

컴퓨터정보과 파이선 프로그래밍

06주차 – 제어문 & 자료형

enumerate() 함수

- iterable 객체를 인자로 받아, 각 요소의 인덱스 번호와 요소 자체를 tuple 형태로 이루어진 iterator 객체를 반환

```
scores = [10, 20, 30]
```

```
i = 1
```

```
for score in scores:  
    print(f"{i}:{score}")  
    i+=1
```

```
for i in range(len(scores)):  
    print(f"{i+1}:{scores[i]}")
```

```
for i, score in enumerate(scores):  
    print(f"{i+1}:{score}")
```

- iterable : 반복할 수 있는 객체
 - list, tuple, str, ...
- iterator: 순차적으로 값을 하나씩 반환할 수 있는 객체
 - range, enumerate, ...

중첩 반복문

```
data = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
for d1 in data:  
    for d2 in d1:  
        print(d2, end=" ")
```

```
for i in range(len(data)):  
    for j in range(len(data[i])):  
        print(data[i][j], end=" ")
```

```
for i, d1 in enumerate(data):  
    for j, d2 in enumerate(d1):  
        print(f"data[{i}][{j}]={d2}", end=" ")
```

자료형과 기본 연산 (2장)

그룹	자료형	생성자	리터럴 표현	내용
수치형 (Number)	Integer	int()	1	정수
	Floating point number	float()	1.1	실수
	Complex Number	complex()	2 + 3j	복소수
논리형 (Boolean)	Boolean	bool()	True	참(True)과 거짓(False)만 값을 가짐
군집형 (Collection)	String	str()	" 1 " 혹은 ' 1 '	문자열 (순서 0, 수정 X, 중복 0, 구성요소한정 0)
	List	list()	[1,2]	리스트 (순서 0, 수정 0, 중복 0, 구성요소한정 X)
	Tuple	tuple()	(1,2)	튜플 (순서 0, 수정 X, 중복 0, 구성요소한정 X)
	Set	set()	{1,2}	집합, 세트 (순서 X, 수정 0, 중복 X, 구성요소한정 X)
	Dictionary	dict()	{1: " 1 ", 2: " 2 " }	사전, 딕셔너리 (순서 X, 수정 0, 중복 X, 구성요소한정 X) {key:value, ... , keyn:valuen}

딕셔너리

- Dictionary (사전) p.93
 - 연관배열 associative array, 해쉬 hash 라고도 불림
 - ★ 순서가 없음
 - Key : Value, 쌍 pair로 이루어진 정보가 0개 이상 모여있음
 - 리스트, 튜플과 같이 순차(인덱스 사용) 방식을 사용하지 않고, Key를 통해서만 연관된 Value를 얻어올 수 있는 구조.
 - 중괄호와 키:값 형태로 구성
 - { key1:value1 , key2:value2, ... keyN:valueN }
 - 키는 중복 될 수 없음 (단, 값은 동일해도 됨)
 - 키는 변하지 않는 값이어야 함 (list vs. tuple) p.98
- 기본구조

```
1 myinfo = {'name': 'kim inha', 'age': 23}
2 print("이름:", myinfo['name'])
3 print("나이:", myinfo['age'])
```

이름: kim inha
나이: 23

딕셔너리

- 빈 딕셔너리

```
myInfo = {} #dict()
```

- 키-값 추가 p.94

```
myInfo['name'] = '김인하'
```

```
myInfo['age'] = 23
```

- 값 수정 (기존 키에 값을 넣으면 수정)

```
myInfo['age'] = 24 # 수정
```

```
myInfo['height'] = 163.2 # 추가
```

