

파이썬 기초

12 파일 다루기



홍필두 교수
(파이썬기초)



학습내용

- 01 파일
- 02 데이터베이스 기초
- 03 데이터베이스 다루기





학습목표

- 파일에 대하여 이해하고 사용할 수 있다.
- 데이터베이스에 기본적 사항에 대하여 이해하고 사용할 수 있다.
- 데이터베이스를 파이썬 프로그램 내에서 사용할 수 있다



사전 학습

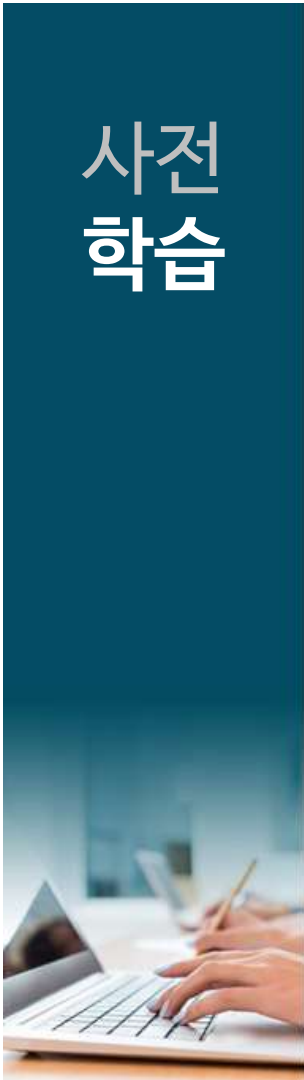
“표준 모듈 활용”

파일은 정보를 저장하는 기본 단위이며, 문서나 이미지, 실행 파일 등 모두 파일의 형태로 저장되어 있습니다.

또 데이터베이스는 자료를 효율적으로 보관하고 제어 사용하는데 매우 편리하고 다양한 기능을 제공합니다. 데이터베이스는 대용량의 데이터도 쉽게 다룰 수 있으며, 직관적인 데이터베이스 언어를 통하여 데이터 관리를 쉽게 처리할 수 있도록 해줍니다.

이번 강의에서는 이러한 파일과 데이터베이스를 사용하는 방법에 관한 몇 가지 사항을 배웁니다.

여러분은 먼저 해당 내용에 대하여 검색을 통하여 미리 알아보고 조사해 본 후, 본 강의를 듣도록 합시다.





01

파일

- 1) 파일 다루기
- 2) 파일 쓰기
- 3) 파일 읽기

1) 파일 다루기

파일

정보를 저장하는 기본 단위

└ 문서나 이미지, 실행 파일 등 모두 파일의 형태로 저장

1) 파일 다루기

파일 처리 순서

01 파일을 연다(Open)

- 옵션을 줄 수 있음
 - 새로 생성, 있으면 안 열기,
기존 파일에 추가 작성(Append), 읽기
권한으로만 열기, 읽기쓰기 권한으로 열기

02 파일을 읽는다/쓴다(Read/Write)

03 파일을 닫는다(Close)

1) 파일 다루기

파일 처리 순서

01 파일을 연다(Open)

02 파일을 읽는다/쓴다(Read/Write)

- 이진파일 (이미지, 실행파일)
 - 1개씩(char a) 읽어서 어느 정도 크기 만큼(buffer) 모아서 작업
- 문서파일
 - 개행문자 단위(한 줄)씩 읽어서 작업

03 파일을 닫는다(Close)

1) 파일 다루기

파일 처리 순서

01 파일을 연다(Open)

02 파일을 읽는다/쓴다(Read/Write)

03 파일을 닫는다(Close)

- 더 이상 할 작업이 없는 경우 반드시 닫아야 함
- 열어 두면 다른 프로그램(프로세서)가 작업할 수 없음

1) 파일 다루기

파일을 Open하는 경우, 옵션 지정 방법

예

- `open("abc.txt", "wt")` : abc.txt라는 텍스트파일을 쓰기모드로 열어라
- `open("abc.exe", "rb")` : abc.exe라는 이진파일을 읽기모드로 열어라
- `open("abc.exe", "rwb")` : abc.exe라는 이진파일을 읽고 쓰기모드로 열어라

모드	내용
r	읽기모드로 파일 열기
w	쓰기모드로 파일 열기, 이미 파일이 있다면 덮어씀
a	쓰기모드로 파일 열기, 이미 파일이 있다면 뒤에 이어서 씀
x	쓰기모드로 파일 열기, 이미 파일이 있다면 에러를 발생



