

Exercise 10. 사용자 정의 함수 (Functions)

Section 10.1 - 함수 정의와 호출

1. 1부터 5까지의 합을 계산해서 반환하는 함수를 작성하세요. 함수명은 'sum_one_to_five'입니다.

15

출력예시

힌트: *for문이나 직접 계산으로 1+2+3+4+5의 결과를 return하세요.*

```
# 1부터 5까지의 합을 반환하는 함수
def sum_one_to_five():
    total = 0
    for i in range(1, 6):
        total += i
    return total

result = sum_one_to_five()
print(result)
```

해설

2. 문자열 "Python Programming"을 대문자로 변환해서 반환하는 함수를 작성하세요. 함수명은 'convert_to_upper'입니다.

PYTHON PROGRAMMING

출력예시

힌트: *.upper() 메서드를 사용해서 문자열을 대문자로 변환하세요.*

```
# 문자열을 대문자로 변환하는 함수
def convert_to_upper():
    text = "Python Programming"
    return text.upper()

result = convert_to_upper()
print(result)
```

해설

3. 두 함수를 작성하세요. 'create_list' 함수는 [10, 20, 30]을 반환하고, 'print_list' 함수는 리스트를 받아서 출력합니다. 두 함수를 연결해서 사용하세요.

```
[10, 20, 30]
```

출력예시

힌트: 첫 번째 함수의 반환값을 두 번째 함수의 인자로 전달하세요.

```
# 두 함수를 연결해서 사용
def create_list():
    return [10, 20, 30]

def print_list():
    my_list = create_list()
    print(my_list)

print_list()
```

해설

4. 구구단 3단을 문자열로 만들어서 반환하는 함수를 작성하세요. 함수명은 'get_multiplication_table'이고, "3 x 1 = 3\n3 x 2 = 6\n3 x 3 = 9" 형태로 반환합니다.

```
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
```

출력예시

힌트: for문을 사용해서 문자열을 만들고, \n으로 줄바꿈을 추가하세요.

```
# 구구단 3단을 반환하는 함수
def get_multiplication_table():
    result = ""
    for i in range(1, 4): # 1, 2, 3만 출력
        result += f"3 x {i} = {3 * i}\n"
    return result.strip() # 마지막 줄바꿈 제거

table = get_multiplication_table()
print(table)
```

해설

Section 10.2 - 매개변수와 인자

5. 직사각형의 가로와 세로를 매개변수로 받아서 넓이를 계산하는 함수를 작성하세요. 함수명은 'calculate_area'이고 가로 5, 세로 3일 때의 넓이를 출력하세요.

15

출력예시

힌트: `width * height`로 넓이를 계산하세요.

```
# 직사각형 넓이를 계산하는 함수
def calculate_area(width, height):
    return width * height

area = calculate_area(5, 3)
print(area)
```

해설

6. 이름을 매개변수로 받아서 인사말을 출력하는 함수를 작성하세요. 함수명은 'greet_person'이고 "김철수"를 인자로 전달해서 호출하세요.

안녕하세요, 김철수님!

출력예시

힌트: 매개변수 `name`을 받아서 인사말에 포함시키세요.

```
# 매개변수를 받는 함수
def greet_person(name):
    print(f"안녕하세요, {name}님!")

greet_person("김철수")
```

해설

7. 온도와 단위를 매개변수로 받아서 섭씨를 화씨로 또는 화씨를 섭씨로 변환하는 함수를 작성하세요. 함수명은 'convert_temperature'입니다.

32.0

출력예시

힌트: $C \text{ to } F: (C * 9/5) + 32$, $F \text{ to } C: (F - 32) * 5/9$ 공식을 사용하세요.

```
# 온도 변환 함수
def convert_temperature(temp, unit):
    if unit == "C":
        # 섭씨를 화씨로 변환
        return (temp * 9/5) + 32
    elif unit == "F":
        # 화씨를 섭씨로 변환
        return (temp - 32) * 5/9

result = convert_temperature(0, "C") # 섭씨 0도를 화씨로
print(result)
```

해설

8. 시작값과 끝값을 매개변수로 받아서 그 범위의 숫자들을 리스트로 반환하는 함수를 작성하세요. 함수명은 'create_range_list'입니다.

