

# Function and Console I/O

---

TEAMLAB director

최성철

# function

## -어떤 일을 수행하는 코드의 덩어리

```
# 사각형의 넓이를 구하는 함수
def calculate_rectangle_area (x , y):
    return x * y          # 가로, 세로를 곱해서 반환
```

- 반복적인 수행을 1회만 작성 후 호출
- 코드를 논리적인 단위로 분리
- 캡슐화: 인터페이스만 알면 타인의 코드 사용

- 함수 이름, parameter, indentation, return value(optional)

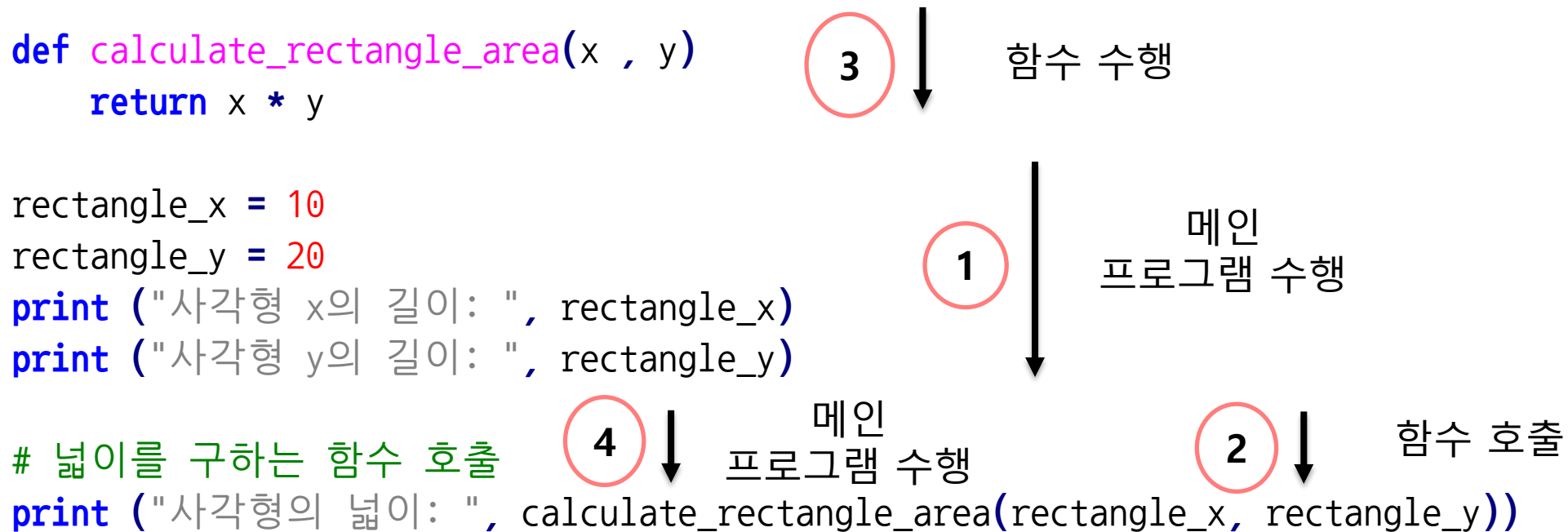
```
def 함수 이름 (parameter #1, ..., ):
    수행문 #1(statements)
    수행문 #2(statements)
    return <반환값>
```

```
def calculate_rectangle_area(x , y)
    return x * y

rectangle_x = 10
rectangle_y = 20
print ("사각형 x의 길이: ", rectangle_x)
print ("사각형 y의 길이: ", rectangle_y)

# 넓이를 구하는 함수 호출
print ("사각형의 넓이: ", calculate_rectangle_area(rectangle_x, rectangle_y))
```

- 함수 부분을 제외한 메인 프로그램부터 시작
- 함수 호출 시 함수부분을 수행 후 되돌아옴



- 프로그래밍의 함수와 수학의 함수는 유사함
- 모두 입력 값과 출력 값으로 이루어짐

$f(x) = 2x + 7, g(x) = x^2$  이고  $x = 2$ 일 때  
 $f(x) + g(x) + f(g(x)) + g(f(x))$ 의 값은?