모드	내용
t	텍스트 파일
b	바이너리(이진)파일

1) 파일 다루기



파일을 읽고 쓰는 프로그램은
파일의 제어를 시스템 운영체제(윈도우 등)에 의존함



발생되는 에러에 대하여
예외 에러 처리(try / except / finally)를 하여야 함



시스템에서 에러가 발생하면
except(에러 종류) 블록으로 점프하여 처리됨



에러 발생 및 정상실행 후
맨 마지막에 finally 블록을 실행

1) 파일 다루기

```
f=0
try:
    f=open("LICENSE.txt","rt")
    while True:
        r=f.readline()
        if not r:
            break
        print(r)
except FileNotFoundError:
    print("파일없음")
finally:
    if(f): f.close()
    print("프로그램 끝")
```

2) 파일 쓰기

파일을 쓰는 경우 처리 순서

01 파일 열기

02 하나의 문자열을 원하는 모양으로 만듦

03 해당 문자열 한 줄을 파일에 쓰기

04 쓰기 완료

05 파일 닫기



2) 파일 쓰기

 100명의 국어, 영어, 수학 점수를 임의로 만든 후
csv형태의 파일의 기록하는 예제

```
f=open("score.txt", "wt")
for i in range(100):
    # 아래와 같은 형태를 만들기 위하여 str.format 함수를 사용할 수 있다.
    # one_person="{0:s},{1:d},{2:d},{3:d}\n"
    #.format("홍길동"+str(i),r.randrange(0,100),r.randrange(0,100),r.randrange(0,100))

    one_person="홍길동"+str(i)+","+"+str(r.randrange(0,100))+","+"+str(r.randrange(0,100))
    +","+"+str(r.randrange(0,100)))+"\n"
    f.write(one_person) #한 줄씩 쓴다
    print(one_person)
f.close()
```

3) 파일 읽기

파일을 읽는 경우 처리 순서



3) 파일 읽기

앞서 만든 csv파일을 한 줄씩 읽어서 처리하는 예제

```
f=0
f=open("score.txt","rt")
while True:
    one_person=f.readline() #한 줄씩 읽는다
    if not one_person: break #읽을 내용이 없으면 루프탈출
    one_person=one_person.replace("\n","") #줄 바꿈문자 없애기
    item=one_person.split(",") #콤마 구분자로 찢르기
    print("="*10)
    print("이름: ", item[0])
    print("국어: ", item[1])
    print("영어: ", item[2])
    print("수학: ", item[3])
    print("="*10)
f.close()
```

```
>>>
RESTART: ***.py
---
=====
이름:   홍길동99
국어:   36
영어:   37
수학:   0
=====
```


02

데이터베이스 기초

- 1) 데이터베이스의 정의
- 2) Table, Record, Field
- 3) 데이터베이스 (DB)와 데이터베이스 시스템 (DBMS)
- 4) 데이터베이스를 다루는 언어
- 5) 데이터베이스 처리 실습

1) 데이터베이스의 정의

데이터베이스

- 관련 있는 데이터의 저장소
- 여러 사람이나 응용시스템에 의해 참조 가능하도록 서로 논리적으로 연관되어 통합 관리되는 데이터의 모임

→ 통합된(Integrated) 관련(Related) 있는
데이터 중복(Redundancy)을 최소화하여 보조기억장치에 저장함

1) 데이터베이스의 정의

데이터베이스의 예

- ✓ 수업 사이트 내 자유게시판에 글 모음
- ✓ 엑셀로 작성한 수강생 리스트
- ✓ 아래아 한글로 작성한 주소록
- ✓ 통계분석을 위한 10년치 주가 데이터
- ✓ 회원 가입 시 받아오는 회원가입정보
- ✓ 극장의 좌석을 예약하기 위한 티켓 발급 정보

1) 데이터베이스의 정의

데이터베이스와 테이블

- 3학년 10반 중간고사 성적 테이블 :
- 3학년 학교활동 데이터베이스 : 출석부테이블, 성적테이블, 학생부테이블, 상벌테이블, 체력검정테이블
- 즉 데이터베이스는 여러 개의 테이블을 포함한 개념

2) Table, Record, Field

열, column, field

이름	학번	국어	영어	수학
나연	20990001	95	100	95
정연	20990002	95	95	95
모모	20990003	100	100	100
사나	20990004	100	95	90
지효	20990005	80	100	70
미나	20990006	100	100	70
다현	20990007	70	70	70
채영	20990008	80	90	80
쯔위	20990009	60	100	80

행, row,
record

- Record + Field
= Table < Database
< Databases
- Key , Index, unique
용어
- 우리 반에는 나연이가
두 명이고 학번은
다르다, 모모의 성적
찾기, 미나의 성적이
두 번 입력되면 안됨
➡ 이러한 문제는
Database를
활용하여 쉽게 해결

3) 데이터베이스(DB)와 데이터베이스 시스템(DBMS)

DB(Database)

DB(Database)

- 실 세계의 정보들(회원목록, 도서목록, 비디오목록 등...)
- 정보를 다루는 개념, 이론적인 사항

3) 데이터베이스(DB)와 데이터베이스 시스템(DBMS)