[3, 4, 5, 6, 7]

출력예시

힌트: `range(start, end+1)`을 사용해서 리스트로 변환하세요.

```
# 범위 리스트를 생성하는 함수
def create_range_list(start, end):
    return list(range(start, end + 1))

result = create_range_list(3, 7)
print(result)
```

해설

Section 10.3 - 매개변수의 기본값

9. 학생의 성적을 평가하는 함수를 작성하세요. 점수는 필수이고, 과목명(기본값: "시험"), 만점(기본값: 100)은 선택사항입니다. 함수명은 'evaluate_score'입니다.

수학 성적: 85/100 (85.0%)

출력예시

힌트: 백분율은 (점수/만점)*100으로 계산하세요.

```
# 성적 평가 함수
def evaluate_score(score, subject="시험", max_score=100):
    percentage = (score / max_score) * 100
    return f"{subject} 성적: {score}/{max_score} ({percentage}%)"

result = evaluate_score(85, subject="수학")
print(result)
```

해설

10. 로그 메시지를 출력하는 함수를 작성하세요. 메시지는 필수이고, 레벨(기본값: "INFO"), 시간(기본값: "12:00")은 선택사항입니다. 함수명은 'log'입니다.

[14:30] ERROR: 오류가 발생했습니다

출력예시

힌트: f-string을 사용해서 로그 형식을 만드세요.

```
# 로그 출력 함수
def log(message, level="INFO", time="12:00"):
    return f"[{time}] {level}: {message}"

result = log("오류가 발생했습니다", level="ERROR", time="14:30")
print(result)
```

해설

11. 파일 경로를 생성하는 함수를 작성하세요. 파일명은 필수이고, 확장자(기본값: ".txt"), 폴더(기본값: "documents")는 선택사항입니다. 함수명은 'create_filepath'입니다.

photos/image.jpg

출력예시

힌트: 문자열 연결을 사용해서 경로를 만드세요.

```
# 파일 경로 생성 함수
def create_filepath(filename, extension=".txt", folder="documents"):
    return f"{folder}/{filename}{extension}"

path = create_filepath("image", extension=".jpg", folder="photos")
print(path)
```

해설

12. 문자열을 포매팅하는 함수를 작성하세요. 텍스트는 필수이고, 패딩문자(기본값: " "), 총길이(기본값: 20)는 선택사항입니다. 함수명은 'format_text'입니다.

```
===Hello===
```

출력예시

힌트: `.center()` 메서드를 사용해서 텍스트를 가운데 정렬하세요.

```
# 텍스트 포매팅 함수
def format_text(text, padding=" ", width=20):
    return text.center(width, padding)

result = format_text("Hello", padding="=", width=11)
print(result)
```

해설

Section 10.4 - 함수의 타입 힌트

13. 나이를 받아서 성인 여부를 불린값으로 반환하는 함수를 타입 힌트와 함께 작성하세요. 함수명은 'is_adult'입니다.

```
True
```

출력예시

힌트: `bool` 타입 힌트를 사용하고, 18세 이상이면 `True`를 반환하세요.

```
# 성인 여부 확인 함수 (타입 힌트 포함)
def is_adult(age: int) -> bool:
    return age >= 18

result = is_adult(20)
print(result)
```

해설

14. 문자열 리스트를 받아서 가장 긴 문자열의 길이를 정수로 반환하는 함수를 타입 힌트와 함께 작성하세요. 함수명은 'find_max_length'입니다.

```
words = ['apple', 'banana', 'cherry', 'date']
```

