

4. 파일과 배열 안의 데이터: 데이터 정렬하기

순천향대학교 컴퓨터공학과
이 상 정

프로그래밍 기초

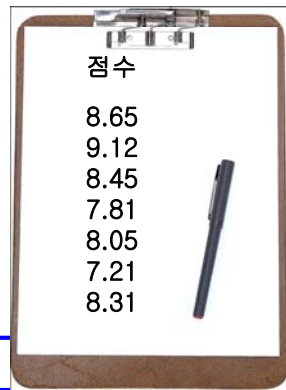
학습 내용

- 프로그램이 커지면 **데이터 처리** 필요성도 커짐
 - 데이터를 **개별 변수**에만 사용 저장할 수 없음
 - 대용량의 데이터 처리를 쉽게 해 주는 **자료구조**를 사용
- 데이터는 종종 하드디스크에 저장되어 있는 **파일**에서 가져옴
 - 파일 안의 데이터 처리 필요



서핑 대회 예

- 매년 열리는 서핑대회 서파톤(Surf-A-Thon)
- 주최 측에서 점수를 기록하는 파이썬 프로그램 요청
- 심판이 참가자들의 점수를 매기면 점수는 result.txt 파일에 저장
 - 각 참가자들의 점수는 한 라인에 기록
 - 각 라인의 점수를 읽어 서파톤 최고의 점수를 찾아내야 함



result.txt 파일



파일 데이터 접근

- 파일 데이터를 접근하기 위해 open, for, close 패턴 사용
- open() 내장 함수
 - 지정된 파일을 열고 파일 핸들(객체)를 리턴
 - 파일 작업 시 파일 이름 대신 파일 핸들을 사용
 - open() 함수 사용 예
`result_f = open("result.txt");`
- for 루프를 사용하여 파일 안에서 한 번에 한 라인씩 읽음

```

result_f = open("result.txt");           # 파일을 열음
for each_line in result_f:              # each_line 변수에 읽은 라인(스트링) 저장
    print(each_line)                    # 더 이상 읽을 라인이 없으면 루프 종료
result_f.close()                         # 파일 사용 종료 시 파일 닫음

```



서핑대회 코드

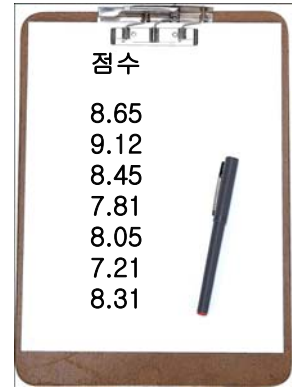
```

highest_score = 0
result_f = open("results.txt")

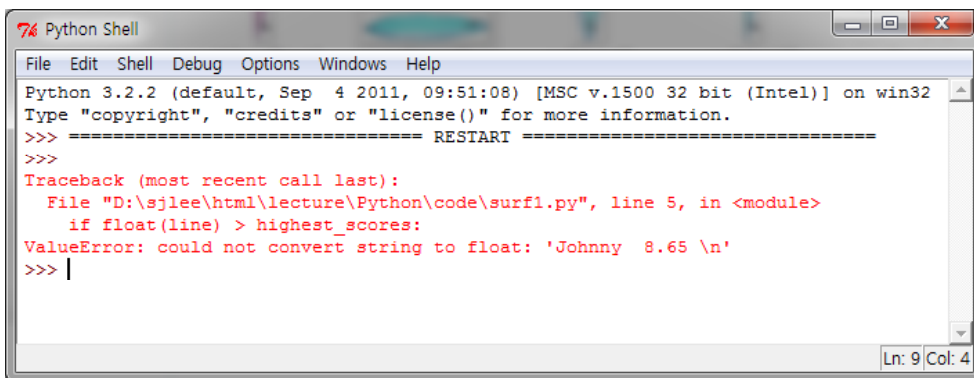
for line in result_f:
    if float(line) > highest_score:
        highest_score = float(line)

result_f.close()
print("The highest score was:")
print(highest_score)

```



시험주행



❑ 에러 원인은? ValueError ?