DBMS(Data Base Management System)

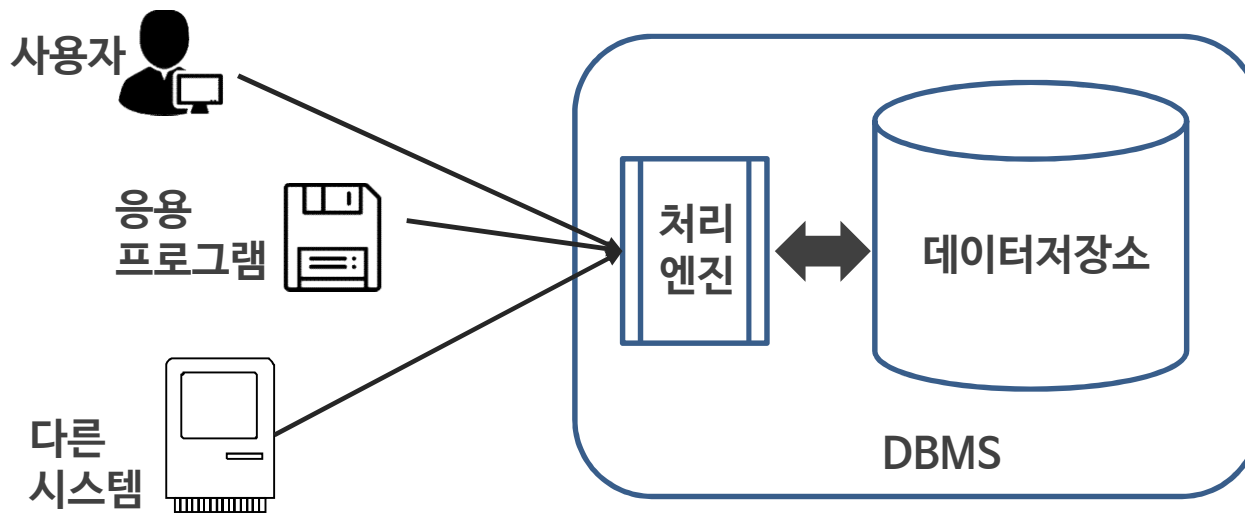
DBMS(Data Base Management System)

- Database의 소프트웨어 및 하드웨어 시스템
- 데이터베이스를 관리해 주는 시스템
- 시스템 내의 데이터베이스 개념으로 데이터를 관리할 수 있도록 구성된 소프트웨어

예 오라클, MySQL, MSSQL

3) 데이터베이스(DB)와 데이터베이스 시스템(DBMS)

 DBMS(Data Base Management System)



4) 데이터베이스를 다루는 언어

데이터베이스를 다루는 언어의 분류

구분	설명	예시
DDL (Data Definition Language)	데이터와 그 구조를 정의	<ul style="list-style-type: none">• CREATE : 데이터베이스 객체를 생성• DROP : 데이터베이스 객체를 삭제• ALTER : 기존에 존재하는 데이터베이스 객체를 다시 정의• RENAME : 데이터베이스의 컬럼 이름을 변경• TRUNCATE : 테이블을 최초 생성된 초기상태로 만들며, ROLLBACK이 불가능
DML (Data Manipulation Language)	데이터의 검색, 수정, 삭제 등을 처리	<ul style="list-style-type: none">• INSERT : 테이블에 데이터를 입력• DELETE : 테이블에 데이터를 삭제• UPDATE : 기존에 존재하는 테이블의 데이터 수정• SELECT : 테이블로부터 데이터를 검색• COMMIT : 변경사항을 저장• ROLLBACK : 변경사항을 취소

4) 데이터베이스를 다루는 언어

데이터베이스를 다루는 언어의 분류

구분	설명	예시
DCL (Data Control Language)	데이터베이스 사용자의 권한을 제어	<ul style="list-style-type: none">• GRANT : 데이터베이스 객체에 권한을 부여• REVOKE : 이미 부여된 데이터베이스 객체의 권한을 취소


5) 데이터베이스 처리 실습



데이터베이스 > 전체테이블 > 하나의 테이블을 다루는 방법 실습



가장 간단한 sqlite3를 이용하여 실습

- 데이터베이스 접속
 - 데이터베이스 생성, 삭제
 - 테이블 생성, 수정, 삭제, 검색
 - 한 테이블에 대하여 자료조회, 자료입력, 자료수정, 자료삭제
 - 기타 복잡한 사용 실습 등
- 

