

문제해결을 위한 코딩 첫걸음

5장 리스트_튜플_딕셔너리에 대해 알아봅시다(2)

한성대학교 노은희 교수



학습 목표

- 리스트를 이해하고 리스트의 인덱싱, 리스트 슬라이싱, 리스트 연산에 이해합니다.
- 리스트관련 메서드 함수에 대해 이해합니다.
- 튜플을 이해하고, 튜플 요소를 생성, 접근하는 방법을 이해합니다.
- 딕셔너리를 이해하고 딕셔너리 요소를 생성, 삭제, 접근하는 방법을 이해합니다.



리스트 관련 메서드 함수

- 리스트에 요소 삽입 : insert(인덱스, 값)

```
a = [3,5,7,9]
```

```
a.insert(0,1)
```

```
print(a)
```

a리스트 0번 인덱스에 값 삽입

[1, 3, 5, 7, 9]

[실행결과]

0번 인덱스에 1 삽입

- 리스트 확장 : extend(list)

```
>>> a = ["카카오톡", "페이스북", "인스타그램"]
```

```
>>> a.extend(["트위터", "라인"])
```

```
>>> a
```

```
['카카오톡', '페이스북', '인스타그램', '트위터', '라인']
```

a리스트 뒤에 연결, 확장됨



리스트 관련 메서드 함수

- 리스트에 요소 추가 : append(값)

[소스코드] 5-6.py

```
future=[]  
future.append("AI")  
future.append("빅데이터")  
future.append("자율주행")  
future.append("사물인터넷")
```

future 리스트에 값 삽입

```
print("미래 기술 핵심분야:")  
print(future, sep=", ")
```

출력할 때 '와 공백 삽입

[실행결과]

미래 기술 핵심분야:
['AI', '빅데이터', '자율주행', '사물인터넷']



리스트 정렬하기

- 리스트 정렬하기 : `sort()`
- 리스트의 항목을 순서대로 정렬 후 재저장
- 기본 정렬 : 오름차순 정렬
- 역순 정렬 : `reverse()` 이용, 내림차순 정렬

[소스코드] 5-8.py

```
future=[]  
future.append("AI")  
future.append("빅데이터")  
future.append("자율주행")  
future.append("사물인터넷")
```

```
print("미래 기술 핵심분야:")  
print(future, sep=", ")
```

```
print("정렬후==>:")  
future.sort()  
print(future, sep=", ")
```

future 리스트 내용을 오름차순 정렬

[실행결과]

```
미래 기술 핵심분야:  
['AI', '빅데이터', '자율주행', '사물인터넷']  
정렬후==>:  
['AI', '빅데이터', '사물인터넷', '자율주행']
```



리스트 요소 제거

- 리스트 요소 제거
 - `del 리스트[x]` : 리스트의 x번째 요소 값 삭제
 - `remove(값)` : 리스트에서 값을 찾아서 삭제
 - `pop()` : 리스트의 마지막 값을 삭제, 삭제한 값 반환
 - `clear()` : 리스트의 모든 값 삭제



따라 해보기 : 리스트 요소 제거

- 리스트 요소 제거

[소스코드] 5-9.py

```
rainbow = ["빨", "주", "노", "초", "파", "남", "보"]  
print("원본==>", rainbow, "\n")  
#del을 이용하여 삭제하기  
del rainbow[0] 0번 요소 값 삭제  
print("del을 이용한 삭제 ==>", rainbow, "\n")  
#remove()을 이용하여 삭제하기  
rainbow.remove("주") "주" 내용 값 검색하여 삭제  
print("remove을 이용한 삭제 ==>", rainbow, "\n")  
#pop()을 이용하여 삭제하기  
rainbow.pop() 맨 뒤의 요소 값 삭제  
print("pop을 이용한 삭제 ==>", rainbow, "\n")  
#clear()을 이용하여 삭제하기  
rainbow.clear() 전체 요소 값 삭제  
print("clear를 이용한 삭제 ==>", rainbow, "\n")
```