result.txt 파일 내용 확인

- result.txt 파일에는 점수와 함께 선수 이름도 기록

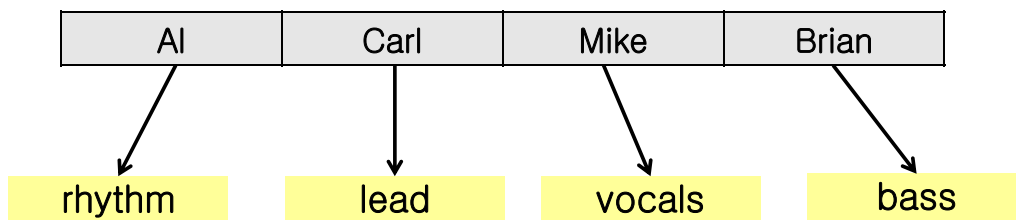
Jonny	8.65
Juan	9.12
Joseph	8.45
Stacey	7.81
Aideen	8.05
Zack	7.21
Aaron	8.31
.	.

- 파일을 읽을 때 각 라인의 데이터 분리해야 함
 - 파이썬 스트링은 split() 메소드 내장

split() 메소드

- split() 메소드는 여러 단어로 구성된 스트링을 분할
- 다중 할당(multiple assignment)을 사용하여 분리된 단어를 여러 변수에 저장
- 사용 예

```
rock_band = "Al Carl Mike Brian"
(rhythm, lead, vocals, bass) = rock_band.split()
```



서핑대회 코드- 데이터 분리 [실습 1]



```

highest_score = 0
result_f = open("results.txt")

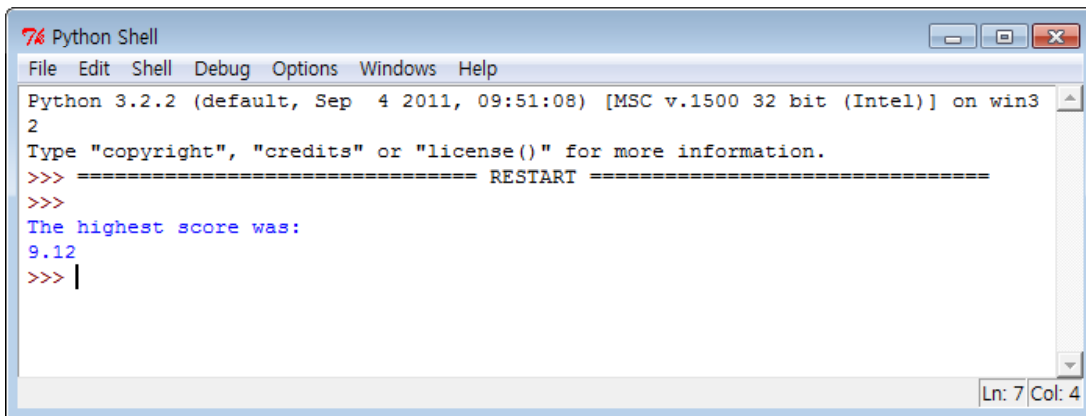
for line in result_f:
    (name, score) = line.split()
    if float(score) > highest_score:
        highest_score = float(score)

result_f.close()
print("The highest score was:")
print(highest_score)

```



시험주행





보충학습 - 파일 쓰기 [실습 2]

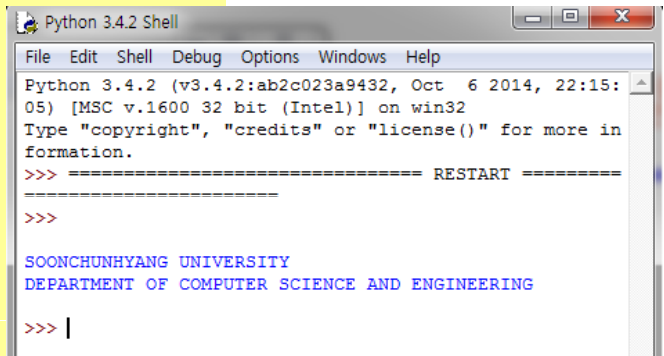
```

text = """
Soonchunhyang University
Department of Computer Science and Engineering
"""

# 파일 쓰기
f = open("test.txt", "w")
f.write(text)
f.close()

# 파일 읽기
f = open("test.txt")
lines = f.read()
print(lines.upper())

```



데이터 정렬

□ 2등과 3등도 알고 싶다면?