5) 데이터베이스 처리 실습



[DB Browser for SQLite]를 다운받아 설치



임의에 데이터베이스 (mydata.db)를 신규로 만들면
해당 파일이 Database임

5) 데이터베이스 처리 실습

 아래의 내용을 하나씩 실습함

테이블 생성 삭제(create table/drop table)

```
>>>create table examtable(₩  
    name varchar(20),₩  
    studentid int not null primary key,₩  
    kor    int,₩  
    eng    int,₩  
    mat    int);
```

```
>>>drop table examtable;
```

```
>>>create table examtable(₩  
    name varchar(20),₩  
    studentid int not null primary key,₩  
    kor    int,₩  
    eng    int,₩  
    mat    int);
```

5) 데이터베이스 처리 실습

아래의 내용을 하나씩 실습함

한 테이블에 대하여 자료조회, 자료입력, 자료수정, 자료삭제

```
>>>select * from examtable;  
>>> insert into examtable (name, studentid, kor, eng, mat) values ("나연", 20990001, 95, 100, 95);  
>>>select * from examtable;
```

```
>>> insert into examtable (name, studentid, kor, eng, mat) values ("정연", 20990002, 95, 95, 95);  
>>>select * from examtable;
```

```
>>> insert into examtable (name, studentid, kor, eng, mat) values ("모모", 20990003, 100, 100, 100);  
>>>select * from examtable;
```

```
>>> insert into examtable (name, studentid, kor, eng, mat) values ("사나", 20990004, 100, 95, 90);  
>>>select * from examtable;
```

```
>>> insert into examtable (name, studentid, kor, eng, mat) values ("지효", 20990005, 80, 100, 70);  
>>>select * from examtable;
```

```
>>> insert into examtable (name, studentid, kor, eng, mat) values ("미나", 20990006, 100, 100, 70);  
>>>select * from examtable;
```

```
>>> insert into examtable (name, studentid, kor, eng, mat) values ("다현", 20990007, 70, 70, 70);  
>>>select * from examtable;
```

```
>>> insert into examtable (name, studentid, kor, eng, mat) values ("채영", 20990008, 80, 90, 80);  
>>>select * from examtable;
```

```
>>> insert into examtable (name, studentid, kor, eng, mat) values ("쯔위", 20990009, 60, 100, 80);  
>>>select * from examtable;
```

- 보기

```
>select * from  
examtable;
```

- 입력

```
>insert into  
examtable(a,b,c)  
values(va,vb,vc);
```

5) 데이터베이스 처리 실습

아래의 내용을 하나씩 실습함

한 테이블에 대하여 자료조회, 자료입력, 자료수정, 자료삭제

```
>>>select * from examtable where name="다현";
```

```
>>>select * from examtable where studentid > 209902;
```

```
>>>select * from examtable where mat < 95;
```

```
>>>select * from examtable where mat > 95 and eng >95;
```

```
>>>select * from examtable where mat > 95 or eng >95;
```

```
>>>update examtable set eng = 95,mat=91 where name = " 다현";
```

```
>>>update examtable set eng = 95,mat=91 ;
```

```
>>> delete from examtable where name = " 다현";
```

- where절 사용

03

데이터베이스 다루기

- 1) 파이썬에서 sqlite3 사용
- 2) 파이썬에서 sqlite3 사용 실습

1) 파이썬에서 sqlite3 사용



파이썬에서는 sqlite3 데이터베이스가 기본으로 포함되어 있음



다음과 같은 방법으로 프로그램 내에서 데이터베이스를 사용함

파이썬에서 데이터베이스를 다루는 경우 처리 순서

- | | |
|-----------|----------------|
| ① connect | ④ {fetchall} |
| ② cursor | ⑤ cursor.close |
| ③ execute | ⑥ con.close |

1) 파이썬에서 sqlite3 사용

```
con = sqlite3.connect("db파일명")  #DB연결,  
패스까지 써줌 기본은 현 소스파일의 디렉토리  
cursor = con.cursor()             #데이터베이스  
사용공간 만듦  
  
~~이부분에 sql쿼리 작성  
#cursor.execute("sql문")  
  
cursor.close()  
con.close()
```

2) 파이썬에서 sqlite3 사용 실습

테이블 생성, 삭제

```
import sqlite3 as s
con=s.connect("my01.db")
cursor = con.cursor()

cursor.execute("drop table examtable;") #table 삭제
con.commit()

cursor.execute( " \                                #table 생성
create table examtable( \
    name varchar(20),\
    studentid int not null primary key,\
    kor      int,\
    eng      int,\
    mat      int) \
    ")

con.commit() #버퍼에 처리된 내용을 기록 commit
cursor.close()
con.close()
```