$f(2) = 11, g(2) = 4, f(g(x)) = 15, g(f(x)) = 121$   
 $11 + 4 + 15 + 121 = 151$

이 공식을 파이썬으로 작성하면?

```
def f(x):  
    return 2 * x + 7  
def g(x):  
    return x ** 2  
x = 2  
print (f(x) + g(x) + f(g(x)) + g(f(x)))
```

↓ f(x) 함수 선언

↓ g(x) 함수 선언



- parameter : 함수의 입력 값 인터페이스

```
def f(x):  
    return 2 * x + 7
```

- argument: 실제 Parameter에 대입된 값

```
>>> print(f(2))  
11
```

- parameter 유무, 반환 값(return value) 유무에 따라 함수의 형태가 다름

	parameter 없음	parameter 존재
반환 값 없음	함수 내의 수행문만 수행	parameter를 사용, 수행문만 수행
반환 값 존재	parameter없이, 수행문 수행 후 결과값 반환	parameter를 사용하여 수행문 수행 후 결과값 반환

# console in/out

# 어떻게 프로그램과 데이터를 주고 받을 것인가?

## 터미널 - mouse가 아닌 키보드로 명령을 입력 프로그램 실행



**Graphical User Interface**



**Command Line Interface**

**Graphic User Interface (GUI)와 달리 Text를 사용하여 컴퓨터에 명령을 입력하는 인터페이스 체계**

**Windows – CMD window, Windows Terminal  
Mac, Linux – Terminal**

**윈도우 cmdr 도 권장(<https://cmdr.net/>)**

## input() 함수는 콘솔창에서 문자열을 입력 받는 함수

console\_test.py

```
print ("Enter your name:")  
somebody = input() # 콘솔창에서 입력한 값을 somebody에 저장  
print ("Hi", somebody, "How are you today?")
```

### 실행

```
python console_test.py  
Enter your name:  
Sungchul Choi  
Hi Sungchul Choi How are you today?
```

## coma(,) 사용할 경우 print 문이 연결됨

```
>>> print ("Hello World!", "Hello Again!!!") # , 사용
Hello World! Hello Again!!! # 실행 시 두 문장이 연결 돼서 출력됨
```

## 숫자입력 받기

```
temperature = float(input("온도를 입력하세요 :")) # 입력 시 바로 형 변환 하기
print(temperature)
```

**temperature.py**

```
python temperature.py
온도를 입력하세요 : 103
103.0
```

**실행**



# print formatting

**형식(format)에 맞춰서  
출력을 할 때가 있음**

	A	B	C	D	
1	Date	Daily Income	Daily Expenses	Percent Gain/Loss	
2	1-May	\$322.00	\$146.00	221%	
3	2-May	\$371.00	\$135.00	275%	
4	3-May	\$345.00	\$467.00	74%	
5	4-May	\$345.00	\$216.00	160%	
6	5-May	\$150.00	\$269.00	56%	
7	6-May	\$116.00	\$481.00	24%	
8	7-May	\$440.00	\$203.00	217%	

# print 문을 활용해서 결과 formatting 하기

프린트 문은 기본적인 출력 외에 출력의 양식을 형식을 지정 가능

(1) % string (2) format 함수 (3) fstring

```
print(1,2,3)
print("a" + " " + "b" + " " + "c")
print("%d %d %d" % (1,2,3))
print("{} {} {}".format("a", "b", "c"))
print(f"value is {value}")
```

## 일반적으로 %-format 과 str.format() 함수를 사용함

```
print('%s %s' % ('one', 'two'))
```

```
print('{} {}'.format('one', 'two'))
```

```
print('%d %d' % (1, 2))
```

```
print('{} {}'.format(1, 2))
```

