

# 제9강 Organizing Files

# 파일 관리 작업 사례

- 한 폴더 내 모든 하위 폴더안의 PDF 파일들을 복사하는 작업 (PDF 파일만을 대상으로)
- 수백 개의 파일 중 'spam001.txt', 'spam002.txt', 'spam003.txt' 등의 파일명에서 선행하는 0을 제거하는 작업
- 여러 폴더의 내용을 하나의 ZIP 파일로 압축하는 작업 (간단한 백업 시스템으로 사용할 수 있음)

# 학습 목차

- 파일과 폴더의 복사 이동 삭제
- 휴지통 액세스
- ZIP 압축과 해제
- 파일 정보
- 파일 비교

# 파일과 폴더 복사와 이동

`shutil.copy(src, dst)`

# 파일 하나 복사

`shutil.copytree(src, dst)`

# 폴더 전체 복사, 하위 폴더까지 한꺼번에

`shutil.move(src, dst)`

# 파일 또는 폴더 이동( 또는 이름 변경)

```
import shutil
```

```
shutil.copy('kitty.jpg', 'new_kitty.jpg')
```

```
# file 하나 복사
```

```
shutil.copytree('.', 'new_folder')
```

```
# 폴더 전체 복사, 하위 폴더까지 한번에
```

```
shutil.move('new_kitty.jpg', 'old_kitty.jpg')
```

```
# 파일 이동
```

```
shutil.move('new_folder', 'old_folder')
```

```
# 폴더 통째로 이동, 결국 폴더 이름 변경
```

```
shutil.move('old_kitty.jpg', 'old_folder')
```

```
# 파일을 폴더 안으로 이동
```

# 파일과 폴더 삭제

`os.unlink(path)` # 파일 삭제

`os.remove(path)` # 파일 삭제

`os.rmdir(path)` # 폴더 삭제(폴더가 비어 있어야 함)

`shutil.rmtree(path)` # 폴더를 무조건 삭제

## 삭제는 항상 주의 필요!

```
import os
for filename in os.listdir():
    if filename.endswith('.rxt'):
        #os.unlink(filename)
        print(filename)
```

# 휴지통 보내기 - send2trash 모듈 이용

```
from send2trash import send2trash  
  
open('garbage.txt', 'w').close()  
send2trash('garbage.txt')
```

# 휴지통 보내기 - winshell 모듈 이용

```
# 먼저 pywin32 와 winshell 모듈을 설치해야 함.  
import winshell  
  
open('garbage.txt', 'w').close()  
winshell.delete_file('garbage.txt')
```



# 휴지통 비우기

```
trash = winshell.recycle_bin() # recycle_bin() iterator
if list(trash):                 # 휴지통이 비어있지 않으면.
    trash.empty(confirm=False, show_progress=False, sound=False)
```

# 휴지통 내용 확인

```
for item in winshell.recycle_bin():  
    print(item.real_filename(), item.original_filename(), item.attributes())
```

```
# stream→file,      folder→directory
```

```
W:\$RECYCLE.BIN\S-1-5-21-893264264-912079751-901045600-1002\$_RJ1TZE9.txt W:\WorkCodingLive\2023-Script-  
Language\src\LEC09-OrganizingFiles\garbage {'canmoniker', 'capabilitymask', 'filesystem', 'hasstorage', 'canmove',  
'candelete', 'storagecapmask', 'stream', 'haspropsheet'}  
W:\$RECYCLE.BIN\S-1-5-21-893264264-912079751-901045600-1002\$_RL42DX5 W:\WorkCodingLive\2023-Script-  
Language\src\LEC09-OrganizingFiles\garbageonly {'canmoniker', 'capabilitymask', 'filesystem', 'hasstorage', 'canmove',  
'candelete', 'storagecapmask', 'stream', 'haspropsheet'}  
W:\$RECYCLE.BIN\S-1-5-21-893264264-912079751-901045600-1002\$_RWB3980 W:\WorkCodingLive\2023-Script-  
Language\src\LEC09-OrganizingFiles\test_folder {'capabilitymask', 'filesysancestor', 'filesystem', 'canmove',  
'candelete', 'storagecapmask', 'folder', 'haspropsheet'}
```

# 휴지통 내용 복구

```
for item in winshell.recycle_bin():  
    item undelete()
```

# 문제점은? 확장자는 사라진다.

# Zip 파일 읽기

```
import zipfile
import os

zf = zipfile.ZipFile('test.zip')
zf.namelist()
zinfo = zf.getinfo('file_handling.py')
zinfo.file_size
zinfo.compress_size
zf.close()
zinfo
```

# Zip 파일 추출

```
# 모두 다 압축 해제
zf = zipfile.ZipFile('test.zip')
zf.extractall()
zf.extractall('c:/outfolder')
zf.close()

# 파일 하나만 압축 해제
zf.extract('file_handling.py')

# 특정 폴더에 압축 해제
zf.extract('file_handling.py', 'c:/outfolder')
```

# Zip 파일 만들고 추가하기

# 새로운 압축 파일 만들고 파일 추가

```
zf = zipfile.ZipFile('new.zip', 'w')  
zf.write('kitty.jpg', compress_type=zipfile.ZIP_DEFLATED)  
zf.close()
```

# 기존 압축 파일에 추가

```
zf = zipfile.ZipFile('new.zip', 'a')  
zf.write('zip_jpg.py')  
zf.close()
```

# 파일 정보

```
os.path.getsize('new.zip')    # 파일 크기
ctime = os.path.getctime('new.zip')  # 생성 시각, 1970년 1월 1일 0시를 기준으로 한 경과 시간
mtime = os.path.getmtime('new.zip')  # 최종 변경 시각
```

```
import time
time.ctime(ctime)
```

```
import datetime
datetime.datetime.fromtimestamp(ctime).strftime('%Y%m%d')
```

```
datetime.datetime.fromtimestamp(ctime).strftime('%c')
datetime.datetime.fromtimestamp(ctime).strftime('%X')
```

# 파일과 폴더 비교 - filecmp 모듈

`filecmp.cmp(f1, f2, shallow=True)` # 파일 비교

`filecmp.cmpfiles(dir1, dir2, common, shallow=True)` # 폴더 비교

`filecmp.dircmp(dir1, dir2)` # 폴더 비교 객체

```
from filecmp import dircmp
def print_diff_files(dcmp):
    for name in dcmp.diff_files:
        print("diff_file %s found in %s and %s" % (name, dcmp.left,
            dcmp.right))
    for sub_dcmp in dcmp.subdirs.values():
        print_diff_files(sub_dcmp)

dcmp = dircmp('dir1', 'dir2')
print_diff_files(dcmp)
```