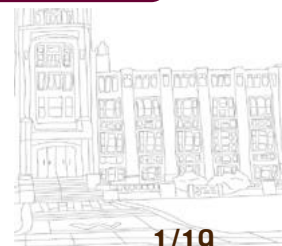




## 14장. 파일

파이썬 정보



## ❖ 목차

- 1. 파일 입출력
- 2. 파일 관리
- 3. 데이터베이스



# 1. 파일 입출력

## ❖ 파일 쓰기

### ■ 파일

- 정보를 저장하는 기본 단위
- 문서, 이미지, 멀티미디어 자료 등을 모두 보관

### ■ 파일 입출력

- 오픈
  - open(파일경로, 모드)
- 모드
  - 파일로 무엇을 할 것인지 지정

모드	설명
r	파일을 읽는다. 파일이 없으면 예외가 발생한다.
w	파일에 기록한다. 파일이 이미 있으면 덮어쓴다.
a	파일에 데이터를 추가한다.
x	파일에 기록하되 파일이 이미 있으면 실패한다.



# 1. 파일 입출력

## ■ open 함수

- 파일 입출력 준비 및 파일 객체 반환
- 출력할 데이터 인수로 전달하여 write 메서드 호출
- 파일 사용 후 close 메서드로 닫기

write

```
f = open("live.txt", "wt")
f.write("""삶이 그대를 속일지라도
슬퍼하거나 노하지 말라!
우울한 날들을 견디면
믿으라, 기쁨의 날이 오리니""")
f.close()
```



# 1. 파일 입출력

## ❖ 파일 읽기

- 적절한 함수 골라 파일 데이터 읽기
- **read** 함수

read

```
try:
    f = open("live.txt", "rt")
    text = f.read()
    print(text)
except FileNotFoundError:
    print("파일이 없습니다.")
finally:
    f.close()
```



# 1. 파일 입출력

## ■ readline 함수

- 한 줄씩 읽기
- 파일 마지막에 빈 문자열 반환

readline

```
f = open("live.txt", "rt")
text = ""
line = 1
while True:
    row = f.readline()
    if not row: break
    text += str(line) + " : " + row
    line += 1
f.close()
print(text)
```

실행결과

```
1 : 삶이 그대를 속일지라도
2 : 슬퍼하거나 노하지 말라!
3 : 우울한 날들을 견디면
4 : 믿으라, 기쁨의 날이 오리니
```



# 1. 파일 입출력

## ❖ 입출력 위치

### ■ 순차 접근

- 파일을 순서대로 읽음

### ■ 임의 접근

- 입출력 위치 바뀌가며 원하는 부분을 자유롭게 읽음

### ■ tell 함수

- 입출력 위치 조사

### ■ seek 함수

- 위치 변경



# 1. 파일 입출력

## ❖ 내용 추가

### ■ a 모드

- 기존 내용 그대로 유지하고 뒤에 덧붙임

### ■ w 모드

- 파일 이미 있는 경우 덮어쓰고 새로 만들

#### append

```
f = open("live.txt", "a")  
f.write("\n\n푸쉬킨 형님의 말씀이었습니다.")  
f.close()
```

#### with as

```
with open("live.txt", "rt") as f:  
    text = f.read()  
print(text)
```





## 2. 파일 관리

### ❖ 파일 관리 함수

함수	설명
shutil.copy(a, b)	파일을 복사한다.
shutil.copy(a, b)	디렉토리를 복사한다. 서브 디렉토리까지 전부 복사한다.
shutil.move(a, b)	파일을 이동한다.
shutil.rmtree(path)	디렉토리를 삭제한다.
os.rename(a, b)	파일 이름을 변경한다.
os.remove(f)	파일을 삭제한다.
os.chmod(f, m)	파일의 퍼미션을 변경한다.
shutil.chown(f, u, g)	파일의 소유권을 변경한다.
os.link(a, b)	하드 링크를 생성한다.
os.symlink(a, b)	심볼릭 링크를 생성한다.

#### filecopy

```
import shutil

shutil.copy("live.txt", "live2.txt")
```



## 2. 파일 관리

### ❖ 디렉토리 관리 함수

함수	설명
<code>os.chdir(d)</code>	현재 디렉토리를 변경한다.
<code>os.mkdir(d)</code>	디렉토리를 생성한다.
<code>os.rmdir(d)</code>	디렉토리를 제거한다.
<code>os.getcwd()</code>	현재 디렉토리를 조사한다.
<code>os.listdir(d)</code>	디렉토리의 내용을 나열한다.
<code>glob.glob(p)</code>	패턴과 일치하는 파일의 목록을 나열한다.

