

## Exercise 7. While문과 Tuple

### Section 7.1 - Tuple 선언하기

1. 학생의 이름 "김철수", 나이 20, 학과 "컴퓨터공학"을 하나의 tuple로 만들고 출력하세요.

```
('김철수', 20, '컴퓨터공학')
```

출력예시

**힌트:** 문자열은 따옴표로 감싸고, 숫자는 그대로 써주세요.

```
# 학생 정보를 tuple로 선언
student = ('김철수', 20, '컴퓨터공학')
print(student)
```

해설

2. 숫자 42 하나만 들어있는 tuple을 만들어보세요. (coma 사용에 주의!)

```
(42,)
```

출력예시

**힌트:** 요소가 하나인 tuple을 만들려면 콤마(,)가 필요합니다.

```
# 요소 하나인 tuple 선언 (coma 필수!)
single_tuple = (42,)
print(single_tuple)
```

해설

3. 괄호 없이 1, 2, 3을 이용해서 tuple을 만들고, 그 타입을 확인해보세요.

```
(1, 2, 3)
```

출력예시

**힌트:** Python에서는 콤마로 구분된 값들이 자동으로 tuple이 됩니다.

```
# 괄호 없이 tuple 선언
numbers = 1, 2, 3
print(numbers)
print(type(numbers))
```

해설

4. 빈 tuple을 만들어 empty\_tuple 변수에 저장하고, 그 타입을 확인해보세요.

()

출력예시

**힌트:** 빈 tuple은 ()로 만들고, type() 함수로 타입을 확인할 수 있습니다.

```
# 빈 tuple 선언
empty_tuple = ()
print(empty_tuple)
print(type(empty_tuple))
```

해설

5. 게임 캐릭터의 체력 100, 마나 50, 공격력 25를 하나의 tuple로 만들어 character\_stats에 저장하고 출력하세요.

(100, 50, 25)

출력예시

**힌트:** 체력, 마나, 공격력 순서로 숫자를 tuple에 넣으세요.

```
# 캐릭터 스탯을 tuple로 선언
character_stats = (100, 50, 25)
print(character_stats)
```

해설

## Section 7.2 - Tuple 언패킹하기

6. 학생 정보와 성적이 담긴 tuple (('이철수', 22), ('수학', 95))을 각각 언패킹해서 출력하세요.

```
student_data = (('이철수', 22), ('수학', 95))
# 여기에 코드 작성
```

학생: 이철수 (22세)  
과목: 수학, 점수: 95점

출력예시

**힌트:** 먼저 큰 tuple을 언패킹하고, 각각을 다시 언패킹하세요.

```
student_data = (('이철수', 22), ('수학', 95))
student_info, grade_info = student_data
name, age = student_info
subject, score = grade_info

print(f"학생: {name} ({age}세)")
print(f"과목: {subject}, 점수: {score}점")
```

해설

7. 날짜 tuple (2024, 12, 25)를 year, month, day 변수에 언패킹하고 "2024년 12월 25일" 형태로 출력하세요.

```
date = (2024, 12, 25)
# 여기에 코드 작성
```

2024년 12월 25일

출력예시

**힌트:** 언패킹 후 *f-string*으로 출력하세요.

```
date = (2024, 12, 25)
year, month, day = date
print(f"{year}년 {month}월 {day}일")
```

해설

8. 학생 정보 tuple ('박영희', 19, 'A+')을 name, age, grade 변수에 언패킹하고 출력하세요.

```
student = ('박영희', 19, 'A+')
# 여기에 언패킹 코드 작성
```

이름: 박영희  
나이: 19  
성적: A+

출력예시

**힌트:** *name, age, grade = student* 형태로 언패킹하세요.

```
student = ('박영희', 19, 'A+')
name, age, grade = student
print(f"이름: {name}")
print(f"나이: {age}")
print(f"성적: {grade}")
```

해설

9. 게임 캐릭터 스탯 tuple (150, 80, 35)을 hp, mp, attack 변수에 언패킹하고 상태를 출력하세요.

```
character = (150, 80, 35)
# 여기에 코드 작성
```

체력: 150, 마나: 80, 공격력: 35

출력예시

**힌트:** 언패킹 후 한 줄로 출력하세요.

```
character = (150, 80, 35)
hp, mp, attack = character
print(f"체력: {hp}, 마나: {mp}, 공격력: {attack}")
```

해설

10. 좌표 tuple (3, 7)을 x, y 변수에 언패킹하고 각각 출력하세요.

```
point = (3, 7)
# 여기에 언패킹 코드 작성
```

```
x: 3
y: 7
```

출력예시

**힌트:** `x, y = point` 형태로 언패킹할 수 있습니다.

```
point = (3, 7)
x, y = point
print(f"x: {x}")
print(f"y: {y}")
```

해설

## Section 7.3 - Tuple과 빌트인 함수

11. 구매한 물건 가격 tuple (1500, 3000, 2500, 4000)의 총합을 구해서 출력하세요.

```
prices = (1500, 3000, 2500, 4000)
# 여기에 코드 작성
```

```
총 금액: 11000원
```

