

# 사용자정의 함수 예제

9주차\_02\_01

한 동 대 학 교  
김경미 교수

# 학습목표

2

- ▶ 다양한 사용자 함수 이해하기
- ▶ 다양한 사용자 함수 활용하기

# 사용자 정의 함수 형태

3

Parameter	Return	함수 형태
○	○	Return값 받을변수 =함수이름(Parameter)
X	○	Return값 받을변수=함수이름()
○	X	함수이름(Parameter)
X	X	함수이름()

# 사용자 정의 함수

no parameter, no return value I

```
def print_script():  
    print("Blessed are the poor in spirit,")  
    print("for theirs is the kingdom of heaven.")  
    print(" ")
```

```
print("result of first function call, ")  
print("=" * 25)  
print_script()    # first function call
```

```
def repeat_script():  
    print_script()  
    print_script()
```

```
print("result of second function call, ")  
print("=" * 25)  
repeat_script()    # second function call
```

```
>>>  
===== RESTART: E:/1_Works/2017Work/KMooC  
=====  
result of first function call,  
=====  
Blessed are the poor in spirit,  
for theirs is the kingdom of heaven.  
  
result of second function call,  
=====  
Blessed are the poor in spirit,  
for theirs is the kingdom of heaven.  
  
Blessed are the poor in spirit,  
for theirs is the kingdom of heaven.  
  
>>> |
```

# 사용자 정의 함수

no parameter, no return value 2

```
def Handong1():  
    print("Why not")
```

```
def Handong2():  
    print("the World")
```

```
Handong1()  
print("Change")  
Handong2()
```

```
Why not  
Change  
the World
```

# 사용자 정의 함수

## parameters, no return value I

```
def print_chr(times) :  
    for i in range(times) :  
        print("*")
```

```
print("result of first function call, ")  
print("=" * 25)  
print_chr(5)
```

```
def print_line(t) :  
    print("*" * t)
```

```
print("result of second function call, ")  
print("=" * 25)  
print_line(10)  
print_line(5)
```

```
>>>  
===== RESTART: E:/1_Works/2017  
=====  
result of first dunction call,  
=====  
*  
*  
*  
*  
*  
result of second function call,  
=====  
*****  
*****  
>>> |
```

# 사용자 정의 함수

parameters, no return value 2

```
def repeat(num1,num2):  
    sum = num1+num2  
    print("#"*sum)  
    print("Sum of numbers is",sum)  
    print("#"*sum)
```

```
repeat(7,10)  
repeat(5,6)  
repeat(1,3)
```

```
#####  
Sum of numbers is 17  
#####  
#####  
Sum of numbers is 11  
#####  
####  
Sum of numbers is 4  
####
```

# 사용자 정의 함수

parameters, return value I

```
def plus(num1, num2) :  
    return num1 + num2
```

```
result_plus = plus(2, 12)  
print("result of plus(2, 12) ")  
print(result_plus)
```

```
def mul(num1, num2) :  
    return num1 * num2
```

```
result_mul = mul(2, 12)  
print("result of mul(2, 12) ")  
print(result_mul)
```

```
>>>  
===== RESTART: E:/1_Works,  
=====  
result of plus(2, 12)  
14  
result of mul(2, 12)  
24  
>>>
```



# 사용자 정의 함수

## parameters, return value 2

```
def calcul_age(year):  
    age = 2023-year+1  
    return age
```

```
year = int(input("Enter the year:"))  
print("age is",calcul_age(year))
```

```
Enter the year:1999  
age is 23
```

# 연습문제 I

10

- ▶ 두 수를 파라미터로 받아,
- ▶ 두 수를 서로 더한 값과 곱한 값을 더하여 리턴하는 함수 만들기
- ▶ 파라미터로 보내지는 값은 3과 5이다

# 연습문제 1, 코드

11

```
def multi(num1, num2):  
    add = num1 + num2  
    mul = num1 * num2  
    total = add + mul  
    return total
```

```
result = multi(3,5)  
print(result)
```

# 연습문제 2

12

- ▶ 2개의 수를 파라미터로 받아
- ▶ 2개 수의 최대공약수를 찾는다
- ▶ 결과를 출력하는 함수는 `print_gcd()`, 결과를 리턴하는 함수는 `return_gcd()` 작성하고 한번씩 실행하시오

# 연습문제 2, 코드

13

```
import math
```

```
def print_gcd(num1, num2):  
    print(math.gcd(num1, num2))
```

```
def return_gcd(num1, num2):  
    return math.gcd(num1, num2)
```

```
print_gcd(6, 18)  
print(return_gcd(7, 21))
```

# 강의 요약

14

- ▶ 사용자함수 정의 연습하기
  - ▶ parameter과 return 값의 유무에 따라

# 목표 달성 질문

15

- ▶ 함수의 종류에 대해 설명해 주세요

# 감사합니다