2) 파이썬에서 sqlite3 사용 실습

데이터 삽입(insert)

```
import sqlite3 as s
con=s.connect("my01.db")
cursor = con.cursor()

cursor.execute("insert into examtable (name, studentid, kor, eng, mat) values (\\"나연\\", 20990001, 95, 100, 95);")
#테이블내 레코드 삽입
cursor.execute("insert into examtable (name, studentid, kor, eng, mat) values (\\"정연\\", 20990002, 90, 90, 85);")
cursor.execute("insert into examtable (name, studentid, kor, eng, mat) values (\\"모모\\", 20990003, 95, 95, 75);")
cursor.execute("insert into examtable (name, studentid, kor, eng, mat) values (\\"사나\\", 20990004, 85, 90, 90);")
cursor.execute("insert into examtable (name, studentid, kor, eng, mat) values (\\"지효\\", 20990005, 100, 90, 85);")
cursor.execute("insert into examtable (name, studentid, kor, eng, mat) values (\\"미나\\", 20990006, 90, 80, 90);")
cursor.execute("insert into examtable (name, studentid, kor, eng, mat) values (\\"다현\\", 20990007, 75, 90, 100);")
cursor.execute("insert into examtable (name, studentid, kor, eng, mat) values (\\"채영\\", 20990008, 95, 70, 90);")
cursor.execute("insert into examtable (name, studentid, kor, eng, mat) values (\\"쯔위\\", 20990009, 85, 90, 90);")

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

2) 파이썬에서 sqlite3 사용 실습

데이터 선택 (select)

select에 경우 fetch를 사용하여 데이터를 반복으로 가져옴

```
import sqlite3 as s
con=s.connect("my01.db")
cursor = con.cursor()

cursor.execute("select * from examtable")

one_rec=cursor.fetchall() #해당 데이터를 전부 가져옴

for field in one_rec: #해당 데이터를 반복문을 통해 하나씩 처리
    kor=field[2]
    eng=field[3]
    mat=field[4]
    print("이름:%s 학번:%s 국어:%3d 영어:%3d 수학:%3d 총점:%3d
평균:%3d"%(field[0],field[1],kor,eng,mat,kor+eng+mat,int((kor+eng+mat)/3)))

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

```
>>>
RESTART: ***.py
이름:나연 학번:20990001 국어: 95 영어:100 수학: 95 총점:290 평균: 96
이름:정연 학번:20990002 국어: 90 영어: 90 수학: 85 총점:265 평균: 88
이름:모모 학번:20990003 국어: 95 영어: 95 수학: 75 총점:265 평균: 88
이름:사나 학번:20990004 국어: 85 영어: 90 수학: 90 총점:265 평균: 88
이름:지호 학번:20990005 국어:100 영어: 90 수학: 85 총점:275 평균: 91
이름:미나 학번:20990006 국어: 90 영어: 80 수학: 90 총점:260 평균: 86
이름:다현 학번:20990007 국어: 75 영어: 90 수학:100 총점:265 평균: 88
이름:채영 학번:20990008 국어: 95 영어: 70 수학: 90 총점:255 평균: 85
이름:쓰위 학번:20990009 국어: 85 영어: 90 수학: 90 총점:265 평균: 88
```

04

실습하기 I

- 1) 파일 - 파일 다루기
- 2) 파일 - 파일 읽기, 파일 쓰기
- 3) 데이터베이스 기초 - 도구설치, 테이블 생성, 삭제

실습내용

- 1) 파일 - 파일 다루기
- 2) 파일 - 파일 읽기, 파일 쓰기
- 3) 데이터베이스 기초 - 도구설치, 테이블 생성, 삭제

05

실습하기 II

- 1) 데이터베이스 기초 - DB에서 자료조회, 입력, 수정, 삭제
- 2) 데이터베이스 다루기 - 파이썬에서 sqlite3 실습 1
- 3) 데이터베이스 다루기 - 파이썬에서 sqlite3 실습 2

실습내용

- 1) 데이터베이스 기초 - DB에서 자료조회, 입력, 수정, 삭제
- 2) 데이터베이스 다루기 - 파이썬에서 sqlite3 실습 1
- 3) 데이터베이스 다루기 - 파이썬에서 sqlite3 실습 2



