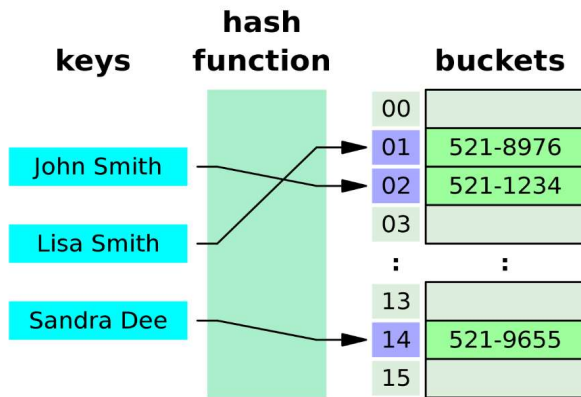


13. 사전 과 집합

사전(연관 배열)

- 키 와 값의 쌍을 저장 하는 대용량의 자료 구조
- . 해시 알고리즘 을 사용하여 데이터 가 정렬 되어 보관 되므로 대용량 데이터 를 관리 할때 빠르게 결과를 얻을 수 있다.



출처: https://en.wikipedia.org/wiki/Hash_table



< 출처 : <https://blog.hexabrain.net/245> >

- * 사전 : Key 의 리스트 와 Value 의 리스트를 서로 연결 시켜 데이터를 관리하기 쉽도록 만든 자료형(메모리 비용이 크다)

- > 리스트 및 배열 등은 선형 탐색 구조 형식
- . 따로 조작 할 필요가 없어 단순하지만 데이터를 관리하는데 다소 비효율적이다.
- > 해시 알고리즘
- . Key 를 일정한 길이의 주소값 으로 변형하여 값과 연결.
- > 리스트 전화번호부 : 전화번호를 찾기 위하여 처음부터 끝까지 검색
- > 사전 전화번호부 : 이름을 알고있으면 곧바로 접근
- > 물론 리스트의 index 를 알고있으면 직접 접근
- 숫자로 된 값의 데이터를 모두 기억하기는 어려움.

사전 의 기본 형식

```
Chap01_intro > 10.Dictionary[사전].py > ...
1 dic = { 'boy' : '소년', 'school' : '학교', 'book' : '책' }
2 print(dic)
3 print(dic['boy'])
```

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_10.Dictionary[사전].py

```
{'boy': '소년', 'school': '학교', 'book': '책'}
```

키	값
boy	소년
school	학교
book	책

키

- > 값을 찾는 기준
- > 중복되지 않는 고유한 값을 가진다.
- > 읽기 전용 속성으로 변경 할 수 없다. (튜플은 '키'로 사용 가능 하지만 리스트 는 '키'로 사용 할 수 없다.)

값

- > 키 에 할당 되는 데이터
- > 데이터 변경이 가능하다

키를 통한 값의 표현

- > 3행 과 같이 사전 객체 에 ['키'] 를 입력하여 키 에 등록 된 값을 호출 할 수 있다.

값 찾기

.get() 메서드 를 통한 Key 의 값 찾기

```
Chap01_intro > 10.Dictionary[사전].py > ...
1 dic = { 'boy' : '소년', 'school' : '학교', 'book' : '책' }
2 print(dic.get('boy'))
3 print(dic.get('student'))
4 print(dic.get('student'), '등록되지 않은 key 입니다.')
5
```

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_10.Dictionary[사전].py

```
소년
None
None 등록되지 않은 key 입니다.
```

- > 2행 : get 메서드를 통해 key 의 값을 찾을 수 있다.
- > 3행 : 존재하지 않는 key 를 검색 시 None 를 반환한다
- > 4행 : 존재 하지 않을 경우의 메시지 처리를 구현한다.