함수	설명
<code>os.path.isabs(f)</code>	절대 경로인지 조사한다.
<code>os.path.abspath(f)</code>	파일의 절대 경로를 구한다.
<code>os.path.realpath(f)</code>	원본 파일의 경로를 구한다.
<code>os.path.exists(f)</code>	파일의 존재 여부를 조사한다.
<code>os.path.isfile(f)</code>	파일인지 조사한다.
<code>os.path.isdir(f)</code>	디렉토리인지 조사한다.



## 2. 파일 관리

listdir

```
import os

files = os.listdir("c:\\Test")
for f in files:
    print(f)
```

실행결과

다비치-안녕이라고 말하지마.mp3  
멜로디데이-All About.mp3  
싸이 - 강남스타일.mp3  
아이유-너랑 나.mp3  
알리 - 서약.mp3  
인증샷-수영복.jpg



## 2. 파일 관리

### ❖ 파일 관리 유틸리티

#### ■ Ex) 파일명 일괄 수정

changenname

```
import os

path = "c:\\\\Test"
files = os.listdir(path)
for f in files:
    if (f.find("-") and f.endswith(".mp3")):
        name = f[0:-4]
        ext = f[-4:]
        part = name.split("-")
        newname = part[1].strip() + "-" + part[0].strip() + ext
        print(newname)
        os.rename(path + "\\\" + f, path + "\\\" + newname)
```

실행결과

안녕이라고 말하지마-다비치.mp3  
All About-멜로디데이.mp3  
강남스타일-싸이.mp3  
너랑 나-아이유.mp3  
서약-알리.mp3



## 2. 파일 관리

- **lsdir 함수**
  - 모든 파일 목록 조사한 후 각 파일 순회
  - 해당되는 파일 재조립
- **split 함수**
  - - 기준으로 파일명을 두 조각으로 나눔
  - 순서 바꾸어 합치기 위함
- **strip 함수**
  - 불필요한 공백 제거



# 3. 데이터베이스

## ❖ SQLite

- 데이터베이스 관리 시스템 (DBMS)
- 무료 이용 가능
- <http://www.sqlite.org>

## ❖ 테이블 생성

- 스크립트로 DB 생성
- **connect** 메서드
  - DB 파일과 연결하여 데이터베이스 열기
- **cursor** 메서드
  - 커서 객체 구함
- **execute** 메서드
  - SQL 명령 수행

```
con = sqlite3.connect(DB 파일)
cursor = con.cursor()
```

# 여기서 SQL 명령을 실행한다.

```
cursor.close()
con.close()
```



### 3. 데이터베이스

makedb

```
import sqlite3

con = sqlite3.connect('addr.db')
cursor = con.cursor()

cursor.execute("DROP TABLE IF EXISTS tblAddr")
cursor.execute("""CREATE TABLE tblAddr
    (name CHAR(16) PRIMARY KEY, phone CHAR(16), addr TEXT)""")

cursor.execute("INSERT INTO tblAddr VALUES ('김상형', '123-4567', '오산')")
cursor.execute("INSERT INTO tblAddr VALUES ('한경은', '555-1004', '수원')")
cursor.execute("INSERT INTO tblAddr VALUES ('한주완', '444-1092', '대전')")

con.commit()

cursor.close()
con.close()
```



# 3. 데이터베이스

## ❖ 조회

- 데이터 제대로 저장되어 있는지 확인
- **SELECT 명령**

selectdb

```
import sqlite3

con = sqlite3.connect('addr.db')
cursor = con.cursor()

cursor.execute("SELECT * FROM tblAddr")
table = cursor.fetchall()
for record in table:
    print("이름 : %s, 전화 : %s, 주소 : %s" % record)

cursor.close()
con.close()

이름 : 김상형, 전화 : 123-4567, 주소 : 오산
이름 : 한경은, 전화 : 555-1004, 주소 : 수원
이름 : 한주완, 전화 : 444-1092, 주소 : 대전
```





# 3. 데이터베이스

## ■ fetchall 메서드

- 모든 레코드 한꺼번에 읽어 리스트로 반환

## ■ fetchone 메서드

- 한 레코드씩 순서대로 읽음

selectdb2

```
import sqlite3

con = sqlite3.connect('addr.db')
cursor = con.cursor()

cursor.execute("SELECT * FROM tblAddr")
while True:
    record = cursor.fetchone()
    if record == None:
        break
    print("이름 : %s, 전화 : %s, 주소 : %s" % record)

cursor.close()
con.close()
```



# 3. 데이터베이스

## ❖ 수정 및 삭제

- 특정 필드값 수정

- UPDATE 명령

- WHERE 절과 함께 사용

updatedb

```
import sqlite3

con = sqlite3.connect('addr.db')
cursor = con.cursor()

cursor.execute("UPDATE tblAddr SET addr = '제주도' WHERE name = '김상형'")
con.commit()

cursor.close()
con.close()
```

- DELETE 명령

- 레코드 삭제





# Thank You !

파이썬 정복

