

이번 시간에서는 지금까지 배운 내용을 바탕으로 DB 활용하는 예제를 만들어보겠습니다.

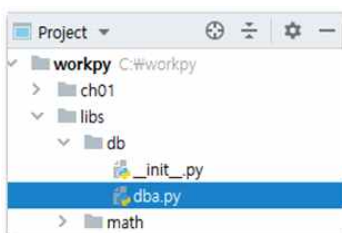
1. 전화번호부 예제

테이블을 생성한 뒤 원하는 테이블에 데이터를 입력, 조회, 삭제, 수정하는 예제입니다.

```
C:\workpy\venv\Scripts\python.exe C:/workpy/ch01/p20.py
a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:a
a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:1
이름?홍길동
전화번호?010-1212-9999
a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:2
*****데이터출력*****
홍길동 010-1212-9999
a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:0
프로그램을 종료합니다.
```

2. 모듈 만들기

libs 디렉터리 밑에 db 패키지를 만들고, db 패키지 안에 dba.py 파일을 만듭니다.



```
import sqlite3

def getConn(dbpath):
    conn = sqlite3.connect(dbpath)

    return conn
```

경로를 매겨 변수로 받는다.
db 접속과 관련된 정보를 가진
커넥션을 생성해서 리턴

dba.py

```
import sqlite3
```

```
def getConn(dbpath):
```

```
    conn = sqlite3.connect(dbpath) # db 접속과 관련 된 정보를 가진 커넥션 객체생성
    return conn
```

3. main

ch01 디렉터리 안에 p20.py 파일을 만듭니다. 예제가 길어서 전체 소스를 먼저 작성하고 설명을 하겠습니다.

p20.py

```
import sqlite3
from libs.db.dba import getConn

class Db:
    # 클래스 변수
    conn = getConn('c:/test/phone.db') # 커넥션을 가지고 옵니다.
    cur = conn.cursor()                # 커넥션을 이용해서 커서를 생성합니다.

    def createtable(self):                # 테이블 생성
        Db.cur.execute('''
        create table tell(name text, no text)
        ''')

    def insert(self):
        self.name = input('이름?')
        self.no = input('전화번호?')
        self.sql = ''
        insert into tell values(?,?)
        ''
        Db.cur.execute(self.sql,
                        (self.name, self.no))

    def select(self):
        self.sql = 'select * from tell'
        Db.cur.execute(self.sql)
        rs = Db.cur.fetchall()
        print('*****데이터출력*****')

        for k, v in rs:
            print(k, v)

    def delete(self):
        self.name = input('이름?')
        self.sql = 'delete from tell where name=?'
        Db.cur.execute(self.sql, (self.name,))

    def update(self):
        self.name = input('이름?')
        self.no = input('번호?')
        self.sql = 'update tell set no=? where name=?'
        Db.cur.execute(self.sql, (self.no, self.name))
```

```

def main():
    d = Db() # 클래스 객체 생성

    while True:
        n = input('a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:')

        if n=='a':
            d.createtable()
        if n=='1':
            d.insert()
        if n=='2':
            d.select()
        if n=='3':
            d.delete()
        if n=='4':
            d.update()
        if n=='0':
            d.conn.commit()
            d.conn.close()
            break
    print('프로그램을 종료합니다.')

if __name__=='__main__':
    main()

```

클래스 Db를 만들고, 클래스 변수로 커넥션과 커서를 생성합니다.

```

import sqlite3
from libs.db.dba import getConn

```

```

class Db:
    # 클래스 변수
    conn = getConn('c:/test/phone.db')
    cur = conn.cursor()

    def createtable(self):
        Db.cur.execute(''
            create table tell
            (name text, no text)
            '')

```

→ 'sqlite3' 'dba' 모듈을 import

'] 'dba' 모듈의 getConn()을 호출해서 커넥션을 가져온다.
커넥션을 이용해서 커서를 생성한다.

'] 테이블을 생성한다.
필드는 이름과 전화번호.

insert 메서드는 이름과 번호를 입력받아서 테이블에 삽입합니다.

```
def insert(self):
    self.name=input('이름')
    self.no=input('번호')
    self.sql=''
    insert into tell values (?,?)'''
    Db.cur.execute(self.sql,
                    (self.name,self.no))
```

이름과 전화번호를 입력 받아
insert 쿼리를 통해서 테이블에 데이터를 넣는다.