## "%datatype" % (variable) 형태로 출력 양식을 표현

```
print("I eat %d apples." % 3)
print("I eat %s apples." % "five")
number = 3; day="three"
print("I ate %d apples. I was sick for %s days."
      % (number, day))
print("Product: %s, Price per unit: %f." % ("Apple", 5.243))
```

# %-format

print formatting

```
print("Art: %5d, Price per Unit: %8.2f" % (453, 59.058))
```

output

String Modulo Operator

```
Art:   453, Price per Unit:   59.06
```

type	설명
%s	문자열 (String)
%c	문자 1개(character)
%d	정수 (Integer)
%f	부동소수 (floating-point)
%o	8진수
%x	16진수
%%	Literal % (문자 % 자체)

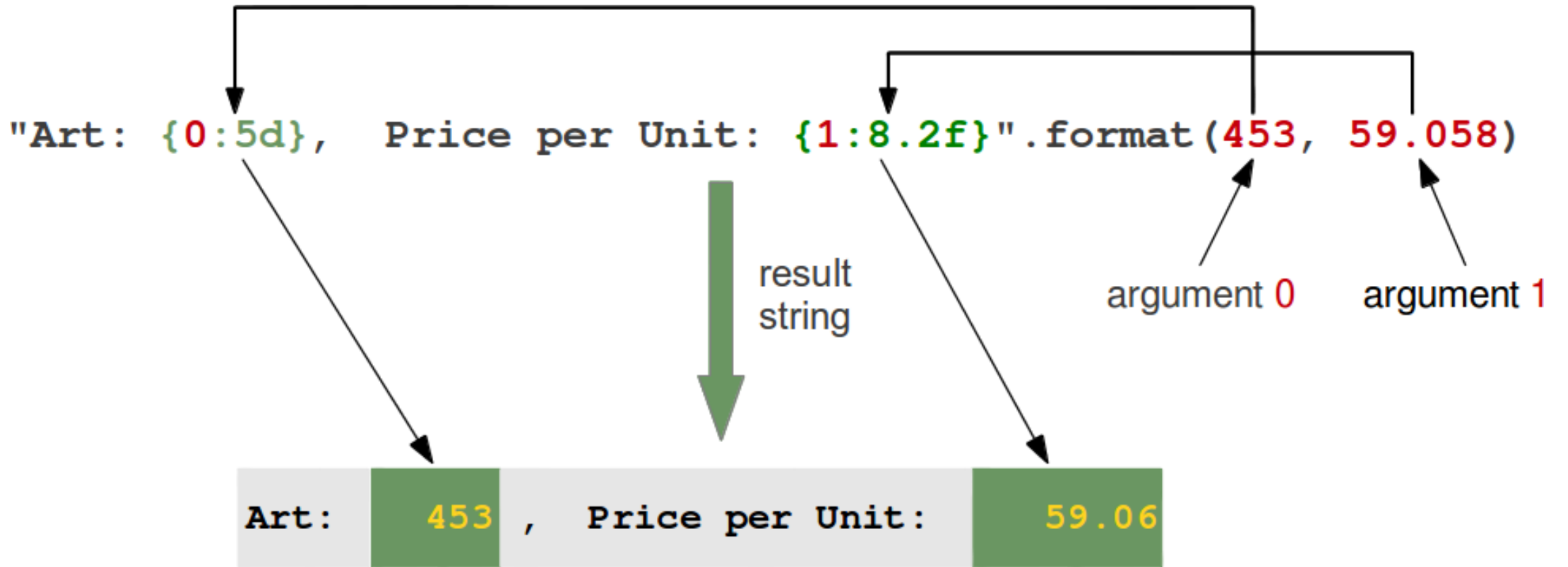


"~~~~{datatype}~~~~".format(argument)

```
age = 36; name='Sungchul Choi'
print("I ' m {0} years old.".format(age))
print("My name is {0} and {1} years old.".format(name,age))
print(
    "Product: {0}, Price per unit: {1:.3f}.".format(
        "Apple", 5.243))
```

# str.format() 함수

print formatting



[https://www.python-course.eu/python3\\_formatted\\_output.php](https://www.python-course.eu/python3_formatted_output.php)