- 삭제 p.95

```
del myInfo['height']
```

- 키를 통한 값 접근하기 p.96

```
print(f"나의 이름은 {myInfo['name']}입니다.")
```

딕셔너리

▪ 딕셔너리 관련 메소드 (1)

dicts = {1:'one', 2:'two'}

- Key의 순차구조 자료형 가져오기 : `keys()` p.98
 - 자료형 : `dict_keys`
 - `dicts.keys()`
- Value의 순차구조 자료형 가져오기 : `values()` p.99
 - 자료형 : `dict_values`
 - `dicts.values()`
- (Key,Value)의 순차구조 자료형 가져오기 : `items()` p.99
 - 자료형 : `dict_items` (Key와 Value를 tuple로 묶은 객체)
 - `dicts.items()`

위 세 가지 자료형으로 가져온 객체는 `list()` 생성자를 이용하여 `list`자료형으로 생성할 수 있다.

- 모두 지우기 : `clear()` p.100
 - `dicts.clear()`

딕셔너리

- 딕셔너리 관련 메소드 (2)

```
myInfo = {1:'one', 2:'two'}
```

- Key를 통해 Value 얻기 : get() p.100
print(f"나의 키는 {myInfo['height']}cm 입니다.")

```
Traceback (most recent call last):  
  File "E:/PythonClass/Week3/test01.py", line 13, in <module>  
    print(f"나의 키는 {myInfo['height']}cm 입니다.")  
KeyError: 'height'
```

- 존재하지 않는 키에 접근하면 에러 발생
- **get(키)**를 사용하면, 키가 없을 경우 **None(뜻:아무것도 없다)** 값을 반환한다.
height = myInfo.get('height')
if None != height: # if not height:
 print(f"나의 키는 {height}cm 입니다.")
else:
 print("아직 키 정보가 없습니다.")

딕셔너리

- 딕셔너리 관련 연산자

```
myInfo = {1: 'one ', 2: 'two '}
```

- Key가 딕셔너리 안에 있는지 조사하기 : in / not in p.101

```
a = '1' in myInfo
```

```
b = 1 in myInfo
```

```
c = 1 not in myInfo
```

딕셔너리

- 순회
 - 키-값 쌍 순회
 - 키 순회
 - 값 순회
- 중첩
 - 리스트의 Element가 딕셔너리
 - 딕셔너리의 Value가 리스트
 - 딕셔너리의 Value가 딕셔너리

딕셔너리

- 키 순회 p.98

- dict.keys() 이용

```
1 fav_fruits = {  
2     '김인하': '딸기',  
3     '이물류': '귤',  
4     '최کم정': '복숭아',  
5     '박정석': '키위'  
6 }  
7  
8 for key in fav_fruits.keys():  
9     print(f"{key}이(가) 좋아하는 과일은 {fav_fruits[key]}입니다.")  
10  
11 #print(fav_fruits.keys())  
12 #print(list(fav_fruits.keys()))
```

- keys()없이 사용해도 키 순회로 동작한다.

```
8 for key in fav_fruits:  
9     print(f"{key}이(가) 좋아하는 과일은 {fav_fruits[key]}입니다.")
```

김인하(가) 좋아하는 과일은 딸기입니다.
이물류(가) 좋아하는 과일은 귤입니다.
최کم정(가) 좋아하는 과일은 복숭아입니다.
박정석(가) 좋아하는 과일은 키위입니다.
dict_keys(['김인하', '이물류', '최کم정', '박정석'])
['김인하', '이물류', '최کم정', '박정석']

딕셔너리

- 값 순회 p.99
 - dict.values() 이용

```
1 fav_fruits = {  
2     '김인하': '딸기',  
3     '이물류': '귤',  
4     '최컴정': '복숭아',  
5     '박정석': '귤'  
6 }  
7  
8 for value in fav_fruits.values():  
9     print(f"좋아하는 과일은 {value}에 투표해주셔서 감사합니다.")  
10  
11 #print(fav_fruits.values())  
12 #print(list(fav_fruits.values()))
```

