**< 함수1 >**

* **매개변수 입력 받는 여러 방법 (디폴트, 키워드, 가변 매개변수)**

1. 숫자에 대해 제곱을 구하는 함수 **squre**를 정의하고, 다음과 같이 숫자를 콤마(,)로 구분해 입력하면 정의한 함수를 이용해 제곱 값을 출력하는 프로그램을 작성하세요.

ex) 입력 : 2, 3, 4

출력 : squre(2) ==> 4

squre(3) ==> 9

squre(4) ==> 16

* 1. 콤마(,)를 포함하여 입력 받은 값(문자열)을, **콤마(, )기준으로 쪼개서** 리스트 **num\_list**에 저장하는 코드를 작성하세요.
  2. 입력 받은 숫자의 개수가 몇 개인지 구하고 그 개수를 변수 **num**에 저장하는 코드를 작성하세요. (리스트 원소의 개수, len( ) 함수 사용)
  3. 입력 받은 숫자의 개수 **num**과 1-2에서 구한 리스트 **num\_list**를 매개변수로 받는 함수 **squre**를 정의하세요.

(단, return 값은 없으며, 함수 내에서 출력 형식은 위의 ‘ex) squre(2) ==> 4’와 같은 문장을 num만큼 print한다. int( )함수 사용.)

* 1. 함수 **squre**를 호출하세요. (매개변수로는 **num\_list**와 **num**)

1. 다음 코드의 출력 결과를 예측해보세요. (손으로 풀어주세요! IDLE 사용 X)

def total(\*numbers):

tot = 0

for n in numbers:

tot += n

return tot

t = total(1,2)

print(t)

t = total(1,5,2,6)

print(t)

1. 다음 코드의 출력 결과를 예측해보세요. (손으로 풀어주세요! IDLE 사용 X)

def calc(\*numbers):

count = 0

tot = 0

for n in numbers:

count += 1

tot += n

return count, tot

count, sum = calc(1,5,2,6)

print(count, sum)

1. 다음 코드의 출력 결과를 예측해보세요. (손으로 풀어주세요! IDLE 사용 X)

def func(\*args, \*\*dic):

print(args[1])

print(dic['z'])

func(1, 2, 3, x='xx', y='yy', z='zz')

* **호출자에게 return (반환)**

1. 다음 코드의 출력 결과를 예측해보세요. (손으로 풀어주세요! IDLE 사용 X)

def not\_ten(a):

if a == 10:

return

print('%d입니다.' %a)

not\_ten(5)

print(not\_ten(10))

1. 다음의 리스트 **num\_list = [1, 2, 3, 6, 7, 9]**에서 특정 숫자를 찾는 함수 **isIncluded**를 정의하고, 이 함수를 이용해 임의의 숫자의 포함 여부(**True, False**)를 출력(반환)하는 프로그램을 작성하세요. (임의의 숫자 하나로 함수 호출하여 return값 출력하기)

ex) 입력 : 5 🡺 출력(isIncluded(5)) : False

입력 : 9 🡺 출력(isIncluded(9)) : True****