

SINCE 1993

Powered By AI

S

S

F

튜플, 세트



# 리스트 복습

리스트란, 하나의 자료형에 여러 변수의 값을 저장하는 자료형.

리스트의 선언

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



# 튜플의 정의

**튜플이란,  
여러 값을 하나의 변수에 저장할 수 있는 자료형  
하지만, 변경이 불가능하기에 (튜플 == 변경을 못하는 리스트)**



# 튜플의 선언

```
>>> t1 = ()  
>>> t2 = (1, )  
>>> t3 = (1, 2, 3)  
>>> t4 = 1, 2, 3  
>>> t5 = ('a', 'b', ('ab', 'cd'))
```

- 튜플은 소괄호를 사용하지만, 소괄호를 빼도 선언 가능함.
- 빈 튜플의 선언은 ()만 사용하지만 값이 한 개만 있는 경우는 꼭 (1, ) 처럼 값 뒤에 콤마를 붙여야 함.
- 튜플은 값을 삭제하거나 수정할 수 없기 때문에 수정과 삭제를 시도하면 오류 발생함.



# 튜플 다루기

```
t1 = (1, 2, 3)
t2 = (4, 5)

print(t1[0])      # 인덱싱: 1
print(t1[1:])     # 슬라이싱: (2, 3)
print(t1 + t2)    # 덧셈: (1, 2, 3, 4, 5)
print(t1 * 2)     # 곱셈: (1, 2, 3, 1, 2, 3)
```

리스트와 같이 인덱싱, 슬라이싱, 덧셈, 뺄셈 모두 가능.  
하지만 값을 변경하는 `append()`, `remove()`, `sort()` 등의 함수 사용불가



# 세트의 정의

세트란,  
중복을 허용하지 않고, 순서가 없는 파이썬의 자료형.  
수학의 " 집합 " 개념을 따르며, 집합연산을 지원한다.



# 세트의 선언

```
>>> s = {1, 2, 3, 4, 5, 2, 3}
>>> s
{1, 2, 3, 4, 5} #중복불허
```

- 세트는 {} 중괄호를 사용해서 선언함.
- 세트는 순서가 없기에 s[2]같은 인덱싱을 지원하지 않음.
- 중복을 허용하지 않아 2, 3 이 삭제됨.



# 세트의 활용

```
>>> s1 = set([1,2,3])
>>> l1 = list(s1)
>>> l1
[1, 2, 3]
>>> l1[0]
1
>>> t1 = tuple(s1)
>>> t1
(1, 2, 3)
>>> t1[0]
1
```

- 세트도 자료형의 집합이기에 리스트나 튜플로 변경 가능
- 하지만, 순서가 없어 리스트나 튜플로 바뀐 후의 순서는 예측 불가  
=> 필요시 정렬해서 바꾸기





# 세트의 집합연산

```
a = {1, 2, 3}
b = {3, 4, 5}

print(a | b)    # 합집합: {1, 2, 3, 4, 5}
print(a & b)    # 교집합: {3}
print(a - b)    # 차집합: {1, 2}
print(a ^ b)    # 대칭차집합: {1, 2, 4, 5}
```

- 위의 기호로 합집합, 교집합, 차집합, 대칭차집합 등의 집합연산 가능



# 집합 관련 함수들

값 1개 추가하기  
(add)

```
>>> s1 = set([1, 2, 3])  
>>> s1.add(4)  
>>> s1  
{1, 2, 3, 4}
```

값 여러 개 추가하기  
(update)

```
>>> s1 = set([1, 2, 3])  
>>> s1.update([4, 5, 6])  
>>> s1  
{1, 2, 3, 4, 5, 6}
```

특정 값 제거하기  
(remove)

```
>>> s1 = set([1, 2, 3])  
>>> s1.remove(2)  
>>> s1  
{1, 3}
```



# 리스트 vs 튜플 vs 세트

	리스트	튜플	세트
기호	[ ]	( )	{ }
순서	있음	있음	없음
인덱싱/슬라이싱	가능	가능	불가능
중복	가능	가능	불가능
변경	가능	불가능	가능



SSF Powered By  
AI

실습