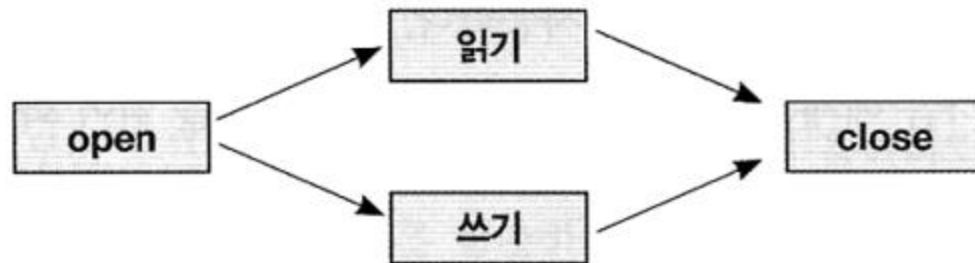

파일

파일 입출력

❖ 파일 쓰기

- open(파일경로, 모드)
- 모드
 - r : 읽기, 파일이 없는 경우 예외 발생
 - w : 쓰기, 파일이 없으면 새로 생김
 - a : 추가
 - x : 쓰기용으로 여나 기존 파일이 있는 경우 실패
 - t : text 모드로 열기
 - b : binary 모드로 열기



파일 입출력

❖ 파일 쓰기

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

- live.txt

삶이 그대를 속일지라도
슬퍼하거나 노하지 말라!
우울한 날들을 견디면
믿으라, 기쁨의 날이 오리니

파일 입출력

❖ 파일 읽기

- `f.read()` -> 파일 전체 내용
- `f.read(n)` -> n개의 내용
- `f.readline()` -> 한 줄
- `f.readlines()` -> 전체 라인 리스트
 - 각 라인의 끝에 개행 문자가 들어 있음

파일 입출력

❖ 파일 읽기

- o live.txt

삶이 그대를 속일지라도
슬퍼하거나 노하지 말라!
우울한 날들을 견디면
믿으라, 기쁨의 날이 오리니

파일 입출력

❖ 파일 읽기

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

삶이 그대를 속일지라도
슬퍼하거나 노하지 말라!
우울한 날들을 견디면
믿으라, 기쁨의 날이 오리니

파일 입출력

❖ 파일 읽기

```
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)
```

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

파일 입출력

❖ 파일 읽기

```
f = open("live.txt", "rt")
rows = f.readlines()

for row in rows:
    print(row, end="")

f.close()
```

삶이 그대를 속일지라도
슬퍼하거나 노하지 말라!
우울한 날들을 견디면
믿으라, 기쁨의 날이 오리니

파일 입출력

❖ 파일 읽기

```
f = open("live.txt", "rt")

for line in f:
    print(line, end="")

f.close()
```

삶이 그대를 속일지라도
슬퍼하거나 노하지 말라!
우울한 날들을 견디면
믿으라, 기쁨의 날이 오리니

파일 입출력

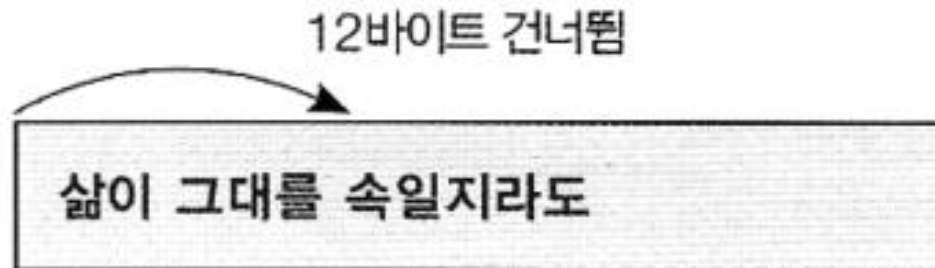
❖ 입출력 위치

- seek(위치, 기준)
 - 위치
 - 기준으로부터 얼마나 떨어진 곳인지 바이트 단위로 지정
 - 한글의 경우 주의 필요
 - 기준
 - 0: 파일의 처음 위치
 - 1: 현재 위치
 - 2: 파일의 끝 위치

파일 입출력

❖ 입출력 위치

```
f = open("live.txt", "rt")  
  
f.seek(12, 0)    # 12 바이트가 문자 중간 위치에 있으므로  
text = f.read()  # 예외 발생  
f.close()  
  
print(text)
```



파일 입출력

❖ 내용 추가

- w 모드
 - 기존에 파일이 존재하는 경우 내용을 모두 지우고 다시 작성
- a 모드
 - 기존에 파일이 존재하는 경우 파일의 끝에 내용을 추가

