

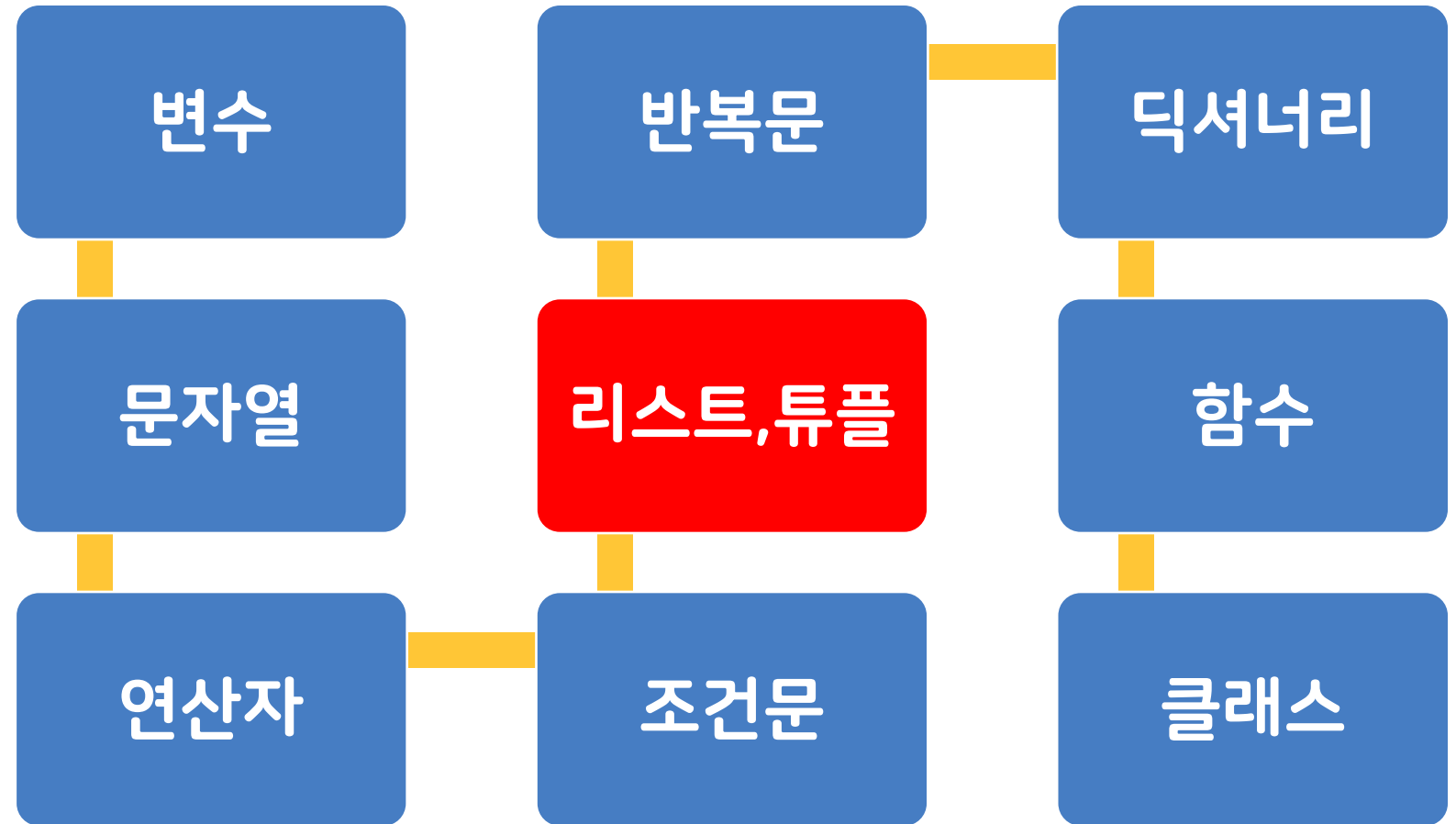


스마트인재개발원
Smart Human Resources Development

최성우 연구원



수업 진행방향





학습목표

- 리스트와 튜플에 대해 이해 할 수 있다.
- 리스트 함수를 활용 할 수 있다.