학습활동

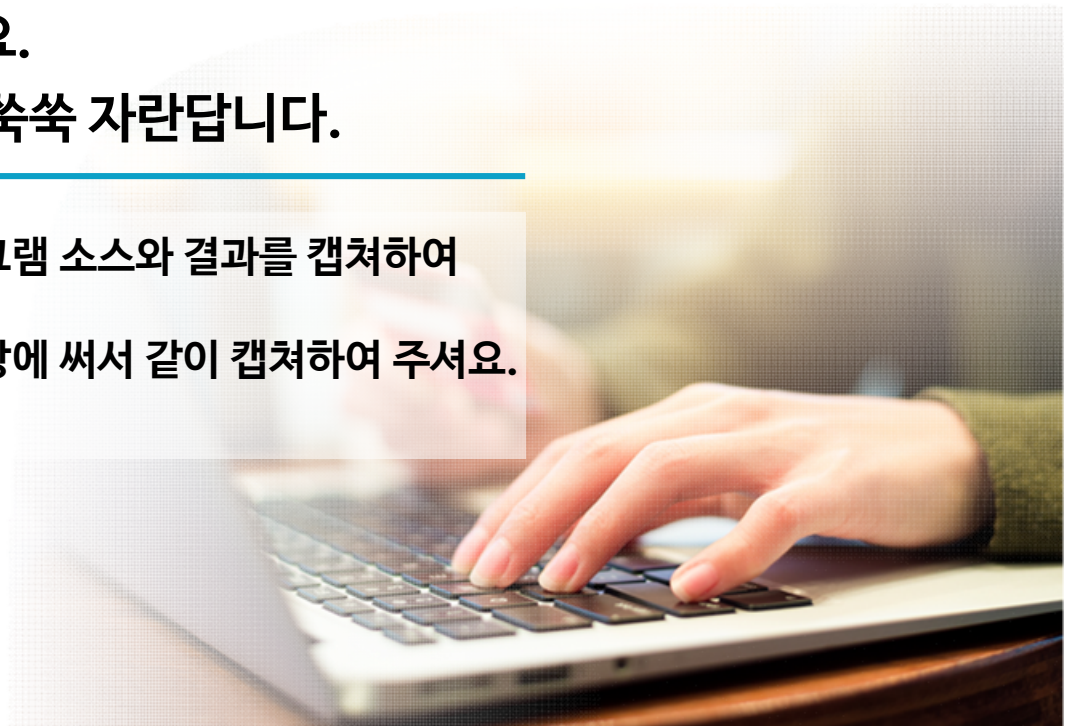
일시정지 버튼을 누른 후, 아래의 학습활동에 참여하세요.

Q

오늘 배운 내용을 스스로 실습하여
자유게시판에 올려 주세요.

이렇게 정리하면 실력이 쑥쑥 자란답니다.

- ① 본인이 실습한 내용을 프로그램 소스와 결과를 캡처하여 올려주세요.
- ② 본인의 학번과 이름을 메모장에 써서 같이 캡처하여 주세요.
- ③ 그리고 설명도 달아 주세요.





학습활동에 대한 교수님 의견

Q

오늘 배운 내용을 스스로 실습하여 자유게시판에 올려 주세요.
이렇게 정리하면 실력이 쑥쑥 자란답니다.

A

[오늘 학습한 내용의 실습 사항]

- ① 파일 - 파일 다루기
- ② 파일 - 파일 읽기, 파일 쓰기
- ③ 데이터베이스 기초 - 도구설치, 테이블 생성, 삭제
- ④ 데이터베이스 기초 - DB에서 자료조회, 입력, 수정, 삭제
- ⑤ 데이터베이스 다루기 - 파이썬에서 sqlite3 실습 1
- ⑥ 데이터베이스 다루기 - 파이썬에서 sqlite3 실습 2

학습 평가

Q1

Q2

Q3

Q4

Q5

Q1

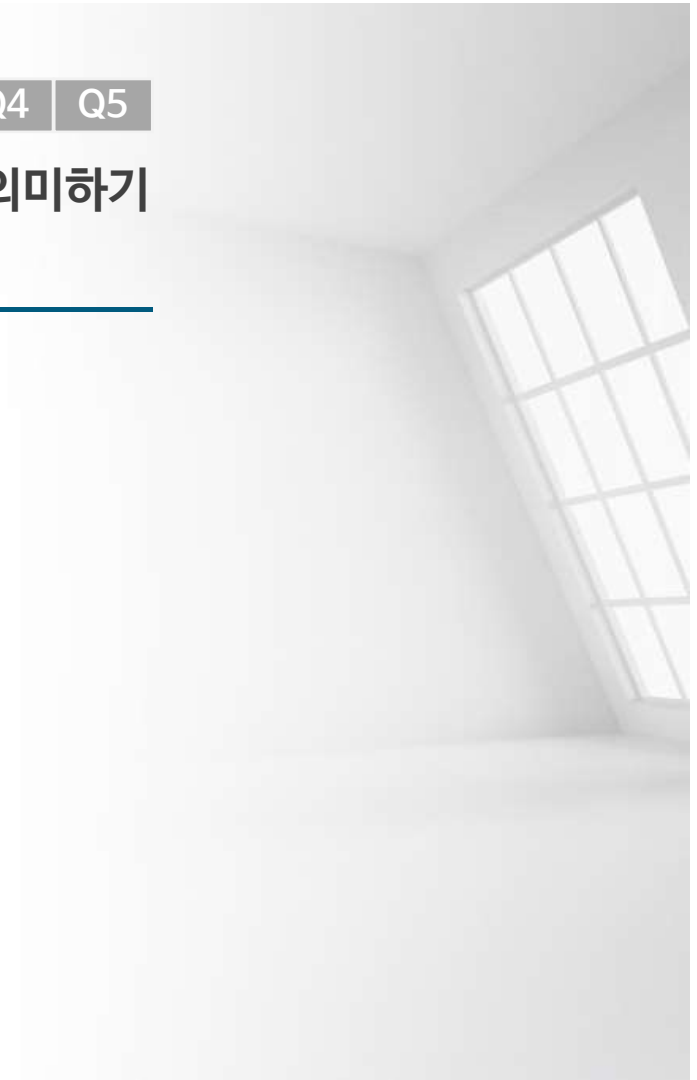
다음 중 이진파일을 open할 경우를 의미하기 위하여 옵션으로 주는 문자는?

