

7. 파이썬 자료형 - 리스트

Made By. 김규일

- 파이썬의 리스트는 다른 언어의 배열과 같다.
- 다른 언어와 다른 점은 리스트의 각 요소들의 데이터 타입이 달라도 된다는 점이다.
- 따라서 파이썬의 리스트는 정말 많이 사용하는 자료형 중 하나이니 꼭 익히길 바란다.

```
>>> a = []
>>> b = [1, 2, 3]
>>> c = ['Life', 'is', 'too', 'short']
>>> d = [1, 2, 'Life', 'is']
>>> e = [1, 2, ['Life', 'is']]
```

- 리스트의 생김새는 위와 같다. d, e 처럼 리스트 안의 각 요소들은 같은 데이터 타입이 아니어도 된다.

리스트 인덱싱과 슬라이싱

- 리스트의 인덱싱과 슬라이싱 또한 문자열과 비슷하다.
- 리스트의 인덱싱 또한 0부터 시작하고 마지막 값은 -1부터 시작한다.
- 중첩된 리스트의 인덱싱은 다음과 같다.

```
>>> a = [1, 2, 3, ['a', 'b', 'c']]
>>> a[0]
1
>>> a[-1]
['a', 'b', 'c']
>>> a[3]
['a', 'b', 'c']
```

- 위의 리스트의 형태는 인덱스 0에는 1, 인덱스 1에는 2, 인덱스 2에는 3, 인덱스 4에는 리스트가 들어있는 형태이다.

- 즉 a[3]처럼 인덱스 3을 꺼내게 된다면 3번째 요소인 리스트 ['a', 'b', 'c']가 나오게 된다.
- 또한 2차원 배열처럼 활용하여 아래처럼 인덱싱을 할 수 있다.

```
>>> a[-1][0]
'a'
>>> a[-1][1]
'b'
>>> a[-1][2]
'c'
```

- 리스트의 슬라이싱 또한 문자열의 슬라이싱과 비슷하다.

```
>>> a = [1, 2, 3, 4, 5]
>>> a[0:2]
[1, 2]

>>> a = "12345"
>>> a[0:2]
'12'

>>> a = [1, 2, 3, 4, 5]
>>> b = a[:2]
>>> c = a[2:]
>>> b
[1, 2]
>>> c
[3, 4, 5]
```

리스트 연산하기

- 리스트 또한 + 기호를 통해 연결할 수 있고 * 기호를 통해 반복할 수 있다.
- 리스트 더하기

```
>>> a = [1, 2, 3]
>>> b = [4, 5, 6]
>>> a + b
[1, 2, 3, 4, 5, 6]
```

- 리스트 반복하기

```
>>> a = [1, 2, 3]
>>> a * 3
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

- 리스트 길이 구하기

```
>>> a = [1, 2, 3]
>>> len(a)
3
```

리스트의 수정과 삭제

- 리스트 값 수정하기

```
>>> a = [1, 2, 3]
>>> a[2] = 4
>>> a
[1, 2, 4]
```

인덱스 번호를 이용하여 수정할 수 있다.

- del 함수를 이용해서 요소 삭제하기

```
>>> a = [1, 2, 3]
>>> del a[1]
>>> a
[1, 3]
```

인덱스 번호와 del 함수를 이용하여 삭제

```
>>> a = [1, 2, 3, 4, 5]
>>> del a[2:]
>>> a
[1, 2]
```

슬라이싱과 del 함수를 이용하여 삭제

리스트 관련 함수들

- 리스트에 요소 추가(append)

```
>>> a = [1, 2, 3]
>>> a.append(4)
>>> a
[1, 2, 3, 4]
```

append 함수는 리스트의 맨 마지막에 요소를 추가한다.

```
>>> a.append([5,6])
>>> a
[1, 2, 3, 4, [5, 6]]
```

또한 리스트 안의 각 요소에는 데이터 타입이 달라도 되기 때문에 위처럼 다른 자료형을 추가해도 된다.

- 리스트 정렬(sort)

```
>>> a = [1, 4, 3, 2]
>>> a.sort()
>>> a
[1, 2, 3, 4]
```

숫자를 오름차순으로 정렬한다.

```
>>> a = ['a', 'c', 'b']
>>> a.sort()
>>> a
['a', 'b', 'c']
```

문자의 경우 알파벳 순서, 한글의 경우 ㄱㄴㄷ 순서로 정렬

- 리스트 뒤집기(reverse)

```
>>> a = ['a', 'c', 'b']
>>> a.reverse()
>>> a
['b', 'c', 'a']
```

- 인덱스 반환(index)

```
>>> a = [1,2,3]
>>> a.index(3)
2
>>> a.index(1)
0
```

index 함수 안의 값이 리스트에 존재하면 그 위치를 반환한다. 값이 없다면 오류를 발생

- 리스트에 요소 삽입(insert)

```
>>> a = [1, 2, 3]
>>> a.insert(0, 4)
>>> a
[4, 1, 2, 3]
```

insert(a, b) 는 a의 위치에 b를 삽입

- 리스트 요소 제거(remove)

```
>>> a = [1, 2, 3, 1, 2, 3]
>>> a.remove(3)
>>> a
[1, 2, 1, 2, 3]
```

remove() 함수 안의 값을 삭제. 단, 첫 번째로 나오는 값 하나를 삭제

- 리스트 요소 끄집어내기(pop)

```
>>> a = [1,2,3]
>>> a.pop()
3
>>> a
[1, 2]
```

맨 마지막 요소를 꺼내고 그 값은 삭제한다.

- 리스트에 포함된 요소 개수 세기(count)

```
>>> a = [1,2,3,1]
>>> a.count(1)
2
```

```
## count() 함수 안의 값의 개수를 센다.
```

- 리스트 확장(extend)

```
>>> a = [1,2,3]
>>> a.extend([4,5])
>>> a
[1, 2, 3, 4, 5]
>>> b = [6, 7]
>>> a.extend(b)
>>> a
[1, 2, 3, 4, 5, 6, 7]
```

```
## extend() 안의 리스트를 연결한다. 다만 extend() 안에는 리스트만 올 수 있다.
```