- 데이터가 원하는 순서대로 배치되면 쉽게 알 수 있음
=> 정렬(sorting) 이라고 함

정렬되지 않은 원시 데이터

8.65
9.12
8.45
7.81
8.05
7.21
8.31



정렬된 사본 데이터

9.12
8.65
8.45
8.31
8.05
7.81
7.21

1 등

2 등

3 등

□ 데이터 보관 2가지 방법

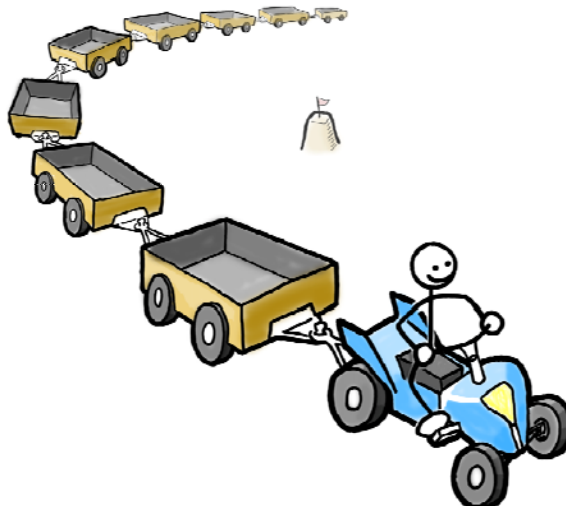
1. 데이터를 **디스크 파일**로 보관
 - 많은 데이터 저장, 영구적, 느림
2. 데이터를 **메모리**에 보관
 - 영구적이지 않음, 빠름
 - 프로그램의 **변수**는 메모리에 데이터 보관

□ 데이터 정렬은 빠른 **메모리**에서 수행

- 데이터를 파일에서 메모리로 읽음
- 많은 데이터를 개개의 변수에 저장하지 않고 하나의 **배열** 변수에 저장

□ 배열은 일종의 **집합 변수** 또는 **데이터 구조체**

- 전체 데이터를 **하나의 덩어리**로 묶어서 이름을 부여
- 배열 자체는 하나의 변수



□ 파이썬은 배열(array)을 리스트(list)로 제공

□ 배열 사용 예

```
my_words = ["Dudes", "and"]           # 배열 생성
print (my_words[0])                   # Dudes 출력
print(my_words[1])                    # and 출력

my_words.append("Bettys")              # 배열 뒤에 데이터 추가
print(my_words[2])                     # Bettys 출력
```

서핑대회 코드 - 처음 3개 점수 출력



```
scores = []
result_f = open("results.txt")

for line in result_f:
    (name, score) = line.split()
    scores.append(float(score))

result_f.close()
print(scores[0])
print(scores[1])
print(scores[2])
```




시험주행

배열 데이터 정렬은?

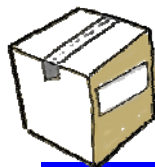


```

Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.2 (default, Sep 4 2011, 09:51:08) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
8.65
9.12
8.45
>>> |

```

- 8.65
- 9.12
- 8.45
- 7.81
- 8.05
- 7.21
- 8.31



파이썬 배열 메서드

메서드	메서드가 하는일
count()	배열 안에 특정 값이 몇 개 있는지 알려줌
extend()	일련의 항목들을 배열에 추가
index()	항목을 검색하고 해당 인덱스를 반환
insert()	지정한 인덱스 위치에 항목을 추가
pop()	배열의 제일 마지막 항목을 제거하고 반환
remove()	배열의 첫 번째 항목을 제거하고 반환
reverse()	배열 항목들의 순서를 반대로 만듦
sort()	배열을 지정한 순서(낮은 것부터 높은 순으로)로 정렬



