12.Python 파일 입출력

강사 강태광

1. Python 파일 입출력

1] **파일 입출력 방법**

- 파일을 열어서 읽고, 쓰고, 덧붙이는 방법

-open(filename, mode) 내장 함수로
filename 이름을 지닌 file 객체를 얻는다.

-얻어진 파일 객체에서 자료를 읽거나, 쓰거나,
덧붙이는 작업 수행

-모든 작업이 끝나면 close()를 호출하여
작업 프로세스의 자원 점유 해제

- open 내장 함수의 두번째 인자 mode 설명

- 두번째 인자 mode 생략시에는 읽기 전용(r) 모드로 설정

Mode	간단 설명	자세한 설명
'r'	읽기 전용 (기본 모드)	파일 객체를 읽기 모드로 생성하고, 파일 포인터를 파일 처음 위치에 놓는다.
'w'	쓰기 전용	새로운 파일을 쓰기 모드로 생성하거나 해당 파일이 이미 존재하면 내용을 모두 없에면서 쓰기 모드로 생성하고, 파일 포인터를 파일 처음 위치에 놓는다.
'a'	파일 끝에 추가	이미 존재하는 파일을 쓰기 모드로 생성하거나 파일이 존재하지 않으면 새롭게 파일을 생성하면서 쓰기 모드로 생성하고, 파일 포인터를 파일의 마지막 위치에 놓는다. 그래서, 이후 작성되는 내용은 파일의 뒷 부분에 추가된다.

예시 1

```
s = """Its power: Python developers typically report
they are able to develop applications in a half
to a tenth the amount of time it takes them to do
the same work in such languages as C."""
f = open('t.txt', 'w')
f.write(s) # 문자열을 파일에 기록
f.close()
```

결과

---

t.txt 파일에 s 내용 저장

1. Python 사전 활용법

2] **해쉬 기법**
- 사전을 출력하면 각 아이템들이 임의의 순서로 출력된다.
- 새로운 아이템이 들어오면 키 내용에 따라 그 순서가 달라진다.
- 내부적으로 키 내용에 대해 해쉬(Hash) 기법을 사용(검색 속도가 매우 빠름)
- 키와 값 매핑에 대한 아이템을 삭제할 때에는 del과 함께 키값 명시
- 사전 내 아이템 순서는 존재하지 않음

예시2

```
def add(a, b):
    return a + b
def sub(a, b):
    return a – b

action = {0: add, 1: sub} # 함수 이름을 사전의 값으로 사용
print (action[0](4, 5))  # add(a, b)  호출
print (action[1](4, 5))  # sub(a, b)  호출

action2 = {add: 1, sub: 2} # 함수 이름을 사전의 키로 사용
print (action2[add])         # 검색을 add로 하여 add의 value 값 1 출력
```

결과

```
--------------------------
9
-1
1
```

1. Python 사전 활용법

3] **dict 내장함수**
- 사전을 생성하는 다른 방법: 내장함수 dict() 사용
- 내장함수 zip
  ① zip의 원소로 시퀀스 자료형 2개 사용
  ② 두 개의 자료를 순서대로 쌍으로 묶은 튜플들의 리스트 반환

예시2

```
keys = ['one', 'two', 'three']
values = (1, 2, 3)
print (zip(keys, values))          # zip(): 두 개의 자료를 순서대로 쌍으로 묶은 튜플들의 리스트 반환
print (dict(zip(keys, values)))  # zip 함수를 dict 함수의 원소로 사용 가능
```

결과

```
--------------------------
<zip object at 0x000000AF20F798C8>
{'one': 1, 'two': 2, 'three': 3}
```

2. Python 사전 method

4] **dict Method (중요)**
- D.keys(): 사전 D에서 키들을 리스트로 반환
- D.values(): 사전 D에서 값들을 리스트로 반환
- D.items(): 사전 D에서 각 아이템을 튜플형태로 가져와 리스트로 반환
- key in D: 사전 D안에 key를 키값을 가진 아이템이 있는지 확인

예시2

```
phone = {'jack':9465215, 'jin':1111, 'Joseph':6584321}
print ((phone).keys()) # 키의 리스트 반환
print ((phone).values()) # 값들의 리스트 반환
print ((phone).items()) # (키, 값)의 리스트 반환
```

결과

```
--------------------------
dict_keys(['jin', 'Joseph', 'jack'])
dict_values([1111, 6584321, 9465215])
dict_items([('jin', 1111), ('Joseph', 6584321), ('jack', 9465215)])
```

2. Python 사전 method

5] **루프를 이용한 사전 내용 참조**
- 사전의 모든 키값을 순차적으로 참조하는 방법

예시2

```
D = {'a':1, 'b':2, 'c':3}
for key in D.keys():      # key 값만 리스트 안에 담겨 리턴
    print key, D[key]    # key에 해당하는 value 값이 반환됨
```

결과

```
---------------------------
a 1
c 3
b 2
```