

Day 4 - Python Collection Type (Tuple)

- 여러 가지 데이터 타입의 값을 저장할 수 있는, 변경 불가능(immutable)한 순서가 있는 컬렉션이다.

@@ 단수일 경우 콤마를 넣어야지 튜플로 인식을 한다.

```
tu1 = (1)
print(tu1)
print(type(tu1))    # 요소가 1개일 경우, 콤마를 안넣을시 int로 인식을 한다 계산실패의 괄호로 인식

tu1 = (1,)
print(tu1)
print(type(tu1))    # 튜플로 인식된다.
```

```
1
<class 'int'>
(1,)
<class 'tuple'>
```

리스트 → 튜플 변환:

```
tu3 = tuple([1, 3, 5, 7])
print(tu3)
print(type(tu3))
```

```
(1, 3, 5, 7)
<class 'tuple'>
```

튜플 → 리스트 변환:

```
li3 = list(tu3)
print(li3)
print(type(li3))
```

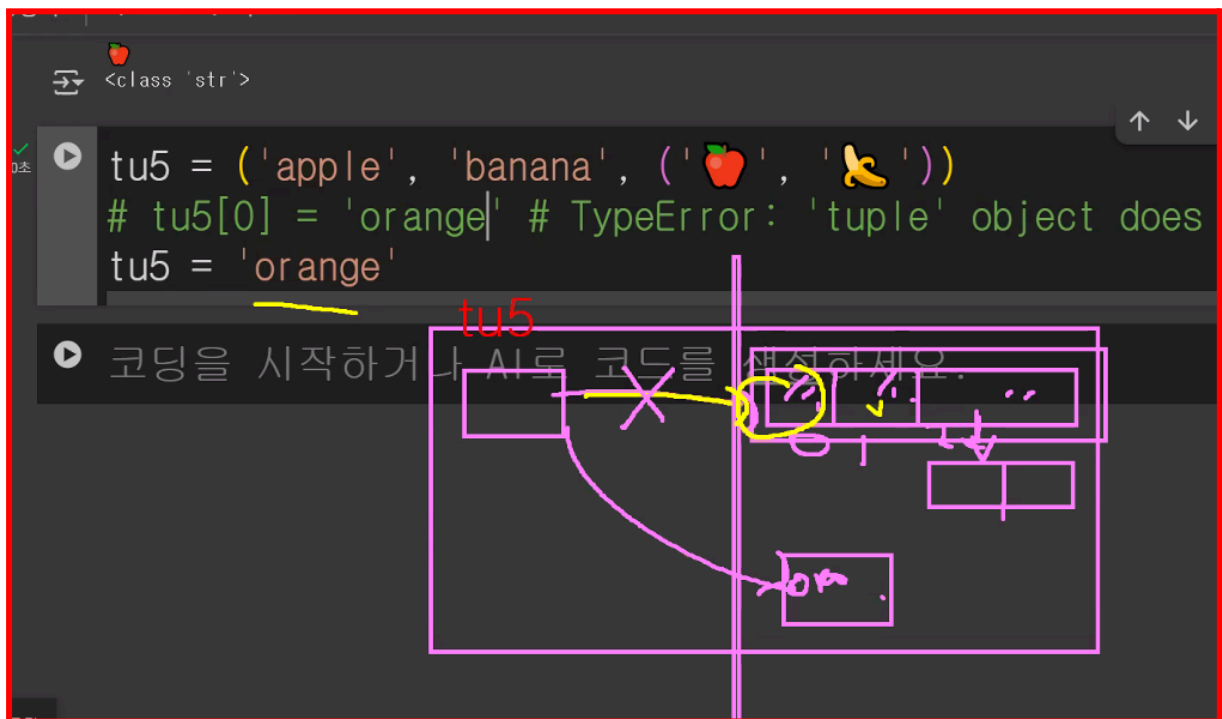
```
[1, 3, 5, 7]
<class 'list'>
```

기본적으로 여러 숫자를 괄호가 없는 형태로 놔두면 튜플로 인식된다:

```
tu4 = 1, 3, 5, 7
print(tu4)
print(type(tu4))
```

```
(1, 3, 5, 7)
<class 'tuple'>
```

리스트와 다르게 할당된 객체의 변경이 불가능하다.