리스트(list) 란?

- 파이썬의 자료구조 형태중 하나
- 순서가 있는 수정가능한 객체의 집합
- 대괄호([])로 작성되어지며, 리스트 내부의 값은 콤마(,)으로 구분
- 추가, 수정, 삭제 가능

리스트명 = [요소1, 요소2, 요소3, ...]

```
a = []
```

```
b = [1, 2, 3]
```

```
c = ['My', 'name', 'is', 'MH']
```

```
d = [1, 2, 'My', 'name']
```

```
e = [1, 2, ['My', 'name']]
```

인덱싱(indexing)

- 무엇인가를 '가리킨다'는 의미 : 하나의 인덱스에 있는 값을 표시

슬라이싱(Slicing)

- 무엇인가를 '잘라낸다'는 의미 : 일정 범위의 인덱스에 있는 값을 표시

리스트[인덱스]

- 인덱스에 위치한 값 반환

```
list1 = [2, 5, 7, 9, 10]
print(list1[0])
print(list1[3])
print(list1[2]+list1[-1])
```

```
list2 = [1, 2, 3, ['a', 'b', 'c']]
```

- 리스트 인덱싱을 사용하여 변수 temp에 ['a', 'b', 'c']를 저장하고 출력하시오.
- list2에서 문자 'b'만 뽑아서 출력하시오.

리스트[start 인덱스 : end 인덱스]

- start 인덱스부터 end 인덱스 바로 전까지 값 반환 ($\text{start} \leq x < \text{end}$)

```
list3 = [0, 1, 2, 3, 4]  
list3[1:3]
```

```
[1, 2]
```

```
list3[:2]
```

```
[0, 1]
```

```
list3[3:]
```

```
[3, 4]
```

```
list3[3:4]
```

```
[3]
```

```
list4 = [1, 2, 3]  
list5 = [3, 4, 5, 6]  
list4 + list5
```

```
[1, 2, 3, 3, 4, 5, 6]
```

리스트.append(값)

- 맨 뒤에 값 추가

```
list5 = [0, 1, 2, 3, 4]  
list5.append(5)  
list5
```

```
[0, 1, 2, 3, 4, 5]
```

```
list5.append(6)  
list5
```

```
[0, 1, 2, 3, 4, 5, 6]
```

append() 실습

```
list_ = ['사과', '포도', ['수박', '멜론'], '복숭아', '딸기', '오렌지']
```

리스트 list_ 에 담긴 '사과', '수박', '오렌지' 를 인덱싱하여 각각 임의의 변수에 저장하고, 비어 있는 리스트 choice_list에 append 함수를 이용하여 추가한 후 아래와 같이 출력하시오.

1	choice_list
---	-------------

['사과', '오렌지', '수박']

리스트.insert(**인덱스**, **값**)

- 인덱스 위치에 값 추가

```
list5 = [0, 1, 2, 3, 4]
list5.insert(1, 5)
list5
```

```
[0, 5, 1, 2, 3, 4]
```

```
list5.insert(5, 6)
list5
```

```
[0, 5, 1, 2, 3, 6, 4]
```

insert() 실습

```
1 music_list = ["LOVE DIVE", "아이브"],  
2             ["TOMBOY", "(여자)아이들"],  
3             ["That That", "싸이"]]  
4 music_list
```

- music_list 인덱스가 1인 곳에
["사랑인가봐", "멜로망스"]를 추가하시오.
- music_list 인덱스 값이 3인 곳에
["봄여름가을겨울", "빅뱅"]를 추가하시오.

```
[['LOVE DIVE', '아이브'],  
 ['사랑인가봐', '멜로망스'],  
 ['TOMBOY', '(여자)아이들'],  
 ['봄여름가을겨울', '빅뱅'],  
 ['That That', '싸이']]
```

insert() 실습

정수 형태의 **인덱스 값**과 **노래 제목**,
가수명도 각각 입력 받은 후 변수에
저장하고, insert함수를 이용하여 리스트
music_list에 **새로운 값**을 추가하시오.