1 a

2 t

3 b

4 2



학습 평가

Q1

Q2

Q3

Q4

Q5

Q1

다음 중 이진파일을 open할 경우를 의미하기 위하여 옵션으로 주는 문자는?

1 a

2 t

☒ b

4 2

정답

3번

해설

바이너리(이진)파일을 의미하는 것은 b입니다.

학습 평가

Q1

Q2

Q3

Q4

Q5

Q2

다음 중 파일처리의 실행 에러를 처리하기 위하여 사용되지 **않은** 문장은?

1 try:

2 except FileNotFoundError:

3 catch:

4 finally:



학습 평가

Q1

Q2

Q3

Q4

Q5

Q2

다음 중 파일처리의 실행 에러를 처리하기 위하여 사용되지 **않은** 문장은?

1 try:

2 except FileNotFoundError:

☒ 3 catch:

4 finally:

정답

3번

해설

파일 처리시 발생하는 에러에 대하여 예외 에러 처리 (try / except / finally)를 하여야 합니다.

학습 평가

Q1

Q2

Q3

Q4

Q5

Q3

다음 데이터베이스를 다루는 언어 중
그 성격이 다른 언어는 ?

- 1 INSERT
- 2 GRANT
- 3 COMMIT
- 4 UPDATE



학습 평가

Q1

Q2

Q3

Q4

Q5

Q3

다음 데이터베이스를 다루는 언어 중
그 성격이 다른 언어는 ?

1

INSERT



GRANT

3

COMMIT

4

UPDATE

정답

2번

해설

GRANT는 DCL (Data Control Language)로
데이터베이스 사용자 권한을 제어하는 언어입니다.

학습 평가

Q1

Q2

Q3

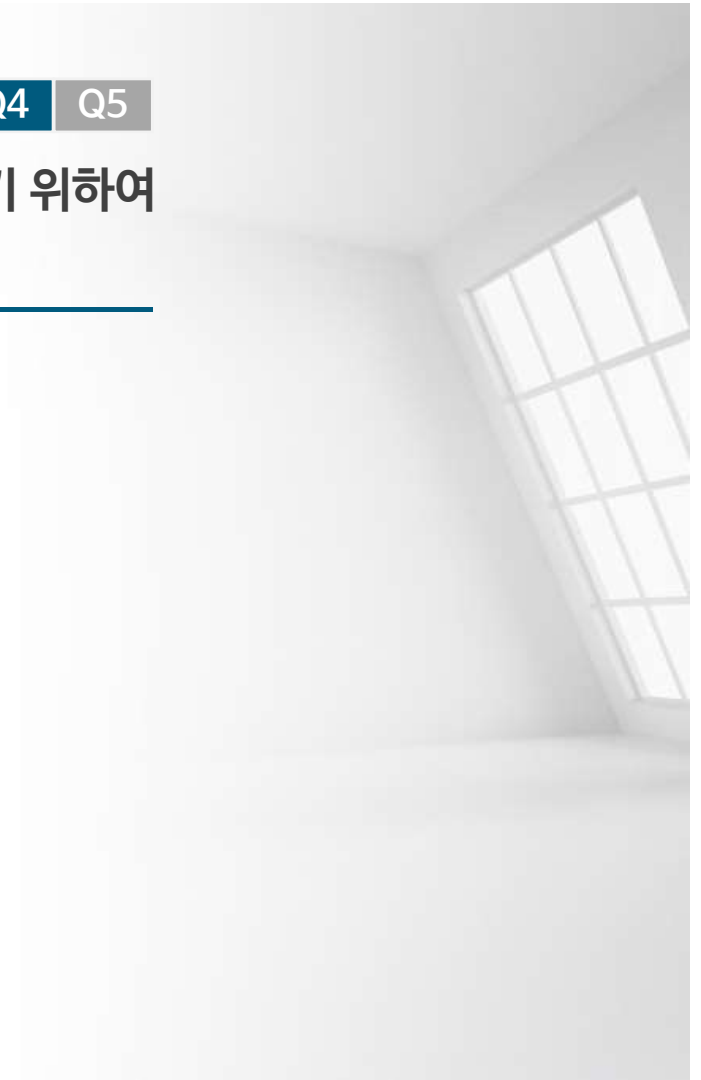
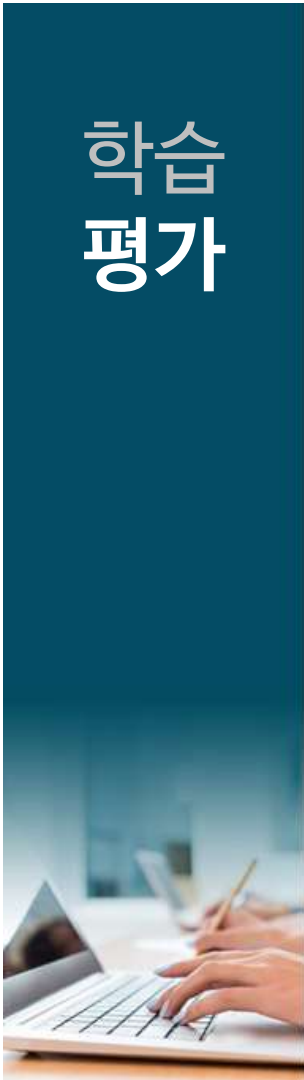
Q4

