DAY 11

함수를 정의하고 호출하기

모두의 파이썬 20일 만에 배우는 프로그래밍 기초

길벗

함수를 정의하고 호출하기

01 함수를 정의하고 호출하는 프로그램

02 인자가 있는 함수

03 결괏값이 있는 함수

01

1. 함수를 정의하고 호출하는 프로그램

» 함수 : 자주 사용하는 프로그램의 일부분을 블록으로 분리해서 여러 번 사용할 수 있게 해 줌

- 함수를 정의한다' : 함수가 어떤 기능을 할지 파이썬에 알려 주는 것
- '함수를 호출한다' : 만들어진 함수를 실제로 사용하는 것

```
def 함수 이름(인자):          # 인자 없을 때 생략 가능
    함수의 내용
    return 함수의 결과값    # 결과값 없을 때 생략 가능
```

01

1. 함수를 정의하고 호출하는 프로그램

» 예제

```
def hello():                          # hello 함수를 정의
    print("Hello Python!")

hello()                               # hello 함수를 호출
hello()
hello()
```

» 실행결과

```
Hello Python!
Hello Python!
Hello Python!
```

02

2. 인자가 있는 함수

```
def hello():                    # 이름을 인자로 전달받아 Hello와 함께 출력하는 함수
    print("Hello" , name)

hello2("Justin")                # Justin을 인자값으로 넣어 hello2 함수를 호출
hello2("John")                  # John을 인자값으로 넣어 hello2 함수를 호출
hello2("Mike")                  # Mike를 인자값으로 넣어 hello2 함수를 호출
```

» 실행결과

```
Hello Justin
Hello John
Hello Mike
```

3. 결괏값이 있는 함수

```
def square(a):
    c = a * a
    return c

s1 = 4
s2 = square(s1
print(s1, s2)
```

a의 제곱(a*a)을 구하는 함수

s1(4)의 제곱을 구하는 함수를 호출해 결과를 s2에 저장

실행결과

```
4 16
```

3. 결괏값이 있는 함수

```
def triangle(a, h):                # 밑변이 a이고 높이가 h인 삼각형의 넓이를 구하는 함수
    c = a * h / 2
    return c


print(triangle(3, 4))              # 밑변이 3이고 높이가 4인 삼각형의 넓이를 출력
```

실행결과

?

3. 결괏값이 있는 함수

인자는 종류가 두 가지예요?

다음 프로그램을 잠깐 볼까요?

```
def square(n):
    return n*n

print(square(3))
```

- 어떤 수를 변수 n으로 전달받아 n의 제곱값(n*n)을 결과로 돌려주는 함수인 square를 정의하였습니다.
- 이후, 이 함수에 3이라는 값을 넣어서 호출해 출력하는 프로그램입니다.
- 여기서 square 함수를 정의할 때 사용한 n과 square 함수를 호출할 때 사용한 3은 둘 다 '인자'입니다.

3. 결괏값이 있는 함수

인자는 종류가 두 가지예요?

다음 프로그램을 잠깐 볼까요?

```
def square(n):
    return n*n

print(square(3))
```

- n과 같이 함수에서 사용되는 값을 정의하는 인자를 '형식 인자' 또는 '매개변수'라 합니다.
- 3과 같이 함수를 호출할 때 실제로 사용되는 값을 '실 인자' 또는 '인자/인수'라고 부릅니다.
- 엄밀하게 말하면 조금 다른 개념이지만, 처음 프로그래밍을 배우는 단계에서는 이 둘을 구분하면 오히려 혼란스러울 수 있습니다.
- 따라서 이 책에서는 그냥 '인자'라고 부르겠습니다.

DAY 12
함수 응용하기

모두의 파이썬 20일 만에 배우는 프로그래밍 기초

길벗

함수 응용하기

01 1부터 n까지의 합을 구하는 함수

02 1부터 n까지 곱을 구하는 함수

03 다각형을 그리는 함수

01

1. 1부터 n까지의 합을 구하는 함수

```
def sum_func(n):
    s = 0                       # 합을 구하기 위한 변수 s(시작 값을 0으로 지정)
    for x in range(1, n+1):     # range(1, n+1)로 1, 2, ..., n까지 반복(n+1은 제외)
        s = s+x                 # 지금까지 계산된 s 값에 x를 더해서 다시 s에 저장
    return s                    # 계산된 s 값을 결괏값으로 돌려줌

print(sum_func(10))
print(sum_func(100))
```

실행결과

```
55
5050
```

Lab! 원하는 수 x 부터 원하는 수 y까지의 합을 구하는 프로그램으로 수정!

02

2. 1부터 n까지 곱을 구하는 함수

```
def factorial(n):
    fact = 1                    # 곱을 구하기 위한 변수 fact(시작 값을 1로 지정).
    for x in range(1, n+1):     # range(1, n+1)로 1, 2, ..., n까지 반복(n+1은 제외)
        fact = fact * x         # 지금까지 계산된 값에 x를 곱해 fact에 다시 저장
    return fact                 # 계산된 fact 값을 돌려줌

print(factorial(5))
print(factorial(10))
```

» 실행결과

```
120
3628800
```

Lab! 해당 프로그램의 순서도를 작성해 보자!

02

2. 1부터 n까지 곱을 구하는 함수

함수이름 Factorial(팩토리얼)이 뭐예요?

- 1부터 n까지의 양의 정수를 모두 곱한 것을 수학에서는 n 팩토리얼(Factorial)이라 부릅니다. '계승'이라고 부르기도 합니다.
- 느낌표 기호(!)를 사용해서 n!로 표시합니다.
- 예:
  - 2 팩토리얼 : 2! = 1*2 = 2
  - 5 팩토리얼 : 5! = 1*2*3*4*5 = 120
  - 10 팩토리얼 : 10! = 1*2*3*4*5*6*7*8*9*10 = 3628800
  - 단, 0!은 1이라고 약속합니다.

팩토리얼은 중고등학교 수학에서 순열이나 확률을 배울 때 자주 사용하는 계산법이니 기억해 두세요.