1	music_list
---	------------

```
[['LOVE DIVE', '아이브'],  
 ['사랑인가봐', '멜로망스'],  
 ['TOMBOY', '(여자)아이들'],  
 ['봄여름가을겨울', '빅뱅'],  
 ['That That', '싸이']]
```

1	index_value =	?
2	song =	?
3	singer =	?
4		
5	music_list.	?
6	music_list	

인덱스 입력 >> 4

노래 제목 >> 나의X에게

가수명 >> 경서

```
[['LOVE DIVE', '아이브'],  
 ['사랑인가봐', '멜로망스'],  
 ['TOMBOY', '(여자)아이들'],  
 ['봄여름가을겨울', '빅뱅'],  
 ['나의X에게', '경서'],  
 ['That That', '싸이']]
```

```
list6 = [0, 1, 2, 3, 4]  
list6
```

```
[0, 1, 2, 3, 4]
```

```
print("수정 전 :", list6[1])  
list6[1] = 7  
print("수정 후 :", list6[1])
```

```
수정 전 : 1  
수정 후 : 7
```

```
list6
```

```
[0, 7, 2, 3, 4]
```

```
list6[2:4] = 7
```

TypeError

<ipython-input-46-913ecdac98a2> in <module>
----> 1 list6[2:4] = 7

TypeError: can only assign an iterable

```
print(list6[2:4])  
list6[2:4] = [7]  
list6
```

```
[2, 3]
```

```
[0, 7, 7, 4]
```



```
array = [1, 2, 3, 4, 5]
```

- 리스트 array에 숫자 5를 7으로 바꾸시오.

```
[1, 2, 3, 4, 7]
```

← 하나의 값을 하나로 수정

- 리스트 array에 숫자 2를 리스트['a', 'b', 'c']로 바꾸시오.

```
[1, ['a', 'b', 'c'], 3, 4, 7]
```

← 하나의 값을 리스트로 수정

- 리스트 array에 4,7을 'd', 'e', 'f', 'g'로 바꾸시오.

```
[1, ['a', 'b', 'c'], 3, 'd', 'e', 'f', 'g']
```

← 여러 개의 값을 여러 개로 수정

- 리스트 array에 1을 'h', 'i', 'j' 로 바꾸시오

```
['h', 'i', 'j', ['a', 'b', 'c'], 3, 'd', 'e', 'f', 'g']
```

← 하나의 값을 여러 개로 수정

del 키워드 이용

```
list7 = [0, 1, 2, 3, 4, 5]  
del list7[1]  
list7
```

```
[0, 2, 3, 4, 5]
```

```
list7 = [0, 1, 2, 3, 4, 5]  
del list7[1:5]  
list7
```

```
[0, 5]
```

리스트.remove(값)

```
list7 = ['a', 'b', 'c', 'd', 'e']  
list7.remove('b')  
list7
```

```
['a', 'c', 'd', 'e']
```

```
list7.remove('b')
```

```
-----  
--  
ValueError  
t)  
<ipython-input-42-ac74ba81fef3  
----> 1 list7.remove('b')
```

```
ValueError: list.remove(x): x
```

리스트.sort()

- 리스트에 있는 값을 **오름차순**으로 정렬

```
list8 = [9, 77, 13, 51, 100, 3]  
list8
```

```
[9, 77, 13, 51, 100, 3]
```

```
list8.sort()  
list8
```

리스트.reverse()

- 리스트에 있는 값을 **역순**으로 뒤집음

```
list9 = [9, 77, 13, 51, 100, 3]  
list9
```

```
[9, 77, 13, 51, 100, 3]
```

```
list9.reverse()  
list9
```

리스트.sort() 와 리스트.reverse() 이용

- 리스트에 있는 값을 **내림차순**으로 **정렬**

```
list10 = [9, 77, 13, 51, 100, 3]  
list10
```

```
[9, 77, 13, 51, 100, 3]
```

```
list10.sort()  
list10
```

```
[3, 9, 13, 51, 77, 100]
```

```
list10.reverse()  
list10
```

```
[100, 77, 51, 13, 9, 3]
```

리스트.index()

