

# 프로그래밍특론

함수

# 함수(FUNCTION)

국민대학교  
이정미 교수

kook906@kookmin.ac.kr

# 학습내용

함수란?

내장함수

매개변수와 인수, 리턴값

기타

# 1. 함수(function)란?

# 함수란?

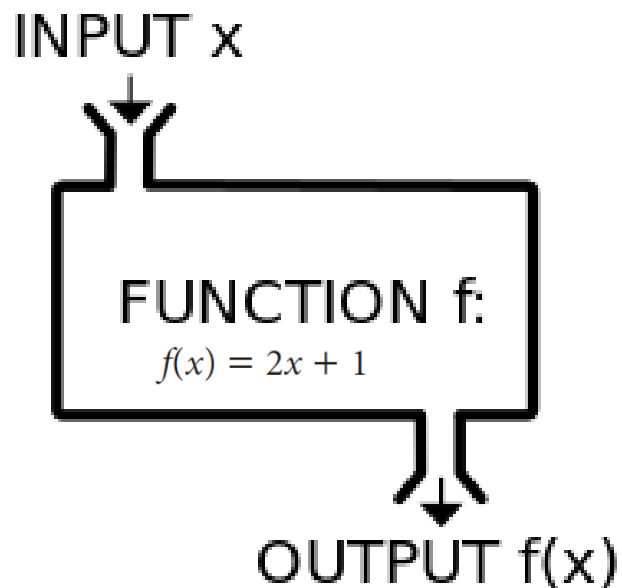
- 정의

- 일정한 작업을 수행하는 코드블럭으로 보통 반복적으로 사용되는 코드들의 묶음
- 여러 입력을 받아 특정한 기능을 수행하고 결과값을 반환하는 코드

- 함수를 사용하는 이유

- 프로그래밍 시, 똑같은 내용을 반복적으로 작성하는 경우가 많음. 반복되는 부분이 있을 경우, 반복되는 부분을 하나로 묶어 "어떤 입력값이 주어졌을 때, 어떤 계산 과정을 거쳐 결과값(리턴값)을 돌려준다"는 내용을 함수로 작성
- 함수 사용 시, 프로그램의 흐름 파악이 용이함. 어떤 계산과정을 하나로 묶어 블럭화한 것이기 때문에, 흐름 파악이 쉽고 오류 발생 시 오류를 찾기도 쉬움

# 입력과 출력이 있는 형태의 함수



# 함수란?

- 사용자 정의 함수의 함수 정의와 함수 호출
  - 사용자 정의 함수(user-defined functions)와 내장 함수(built-in functions)로 구분
  - `print( )`와 `input( )`, `str( )`, `int( )`, `float( )`, `len( )` 등과 같은 내장 함수 사용
- 함수 머리는 키워드 `def`
  - 함수 이름과 괄호, 괄호 사이의 인자들
  - 함수 정의의 몸체인 블록을 예고하는 콜론(:)이 필요
- 함수 몸체는 들여쓰기로 시작
  - 기능을 수행하는 문장들
  - 마지막으로 키워드 `return`에 의한 반환 값을 돌려 주는 구문이 필요
- 인자(arguments)와 `return` 문장은 선택적(optional)

# 함수란?

## 문법

```
def 함수명(입력 값):  
    <수행내용1>  
    <수행내용2>  
    return 결과값
```

```
def name(parameter):  
    <수행내용1>  
    <수행내용2>  
    return 결과값
```



# 입력과 출력이 비어 있는 함수

```
01 def hello() :  
02     print("안녕 파이썬")  
03     print("즐거운 코딩시간이야")  
04  
05 print("Study Python Function")  
06 hello()  
07 print("Understand Function")
```

Study Python Function

안녕 파이썬

즐거운 코딩시간이야

Understand Function

Goodbye라는 함수를 추가해봅시다.  
(Goodbye 함수는 '다음에 또 만나요'라고 출력합니다.)

