

▼ 파이썬 창에서 DB 언어 사용하기

- CREATE, INSERT 등은 SQL과 같은 DB에서 사용되는 언어
- %mkdir(=make directory) : 폴더 생성
- %ls : 폴더 내용 확인
- cd : 작업폴더 이동 (위치 이동)

```
%mkdir database
```

```
%ls # 작업폴더 파일(폴더포함) 확인
```

```
import sqlite3
```

```
db_name = './database/my_books.db'
```

```
def create_table(db_name, db_sql):  
    """  
    데이터베이스 테이블을 생성하는 함수  
    Args:  
        db_name : Database Name  
        db_sql : Query for creating Table  
    Returns :  
        is_success : Boolean  
    """  
    is_success = True  
  
    # 따옴표(""" """)에 넣는 문장은 내부주석이 됨  
  
    try :  
        # 데이터베이스 커백션 생성  
        conn = sqlite3.connect(db_name)  
  
        # 커서 확보  
        cur = conn.cursor()  
  
        # 테이블 생성  
        cur.execute(db_sql)  
  
    except:  
        is_success = False  
        print("Database Error!")  
  
    finally :  
        if is_success:  
            # 데이터베이스 반영  
            conn.commit()  
        else:  
            # 데이터베이스 철회  
            conn.rollback()  
  
        # 데이터베이스 커백션 닫기  
        conn.close()  
  
    return is_success
```

```
%rmdir database
```

```
import os  
from os.path import exists
```

```

dir_name = 'database'

if not exists(dir_name) :
    os.mkdir(dir_name)
    print('[%s] 디렉터리를 생성했습니다.' %dir_name)
else:
    print('[%s]은(는) 이미 존재합니다.' %dir_name)

```



[database] 디렉터리를 생성했습니다.

```

# Database 파일 삭제
#db_name = './database/my_books.db'
#os.remove(db_name)

```

```

db_sql = '''
CREATE TABLE my_books (
    title text,
    published_date text,
    publisher text,
    pages integer,
    recommendation integer
)
'''

if create_table(db_name, db_sql):
    print('테이블이 성공적으로 생성되었습니다.')
else :
    print('테이블이 생성되지 않았습니다')

```



테이블이 성공적으로 생성되었습니다.

```

import sqlite3

# 데이터 입력 함수
def insert_books(db_name):
    """
    데이터베이스 테이블에 데이터를 등록하는 함수
    Args:
        db_name : Database Name
    Returns :
        is_success : Boolean
    """
    is_success = True

    try:
        # 데이터베이스 커넥션 생성
        conn = sqlite3.connect(db_name)

        # 커서 확보
        cur = conn.cursor()

        # 데이터 입력 SQL1
        db_sql = "INSERT INTO my_books VALUES ('마케팅트랜드', '2002.03.02', 'A', 200, 0)"
        cur.execute(db_sql)

        # 데이터 입력 SQL2
        db_sql = 'INSERT INTO my_books VALUES (?, ?, ?, ?, ?)'
        cur.execute(db_sql, ('주식회사 4.0', '2016.07.09', 'B', 584, 1))

        # # 데이터 입력 SQL3
        books = [
            ('스타트업', '2011.07.15', 'A', 248, 1),
            ('데이터 마케팅', '2012.08.25', 'A', 296, 1),
            ('인터넷 쇼핑', '2013.08.22', 'B', 526, 0)
        ]
        cur.executemany(db_sql, books)

    except:
        is_success = False
        print("Database Error!")

    finally :
        if is_success:

```

```

        # 데이터베이스 반영
        conn.commit()
    else:
        # 데이터베이스 철회
        conn.rollback()

    # 데이터베이스 커백션 닫기
    # print('Finish process of function.')
    conn.close()

return is_success

```

```

if insert_books(db_name):
    print('데이터가 성공적으로 등록되었습니다.')
else:
    print('데이터가 등록되지 않았습니다')

```



데이터가 성공적으로 등록되었습니다.

```
%cd Anaconda_src
```



C:\Users\W202-017\Anaconda_src

```
%cd database
```



C:\Users\W202-017\Anaconda_src\database

```
%ls
```



C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: AC84-B502

C:\Users\W202-017\Anaconda_src\database 디렉터리

```

2019-09-02 오전 09:56 <DIR>      .
2019-09-02 오전 09:56 <DIR>      ..
2019-09-02 오전 09:56            8,192 my_books.db
                1개 파일            8,192 바이트
                2개 디렉터리  74,332,200,960 바이트 남음

```

```
%cd ..
```



C:\Users\W202-017\Anaconda_src