Q5

Q4

다음 중 DB에 데이터를 새로 입력하기 위하여
사용된 명령어는?

- 1 select
- 2 insert
- 3 delete
- 4 update



학습 평가

Q1

Q2

Q3

Q4

Q5

Q4

다음 중 DB에 데이터를 새로 입력하기 위하여 사용된 명령어는?

1 select

☒ 2 insert

3 delete

4 update

정답

2번

해설

insert into examtable (name, studentid, kor, eng, mat) values ("나연", 20990001, 95, 100, 95); 가 사용되었습니다.

학습 평가

Q1

Q2

Q3

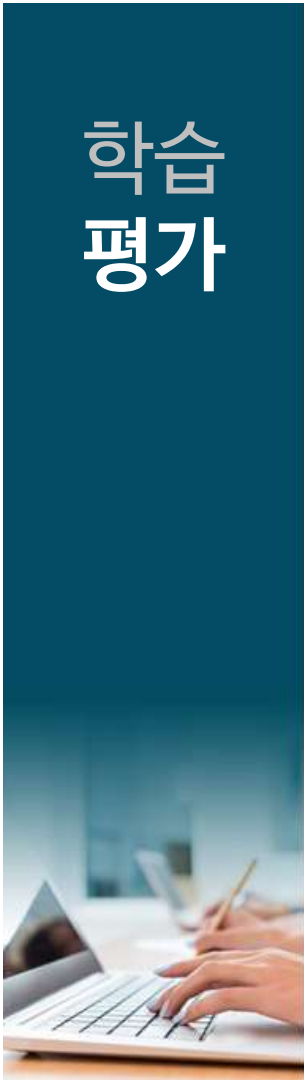
Q4

Q5

Q5

다음 파이썬에서 select 문장으로 선택된 데이터를 가지고 오기 위하여 사용된 명령어는?

- 1 `sqlite3.connect()`
- 2 `con.cursor()`
- 3 `cursor.close()`
- 4 `cursor.fetchall()`



학습 평가

Q1

Q2

Q3

Q4

Q5

Q5

다음 파이썬에서 select 문장으로 선택된 데이터를 가지고 오기 위하여 사용된 명령어는?

1 sqlite3.connect()

☒ 2 con.cursor()

3 cursor.close()

4 cursor.fetchall()

정답

2번

해설

해당 데이터를 전부 가져오기 위하여 fetchall()을 사용합니다.



정리하기

파일

- ✓ 파일은 정보를 저장하는 기본 단위이며, 문서나 이미지, 실행 파일 등 모두 파일의 형태로 저장함
- ✓ 파일을 쓰는 경우 처리 순서
 - 파일 열기 → 하나의 문자열을 원하는 모양으로 만들
→ 해당 문자열 한 줄을 파일에 쓰기 → 쓰기 완료
→ 파일 닫기
- ✓ 파일을 읽는 경우 처리 순서
 - 파일 열기 → 한 줄의 문자열 읽기 → 해당 문자열을 원하는 데로 가공 사용 → 읽기 완료 후 파일 닫기





정리하기

데이터베이스 기초

- ✓ CREATE : 데이터베이스 객체를 생성
- ✓ DROP : 데이터베이스 객체를 삭제
- ✓ INSERT : 테이블에 데이터를 입력
- ✓ DELETE : 테이블에 데이터를 삭제
- ✓ UPDATE : 기존에 존재하는 테이블의 데이터 수정
- ✓ SELECT : 테이블로부터 데이터를 검색





정리하기

데이터베이스 다루기

- ✓ 파이썬에서 데이터베이스를 다루는 경우 처리 순서
 - connect ➡ cursor ➡ execute ➡ {fetchall}
➡ cursor.close ➡ con.close