서핑대회 코드- 높은 점수 3개 출력 [실습 3]



```

scores = []
result_f = open("results.txt")
for line in result_f:
    (name, score) = line.split()
    scores.append(float(score))
result_f.close()

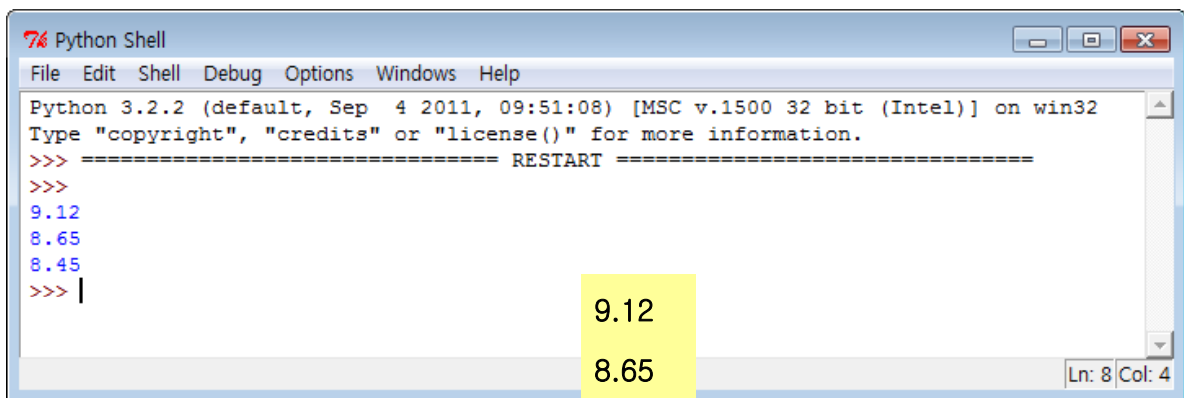
scores.sort()
scores.reverse()

print(scores[0])
print(scores[1])
print(scores[2])

```



시험주행



9.12
8.65
8.45
8.31
8.05
7.81
7.21





보충학습 - 리스트 [실습 4]

- 리스트(list)는 대괄호 []를 사용하여 임의의 자료(객체)를 저장하는 집합적인 자료형

```
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:15:05) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
>>> L = [1,2,3,4,5]
>>> L
[1, 2, 3, 4, 5]
>>> L[3]
4
>>> L[2:4]
[3, 4]
>>> L.append(10)
>>> L
[1, 2, 3, 4, 5, 10]
>>> del L[0]
>>> L
[2, 3, 4, 5, 10]
>>> L.reverse()
>>> L
[10, 5, 4, 3, 2]
>>> L[2] = 8
>>> L
[10, 5, 8, 3, 2]
>>> |
```

순천향대학교 컴퓨터공학

파일과 배열 안의 데이터



보충학습 - 튜플

- 튜플(tuple)는 괄호 ()를 사용하며 리스트와 유사하다. 차이는 튜플은 값을 변경할 수 없다.

```
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:15:05) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> T = (10, 20, 30, 40, 50)
>>> T
(10, 20, 30, 40, 50)
>>> T[0:3]
(10, 20, 30)
>>> T[2]
30
>>> T[2] = 300
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    T[2] = 300
TypeError: 'tuple' object does not support item assignment
>>> |
```



프로그래밍 도구상자 (1)

□ 프로그래밍 도구

- **파일(file)** - 디스크에 저장되어 있는 데이터 읽기
- **리스트(list)** - 대괄호 []를 사용하여 임의의 자료(객체)를 저장하는 집합적인 자료형
- **배열(array)** - 인덱스로 접근할 수 있는 여러 데이터 항목을 갖고 있는 집합 객체 변수
- **정렬(sort)** - 원하는 순서로 집합의 데이터를 배치하는 것



프로그래밍 도구상자 (2)

□ 파이썬 도구

- **open()** - 처리하기 위해 파일을 여는 것
- **close()** - 파일을 닫음
- **read(), write()** - 파일 읽기, 쓰기
- **for** - 어떤 것을 반복
- **string.split()** - 스트링을 여러 부분으로 분할
- **[]** - 배열 인덱스 연산자
- **array.append()** - 배열의 마지막에 항목을 추가
- **array.sort()** - 배열을 낮은 것부터 높은 순으로 정렬
- **array.reverse()** - 배열 안의 데이터들을 역순으로 바꿈

4장 실습

[실습 1] 서핑대회 코드 - 데이터 분리

[실습 2] 보충학습 - 파일 쓰기

[실습 3] 서핑대회 코드- 높은 점수 3개 출력

[실습 4] 보충학습 - 리스트

[실습 5] 파이썬 매뉴얼의 Tutorial의 List 예제 실행



4장 과제

1. 서핑대회 코드를 수정하여 낮은 점수 3개를 파일에 출력
2. 앞에서 배운 내용을 사용한 임의의 프로그램 작성
 - 프로그램 설명
 - 프로그램 소스
 - 실행 결과