

22장

고급 함수 연산

22장 고급 함수 연산

22.1 함수를 두 가지 관점에서 살펴보기

22.2 함수 영역

22.3 내포 함수

22.4 함수를 매개변수로 넘기기

22.5 함수 반환하기

22.6 요약

22.1 함수를 두 가지 관점에서 살펴보기



작성자 관점, 사용자 관점

- » 작성자 관점 - 함수를 작성할 때는 여러 값을 받아서 처리할 수 있게 일반적인 방식으로 작성한다. 이때 함수가 미리 정해진 이름의 변수로 입력을 받는다고 가정하고 그 동작을 일반화한다. 이런 입력을 형식인자라고 한다(다른 말로 매개변수라고도 하며, 인자라는 말을 사용하는 경우도 많다).
- » 사용자 관점 - 함수를 호출할 때는 함수 이름 뒤에 괄호를 붙이고, 괄호 안에 값들을 지정한다. 이런 값들을 실인자라고 부른다.

22.2 함수 영역



22.2 함수 영역

- » 함수 코드 블록 안에서 발생한 일은 그 코드 블록 안에 한정된다.
- » 함수의 매개변수나 함수 내에서 정의된 변수를 함수 영역 밖에서 사용하려고 시도하면 오류가 발생한다.



22.2.1 간단한 영역 규칙

- » 함수 안에서 다른 함수나 메인 프로그램이 사용하는 변수와 같은 이름의 변수를 정의할 수 있다.
- » 파이썬은 이 변수들이 이름만 같을 뿐이지 서로 다른 객체를 가리킨다는 사실을 알고 있다.

```
def 파리의_연인(): ---- 함수 정의
    정은 = "강태영" ---- 함수 안에 있는 정은이라는 변수의 값은 '강태영'임
    print(정은) ---- 강태영 출력

정은 = "위원장" ---- 메인 프로그램에서 실행되는 첫 번째 문장. 정은이라는 변수를 만들고 '위원장'으로 초기화
파리의_연인() ---- 함수 호출이 새 영역을 만들어내고 '강태영'을 출력. 그 후 함수가 None을 반환하면서 영역이 끝남
print(정은) ---- 위원장 출력
```

코드 22-1

다른 영역에서 같은 이름의 변수 정의하기



22.2.2 영역 규칙

» 다음은 (프로그램 안에서 이름이 같은 변수가 둘 이상 있을 때) 각 영역 내에서 사용할 변수를 결정하는 규칙이다.

1. 현재 영역이 함수인 경우 그 이름으로 변수가 정의되어 있거나 그 이름의 매개변수가 있는지 찾아본다. 이름이 같은 변수나 매개변수가 있다면 그 변수를 사용한다. 그렇지 않다면 함수 호출을 둘러싸고 있는 영역을 찾는다. 이 바깥쪽 영역이 함수일 수도 있다.
2. 만약 호출을 둘러싼 영역에 변수가 있다면 그 변수를 사용한다.
3. 호출을 둘러싼 영역에 변수가 있을 때까지 1, 2번 과정을 반복한다. 그러다 보면 결국에는 메인 프로그램 영역에 도달할 것이다. 메인 프로그램 영역을 다른 말로 글로벌 영역(global scope)라고도 한다(우리말로 전역이라고도 한다). 글로벌 영역 바깥에는 아무 영역도 없다. 글로벌 영역에 정의된 변수를 전역 변수라고 부른다.
4. 전역 변수 중에서도 원하는 이름의 변수를 찾을 수 없으면 그 변수가 존재하지 않는다고 오류를 발생시킨다.

» 코드 22-2 ~ 코드 22-5

- 함수 안에서 변수를 선언하지 않고도 변수에 접근(access)할 수 있다. 바깥 영역이나 글로벌 영역에 그 이름의 변수가 존재하는 한, 아무 오류도 나지 않는다.
- 어떤 함수에서 정의하지 않은 새 이름의 변수에 값을 대입하면, 그 함수 영역 안에서 변수를 정의하는 것으로 취급한다. 설명 함수의 바깥 영역에 같은 이름의 변수가 존재해도 새로운 변수가 생긴다는 점에 유의하라.



