12. 파일

1.파일의 이해

파일의 역할

컴퓨터에 의해 처리될 또는 처리된 데이터와 정보가 영지적으로 저장된 상태

- 일련의 연속된 바이트
- 프로그램(파이썬 소스코드)에 읽혀 가공처리

파일의 구성

- 연속된 바이트와 파일의 시작, 파일 포어넣, 파일의 끝(EoF)로 표현

파일의 종류

파일의 종류

데이터가 저장되는 방식에 따라 구분

→텍스트 파일

→바이너리 파일

- 데이터가 저장되는 방식에 따라 구분
  - 1 텍스트 파일
  - 2 바이너리 파일

파일 함수

- 파일의 시작, 파일 포인터, 파일의 끝을 활용하여 데이터 읽기, 쓰기를 위한 함수 및 메소드를 내장

함수	설명
open(): file obj	파일과 연결되어 있는 파일 객체 생성
read( )	특정 개수의 문자로 변환
readline( )	한 라인의 문자열로 반환
readlines( ): list	전체 라인의 문자열을 리스트로 변환
write(): str)	파일에 문자열을 작성
close( )	파일 닫기 및 파일 객체 삭제

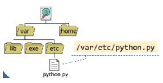
파일 객체 생성

- 구문형식
  - 파일객체_참조변수 = open(파일이름, 모드)
    - 물리적인 파일과 연결된 파일 객체를 생성하고 참조변수에 할당
  - 모드

모드	설명
'r'	읽기 모드
'w'	새로운 파일을 쓰기 모드
'a'	파일의 끝이 데이터를 덧붙이기 모드

파일 이름

- 파일의 고유 식별자 역할 및 저장장치 내부에서 파일의 위치를 표현하는 파일경로를 내포



- /var/etc/python.py

파일 읽기

- 특정 범위의 데이터를 파일에서 읽고 문자열로 변환
  - 파일 포인터의 이동을 동반

```
Hamlet.Written William Shakespeare#Edited by Ba
```

```
b_fp = open("Hamlet_by_Shakespeare.txt", "r")
title = b_fp.read(8)
author = b_fp.readline( )
b_fp.close( )
```

파일 쓰기

- 문자열을 파일 포인터가 위치한 지점에 기록
- w 모드로 존재하는 파일 오픈 시 데이터 삭제

```
p_fp = open("python.txt", "w")
p_fp.write("DOOU")
p_fp.write("python programming\n")
p_fp.close( )
```

```
DOOUpython programming#n
```

데이터 추가

- 파일의 끝에 데이터를 덧붙이는 작업

```
KhOUWn python programming#n by CSWn
```

```
p_fp = open("python.txt", "a")
p_fp.write("\nby CSW")
p_fp.close( )
```

- 파일 오픈 후 파일 포인터를 EoF로 이동
- 존재하지 않는 파일은 write와 동일

파일 읽고 쓰고 수정하는 프로그램

'Khan.txt'파일을 읽고 처리하는 프로그램을 작성하시오

- 모든 내용을 출력하시오.
- 마지막에 "동기스 칸-"를 삽입하시오.

2.파일의 활용

데이터 분석 프로그램

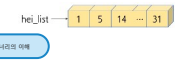
데이터 분석 프로그램

"Hamlet_by_Shakespeare.txt" 파일에 포함된 단어가 출현한 횟수를 출력하는 프로그램을 작성하시오.

```
[the] 984회
[acquaintance] 2회
[say] 7회
[approve] 2회
[eyes] 2회
...
```

시퀀스의 개념

- 순서화된 값의 집합체를 저장할 수 있는 데이터 타입
  - 단일 식별자로 연속된 저장 공간 접근 수단 제공
  - 개별 원소의 값을 수정, 추가, 삭제 가능
  - 원소(element)의 나열을 저장할 수 있는 자료
    - 리스트, 세트, 튜플, 딕셔너리 등



딕셔너리의 이해

- 키와 값의 쌍(pair)을 저장하는 시퀀스

딕셔너리의 이해

키와 값의 쌍(pair)을 저장하는 시퀀스

딕셔너리 생성

- 구문형식
  - 딕셔너리_이름 = {키1:값1, 키2:값2, ..., 키n:값n}
  - 세트, 리스트, 딕셔너리는 키 불가능
- 빈 딕셔너리 생성 구문형식
  - 딕셔너리_이름 = {}
  - 딕셔너리_이름 = dict()

항목의 추가 및 삭제

- 추가 구문형식
  - 딕셔너리_이름[키] = 값
- 삭제 구문 형식
  - del 딕셔너리_이름[키]

딕셔너리 멤버

함수	설명
keys(): tuple	포함된 모든 키를 반환
values(): tuple	포함된 모든 값을 반환
items(): tuple	(키, 값)의 튜플의 튜플로 모든 항목을 반환
clear(): None	모든 항목을 이해
get(key): value	키에 대응하는 값을 반환
pop(key): value	키에 대응하는 값을 반환하고 항목을 삭제
pop.item(): tuple	무작위로 한 (키, 값)을 반환하고 선택된 항목 삭제

딕셔너리 순회

- 순회(traversal)
  - 각각의 항목을 순서대로 한 번씩만 방문하는 과정
- 순회형식
  - for key in dictionary:
    print(key+ ":" + str(dictionary[key]))
    key 사용 불록

데이터 분석 프로그램

"Hamlet_by_Shakespeare.txt" 파일에 포함된 단어가 출현한 횟수를 출력하는 프로그램을 작성하시오.

```
[the] 984회
[acquaintance] 2회
[say] 7회
[approve] 2회
[eyes] 2회
...
```

데이터 분석 프로그램 개선

데이터 분석 프로그램 개선

'Hamlet_by_Shakespeare.txt' 파일에 출현 횟수가 100 이상되는 단어와 출력 횟수를 정렬하여 출력하는 프로그램을 작성하시오

```
[the] 1086 회
[and] 965 회
[to] 756 회
[of] 678 회
[you] 549 회
[i] 529 회
...
```

연습문제

1.

2.

3.