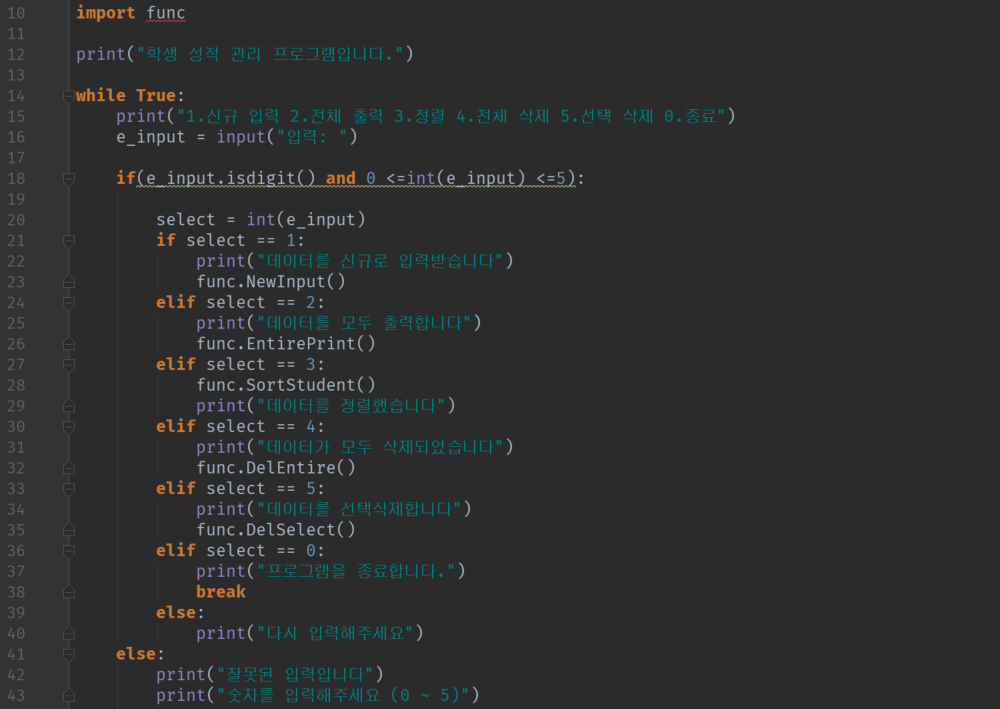
<Python – 학생성적 관리 프로그램>

1. 기능 :
2. 신규입력 : 이름, 영어성적, 수학성적을 입력받은 후 파일에 저장
3. 전체출력 : 현재까지 입력된 모든 학생들의 데이터 출력
4. 정렬 : 현재 데이터를 사전순으로 정렬
5. 전체삭제 : 현재까지 입력된 모든 데이터 삭제
6. 선택 삭제 : 선택삭제 선택 시 현재까지 입력 된 데이터를 순서와 함께 출력 후 데이터의 순서를 입력받아 삭제

\*프로그램 종료 시, 리스트의 전체를 파일로 저장

\*프로그램 실행 시, 파일에 있는 리스트를 모두 불러와서 리스트에 입력

1. 구현 : func.py 내에서 기능을 구현하는 함수 정의 (모듈 사용)
2. 신규입력 : func.py내에 NewInput()으로 정의 (함수사용)
3. 전체출력 : func.py내에 EntirePrint()으로 정의 (함수사용)
4. 정렬 : func.py내에 SortStudent()으로 정의(함수 사용)
5. 전체삭제 : func.py내에 DelEntire()으로 정의(함수 사용)
6. 선택삭제 : func.py내에 DelSelect()으로 정의(함수 사용)
7. 소스코드
8. Main



func.py내에서 정의된 함수들을 사용하기 위해서 처음에 func.py 모듈을 import 한다.

e\_input은 잘못된 입력을 검사하기 위한 변수로 string변수이다.