22.2.2 영역 규칙

» 코드 22-6

```
def odd_or_even(num): ---- 함수 정의는 num이라는 한 매개변수만 받음
    num = num%2 ---- 2로 num을 나눈 나머지
    if num == 1:
        return "odd"
    else:
        return "even"

num = 4 ---- 전역(글로벌 영역) 변수
print(odd_or_even(num)) ---- 함수를 호출하고, 그 반환 값을 출력
odd_or_even(5) ---- 함수를 호출하고 반환 값에 대해 아무 일도 하지 않음
```

코드 22-6

여러 다른 영역 규칙을 보여 주는 함수

22.2.2 영역 규칙

1

```
def odd_or_even (num):
    num = num%2
    if num == 1:
        return "odd"
    else:
        return "even"
```

```
num = 4
print(odd_or_even(num))
print(num)
```

글로벌 영역

odd_or_even: SOME
CODE

num: 4

2

```
def odd_or_even (num):
    num = num%2
    if num == 1:
        return "odd"
    else:
        return "even"
```

```
num = 4
print(odd_or_even(num))
print(num)
```

글로벌 영역

odd_or_even: SOME
CODE

num: 4

odd_or_even(4)의
영역

returns: ???

num: 4

```
def odd_or_even (num):
    num = num%2
    if num == 1:
        return "odd"
    else:
        return "even"
```

```
num = 4
print(odd_or_even(num))
print(num)
```

글로벌 영역

odd_or_even: SOME
CODE

num: 4

odd_or_even(4)의
영역

returns: ???

num: 0

```
def odd_or_even (num):
    num = num%2
    if num == 1:
        return "odd"
    else:
        return "even"
```

```
num = 4
print(odd_or_even(num))
print(num)
```

글로벌 영역

odd_or_even: SOME
CODE

num: 4

odd_or_even(4)의
영역

returns: "even"

num: 0

3

```
def odd_or_even (num):
    num = num%2
    if num == 1:
        return "odd"
    else:
        return "even"
```

```
num = 4
print(odd_or_even(num))
print(num)
```

글로벌 영역

odd_or_even: SOME
CODE

num: 4

출력:

even

4



22.2.2 영역 규칙

» 셀프 체크 22-1

```
def f(a,b):
    x = a+b
    y = a-b
    print(x*y)
    return x/y
```

```
a=1
b=2
x=5
y=6
```

```
print(f(x,y))
print(f(a,b))
print(f(x,a))
print(f(y,b))
```

22.3 내포 함수



22.3 내포 함수

» 내포 함수는 다른 함수의 안쪽에 정의된 함수다.

- 파이썬에서 내포된 함수는 오직 바깥쪽 함수의 영역만 볼 수 있다. 그리고 내포된 함수는 그 함수가 포함된 외부 함수가 호출된 경우에만 영역 내 변수를 볼 수 있다.

» 코드 22-7. sing() 함수 안에 stop() 함수가 내포되어 있다.

- 실제 sing() 함수를 호출해야만 함수가 실행된다. 만약 메인 프로그램 영역에서 stop()을 호출하면 오류가 발생할 것이다.

```
def sing():
    def stop(line): ---- sing() 안에 있는 함수 정의
        print("STOP",line)
    stop("it's hammer time")
    stop("in the name of love")
    stop("hey, what's that sound") ---- sing() 안에서 stop() 함수를 호출
stop() ---- 글로벌 영역 안에 stop()이 없기 때문에 오류가 발생함
sing()
```



22.3 내포 함수

» 셀프 체크 22.2

```
def add_one(a,b):
    x = a+1
    y = b+1
    def mult(a,b):
        return a*b
    return mult(x,y)
a = 1
b = 2
x = 5
y = 6

print(add_one(x, y))
print(add_one(a, b))
print(add_one(x, a))
print(add_one(y, b))
```

22.4 함수를 매개변수로 넘기기



22.4 함수를 매개변수로 넘기기

» 지금까지 객체 타입으로 정수(int), 문자열(string), 실수(float), 불(bool)을 봤다.

- 파이썬에서는 모든것이 객체다.
- 따라서 정의하는 함수도 function이라는 타입의 객체다.
- 어떤 객체라도 함수의 인자로 전달될 수 있으므로, 함수도 인자로 전달할 수 있다.