# 인사하기 함수

```
def 인사하기(name,n):  
    for x in range(n):  
        print("안녕",name)
```

```
인사하기('홍길동',3)
```

```
인사하기('이순신',2)
```

안녕 홍길동

안녕 홍길동

안녕 홍길동

안녕 이순신

안녕 이순신

# 함수 응용하기

```
def hello():  
    print('Hello, Python!')  
  
def helloint():  
    print(77)  
  
print('처음 하는 함수 호출 hello()')  
hello()  
print('함수 호출 hello() 이후')  
  
print(1)  
print(helloint)  
helloint()  
print(3)
```

# 함수 응용하기

```
def hello(name):  
    print('안녕,',name)
```

```
hello('나희')  
sname='기수'  
hello(sname)
```

```
def hello(name='나희'):  
    print('안녕,',name)
```

```
hello()  
hello('기수')
```

# 함수 응용하기

```
def f(x):  
    y=2*x+1  
    return y
```

```
print('f(10)=',f(10))  
print('f(20)=',f(20))  
print('f(30)=',f(30))
```

f(10)= 21  
f(20)= 41  
f(30)= 61

```
def mysum(x,y=0):  
    return x+y
```

```
hap=mysum(5)  
print(hap)
```

```
hap=mysum(5,10)  
print(hap)
```

```
def getprime():  
    return 2,3,4,5
```

```
a,b,c,d = getprime()  
print(a,b,c,d)
```

```
prime=getprime()  
print(prime)
```

## 2. 내장함수

# abs(), pow(), round(), divmod(), min,max,sum

```
print(abs(-3)) #절대값
```

```
print(pow(2,3)) #2의3승 지수(power)
```

```
round(3.1415)
```

```
round(3.1415,3)
```

```
divmod(10,3) # 인자2개의 나누기 몫과 나머지로 구성된 한 쌍의 튜플로 반환 (a//b,a%b)
```

```
min(5,1,10)
```

```
max(5,1,10)
```

```
sum([5,1,10])
```

```
sum([5,1,10],1) # 만약 인자가 2개이면 정수인 두 번째 인자를 더해 반환
```

# sorted(), reversed(),all(),any()

```
count=[20,10,80,50]
```

```
print(sorted(count))
```

```
print(sorted(count,reverse=True))
```

```
b=all([3>4, 'p'in 'py']) # 모든 항목이 참이면 True
```

```
print(b)
```

```
print(all("")) # 항목이 비어 있는 시퀀스는 True 반환
```

```
b=any([3>4, 'p'in 'py']) # 하나의 항목이 참이면 True
```

```
print(b)
```

```
print(any(""))
```



### 3. 매개변수와 인수, 리턴값

# 함수란?

- 매개변수와 인수

- 매개변수: 함수에서 입력으로 전달된 값을 받는 변수
- 인수: 함수를 호출할 때 전달하는 입력값

- 예시

---

```
def sum(a,b):      # a, b는 매개변수
    c = a + b
    return c
```

```
s = sum(5,7)      # 5, 7은 인수
print(s)
```

# 입력값, 리턴값에 따른 함수의 형태

- 4가지 유형

- 입력값과 리턴값이 모두 있는 함수. 가장 일반적인 형태
- 입력값이 없고, 리턴값만 있는 함수
- 입력값이 있고, 리턴값이 없는 함수
- 입력값과 리턴값이 모두 없는 함수

# 입력값, 리턴값에 따른 함수의 형태

- 입력값과 리턴값이 모두 있는 함수

```
File Edit Format Run Options Window Help
def sum(a,b):
    c = a + b
    return c

s = sum(3,4)
print(s)
```

## 실행 결과

7

- 입력값: 3, 4
- 리턴값: 7

# 입력값, 리턴값에 따른 함수의 형태

- 입력값이 없고, 리턴값만 있는 함수

```
File Edit Format Run Options Window Help
def say():
    return "Hello, world"

a = say()
print(a)
```

## 실행 결과

Hello, world

- 입력값: 없음
- 리턴값: Hello, world

# 입력값, 리턴값에 따른 함수의 형태

- 입력값이 있고, 리턴값이 없는 함수

File Edit Format Run Options Window Help

```
def sum(a, b):  
    print(a, b, a+b)
```

```
a = sum(3, 5)
```

## 실행 결과

3 5 8

- 입력값: 3, 5
- 리턴값: 없음

# 입력값, 리턴값에 따른 함수의 형태

- 입력값과 리턴값이 모두 없는 함수

```
File Edit Format Run Options Window Help
def say():
    print('Hello')

say()
```

## 실행 결과

Hello

- 입력값: 없음
- 리턴값: 없음

# 입력값, 리턴값에 따른 함수의 형태

- 입력값과 리턴값이 모두 없는 함수

```
File Edit Format Run Options Window Help
def say():
    print('Hello')

say()
```

## 실행 결과

Hello

- 입력값: 없음
- 리턴값: 없음



# 리턴값에 따른 함수의 형태

# return 차이점

```
def f1(x):  
    a=3  
    b=5  
    y=a*x+b  
    return y
```

```
c = f1(10)  
print(c)
```

35  
35

# return 차이점

```
def f1(x):  
    a=3  
    b=5  
    y=a*x+b  
    print(y)
```

```
d=f1(10)  
print(d) # ?
```

# help()



help()

...