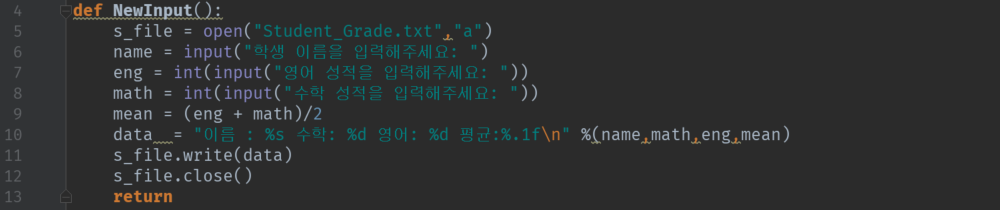
e\_input.isdigit()은 e\_input에 들어간 문자열에 숫자만 입력되면 True, 하나라도 숫자가 아닌 것이 입력되면 False를 반환시켜주는 함수이다.

Input을 받기 전에 e\_input으로 사용자가 올바른 입력을 한다.

입력에 숫자가 아닌 것이 포함되면 else문으로 들어가 ‘잘못된 입력입니다’를 출력한 뒤 다시 입력을 받게 하였다.

입력이 숫자이고 0~5 사이에 숫자라면 e\_input을 정수로 바꿔서 선택에 따른 기능을 한다.

1. func.py
2. NewInput()

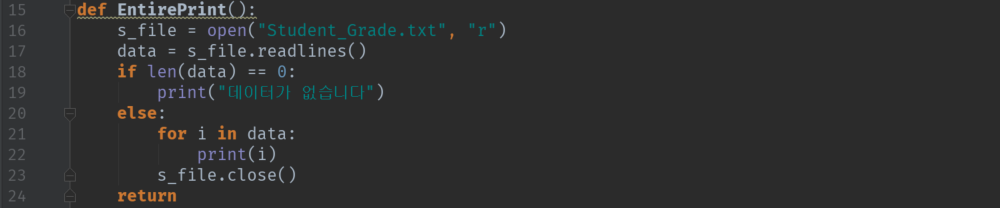


파일 객체는 모두 s\_file이라는 이름으로 생성하고 저장한다. 파일은 Student\_Grade.txt. 라는 이름의 텍스트 파일이다. 신규입력에서는 파일의 마지막에 신규로 입력받은 데이터를 저장해야 하므로 파일모드는 “a”이다.

Name은 학생의 이름을 eng는 학생의 영어성적을 math는 학생의 수학성적을 입력받는 변수이다. Mean은 현재 입력받고 있는 학생의 영어성적과 수학성적의 평균을 나타내는 변수이다.

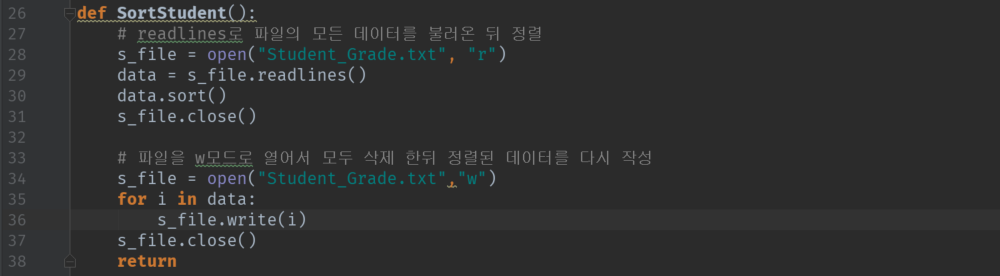
위의 4가지의 변수를 포함하는 문자열을 data라는 문자열 변수에 저장한 후 파일객체에 저장한다. 저장이 안될 수도 있기 때문에 마지막에 파일객체를 닫아준다.

1. EntirePrint()



파일객체를 읽기모드로 불러들인 후 readlines()함수를 써서 파일객체에 저장된 모든 문자열을 리스트로 반환받는다. 반환받은 리스트의 길이가 0인 경우(입력된 데이터가 아무 것도 없는 경우) “데이터가 없습니다”를 프린트 한다. 반환된 리스트에 데이터가 있다면 리스트를 전부 프린트한다.

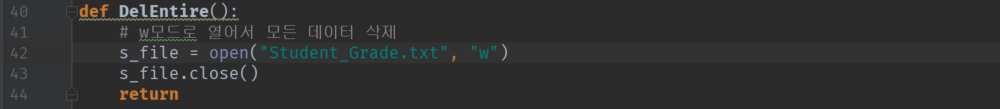
1. SortStudent()



파일객체를 읽기모드로 불러들인 후 readlines()함수를 써서 리스트로 반환받은 후 sort()함수를 써서 정렬한다. sort()한 후 다시 써야 하므로 파일객체를 닫는다.