22.4 함수를 매개변수로 넘기기

- » sandwich 함수는 kind_of_sandwich를 매개변수로 받는다.
- » Sandwich 함수를 호출할 때는 함수 객체를 인자로 넘겨야 한다.

```
def sandwich(kind_of_sandwich): ---- kind_of_sandwich는 매개변수임
    print("-----")
    print(kind_of_sandwich ()) ---- kind_of_sandwich 뒤에 괄호를 붙여서 함수 호출임을 표현함
    print("-----")

def blt():
    my_blt = " 베이컨\n 상추\n 토마토"
    return my_blt

def breakfast():
    my_ec = " 달걀\n 치즈"
    return my_ec

print(sandwich(blt)) ---- 함수 이름(함수 객체 자체)만 사용함
```

22.5 함수 반환하기



22.5 함수 반환하기

» 함수도 객체이므로 함수에서 다른 함수를 반환할 수도 있다.

```
def grumpy(): ---- 함수 정의
    print("I am a grumpy cat:")
    def no_n_times(n): ---- 내포 함수 정의
        print("No", n, "times...")
        def no_m_more_times(m): ---- 내포 함수 정의
            print("...and no", m, "more times")
            for i in range(n+m): ---- no를 n + m번 출력하는 루프
                print("no")
        return no_m_more_times ---- no_n_times는 no_m_more_times를 반환
    return no_n_times ---- grumpy는 no_n_times를 반환
grumpy()(4)(2) ---- 메인 프로그램에 있는 함수 호출
```

코드 22-9 함수에서 다른 함수 객체 반환하기

22.6 요약



22.6 요약

- » 이미 이 책의 앞부분에서 print와 같은 내장 함수를 여러 개 살펴봤고, 사용법을 배웠다. 이장에서는 내장 함수를 호출할 때 인자를 괄호 안에 넣어서 호출한다는 사실과 내장 함수도 연산 결과를 반환 값으로 돌려준다는 사실을 배웠다.
- » 함수 안에 함수를 정의해 내포할 수 있다. 내포된 함수는 그 함수가 정의된 바로 바깥 함수의 영역 안에서만 존재한다.
- » 함수 객체를 다른 객체와 마찬가지로 사용할 수 있다. 함수를 인자로 넘길 수도 있고, 함수가 함수를 반환할 수도 있다.



22.6 요약

» (Q22.1) 다음 코드에 필요한 부분을 작성하라.

```
def area(shape, n):
    # 매개변수로 넓이를 계산하는 함수 shape와 값 n을 입력 받아서
    # 넓이를 계산하는 일반적인 코드를 한 줄로 작성하라

def circle(radius):
    return 3.14*radius**2
def square(length):
    return length*length

print(area(circle,5)) # 함수 호출 예제
```

1. area()를 사용해 반지름(radius) 10인 원(circle)의 넓이를 계산하는 코드를 작성하라.
2. area()를 사용해 한 변의 길이가 5인 정사각형(square)의 넓이를 계산하는 코드를 작성하라.
3. area()를 사용해 지름(diameter, radius의 2배)이 4인 원의 넓이를 계산하는 코드를 작성하라.



22.6 요약

» (Q22.2) 다음 코드에 필요한 부분을 작성하라.

```
def person(age):
    print("I am a person")
    def student(major):
        print("I like learning")
        def vacation(place):
            print("But I need to take breaks")
            print(age,"|",major,"|",place)
        # 적절한 함수를 반환하는 한 줄짜리 코드를 작성하라
    # 적절한 함수를 반환하는 한 줄짜리 코드를 작성하라
```

1. area()를 사용해 반지름(radius) 10인 원(circle)의 넓이를 계산하는 코드를 작성하라.
2. area()를 사용해 한 변의 길이가 5인 정사각형(square)의 넓이를 계산하는 코드를 작성하라.
3. area()를 사용해 지름(diameter, radius의 2배)이 4인 원의 넓이를 계산하는 코드를 작성하라.



22.6 요약

예를 들어 다음과 같이 호출하면

```
person(12)("Math")("beach") # 함수 호출 예제
```

다음과 같은 출력이 나와야 한다.

```
I am a person
I like learning
But I need to take breaks
12 | Math | beach
```

1. 나이(age) 29, 전공(major) "CS", 방학 계획으로 "SouthKorea"를 사용한 person 함수 호출을 작성하라.
2. 23 | "Law" | "NorthKorea"를 마지막에 표시하도록 person 함수를 호출하라.