 모듈

- import glob
- >>>glob.glob('test*.py') # 현재 경로에서 test로 시작하는 모든 목록 반환
- >>>glob.glob('c:/test[0-9]') # test 문자열 다음에 숫자 1개가 오는 목록 반환
- ['c:/test1', 'c:/test9']
- >>>glob.glob('c:/test[0-9]/*.txt') # test1, test9 디렉터리에 포함된 *.txt 파일 반환
- ['c:/test1\1.txt.txt', 'c:/test1\test.tx.txt', 'c:/test9\10.jpg.txt']
- >>>glob.glob('c:/test[0-9]/[0-9].*') # test1, test9 디렉터리에서 숫자 1개 파일 반환

4. glob 모듈

- ['c:/test1\WW1.txt.txt']
- >>> glob.glob('c:/test1/*.txt') # test1 디렉터리에서 *.txt 파일 반환
- ['c:/test1\WW1.txt.txt', 'c:/test1\WWtest.tx.txt']
- >>> glob.glob('c:/test1/?.txt') # test1 디렉터리에서 파일명이 1자인 txt 파일 반환
- ['c:/test1\WW1.txt']
- >>> glob.glob('c:/test1/*.txt', recursive=True) # *.txt 파일을 내림차순 정렬하여 반환
- ['c:/test1\WW1.txt', 'c:/test1\WWtest.txt']

THANK YOU