. In 을 통한 key 의 값 찾기

```
Chap01_intro > 10.Dictionary[사전].py > ...
1  dic = { 'boy' : '소년', 'school' : '학교', 'book' : '책' }
2  if 'student' in dic :
3      print('등록 되어 있는 Key 입니다.')
4  else :
5      print('등록 되지 않는 Key 입니다.')
6
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_...
등록 되지 않는 Key 입니다.

사전 편집

- 값의 삽입, 삭제, 수정
- . 사전 자료 형 편집 의 예

```
Chap01_intro > 10.Dictionary[사전].py > ...
1  dic = { 'boy' : '소년', 'school' : '학교', 'book' : '책' }
2  dic['boy'] = '남자아이'
3  dic['girl'] = '여자아이'
4  del dic['book']
5  print(dic)
6
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_...
{'boy': '남자아이', 'school': '학교', 'girl': '여자아이'}

>2행 : boy 키 값 수정
>3행 : Key 와 값을 삽입
>4행 : book 키 데이터 삭제

- 사전의 Key , 값 의 목록 확인
- . Keys(), Values(), items()

```
55  dic = { 'boy' : '소년', 'school' : '학교', 'book' : '책' }
56  print(dic.keys())
57  print(dic.values())
58  print(dic.items())
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Python\Source> cmd /C "C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe c:\Python\debugpy\adapter\...
dict_keys(['boy', 'school', 'book'])
dict_values(['소년', '학교', '책'])
dict_items([('boy', '소년'), ('school', '학교'), ('book', '책')])

>56행 : 사전 의 Key 를 추출한다.
>57행 사전의 Value 를 추출한다.
>58행 : 사전의 Key 와 Value 를 모두 추출한다.

> Key 를 추출 후 for 문을 통하여 출력 할 수 있다.

```
Chap01_intro > 10.Dictionary[사전].py > ...
1 dic = { 'boy' : '소년', 'school' : '학교', 'book' : '책' }
2 KeyList = dic.keys()
3 for key in KeyList :
4     print(key)
5
6
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_intro/
boy
school
book

- 사전 의 병합

. Update()

> 두 사전의 'book' Key 는 중복되어 후행 하는 사전의 데이터 로 덮어쓰기 된다.

```
Chap01_intro > 10.Dictionary[사전].py > ...
1 dic1 = { 'boy' : '소년', 'school' : '학교', 'book' : '책' }
2 dic2 = { 'student' : '학생', 'teacher' : '선생님', 'book' : '서적' }
3 dic1.update(dic2)
4 print(dic1)
5
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_intro/10.Dictionary
{'boy': '소년', 'school': '학교', 'book': '서적', 'student': '학생', 'teacher': '선생님'}

- 사전 형으로 변환

. dict()

> 2차원 리스트 를 사전으로 변환

* 키 와 값의 쌍으로 이루어진 리스트 또는 튜플 여야 한다.

```
84 li = [['boy', '소년'], ['school', '학교'], ['book', '책']]
85 dic = dict(li)
86 print(dic)
87
88 li = (('boy', '소년'), ('school', '학교'), ('book', '책'))
89 dic = dict(li)
90 print(dic)
91
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

C:\Python\Source> cmd /C "C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe c:\
ebugpy\adapter/../../debugpy\launcher 65036 -- C:\Python\Source\13.Dictionary[사전_집합].py "

{'boy': '소년', 'school': '학교', 'book': '책'}
{'boy': '소년', 'school': '학교', 'book': '책'}

사전 의 활용

- 알파벳 개수 찾기
- 특정 데이터에 포함되어있는 문자의 개수 찾기

```
Chap01_intro > 10.Dictionary[사전].py > ...
1  song = '''
2  Like an apple is my face
3  (사과 같은 내 얼굴)
4  How I look so beautiful
5  (예쁘기도 하구나)
6  Eyes are shiny nose is shiny
7  (눈도 반짝 코도 반짝)
8  And my lips are shiny
9  (입도 반짝 반짝 )
10 ''
11
12 dicWord = dict()
13 for c in song :
14     if c.isalpha() == False:
15         continue
16     c = c.lower()
17     if c not in dicWord :
18         dicWord[c] = 1
19     else :
20         dicWord[c] += 1
21 print(dicWord)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/chap01_intro/10.Dictionary[사전].py
{ 'l': 5, 'i': 9, 'k': 2, 'e': 9, 'a': 7, 'n': 6, 'p': 3, 's': 9, 'm': 2, 'y': 6, 'f': 2, 'c': 1, '사': 1, '과': 1, '같': 1, '은': 1, '내': 1, '얼': 1, '굴': 1, 'h': 4, 'o': 5, 'w': 1, 'b': 1, 'u': 2, 't': 1, '예': 1, '쁘': 1, '기': 1, '도': 4, '하': 1, '구': 1, '나': 1, 'r': 2, '눈': 1, '반': 4, '짝': 4, '코': 1, 'd': 1, '입': 1 }

- > 12행 : 사전 을 생성
- > 14행 : 추출한 문자 가 문자 형식이 아니라면 continue * 한글 : True, 공백이나 숫자, 특수문자를 제외함
- > 16행 : 추출 문자 를 소문자로 변경 (대/소문자 구분을 하지 않고 알파벳의 개수 를 찾기 위하여)
- > 17행 : 추출문자 가 dicWord 사전의 key 로 포함되어있지 않다면
- > 18행 : 추출문자 로 된 key 에 1의 값을 사전에 삽입
- > 20행 : 추출문자 가 이미 사전에 재 하는 Key 라면 해당 key 의 값을 +1 증가

- 정렬을 통한 출력 응용