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

삶이 그대를 속일지라도
슬퍼하거나 노하지 말라!
우울한 날들을 견디면
믿으라, 기쁨의 날이 오리니

푸쉬킨 형님의 말씀

파일 입출력

❖ 파일 예외 처리

```
try:
    f = open("live.txt", "rt")
    text = f.read()
    print(text)
except FileNotFoundError:
    print("파일이 없습니다.")
finally:
    f.close()    # 파일 열기에 실패한 경우 f가 정의되지 않아 또 예외 발생
```

파일 입출력

❖ with ~ as 문 (컨텍스트 매니저)

- open() 함수와 함께 with ~ as문을 사용하면 명시적으로 close() 함수를 호출하지 않아도 파일이 항상 닫힘.

with open(파일이름) as 파일객체:

코드블록
이곳에서 읽거나
쓰기를 한 후
그냥 코드를 빠져나감.

"파일객체 = open(파일이름)"
과 동일

With에 의해 코드 블록을 벗어
날 때 자동 close()

❖ 예제

```
with open('test.txt', 'r') as file:  
    str = file.read()  
    print(str)  
#file.close()
```

파일 입출력

❖ 파일 예외 처리

```
try:
    with open('test.txt', 'r') as file:
        text = file.read()
        print(text)
except Exception as e:
    print(e)
```

파일 관리

❖ 파일 관리 함수

- `shutil.copy(a, b)`
- `shutil.move(a, b)`
- `shutil.rmtree(path)`
- `shutil.chown(f, u, g)`

- `os.rename(a, b)`
- `os.remove(f)`
- `os.chmod(f, m)`
- `os.link(a, b)`
- `os.symlink(a, b)`

파일 관리

❖ 파일 관리 함수

```
import shutil
```

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

파일 관리

❖ 디렉토리 관리함수

- `os.chdir(d)`
- `os.mkdir(d)`
- `os.rmdir(d)`
- `os.getcwd()`

- `os.listdir(d)`
- `glob.glob(pattern)`

- `os.path.isabs(f)`
- `os.path.abspath(f)`
- `os.path.realpath(f)`
- `os.path.exists(f)`
- `os.path.isfile(f)`
- `os.path.isdir(f)`

파일 관리

❖ 디렉토리 관리 함수

```
import os

files = os.listdir('/home/gusu/work')
for f in files:
    print(f)
```

```
data_backup
out
.ipynb_checkpoints
docker-storage
:
```

파일 관리

❖ 디렉토리 관리 함수

```
import os

def dumpdir(path):
    files = os.listdir(path)
    for f in files:
        fullpath = os.path.join(path, f)
        if os.path.isdir(fullpath):
            print("[%s]"%fullpath)
            dumpdir(fullpath)
        else:
            print("\t" + f)

dumpdir("/home/gusu/work/consolas")
```

데이터베이스

❖ SQLite

- 파일기반의 임베디드 데이터베이스
 - 독립적인 DBMS 서버 대신 라이브러리 형태로 프로그램에 포함되어 운영

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

여기서 SQL 명령을 실행

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

데이터베이스

❖ 테이블 생성

- 테이블 : 데이터베이스에 정보 저장 단위

name	phone	addr
김상형	123-4567	오산
한경은	555-1004	수원
한주완	444-1092	대전

데이터베이스

❖ 테이블 생성

```
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
)
""")
```

데이터베이스

❖ 데이터 삽입

```
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()
```

name	phone	addr
김상형	123-4567	오산
한경은	555-1004	수원
한주완	444-1092	대전

데이터베이스

❖ 테이블 조회

```
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

데이터베이스

❖ 테이블 조회

```
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()
```

이름: 한경은, 전화: 555-1004, 주소: 수원
이름: 한주완, 전화: 444-1092, 주소: 대전

데이터베이스

❖ 테이블 조회

```
import sqlite3

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

cursor.execute("SELECT * FROM tblAddr ORDER BY addr")

table = cursor.fetchall()
for record in table:
    print(record)
    print("이름: %s, 전화: %s, 주소: %s"%record)

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

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

데이터베이스

❖ 테이블 조회

```
import sqlite3

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

cursor.execute("SELECT addr FROM tblAddr WHERE name = '김상형'")

record = cursor.fetchone()

if record : print("김상형은 %s에 살고 있습니다."%record)
else : print("김상형은 없는 이름입니다.")

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

김상형은 오산에 살고 있습니다.

데이터베이스

■ 수정 및 삭제

```
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()
```

데이터베이스

■ 수정 및 삭제

```
import sqlite3

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

cursor.execute("DELETE FROM tblAddr WHERE name = '김상형'")
con.commit()

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