쿼리의 의미는 다음과 같습니다.

예) 이름은 홍길동, 전화는 1111-1111을 입력 받았을 때
tell 테이블에 name에는 홍길동, no 에는 1111-1111을 넣어라.
`insert into tell values('홍길동', '1111-1111')`

C:\workpy\venv\Scripts\python.exe C:/workpy/ch01/p20.py

a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:1

이름? **홍길동**

전화번호? **1111-1111**

a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:0

프로그램을 종료합니다.

테이블(T): tell	
name	no
필터	필터
1 홍길동	1111-1111

select 메서드는 데이터조회입니다.

```
def select(self):
    self.sql='select * from tell'

    Db.cur.execute(self.sql)
    rs = Db.cur.fetchall()

    print('*****데이터출력*****')

    for k, v in rs:
        print(k, v)
```

Tell 테이블의 데이터를 모두 가져온다.
결과는 리스트의 형태로 rs에 반환됨.

rs에 있는 데이터를 출력

C:\workpy\venv\Scripts\python.exe C:/workpy/ch01/p20.py

a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:2

*****데이터출력*****

홍길동 1111-1111

김철수 22222-22222

a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:0

프로그램을 종료합니다.

delete 메서드는 데이터 삭제입니다.

```
def delete(self):
    self.name=input('이름')

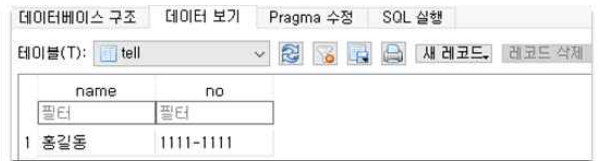
    self.sql=''
    delete from tell where name=?
    '''

    Db.cur.execute(self.sql,
                    (self.name,) )
```

삭제할 이름을 입력 받고, 쿼리를 작성.

쿼리를 실행한다.
execute() 메서드의 첫 번째 인자는 쿼리,
두 번째 인자는 튜플 이다.

```
C:\workpy\venv\Scripts\python.exe C:/workpy/ch01/p20.py
a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:3
이름? 김철수
a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:2
*****데이터출력*****
홍길동 1111-1111
a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:0
프로그램을 종료합니다.
```



name	no
홍길동	1111-1111

update 메서드는 데이터 수정입니다.

```
def update(self):
    self.name=input('이름')
    self.no=input('번호')

    self.sql=''
    update tell set no=? where name=?
    '''

    Db.cur.execute(self.sql,
                    (self.no,self.name) )
```

입력 받은 이름에 해당하는 번호를 수정한다.
이름과 번호를 입력 받는다.

update 쿼리를 작성한다.

쿼리를 실행한다.
execute() 메서드의 첫 번째 인자는 쿼리,
두 번째 인자는 튜플 이다.

```
C:\workpy\venv\Scripts\python.exe C:/workpy/ch01/p20.py
a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:4
이름? 홍길동
번호? 9999-1111
a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:2
*****데이터출력*****
홍길동 9999-1111
a.테이블생성 1.입력 2.조회 3.삭제 4.수정 0.종료:0
프로그램을 종료합니다.
```

main() 함수입니다. main에서 클래스의 객체를 생성합니다.

무한 반복을 하면서 메뉴를 입력받아 해당 메뉴에 맞는 메서드를 실행합니다.

```
def main():
    d = Db()      # 클래스 객체 생성

    while True:
        n = input('a.테이블생성 1.입력 2.조회 3.삭제 4. 수정 0.종료:')

        if n == 'a':
            d.createtable()      # 테이블 생성
        if n == '1':
            d.insert()           # 데이터 삽입
        if n == '2':
            d.select()           # 데이터 조회
        if n == '3':
            d.delete()           # 데이터 삭제
        if n == '4':
            d.update()           # 데이터 수정
        if n == '0':
            d.conn.commit()      # 데이터베이스에 반영
            d.conn.close()      # 커넥션 닫기
            break               # 반복을 멈춤

    print('프로그램을 종료합니다.')
```

메뉴를 입력 받는다.

해당하는 메서드 호출

```
if __name__ == '__main__':
    main()
```

이 파일이 main으로서 동작할 때만 if 안의 코드 실행