```
Chap01_intro > 10.Dictionary[사전].py > ...
1  song = '''
2  Like an apple is my face
3  (사과 같은 내 얼굴)
4  How I look so beautiful
5  (예쁘기도 하구나)
6  Eyes are shiny nose is shiny
7  (눈도 반짝 코도 반짝)
8  And my lips are shiny
9  (입도 반짝 반짝 )
10 ''
11
12 dicWord = dict()
13 for c in song :
14     if c.isalpha() == False:
15         continue
16     c = c.lower()
17     if c not in dicWord :
18         dicWord[c] = 1
19     else :
20         dicWord[c] += 1
21 keylist = list(dicWord.keys())
22 keylist.sort()
23 for c in keylist :
24     cnt = dicWord[c]
25     print(c , ' => ' , cnt)
26
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.22621.213]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/chap01_intro/10.Dictionary[사전].py

a => 7
b => 1
c => 1
d => 1
e => 9
f => 2
h => 4
i => 9
k => 2
l => 5
m => 2
n => 6
o => 5
p => 3
r => 2
s => 9
t => 1
u => 2
w => 1
y => 6
같 => 1
과 => 1
구 => 1
굴 => 1
기 => 1
나 => 1
내 => 1
눈 => 1
도 => 4
반 => 4
쁘 => 1
사 => 1
얼 => 1
예 => 1
은 => 1
입 => 1

- > 21행 : 사전의 Key 를 추출 하여 list 로 생성
- > 22행 : List 를 정렬
- > 23행 : List 순서 대로 사전의 Key 값 출력

. 정렬 의 또다른 예제

```
12 dicWord = dict()
13 for c in song :
14     if c.isalpha() == False:
15         continue
16     c = c.lower()
17     if c not in dicWord :
18         dicWord[c] = 1
19     else :
20         dicWord[c] += 1
21
22 for code in range(ord('a'),ord('z') +1) :
23     c = chr(code)
24     num = dicWord.get(c,0)
25     print(c , ' => ' , num)
```

>22행 : a 와 z 의 아스키코드 번호 로
반복

>23행 : 추출한 코드 를 문자로 변형

>24행 : dicWord 에 포함된 Key 의 값 확인
없을 경우 0 으로 표현

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python

```
a => 7
b => 1
c => 1
d => 1
e => 9
f => 2
g => 0
h => 4
i => 9
j => 0
```

실습

1. 위의 예제 에서 가장 많이 등장하는 문자 를 찾아 출력하세요

```
dict_items([('i', 9), ('e', 9), ('s', 9)])
```

집합

- 여러 가지 값의 모임

. { } 로 표현

> 'korea' 는 중복 되어 마지막 'korea' 는 표현되지 않는다.

```
Chap01_intro > 10.Dictionary[사전_집합].py > ...
1 asia = {'korea', 'china', 'japan', 'korea'}
2 print(asia)
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

{'japan', 'korea', 'china'}

D:\Python>C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:/P
{'korea', 'japan', 'china'}
```

. 다른 자료형을 집합으로 변환 하기 set()

```
Chap01_intro > 10.Dictionary[사전_집합].py > ...
1 print(set("Python"))
2 print(set([22,334,53,23]))
3 print(set(("김범수", "임창정", "박효신")))
4 print(set({'boy': '소년', 'teacher': '선생님'}))
5 test_Set = set()
6 test_Set = {'나비', '매미', '귀뚜라미'}
7 print(test_Set)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:/Pyth
{'o', 't', 'y', 'p', 'n', 'h'}
set([22, 334, 53, 23])
{'박효신', '임창정', '김범수'}
{'teacher', 'boy'}
{'매미', '귀뚜라미', '나비'}
```

>1행 : 문자열 을 집합으로 변환하기

>2행 : 리스트 를 집합으로 변환하기

>3행 : 튜플 을 집합으로 변환하기

>4행 : 사전 을 집합으로 변환하기

* key 값 만 집합으로 변환된다.

