

파이썬 프로그래밍

파일과 디렉토리 다루기



한국기술교육대학교
온라인평생교육원

■ 디렉토리 다루기

1. 디렉토리에 관련된 일반 작업

1) 현재 작업 디렉토리 알아보기

```
import os  
print os.getcwd()
```

```
/Users/yhhan/git/python-e-learning
```

- cwd → current working directory의 약자
- os.getcwd() → 현재 작업하고 있는 디렉토리 불러오기

2) 작업 디렉토리 변경하기

```
os.chdir('/Users/yhhan/Public/')  
print os.getcwd()
```

```
/Users/yhhan/Public
```

- os.chdir() → 작업 디렉토리 변경

■ 디렉토리 다루기

1. 디렉토리에 관련된 일반 작업

3) 디렉토리 만들기

```
import os
```

```
os.mkdir('temp')      # 0755 기본 모드(rwxr-xr-x)로 만들어짐
```

```
os.mkdir('temp2', 0700) # 0700 모드(rwx-----)로 만들어짐
```

```
os.makedirs('temp/level1/level2') #0755 기본 모드, 중간에 필요한 디렉토리도 모두생성
```

```
print os.access('/Users/yhhan/Public/temp', os.F_OK)
```

```
print os.access('/Users/yhhan/Public/temp2', os.F_OK)
```

```
print os.access('/Users/yhhan/Public/temp/level1/level2', os.F_OK)
```

```
-----
OSError                                Traceback (most recent call last)
<ipython-input-13-acd03c396068> in <module>()
      1 import os
      2
----> 3 os.mkdir('temp')      # 0755 기본 모드(rwxr-xr-x)로 만들어짐
      4 os.mkdir('temp2', 0700) # 0700 모드(rwx-----)로 만들어짐
      5 os.makedirs('temp/level1/level2') #0755 기본 모드, 중간에 필요한
                                         디렉토리도 모두생성
```

```
OSError: [Errno 17] File exists: 'temp'
```

- `os.mkdir('temp')` → 현재 작업 디렉토리에 `temp` 디렉토리 생성
- `0700` → 권한 모드 설정
- 소유자에게는 `r`, `w`, `x` 모든 권한 부여
- 일반적인 기본 디렉토리 권한 → `755` 모드
- `7(rwx)`, `5(r이 4, x가 1)`, `5(r이 4, x가 1)`
- `makedirs()` → 재귀적으로 여러 개의 디렉토리 생성
- 이미 `temp` 디렉토리가 존재하기 때문에 `error` 발생

■ 디렉토리 다루기

1. 디렉토리에 관련된 일반 작업

4) 디렉토리 삭제

```
os.rmdir('temp2') #디렉토리에 내용이 없을 때 삭제가능
```

```
os.rmdir('temp') #디렉토리에 다른 파일이 있으면 삭제할 수 없음
```

```
-----  
OSError                                Traceback (most recent call last)  
<ipython-input-43-cb2ef2e59e2c> in <module>()  
----> 1 os.rmdir('temp') #디렉토리에 다른 파일이 있으면 삭제할 수 없음  
  
OSError: [Errno 66] Directory not empty: 'temp'
```

- rmdir() → 디렉토리에 내용이 없을 경우 삭제 가능
- 디렉토리에 내용이 있으면 삭제 불가능 (OS error 발생)

▣ 디렉토리 다루기

1. 디렉토리에 관련된 일반 작업

5) 다단계 디렉토리 삭제

- `os.removedirs(filepath)`
 - `filepath`에 지정된 디렉토리들 중 맨 오른쪽 디렉토리 부터 차례차례로 삭제한다.
 - 디렉토리에 다른 파일이 있으면 삭제하기 않고 중단

```
os.removedirs('temp/level1/level2')
```

- `os.removedirs()` → 다단계 디렉토리 삭제 (↔`makedirs`함수)
- `removedirs`에 적힌 디렉토리의 맨 오른쪽부터 단계별 삭제

6) 하위 디렉토리까지 모두 한번에 삭제

- `shutil.rmtree()`
 - 파일은 물론 하위 디렉토리까지 모두 한번에 삭제 가능
 - 조심해서 사용해야 함

```
import shutil  
shutil.rmtree('temp')
```

- `shutil.rmtree(디렉토리)` → 디렉토리 안 내용 삭제
- `tree` → 디렉토리 모양이 나무인데 나무 자체를 모두 삭제

▣ 디렉토리 다루기

1. 디렉토리에 관련된 일반 작업

7) 디렉토리 복사

- `shutil.copytree(src_filepath, dest_filepath)`
 - 하위 디렉토리와 파일등을 지니고 있는 디렉토리를 복사

```
os.mkdir('temp')  
os.mkdir('temp/temp2', 0700)  
shutil.copytree('temp', 'myweb_backup')
```

- `temp/temp2` → `temp` 밑에 `temp2` 생성
- `copytree(디렉토리, 디렉토리2)` → 디렉토리를 디렉토리2로 복사
- 하위 디렉토리가 복잡하게 존재해도 `copytree`는 모두 복사

▣ 디렉토리 다루기

2. 디렉토리(트리) 탐색하기

- `os.walk(filepath)`
 - `filepath` 부터 시작하여 재귀적으로 모든 하위 디렉토리까지 탐색을 하는 함수
 - 탐색시 발견하는 모든 파일에 대해서는 다음 튜플을 리턴함
 - * (`dirpath`, `dirnames`, `filenames`)
 - + `dirpath`: 탐색하고 있는 디렉토리 경로
 - + `dirnames`: `dirpath` 안에 존재하는 서브 디렉토리의 리스트
 - + `filenames`: `dirpath` 안에 존재하는 파일 리스트

- 아래 예는 현재 디렉토리부터 모든 하위 디렉토리 내에 존재하는 모든 `.pyc` 파일을 삭제하는 프로그램

```
import os
os.chdir('/Users/yhhan/git/python-e-learning')
print os.getcwd()
print
for path, subdirs, files in os.walk(os.getcwd()):
    for fname in files:
        if fname.endswith('.pyc'):
            fullpath = os.path.join(path, fname)
            print 'removing', fullpath
            os.remove(fullpath)
```

```
/Users/yhhan/git/python-e-learning
```

```
removing /Users/yhhan/git/python-e-learning/module_test.pyc
removing /Users/yhhan/git/python-e-learning/mymath.pyc
```

■ 디렉토리 다루기

2. 디렉토리(트리) 탐색하기

- removing → 삭제
- os.walk(디렉토리) → 디렉토리의 모든 하위 디렉토리를 탐색
- path → 방문하고 있는 디렉토리의 위치를 경로명으로 반환
- subdirs → 현재 방문하고 있는 디렉토리의 하위 디렉토리 목록
- files → 현재 방문하고 있는 디렉토리 안 파일 목록
- fname.endswith('.txt') → fname이 txt로 끝나는지 살펴봄
- path + frame = 절대 경로
- 현재 디렉토리에 있는 모든 txt로 끝나는 것 삭제
- os.walk로 다시 올라와서 하위 디렉토리도 동일 작업 진행