6

출력예시

힌트: `max()` 함수와 `key=len`을 사용하거나 반복문을 사용하세요.

```
# 최대 길이 찾기 함수 (타입 힌트 포함)
def find_max_length(words: list) -> int:
    return len(max(words, key=len))

words = ['apple', 'banana', 'cherry', 'date']
max_length = find_max_length(words)
print(max_length)
```

해설

15. 시작과 끝 숫자를 받아서 범위 리스트를 반환하는 함수를 타입 힌트와 함께 작성하세요. 함수명은 'create_number_list'입니다.

```
[3, 4, 5, 6, 7]
```

출력예시

힌트: `range()`를 `list()`로 변환해서 반환하세요.

```
# 숫자 리스트 생성 함수 (타입 힌트 포함)
def create_number_list(start: int, end: int) -> list:
    return list(range(start, end + 1))

numbers = create_number_list(3, 7)
print(numbers)
```

해설

16. 좌표 튜플을 받아서 원점으로부터의 거리를 실수로 반환하는 함수를 타입 힌트와 함께 작성하세요. 함수명은 'calculate_distance'입니다.

5.0

출력예시

힌트: $\text{거리} = \sqrt{x^2 + y^2}$ 공식을 사용하고, `math.sqrt()`나 `**0.5`를 사용하세요.

```
# 거리 계산 함수 (타입 힌트 포함)
def calculate_distance(point: tuple) -> float:
    x, y = point
    return (x**2 + y**2) ** 0.5

distance = calculate_distance((3, 4))
print(distance)
```

해설

Section 10.5 - 가변 매개변수

17. 필수 매개변수와 *args를 함께 사용하는 함수를 작성하세요. 첫 번째 인자는 연산자(+, -, *, /)이고 나머지는 숫자들입니다. 함수명은 'calculate'입니다.

60

출력예시

힌트: 첫 번째 매개변수로 연산자를 받고, *args로 숫자들을 받아서 계산하세요.

```
# 필수 매개변수와 *args를 함께 사용
def calculate(operator, *numbers):
    if operator == "+":
        return sum(numbers)
    elif operator == "*":
        result = 1
        for num in numbers:
            result *= num
        return result

result = calculate("?", 3, 4, 5)
print(result)
```

해설

18. 가변 개수의 점수들을 받아서 통계 정보를 딕셔너리로 반환하는 함수를 작성하세요. 함수명은 'analyze_scores'이고 *args를 사용하세요.

```
{'count': 5, 'total': 425, 'average': 85.0, 'max': 95, 'min': 72}
```

출력예시

힌트: 점수들의 개수, 합계, 평균, 최댓값, 최솟값을 계산해서 딕셔너리로 반환하세요.

```
# *args를 사용한 점수 분석 함수
def analyze_scores(*scores):
    return {
        'count': len(scores),
        'total': sum(scores),
        'average': sum(scores) / len(scores),
        'max': max(scores),
        'min': min(scores)
    }

stats = analyze_scores(85, 92, 78, 95, 72)
print(stats)
```

해설

19. 가변 개수의 숫자들을 받아서 모든 수의 합을 반환하는 함수를 작성하세요. 함수명은 'sum_all'이고 *args를 사용하세요.

```
30
```

출력예시

힌트: *args를 사용하면 여러 개의 위치 인자를 튜플로 받을 수 있습니다.

```
# *args를 사용한 합계 함수
def sum_all(*numbers):
    total = 0
    for num in numbers:
        total += num
    return total

result = sum_all(5, 10, 15)
print(result)
```

해설

20. 가변 개수의 숫자들 중에서 최댓값을 찾는 함수를 작성하세요. 함수명은 'find_maximum'이고 *args를 사용하세요.

45

출력예시

힌트: `max()` 함수를 사용하거나 반복문으로 최댓값을 찾으세요.

```
# *args를 사용한 최댓값 찾기 함수
def find_maximum(*numbers):
    return max(numbers)

result = find_maximum(12, 45, 23, 8, 37)
print(result)
```

해설