

# 파이썬 프로그래밍

19차시

항목의 순서나 내용을  
수정할 수 없는 튜플



## ⚠ 학습개요

- ... 튜플의 이해
- ... 튜플 생성
- ... 튜플에도 적용
  - +, \*
  - 내장 함수 sorted
  - 튜플 자체 제거: del
- ... 프로젝트 lab1, lab2

## ⚠ 학습목표

- ... 튜플은 수정할 수 없다는 특징을 설명할 수 있다.
- ... 튜플을 생성할 수 있다.
- ... 튜플을 연결, 반복할 수 있다.
- ... 튜플을 정렬해 리스트를 생성할 수 있다.

## Chapter 1.

# 튜플의 이해

P Y T H O N   P R O G R A M M I N G

## ⚠ 수정할 수 없는 항목의 나열인 튜플

### + 튜플은 리스트와 달리 항목의 순서나 내용의 수정이 불가능

- 콤마로 구분된 항목 표현
- 각각의 항목은 정수, 실수, 문자열, 리스트, 튜플 등 제한이 없음
  - 튜플은 괄호(parentheses) (...) 사이에 항목을 기술
  - 괄호는 생략 가능

```
singer = ('BTS', '볼빨간사춘기', 'BTS', '블랙핑크', '태연')  
credit = ([2020, 1, 18], [2020, 2, 17])  
space = '밤', '낮', '해', '달'
```

## ⚠ 수정할 수 없는 항목의 나열인 튜플

### + 빈 튜플 생성

- `()`
- 함수 `tuple()`

```
>>> empty = ()  
>>> type(empty)  
<class 'tuple'>  
>>> print(empty)  
()
```

```
>>> empty2 = tuple()  
>>> print(empty2)  
()
```



## ⚠ 수정할 수 없는 항목의 나열인 튜플

+ 항목이 하나인 튜플을 표현할 때는 마지막에 **콤마**를 반드시 붙임

```
>>> inta = 1
>>> tupa = 1,
>>> print(tupa)
(1,)
```

## Chapter 2.

# 튜플 생성

P Y T H O N   P R O G R A M M I N G



## ⚠ K-pop 가수와 곡으로 구성된 튜플의 참조

[코딩실습] K-pop 가수와 곡으로 구성된 튜플의 참조

난이도 기본

```
1. singer = ('BTS', '볼빨간사춘기', 'BTS', '블랙핑크', '태연')
2. song = ('작은 것들을 위한 시', '나만, 봄', '소우주', 'Kill This Love', '사계')
3. print(singer)
4. print(song)
5.
6. print(singer.count('BTS'))
7. print(singer.index('볼빨간사춘기'))
8. print(singer.index('BTS'))
9. print()
10.
11. for _ in range(len(singer)):
12.     print('%s: %s' % (singer[_], song[_]))
```

## ⚠ 일상 코딩: 한 글자의 단어로 슬라이싱 이해하기

**주의** print()는 빈 줄을 삽입하기 위해 사용한다.

**결과**

```
('BTS', '볼빨간사춘기', 'BTS', '블랙핑크', '태연')  
( '작은 것들을 위한 시', '나만, 봄', '소우주', 'Kill This Love', '사계')  
2  
1  
0
```

BTS: 작은 것들을 위한 시  
볼빨간사춘기: 나만, 봄  
BTS: 소우주  
블랙핑크: Kill This Love  
태연: 사계

## Chapter 3.

# 튜플에도 적용

P Y T H O N   P R O G R A M M I N G

## ⚠️ + \* sorted() del

+ +

- 튜플을 연결한 새로운 튜플을 반환

+ \*

- 항목이 횟수만큼 반복된 튜플을 반환

```
>>> kpop = ('BTS', '블랙핑크')
>>> num = (7, 4)
>>> print(kpop + num)
('BTS', '블랙핑크', 7, 4)
>>>
>>> days = ('1학기', '2학기')
>>> print(days * 4)
('1학기', '2학기', '1학기', '2학기', '1학기', '2학기', '1학기', '2학기')
```

## ⚠️ + \* sorted() del

### + 내장 함수 sorted( )

- 튜플 항목의 순서를 정렬한 리스트를 반환

```
>>> fruit = ('사과', '귤', '복숭아', '사과', '파인애플')
>>> tup = sorted(fruit)
>>> print(type(tup), tup)
<class 'list'> ['귤', '복숭아', '사과', '파인애플']
>>>
>>> print(sorted(fruit, reverse=True))
['파인애플', '사과', '복숭아', '귤']
>>> print(fruit)
('사과', '귤', '복숭아', '파인애플')
```