파일객체를 다시 처음부터 써야 하므로 쓰기모드로 연 후 반환받은 리스트를 모두 쓴 뒤 파일객체를 닫는다.

1. DelEntire()



파일객체를 쓰기모드로 열 경우 데이터가 모두 삭제되므로 쓰기모드로 연 후에 파일객체를 닫아준다.

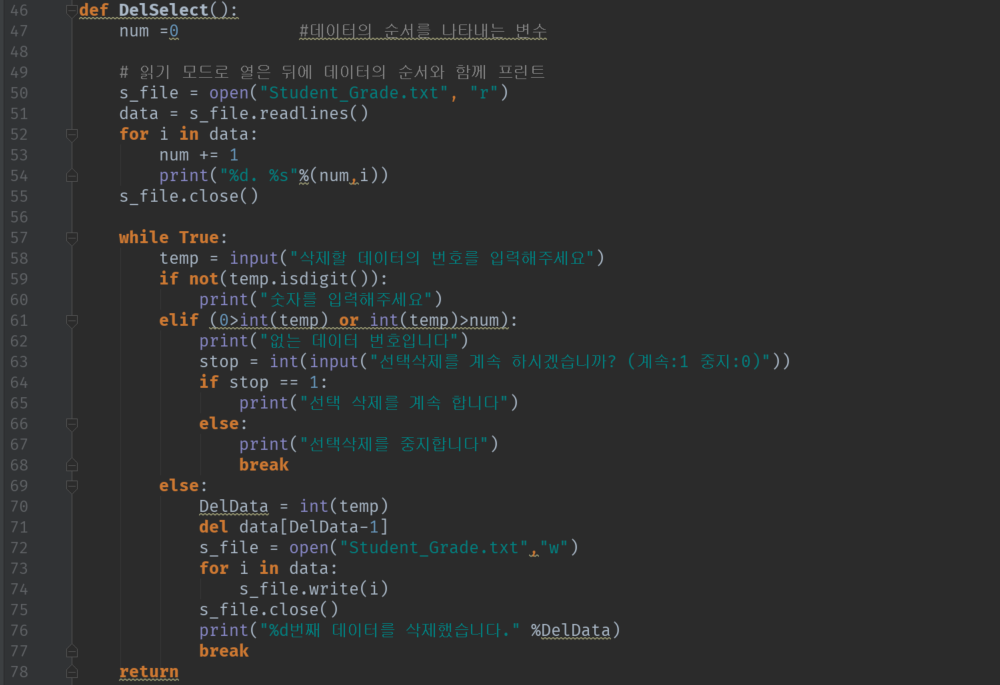
1. DelSelect()

-파일을 읽기모드 연 후 readlines()함수를 써서 리스트로 반환받는다

-받은 데이터를 순서와 함께 프린트해야 하므로 순서를 나타내는 변수 num을 선언한다.

-for문 안에서 num을 하나씩 증가시켜 데이터의 순서를 나타낸다.

-여기까지 하면 data라는 변수에 현재 데이터가 리스트로 모두 저장되어 있고 삭제할 데이터의 순서를 입력받아 해당 데이터를 삭제한 후 다시 쓰기모드로 파일객체를 열어 데이터를 입력해야 한다. 따라서 파일객체를 닫아준다.



파일 객체를 닫은 후에는 삭제할 데이터의 입력을 받는다. 선택삭제를 하고 싶지 않은 경우, 선택삭제를 나가기 위해 데이터의 순서에 없는 번호를 입력하거나 문자열을 입력한다고 가정할 수 있다. 따라서 잘못된 입력이 들어왔을 때 선택삭제를 계속할 지 여부를 묻는다. stop이라는 변수에 1이라는 값이 입력되면 선택삭제를 계속하지만 1이 아닌 경우 선택삭제를 중지한다.

사용자가 데이터의 순서에 있는 번호를 제대로 입력하는 경우, del을 써서 해당 데이터를 삭제한다. 이때 DelData는 1부터 시작하지만 data의 index는 0부터 시작하므로 삭제할 데이터의 index = DelData -1이다.

데이터를 삭제했다면 파일객체를 다시 쓰기모드로 열어서 data에 저장된 데이터를 모두 입력한 후 파일 객체를 닫는다.