함수 만들기

a = 1  
b = 2  
c = -6  
  
# a \* x^2 + b \* x + c = 0, a != 0 인 x에 관한 2차방정식에 대해,  
# 근의 공식은  
  
r1 = (-b + (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
r2 = (-b - (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
  
print('하는 {} 또는 {}'.format(r1, r2))

++

a = 2

b = -6  
c = -8  
  
# 한 번 더 구하려면  
  
r1 = (-b + (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
r2 = (-b - (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
  
print('하는 {} 또는 {}'.format(r1, r2))

def print\_sqrt():  
    r1 = (-b + (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
    r2 = (-b - (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
  
    print('하는 {} 또는 {}'.format(r1, r2))  
  
a = 1  
b = 2  
c = -6  
  
# a \* x^2 + b \* x + c = 0, a != 0 인 x에 관한 2차방정식에 대해,  
# 근의 공식은  
  
print\_sqrt()  
  
  
a = 2  
b = -6  
c = -8  
  
# 한 번 더 구하려면  
  
print\_sqrt()

def function():  
    print('안녕, 함수!')

def function():  
    print('안녕, 함수!')  
function()

def function():  
    print('안녕, 함수!')  
  
  
print()  
print()  
print()

def function():  
    print('안녕, 함수!')  
  
  
print()  # 내장 함수  
print()  
print()  
  
function()  # 직접 만든 함수  
function()  
function()

def : define, 함수를 "정의"한다.

function : 함수 이름은 function 말고 다른 것도 가능

: : "블럭이 들어갈 자리"

함수 만들기-실습

a = 5  
b = 7  
  
def add():  
    result =  a + b  
    print(result)  
  
add()

매개변수

def print\_root():  
    r1 = (-b + (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
    r2 = (-b - (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
  
    print('하는 {} 또는 {}'.format(r1, r2))  
  
#a = 1  
#b = 2  
#c = -6  
  
# a \* x^2 + b \* x + c = 0, a != 0 인 x에 관한 2차방정식에 대해,  
# 근의 공식은  
  
print\_root()  
  
  
a = 2  
b = -6  
c = -8  
  
# 한 번 더 구하려면  
  
print\_root()

def print\_root(a, b, c):  
    r1 = (-b + (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
    r2 = (-b - (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
  
    print('하는 {} 또는 {}'.format(r1, r2))  
  
a = 1  
b = 2  
c = -6  
  
# a \* x^2 + b \* x + c = 0, a != 0 인 x에 관한 2차방정식에 대해,  
# 근의 공식은  
  
print\_root()  
  
  
a = 2  
b = -6  
c = -8  
  
# 한 번 더 구하려면  
  
print\_root()

def print\_root(a, b, c):  
    r1 = (-b + (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
    r2 = (-b - (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
  
    print('해는 {} 또는 {}'.format(r1, r2))  
  
x = 1  
y = 2  
z = -6  
  
# a \* x^2 + b \* x + c = 0, a != 0 인 x에 관한 2차방정식에 대해,  
# 근의 공식은  
  
print\_root(x, y, z)  
  
  
x = 2  
y = -6  
z = -8  
  
# 한 번 더 구하려면  
  
print\_root(x, y, z)

def print\_root(a, b, c):  
    r1 = (-b + (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
    r2 = (-b - (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
  
    print('해는 {} 또는 {}'.format(r1, r2))  
  
x = 1  
y = 2  
z = -6  
  
# a \* x^2 + b \* x + c = 0, a != 0 인 x에 관한 2차방정식에 대해,  
# 근의 공식은  
  
print\_root(x, y, z)  
  
  
x = 2  
y = -6  
z = -8  
  
# 한 번 더 구하려면  
  
print\_root(x, y, z)  
  
def print\_round(number):  
    rounded = round(number)  
    print(rounded)  
     
print\_round(4.6)  
print\_round(2.2)  
print\_round(3.6)

매개변수-실습

def add(a,b):  
    result = a + b  
    print( "{} + {} = {}".format(a,b,result))  
add(10,5)

함수의 값

def add\_10(value):  
    """value에 10을 더한 값을 돌려주는 함수"""  
    result = value + 10  
    return result  
  
add\_10(42)

def add\_10(value):  
    """value에 10을 더한 값을 돌려주는 함수"""  
    result = value + 10  
    return result  
  
n = add\_10(42)  
print(n)

++

n = round(1.5)  
print(n)

def add\_10(value):  
    """value에 10을 더한 값을 돌려주는 함수"""  
    print('return 앞')  
    return 10 # 즉시 종료  
    print('return 뒤')  
    result = value + 10  
    return result  
  
n = add\_10(42)  
print(n)  
     
n = round(1.5)  
print(n)

def add\_10(value):  
    """value에 10을 더한 값을 돌려주는 함수"""  
    if value < 10:  
        return 10 # 즉시 종료  
     
    result = value + 10  
    return result  
  
n = add\_10(5)  
print(n)  
n = add\_10(42)  
print(n)  
     
n = round(1.5)  
print(n)

def add\_10(value):  
    """value에 10을 더한 값을 돌려주는 함수"""  
    if value < 10:  
        result = 10 # 즉시 종료  
    else:  
        result = value + 10  
    return result  
  
n = add\_10(5)  
print(n)  
n = add\_10(42)  
print(n)  
     
n = round(1.5)  
print(n)

def root(a, b, c):  
    r1 = (-b + (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
    r2 = (-b - (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
  
    return r1  
  
x = 1  
y = 2  
z = -6  
  
# a \* x^2 + b \* x + c = 0, a != 0 인 x에 관한 2차방정식에 대해,  
# 근의 공식은  
  
r = root(x, y, z)  
print('근은 {}'.format(r1))

def root(a, b, c):  
    r1 = (-b + (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
    r2 = (-b - (b \*\* 2 - 4 \* a \* c) \*\* 0.5) / (2 \* a)  
  
    return r1, r2  
  
x = 1  
y = 2  
z = -6  
  
# a \* x^2 + b \* x + c = 0, a != 0 인 x에 관한 2차방정식에 대해,  
# 근의 공식은  
  
r1, r2 = root(x, y, z)

print ('근은 {} {}'.format(r1, r2))

함수의 값-실습

def add(a,b):  
    result = a + b  
    return result