[실행결과]

```
원본==> ['빨', '주', '노', '초', '파', '남', '보']  
del을 이용한 삭제 ==> ['주', '노', '초', '파', '남', '보']  
remove을 이용한 삭제 ==> ['노', '초', '파', '남', '보']  
pop을 이용한 삭제 ==> ['노', '초', '파', '남']  
clear를 이용한 삭제 ==> []
```



5.2 튜플(tuple)

- 튜플(tuple)
 - 소괄호 ()안로 생성, () 생략 가능
 - 읽기 전용으로 저장할 때 사용
 - 문자열, 정수, 실수, 리스트, 튜플, 함수 등 가능
- 튜플(tuple) 만들기

형식

튜플 = (값,값,값)

튜플 = 값,값,값

```
>>> tuple1 = (1, 3, 5, 7, 9)
```

```
>>> tuple2 = 1, 3, 5, 7, 9
```

```
>>> tuple1
```

```
(1, 3, 5, 7, 9)
```

```
>>> tuple2
```

```
(1, 3, 5, 7, 9)
```




5.2 튜플(tuple)

- 요소가 한 개 있는 튜플(tuple) 만들기

형식

튜플 = (값,)

튜플 = 값,

끝에 콤마(,) 기입

```
>>> tuple3 = (10,)
>>> tuple3
(10,)
>>> tuple3 = 10
>>> tuple3
10
```

- 문자열, 리스트, 튜플 등 가능

```
>>> tuple3 = ("자바", "파이썬", "자바스크립트", "C언어")
```

```
>>> tuple4 = ((1,3,5,7,9), [2,4,6,8,10])
```

- 튜플의 접근 : 리스트와 동일

```
>>> a = (1,3,5,7,9)
```

```
>>> a[0]
```

```
1
```

```
>>> a[2]
```

```
5
```

```
>>> a = (1,3,5,7,9)
```

```
>>> a[0]= 2
```

튜플은 읽기 전용, 요소 값 변경할 수 없음. Error 발생

TypeError: 'tuple' object does not support item assignment



5.3 딕셔너리(dictionary)

- 딕셔너리(dictionary)
- 중괄호 {}로 묶어서 구성
- 키(key)와 값(value)의 순서쌍을 갖는 자료형

형식

딕셔너리 변수 = {Key1:Value1, Key2:Value2, Key3:Value3, ...}

- 딕셔너리(dictionary) 만들기 예제

```
a = {'Spring':'봄', 'Summer':'여름', 'Autumn':'가을', 'Winter':'겨울'}
```

```
b = {1:'VIP', 2:'골드', 3:'일반'}
```

```
c = {'회원등급':['VIP','골드','회원']}
```



5.3 딕셔너리(dictionary)

- 딕셔너리(dictionary) 쌍 추가

[소스코드] 5-10.py

```
customer = { } ← 공백 딕셔너리 생성
```

```
customer['name'] = '김사랑'
```

```
customer['job'] = '회사원'
```

```
customer['phone'] = '010-2345-9876'
```

```
print(customer)
```

키

값

[실행결과]

```
{'name': '김사랑', 'job': '회사원', 'phone': '010-2345-9876'}
```

[소스코드] 5-11.py

- 딕셔너리 요소 제거

```
customer = { }
```

```
customer['name'] = '김사랑'
```

```
customer['job'] = '회사원'
```

```
customer['phone'] = '010-2345-9876'
```

```
print("원본 ==>", customer)
```

```
del customer['job'] ← 키 'job'을 갖는 요소값 삭제
```

```
print("삭제 후 ==>", customer)
```

[실행결과]

```
원본 ==> {'name': '김사랑', 'job': '회사원', 'phone': '010-2345-9876'}
```

```
삭제 후 ==> {'name': '김사랑', 'phone': '010-2345-9876'}
```



