

Tuple, 튜플

1. 튜플은 ()으로 둘러쌉니다. (리스트는 [])
2. 튜플은 값을 변경할 수 없(immutable)습니다. (리스트는 mutable)

사용방법

```
tuple_data = (1, 2, 3)
print(tuple_data) # (1, 2, 3)
print(type(tuple_data)) # Type -> <class 'tuple'>

# Index 접근
print(tuple_data[0]) # 1
print(tuple_data[1]) # 2

# tuple_data Size(길이)
print(len(tuple_data)) # 3

# tuple 슬라이싱
print(tuple_data[:2]) # (1, 2)
print(tuple_data[0:1]) # (1,)

# `+` 연산
print(tuple_data + (4, 5, 6)) # (1, 2, 3, 4, 5, 6)

# `*` 연산
print(tuple_data * 2) # (1, 2, 3, 1, 2, 3)

tuple_data[0] = 4 # TypeError: 'tuple' object does not support item assignment
print(tuple_data)
```

튜플은 값을 수정하거나 삭제할 수 없기에,

튜플은 값의 변화가 없어야할 때 주로 사용됩니다.

반면, 리스트는 값이 추가되고 변화가 될 수 있을 때 사용됩니다.

이를 염두해두고 사용하시면 됩니다.

심화1]

Packing(패킹)과 Unpacking(언패킹)

패킹은 포장의 의미이고, 언패킹은 포장을 뜯는(언박싱) 의미입니다.

- 패킹 : 하나의 변수에 여러 개의 값을 넣는 것
- 언패킹 : 패킹된 변수에서 여러 개의 값을 꺼내는 것

언패킹에서는 풀어주는 갯수와 받는 갯수가 같아야합니다.

```
packing_data = 1, 2, 3

print(type(packing_data)) # <class 'tuple'>
print(packing_data) # (1, 2, 3)

unpacking_data1, unpacking_data2, unpacking_data3 = packing_data

print(type(unpacking_data1)) # <class 'int'>
print(type(unpacking_data2)) # <class 'int'>
print(type(unpacking_data3)) # <class 'int'>
print(unpacking_data1) # 1
print(unpacking_data2) # 2
print(unpacking_data3) # 3
```