>5행 : 집합의 생성 및 활용

. 집합의 추가 . 제거

```
Chap01_intro > 10.Dictionary[사전_집합].py > ...
1 Set = {'나비', '매미', '귀뚜라미'}
2 Set.add('풍뎅이')
3 print(Set)
4 Set.add('매미')
5 print(Set)
6 Set.remove('나비')
7 print(Set)
8 Set.update({'사마귀', '물방개', '잠자리'})
9 print(Set)
10

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python
{'풍뎅이', '나비', '매미', '귀뚜라미'}
{'풍뎅이', '나비', '매미', '귀뚜라미'}
{'풍뎅이', '매미', '귀뚜라미'}
{'사마귀', '잠자리', '귀뚜라미', '매미', '물방개', '풍뎅이'}
```

>2행 : 풍뎅이 추가

>4행 : 매미 가 있으므로 추가 되지 않음

>6행 : 나비 삭제

>8행 : 다른 집합의 연결 update

집합의 연산

- 차집합, 교집합, 합집합, 배타적 집합

연산	기호	메서드	설명
합집합		union	두 집합의 모든 원소
교집합	&	intersection	두 집합 모두에 있는 원소
차집합	-	difference	왼쪽집합의 원소 중 오른쪽집합의 원소를 뺀 것
배타적 집합	^	symmetric_difference	중복을 제외한 모든 원소

. 연산의 예시

```
Chap01_intro > 10.Dictionary[사전_집합].py > ...
1 two_multi = {2,4,6,8,10,12}
2 three_multi = {3,6,9,12,15}
3
4
5 print('교집합', two_multi & three_multi)
6 print('합집합', two_multi | three_multi)
7 print('차집합', two_multi - three_multi)
8 print('차집합', three_multi - two_multi)
9 print('배타적 집합', two_multi ^ three_multi)
10
```

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:/Python/Chap01_intro/10.Dictionary[사전_집합].py

교집합 {12, 6}
합집합 {2, 3, 4, 6, 8, 9, 10, 12, 15}
차집합 {8, 2, 10, 4}
차집합 {9, 3, 15}
배타적 집합 {2, 3, 4, 8, 9, 10, 15}

>1행 : 2의 배수 집합
>2행 : 3의 배수 집합

>5행 : 중복되는 원소 표현 교집합
>6행 : 중복은 한번만 표현하고 모두 표현
>7행 : 왼쪽 집합에만 있는 원소 표현
>9행 : 중복되는 원소는 제거하고 모두 표현

. 부분 집합, 포함 집합

연산	기호	메서드	설명
부분 집합	<=	issubset	오른쪽집합 내용이 왼쪽 집합에 있는지 조사
진성 부분집합	<		부분집합 이면서 여분의 원소가 더 있는지 확인
포함 집합	>=	issuperset	왼쪽집합 내용이 오른쪽 집합에 있는지 조사
진성 포함집합	>		포함집합이면서 여분의 원소가 더 있음

```
Chap01_intro > 10.Dictionary[사전_집합].py > ...
1 mammal = {'코끼리', '고릴라', '사자', '고래', '사람', '원숭이', '개'}
2 primate = {'사람', '원숭이', '고릴라'}
3
4 if '사자' in mammal : print('사자는 포유류 입니다.')
5 else : print('사자는 포유류 가 아닙니다.')
6
7 print(primate <= mammal)
8 print(primate < mammal)
9 print(primate <= primate)
10 print(primate < primate)
11
12
13
```

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:/Python/Chap01_intro/10.Dictionary[사전_집합].py

사자는 포유류 입니다.
True
True
True
False

>7행 : mammal 이 primate 를 포함 하는가
>8행 : mammal 이 primate 를 포함하고
mammal 의 원소가 더 있는가
>9행 : primate 이 primate 를 포함 하는가
>10행 : primate 이 primate 를 포함 하고
여분의 원소가 더 존재 하는가