- 찾고자 하는 값의 **위치 출력**

```
list11 = ['a', 'b', 'c', 'd', 'e', 'f']  
list11.index('c')
```

2

리스트.pop()

- 마지막 값을 **출력 후** 리스트에서 **제거**

```
list12 = ['a', 'b', 'c', 'd', 'e', 'f']  
list12.pop()
```

'f'

```
list12
```

```
['a', 'b', 'c', 'd', 'e']
```

len(**리스트**)

- 리스트의 값 개수 출력

```
list13 = [0, 1, 2]  
len(list13)
```

```
list14 = ['a', 'b', 'c', 'd', 'e', 'f']  
len(list14)
```

튜플(tuple)이란?

- 파이썬의 자료구조 형태중 하나
- 순서가 있는 집합
- 소괄호(**()**)로 작성되어지며, 튜플의 내부 값은 콤마(,)으로 구분
- 추가, 수정, 삭제 **불가능**

튜플명 = (요소1, 요소2, 요소3, ...)

```
a = ( )
```

```
b = (1, 2, 3)
```

```
c = ('My', 'name', 'is', 'MH')
```

```
d = (1, 2, 'My', 'name')
```

```
e = (1, 2, ('My', 'name'))
```

튜플[인덱스]

- 인덱스에 위치한 값 출력

```
tuple1 = (0, 1, 2, 3, ('a', 'b', 'c'), 5)
```

```
tuple1[2]
```

```
2
```

```
tuple1[4]
```

```
('a', 'b', 'c')
```

튜플[start 인덱스 : end 인덱스]

- start 인덱스부터 end 인덱스 바로 전까지 값 출력 ($\text{start} \leq x < \text{end}$)

```
tuple1 = (0, 1, 2, 3, ('a', 'b', 'c'), 5)
```

```
tuple1[1:3]
```

```
(1, 2)
```

```
tuple1[3:]
```

```
(3, ('a', 'b', 'c'), 5)
```

len(튜플)

- 튜플의 값 개수 출력

```
tuple1 = (0, 1, 2, 3, ('a', 'b', 'c'), 5)  
len(tuple1)
```

6

```
tuple2 = ('a', 'b', 'c', 'd', 'e', 'f')  
len(tuple2)
```

6

튜플은 추가, 수정, 삭제 **불가능**

```
tuple1 = (0, 1, 2, 3, ('a', 'b', 'c'), 5)
```

```
tuple1[0] = 3
```

```
-----  
--  
TypeError                                Traceback (most recent  
t)  
<ipython-input-53-5e0f22de5ab3> in <module>  
----> 1 tuple1[0] = 3  
  
TypeError: 'tuple' object does not support item assignment
```

공통점

- 타입과 상관 없이 일련의 **요소(Element)**들을 가질 수 있다.
- 요소의 순서를 관리한다.

차이점

- 리스트는 **가변적(mutable)**이며, 튜플은 **불변적(immutable)**
- 리스트는 요소의 **개수가 명확하지 않고 유동적일** 경우에 사용
- 튜플은 요소 개수를 **사전에 정확히 알고 있고 변동이 없을** 경우에 사용
(리스트에 비해 시스템 상 소모되는 메모리가 적고 속도가 빠름)

in : 찾고자 하는 값(x)이 포함 되어 있으면 True

not in : 찾고자 하는 값(x)이 포함되어 있지 않으면 True

in	not in
x in 문자열	x not in 문자열
x in 리스트	x not in 리스트
x in 튜플	x not in 튜플

```
str1 = "파이썬 최고"
```

```
"파이썬" in str1
```

True

```
"파이썬" not in str1
```

False

```
list1 = [77, 38, 10]
```

```
33 in list1
```

False

```
33 not in list1
```

True

다음과 같이 문자열에 사용자가 입력한 글자가 몇 개 포함되어 있는지 출력하시오.

```
s = "Hi, My name is SeongWoo"
```

```
a = input("검색할 문자를 입력하세요 : ")  
b = s.
```

검색할 문자를 입력하세요 : hi
hi는 들어있지 않네유...

검색할 문자를 입력하세요 : i
i는 2번 들어있네유!



다음시간에는?

반복문