## ⚠️ + \* sorted() del

### + 문장 del

- 튜플 변수 자체의 제거

```
>>> kpop = ('BTS', '장범준', '블랙핑크', '잔나비')
>>> del kpop
>>> print(kpop)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'kpop' is not defined
```



## ⚠ 요일에 영어 단어를 튜플로 생성해 출력

[코딩실습] 영어 요일 단어로 구성된 튜플 만들기

난이도 기본

```
1. day1 = ('monday', 'tuesday', 'wednesday')
2. day2 = ('thursday', 'friday', 'saturday')
3. # ('sunday')로 하면 튜플이 아니고 문자열
4. day3 = ('sunday', )
5.
6. day = day1 + day2 + day3
7. print(type(day))
8. print(day)
9.
10. day = day1 + day2 + day3 * 3
11. print(day)
```

## ⚠ 요일에 영어 단어를 튜플로 생성해 출력

주의

4번 줄의 콤마는 반드시 필요하다.

결과

```
<class 'tuple'>
('monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday')
('monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday',
'sunday', 'sunday')
```

## Chapter 4.

# 프로젝트 lab1, lab2

P Y T H O N   P R O G R A M M I N G

## ⚠ 프로젝트 Lab1

스포츠 종목과 팀원 수를 리스트로 적절히 출력

난이도 실전

다음과 같이 스포츠 종목과 팀원 수로 구성된 리스트로 적절히 출력하는 프로그램을 작성하자.

```
sports = ['축구', '야구', '농구', '배구']  
# 위 종목에 대응하는 팀원 수를 항목으로 구성  
num = [11, 9, 5, 6]
```

실행결과

축구: 11명 야구: 9명 농구: 5명 배구: 6명

또한 위 두 리스트 sports와 num으로 다음 중첩된 리스트를 만들어 종목과 팀원 수를 적절히 출력하는 코드도 포함 시키자.

```
[['축구', '야구', '농구', '배구'], [11, 9, 5, 6]]
```

마지막으로 두 리스트 sports와 num을 갖고 리스트 컴프리헨션으로 다음 구조의 중첩된 리스트도 만들어 종목과 팀원 수를 적절히 출력하는 코드도 포함시켜 보자.

```
[['축구', 11], ['야구', 9], ['농구', 5], ['배구', 6]]
```

## ⚠ 프로젝트 Lab1

### 문제 이해 (Understanding)

리스트 생성과 리스트 컴프리헨션으로 다양한 구조의 리스트를 만든다.  
만들어진 리스트에서 for문으로 항목을 적절히 출력하도록 한다.  
for문에서 시퀀스로 range(4) 또는 range(2)를 사용할지, 리스트 자체를 사용할지를  
선택하도록 한다.



## 프로젝트 Lab2

## 햄버거 콤보 번호를 받아 주문 가격 표시

난이도 실전

이번에는 햄버거 주문을 구현해 보자. 주문을 위해 콤보의 종류와 수량을 입력 받아 주문한 내역과 가격 그리고 총 주문 가격을 출력하는 프로그램을 작성해 보자. 주문 종류는 오른쪽과 같으며, 0을 입력하면 주문을 종료한다. 콤보 종류와 수량을 한 번에 입력 받으면 주문 내역과 총 가격을 표시하고, 주문이 종료되면 총 주문 가격을 출력한다.

이미 커피 주문은 이전 단원에서 살펴봤다. 이번에는 주문 리스트 가격을 이번 단원에서 배운 리스트나 튜플로 사용하면 훨씬 간결하다. 주문에 사용하는 번호와 menu의 항목 첨자 번호를 같게 하면 편리하다.

물론 price의 가격 리스트와 메뉴가 맞으면 menu[i]의 가격이 price[i]가 된다.

```
menu = ('주문종료', '올인원팩', '투게더팩', '트리오팍', '패밀리팩')
```

```
price = (0, 6000, 7000, 8000, 10000)
```

주문할 콤보 번호와 수량을  
계속 입력하세요!

0	주문종료	
1	올인원팩	6000 원
2	투게더팩	7000 원
3	트리오팍	8000 원
4	패밀리팩	10000 원
>>	2 2	





## ⚠ 프로젝트 Lab2

### 문제 이해 (Understanding)

햄버거의 종류가 1에서 4까지이며, 0이 입력되면 주문을 종료한다.  
표준 입력으로 종류와 수량을 입력 받아 중간 주문 가격을 출력한다.  
주문이 완료되면 총 주문 가격을 출력한다.

# SUMMARY



동양미래대학교  
DONGYANG MIRAE UNIVERSITY

⚠ 튜플의 이해

⚠ 튜플 생성

# SUMMARY



## ⚠ 튜플에도 적용

- ... +, \*
- ... 내장 함수 sorted
- ... 튜플 자체 제거: del

## ⚠ 프로젝트 lab1, lab2