

I 2. Python 사전 활용법

1. Python 사전 활용법

1] 사전 개념

- 집합적 자료형
- 자료의 순서를 정하지 않는 매핑(Mapping)형
 - ① 키(Key)를 이용하여 값(Value)에 접근
 - ② 시퀀스 자료형은 아님
- 키와 값의 매핑 1개를 아이템(item)이라고 부름(인덱싱 X, 사전의 검색 연산)
- 사전(리스트, 튜플과 함께 가장 많이 활용되는 내장자료형)
- 집합적 자료형 = 사전이라는 자료 내 여러 개의 객체 존재 가능
- 매핑형 자료형은 사전이 유일
- 콤마(,) 단위가 사전의 원소 개수

예시 1

```
member = {'basketball':5, 'soccer':11, 'baseball':9}
member['volleyball'] = 7 # 새로운 아이템 설정
member['volleyball'] = 6 # volleyball Key에 해당하는 내용이 변경
print (member)
print (len(member)) # 아이템의 개수 반환
```

결과

```
-----
{'basketball': 5, 'volleyball': 6, 'baseball': 9, 'soccer': 11}
```

```
4
```

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 아이টে을 튜플형태로 가져와 리스트로 반환
- key in D: 사전 D안에 key를 키값을 가진 아이টে이 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
```