출력예시

**힌트:** `sum()` 함수를 사용하면 모든 요소의 합을 구할 수 있습니다.

```
prices = (1500, 3000, 2500, 4000)
total = sum(prices)
print(f"총 금액: {total}원")
```

해설

12. 숫자 tuple (45, 23, 67, 12, 89)을 큰 수부터 작은 수 순으로 정렬해서 출력하세요.

```
numbers = (45, 23, 67, 12, 89)
# 여기에 코드 작성
```

[89, 67, 45, 23, 12]

출력예시

**힌트:** `sorted()` 함수에 `reverse=True` 옵션을 사용하세요.

```
numbers = (45, 23, 67, 12, 89)
sorted_desc = sorted(numbers, reverse=True)
print(sorted_desc)
```

해설

13. tuple (10, 20, 30, 40, 50)의 길이를 구해서 출력하세요.

```
numbers = (10, 20, 30, 40, 50)
# 여기에 코드 작성
```

5

출력예시

**힌트:** `len()` 함수를 사용하면 tuple의 길이를 구할 수 있습니다.

```
numbers = (10, 20, 30, 40, 50)
length = len(numbers)
print(length)
```

해설

14. 점수 tuple (85, 92, 78, 96, 88)에서 최고 점수를 찾아서 출력하세요.

```
scores = (85, 92, 78, 96, 88)
# 여기에 코드 작성
```

최고 점수: 96

출력예시

**힌트:** `max()` 함수를 사용하면 최대값을 구할 수 있습니다.

```
scores = (85, 92, 78, 96, 88)
highest = max(scores)
print(f"최고 점수: {highest}")
```

해설

15. 과일 tuple ('사과', '바나나', '오렌지', '포도')에서 '바나나'가 있는지 확인해보세요.

```
fruits = ('사과', '바나나', '오렌지', '포도')  
# 여기에 코드 작성
```

바나나가 있습니다

출력예시

**힌트:** `in` 연산자를 사용해서 조건문을 만드세요.

```
fruits = ('사과', '바나나', '오렌지', '포도')  
if '바나나' in fruits:  
    print("바나나가 있습니다")  
else:  
    print("바나나가 없습니다")
```

해설

## Section 7.4 - Tuple 메서드

16. 숫자 tuple (1, 5, 3, 5, 7, 5, 9)에서 인덱스 2부터 찾기 시작해서 숫자 5의 위치를 찾아보세요.

```
numbers = (1, 5, 3, 5, 7, 5, 9)  
# 여기에 코드 작성
```

인덱스 2부터 찾은 5의 위치: 3

출력예시

**힌트:** `index(값, 시작위치)` 형태로 사용할 수 있습니다.

```
numbers = (1, 5, 3, 5, 7, 5, 9)  
index_from_2 = numbers.index(5, 2)  
print(f"인덱스 2부터 찾은 5의 위치: {index_from_2}")
```

해설

17. 과일 tuple ('사과', '바나나', '사과', '오렌지', '사과')에서 '사과'의 개수를 세어보세요.

```
fruits = ('사과', '바나나', '사과', '오렌지', '사과')  
# 여기에 코드 작성
```

사과의 개수: 3개

출력예시

**힌트:** 문자열도 `count()` 메서드로 셀 수 있습니다.

```
fruits = ('사과', '바나나', '사과', '오렌지', '사과')  
apple_count = fruits.count('사과')  
print(f"사과의 개수: {apple_count}개")
```

해설

18. 학점 tuple ('A', 'B', 'A', 'C', 'A', 'B')에서 'A' 학점의 개수와 첫 번째 위치를 모두 출력하세요.

```
grades = ('A', 'B', 'A', 'C', 'A', 'B')  
# 여기에 코드 작성
```

A 학점 개수: 3개  
A 학점 첫 위치: 0

출력예시

**힌트:** `count()`와 `index()` 메서드를 둘 다 사용하세요.

```
grades = ('A', 'B', 'A', 'C', 'A', 'B')  
a_count = grades.count('A')  
a_index = grades.index('A')  
print(f"A 학점 개수: {a_count}개")  
print(f"A 학점 첫 위치: {a_index}")
```

해설

19. 동물 tuple ('고양이', '강아지', '토끼', '햄스터')에서 '토끼'의 위치(인덱스)를 찾아보세요.

```
animals = ('고양이', '강아지', '토끼', '햄스터')  
# 여기에 코드 작성
```

토끼의 위치: 2

출력예시

**힌트:** `tuple.index()` 메서드를 사용하면 첫 번째로 나타나는 위치를 찾습니다.

```
animals = ('고양이', '강아지', '토끼', '햄스터')  
rabbit_index = animals.index('토끼')  
print(f"토끼의 위치: {rabbit_index}")
```

해설

20. 숫자 tuple (5, 3, 7, 3, 9, 3)에서 숫자 3이 처음 나타나는 위치를 찾아보세요.

```
numbers = (5, 3, 7, 3, 9, 3)
# 여기에 코드 작성
```

3의 첫 번째 위치: 1

출력예시

**힌트:** `index()` 메서드는 항상 첫 번째로 찾은 위치를 반환합니다.

```
numbers = (5, 3, 7, 3, 9, 3)
first_3_index = numbers.index(3)
print(f"3의 첫 번째 위치: {first_3_index}")
```

해설