## 여유 공간을 지정하여 글자배열 + 소수점 자릿수를 맞추기

```
print("Product: %5s, Price per unit: %.5f." % ("Apple", 5.243))  
print("Product: {0:5s}, Price per unit: {1:.5f}.".format("Apple", 5.243))  
print("Product: %10s, Price per unit: %10.3f." % ("Apple", 5.243))  
print("Product: {0:>10s}, Price per unit: {1:10.3f}.".format("Apple", 5.243))
```

## 해당 표시할 내용을 변수로 표시하여 입력

```
print("Product: %(name)10s, Price per unit: %(price)10.5f." %  
      {"name": "Apple", "price": 5.243})
```

```
print("Product: {name:>10s}, Price per unit:  
{price:10.5f}.".format(name="Apple", price=5.243))
```

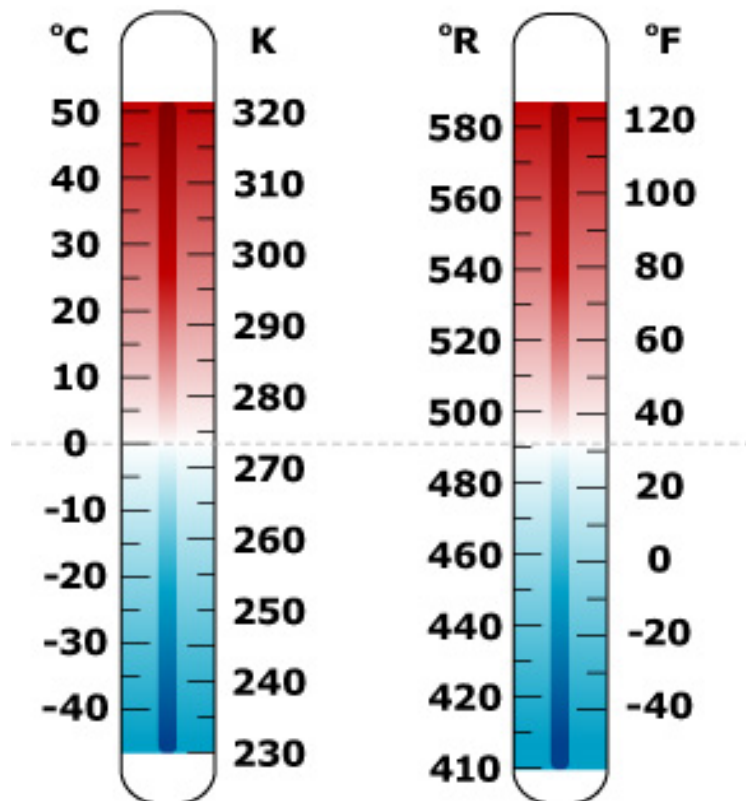
## - python 3.6 이후, PEP498에 근거한 formatting 기법

```
name = "Sungchul"
age = 39
print(f"Hello, {name}. You are {age}.")
print(f'{name:20}')
print(f'{name:>20}')
print(f'{name:*<20}')
print(f'{name:*>20}')
print(f'{name:*^20}')

number = 3.141592653589793
print(f'{number:.2f}')
```

```
Hello, Sungchul. You are 39.
Sungchul
                Sungchul
Sungchul*****
*****Sungchul
*****Sungchul*****
3.14
```

# Lab: Fahrenheit Converter



$$^{\circ}\text{F} = (^{\circ}\text{C} \times 1.8) + 32$$

### 아래와 같이 출력되는 프로그램을 만드시오

```
python fahrenheit.py
```

본 프로그램은 섭씨를 화씨로 변환해주는 프로그램입니다

변환하고 싶은 섭씨 온도를 입력해 주세요:

32.2

섭씨온도 : 32.2

화씨온도 : 89.96

fahrenheit.py

섭씨 온도 변환 공식은:  $((9/5) * \text{섭씨온도}) + 32$



End of Document  
Thank You.