좋아하는 과일은 딸기에 투표해주셔서 감사합니다.
좋아하는 과일은 귤에 투표해주셔서 감사합니다.
좋아하는 과일은 복숭아에 투표해주셔서 감사합니다.
좋아하는 과일은 귤에 투표해주셔서 감사합니다.
dict_values(['딸기', '귤', '복숭아', '귤'])
['딸기', '귤', '복숭아', '귤']

딕셔너리

- 키-값 쌍 순회 p.99
 - dict.Items() 이용

```
1 fav_fruits = {  
2     '김인하': '딸기',  
3     '이물류': '귤',  
4     '최کم정': '복숭아',  
5     '박정석': '키위'  
6 }  
7  
8 for name, fruit in fav_fruits.items():  
9     print(f"{name}이 좋아하는 과일은 {fruit}입니다.")  
10  
11 #print(fav_fruits.items())  
12 #print(list(fav_fruits.items()))
```

김인하이 좋아하는 과일은 딸기입니다.

이물류이 좋아하는 과일은 귤입니다.

최کم정이 좋아하는 과일은 복숭아입니다.

박정석이 좋아하는 과일은 키위입니다.

```
dict_items([('김인하', '딸기'), ('이물류', '귤'), ('최کم정', '복숭아'), ('박정석', '키위')])  
[('김인하', '딸기'), ('이물류', '귤'), ('최کم정', '복숭아'), ('박정석', '키위')]
```

딕셔너리

- 딕셔너리가 요소인 리스트

```
1 alien_0 = {'color': 'green', 'points': 5}
2 alien_1 = {'color': 'red', 'points': 15}
3 alien_2 = {'color': 'blue', 'points': 20}
4
5 aliens = [alien_0, alien_1, alien_2]
6
7 for idx in range(len(aliens)):
8     print(f"{idx+1}번 외계인 색상:{aliens[idx]['color']}")
```

딕셔너리

■ 딕셔너리의 Value가 리스트

```
1 bibimbap = {  
2     '양념': '고추장',  
3     '고명': ['버섯', '계란', '콩나물', '시금치', '육회']  
4 }  
5  
6 print(f"당신이 주문한 비빔밥의 양념은 {bibimbap['양념']}이고, 고명은 ", end="")  
7 print(", ".join(bibimbap['고명']), end=" 입니다.\n")  
8  
9 fav_fruits = {  
10     '김인하': ['딸기', '오렌지'],  
11     '이물류': ['귤', '무화과'],  
12     '최کم정': ['복숭아', '귤', '배'],  
13     '박정석': ['키위', '자두']  
14 }  
15  
16 for name, fruits in fav_fruits.items():  
17     print(f"{name}이 좋아하는 과일은 아래와 같습니다.")  
18     for fruit in fruits:  
19         print(f"\t{fruit}")
```

당신이 주문한 비빔밥의 양념은 고추장이고, 고명은 버섯, 계란, 콩나물, 시금치, 육회 입니다.
김인하이 좋아하는 과일은 아래와 같습니다.

딸기

오렌지

이물류이 좋아하는 과일은 아래와 같습니다.

귤

무화과

최کم정이 좋아하는 과일은 아래와 같습니다.

복숭아

귤

배

박정석이 좋아하는 과일은 아래와 같습니다.

키위

자두

딕셔너리

■ 딕셔너리 Value가 딕셔너리

```
1 students = {  
2     '12210001': {'name': '김인하', 'major': '컴퓨터'},  
3     '12210011': {'name': '김슈숙', 'major': '전자'},  
4     '12210111': {'name': '김슈욱', 'major': '물류'}  
5 }  
6  
7 for number, student in students.items():  
8     print(f"학번:{number}")  
9     print(f"이름:{student['name']}")  
10    print(f"전공:{student['major']}")  
11    print()
```

학번:12210001
이름:김인하
전공:컴퓨터

학번:12210011
이름:김슈숙
전공:전자

학번:12210111
이름:김슈욱
전공:물류