딕셔너리 메서드 : 요소 항목 접근

- 딕셔너리명['key']
 - 키가 존재하지 않으면 Error 발생

```
>>> menu = {1: '김밥', 2: '라면', 3: '불고기'}  
>>> menu[2]  
'라면'
```

- get('key')
 - 딕셔너리명['key']와 동일
 - 키가 존재하지 않으면 아무것도 출력하지 않음

```
>>> menu = {1: '김밥', 2: '라면', 3: '불고기'}  
>>> menu.get(2)  
'라면'
```




딕셔너리 메서드 : 반환하기

- key 목록 반환하기 : keys()
 - 해당 딕셔너리에 존재하는 모든 key 값 반환

```
>>> nation={"한국":"KOREA", "미국":"USA", "중국":"CHINA"}
```

```
>>> nation.keys()
```

nation 딕셔너리의 모든 키 반환

함께
표시됨

```
dict_keys(['한국', '미국', '중국'])
```

- 값 반환하기 : values()

```
>>> nation={"한국":"KOREA", "미국":"USA", "중국":"CHINA"}
```

```
>>> nation.values()
```

nation 딕셔너리의 모든 값 반환

함께
표시됨

```
dict_values(['KOREA', 'USA', 'CHINA'])
```

- 튜플 형태로 목록 반환하기 : items()
 - 해당 딕셔너리에 존재하는 모든 키(key) 와 값(values)의 순서쌍 반환

```
>>> nation={"한국":"KOREA", "미국":"USA", "중국":"CHINA"}
```

```
>>> nation.items()
```

nation 딕셔너리의 모든 키와 값 반환

함께
표시됨

```
dict_items([('한국', 'KOREA'), ('미국', 'USA'), ('중국', 'CHINA')])
```



한영 사전 만들기

- 실행 결과를 참고하여 조건에 맞게 프로그램을 작성해보기
 - animal_dic 딕셔너리를 공백 딕셔너리로 만들기 `animal_dic = {}`
 - animal_dic 딕셔너리에 다음과 같은 데이터를 키와 값으로 추가 저장하기

```
animal_dic['호랑이'] = 'Tiger'
animal_dic['사자'] = 'Lion'
animal_dic['코끼리'] = 'Elephant'
animal_dic['토끼'] = 'Rabbit'
animal_dic['거북이'] = 'Turtle'
```
 - 다음 [실행결과]와 같은 결과가 나오도록 코딩 해보기
 - 한글로 된 키(key)를 입력하면 영문으로 번역된 값(value)이 출력되도록 하기
 - 입력 => 키(key) : 호랑이, 사자, 코끼리, 토끼, 거북이
 - 출력 => 값(value) : Tiger, Lion, Elephant, Rabbit, Turtle

[실행결과]

```
----- 동물 한영 사전입니다. -----
호랑이, 사자, 코끼리, 토끼, 거북이 중 하나를 입력하세요:사자
사자 영어 단어는
==> Lion 입니다
```



한영 사전 만들기 프로그램

```
animal_dic = {}
```

```
animal_dic['호랑이'] = 'Tiger'
```

```
animal_dic['사자'] = 'Lion'
```

```
animal_dic['코끼리'] = 'Elephant'
```

```
animal_dic['토끼'] = 'Rabbit'
```

```
animal_dic['거북이'] = 'Turtle'
```

```
print("----- 동물 한영사전입니다. -----")
```

```
word = input("호랑이, 사자, 코끼리, 토끼, 거북이 중 하나를 입력하세요:")
```

```
print(word, "영어 단어는")
```

```
print("==>", animal_dic[word], "입니다")
```

[실행결과]

----- 동물 한영 사전입니다. -----

호랑이, 사자, 코끼리, 토끼, 거북이 중 하나를 입력하세요:사자

사자 영어 단어는

==> Lion 입니다



마무리

- 이번 차시수업에서는 리스트관련 메서드 함수, 튜플, 딕셔너리에 대해 학습하였습니다.