튜플 안에 리스트를 삽입하는것도 가능하다:



- 또한 튜플안에서 객체를 바꾸는것은 안되지만
- 튜플 안에 있는 리스트는 변경이 가능하다.

인덱싱, 슬라이싱 둘다 리스트처럼 가능하다

```
# 튜플의 슬라이싱은 결과도 튜플로 반환
tu6 = (1, 2, 'apple', 'banana')
print(tu6[1:])
print(tu6[1:3])
```

```
(2, 'apple', 'banana')
(2, 'apple')
```

```
tu5 = ('apple', 'banana', ('🍏', '🍌'))
print(tu5)
print(type(tu5))

print(tu5[0])
print(type(tu5[0]))
print(tu5[2][0])
print(type(tu5[2][0]))
```

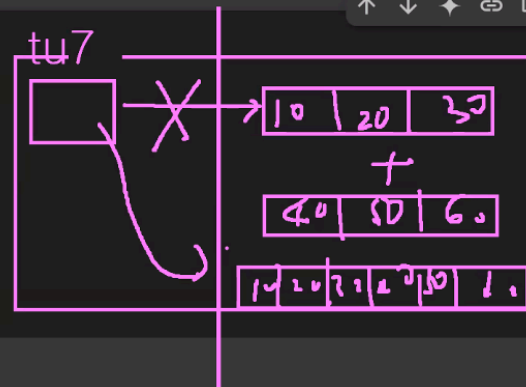
튜플간의 연산

튜플 간의 + 연산으로 리스트를 결합할 수 있고, * 연산으로 리스트를 반복할 수 있습니다.

```
tu7 = (10, 20, 30)
tu8 = (40, 50, 60)
print(tu7 + tu8)
print(tu8 + tu7)

tu7 = tu7 + (40, 50, 60)
print(tu7)
```

```
(10, 20, 30, 40, 50, 60)
(40, 50, 60, 10, 20, 30)
(10, 20, 30, 40, 50, 60)
```



- 튜플 간의 + 연산은
- 메모리에 각각 할당한 후 더해서 새로운 튜플을 메모리에 올린다
- 그리고 다시 재할당한다.

멤버십 테스트

✓ 5. 멤버십 테스트

in 연산자를 사용하여 튜플에 특정 값이 있는지 확인할 수 있습니다.

✓
0초



```
# 컬렉션에 포함되어 있는 요소인지 확인
print(2 in [2, 4, 6, 8, 10])
print(2 in (2, 4, 6, 8, 10))
print(7 in (2, 4, 6, 8, 10))
print("하이" in ("하이", "4", "6", "8", "10"))
```



```
True
True
False
True
```

- 리스트, 튜플 둘다 가능하다.
- 문자열, integer 둘다 가능하다.

sorted() 함수로 튜플 형태 정렬:

✓ 6. 튜플의 정렬

튜플은 sort() 메서드를 제공하지 않습니다. sorted() 함수를 사용합니다.



```
tu9 = (10, 30, 100, 90, 50)
# tu9.sort() # 튜플은 sort 메서드가 존재하지 않음
result = sorted(tu9, reverse = True) # 내림차순하여 리스트로 반환
print(result)
print(tu9) # 변경되지 않았음

# 다시 튜플 형태로 저장
result = tuple(result)
print(result)
```



```
[100, 90, 50, 30, 10]
(10, 30, 100, 90, 50)
(100, 90, 50, 30, 10)
```

- sorted 함수 사용시 리스트 형태로 변환
- 다시 tuple() 로 튜플 형태로 변환가능