Welcome to Python 3.9's help utility!

If this is your first time using Python, you should definitely check out the tutorial on the Internet at <https://docs.python.org/3.9/tutorial/>.

Enter the name of any module, keyword, or topic to get help on writing Python programs and using Python modules. To quit this help utility and return to the interpreter, just type "quit".

To get a list of available modules, keywords, symbols, or topics, type "modules", "keywords", "symbols", or "topics". Each module also comes with a one-line summary of what it does; to list the modules whose name or summary contain a given string such as "spam", type "modules spam".

help>

## 4. 기타

# 지역 변수와 전역 변수

- 지역(local)과 전역(global) 변수
  - 파이썬에서는 변수가 메모리에 생성되고 유지되는 범위(scope)
- 지역 변수(local variables)
  - 함수 내부에서 대입돼 생성된 변수
  - 지역 변수는 함수 외부에서 절대 사용될 수 없음
- 전역 변수(global variables)
  - 함수 외부에서 대입돼 생성된 변수

```
def addone():  
    # 대입에 사용되는 변수는 지역 변수  
    i = 30 # 지역 변수 생성  
    i += 1 # 지역 변수 수정  
    print('\t 지역 변수 i:', i) # 지역 변수 참조
```

# 람다 함수

- 작고 이름이 없는(익명, anonymous ) 함수

- 키워드 lambda 이후에 콤마로 구분된 인자 목록
- 람다 함수는 여러 개의 인자를 취할 수 있지만 표현식은 하나만 가능
- 키워드 return 없이 하나의 표현식 결과값이 반환

```
#람다함수
```

```
lambda x: x+1
```

```
lambda x,y: x+y
```

```
(lambda x,y:x*y)(10,2)
```

```
print(x) #오류
```

# 표준 라이브러리와 내장 함수

- 라이브러리

- 필요한 기능을 담당하는 함수나 클래스를 모아놓은 모듈(modules)의 집합

- 표준 라이브러리(standard library)

- 파이썬과 함께 설치돼 다른 추가적인 작업 없이 사용할 수 있는 라이브러리
- 내장 함수(built-in function)
  - 표준 라이브러리 중 import 없이 바로 함수 호출로 사용, 예 print( ) 함수

- 써드 파티 라이브러리(third-party library)

- 파이썬 패키지 색인 사이트(pypi.org)에서 제공
  - 파이썬 프로그래밍 언어의 방대한 소프트웨어 저장소
- 교육 부문, 과학 및 수치 컴퓨팅, 빅데이터 및 머신 러닝, 웹 및 인터넷 개발, 그래픽, GUI, 게임 등 매우 다양

# 함수 map(function, iterable, ...)

- 함수 map(function, iterable, ...)

- 이터러블(iterable)의 모든 항목에 function을 적용한 후 그 결과를 돌려주는 이터레이터(iterator)를 반환
- 이터러블
  - 항목을 하나씩 차례로 반환할 수 있는 객체(object)
    - 시퀀스인 문자열, 리스트와 튜플 모두 대표적인 이터러블
  - 이터레이터는 시퀀스인 튜플이나 리스트로 변환해 항목을 활용

```
# map() 항목을 하나씩 차례로 반환
def addone(n):
    return n+1

lst=list(map(addone,[10,30,50,20]))
print(lst)
```

# map 응용하기

```
def add(x,y):  
    return x+y
```

```
lst=list(map(add,[10,30,50,20],[1,3,5,2]))  
print(lst)
```

#결과 [11, 33, 55, 22]

```
lst=list(map(lambda x,y:x+y,[10,30,50,20],[1,3,5,2]))  
print(lst)
```



## map 응용하기2

#map() 원 면적 구하기

```
circle=[3,5,7,10]
```

```
area=list(map(lambda r:r*r*3.14,circle))
```

```
for c,a in zip(circle, area):
```

```
    print('반지름',c,'원면적',a) # ('반지름' => 원면적 {}.format(c,a))
```

# dir( )

- **인자 없는 함수 dir( )**
  - 현재 정의된 변수와 함수 이름의 리스트를 반환
- **파이썬 셸을 처음 실행해 dir( )을 입력**
  - 파이썬 셸에서 사용되는 시스템 변수 리스트
  - 시스템에서 사용하는 모듈이나 변수는 앞뒤에 밑줄이 2개 붙어 `__builtins__`처럼 정의

`dir()`

`dir(__builtins__)`

수고 하였습니다.