

# 데이터처리프로그래밍

## 함수 (function)



강원대학교 교육혁신원 송혜정

<hjsong@kangwon.ac.kr>



# function

## ✓ 학습목표

- 함수를 이해하고 활용한다.

## ✓ 학습내용

- 함수
- 함수 정의(definition)
- 함수 호출(function call)
- 변수 범위 (지역변수, 전역변수)
- 매개변수 전달방식 (call by value/call by reference)
- 기본 매개변수 값 / 키워드 인수 / 가변인수
- 함수 반환 값(return)
- 람다(Lambda)



# 강의에 앞서서..

- 본 강의자료는 아래의 자료들을 참고하여 만들어 졌음을 알립니다
  1. 데이터과학을 위한 파이썬 프로그래밍, 최성철, 한빛아카데미,2019
  2. Python (<https://docs.python.org>)
  3. 점프 투 파이썬 (<https://wikidocs.net/book/1>)
  4. 파이썬 for Beginner, 우재남, 한빛아카데미

# function

- 함수(function)

- 특정기능을 수행할 수 있는 명령들을 묶어 놓은 프로그램 모듈
- 매개변수(parameter)를 이용하여 함수 안으로 값을 입력하고  
처리된 결과를 반환(return)

- 함수의 장점

- 반복 명령을 모듈로 작성하여 필요할 때마다 호출
- 기능별로 프로그램을 분할
- 함수 내부의 처리내용을 모르더라도 함수명과 입출력만으로  
필요한 기능을 수행

# function

- 함수 예
  - 두수를 더하는 plus() 함수정의 및 함수 사용(호출) 예

```
# 함수 정의 부분
#두수를 더하는 함수를 정의하여 호출
def plus(v1, v2):
    result = v1 + v2 #매개변수 v1, v2의 값을 더하여 지역변수 result에 저장
    return result    #result 변수값을 반환

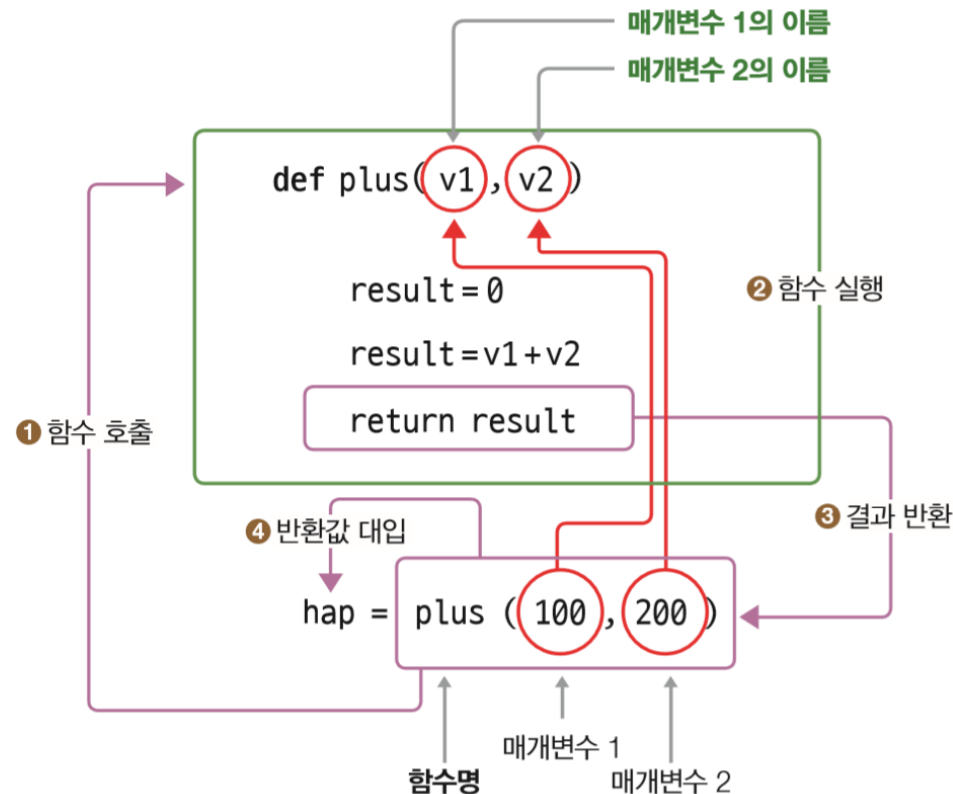
#메인 코드 부분
#함수 호출
hap = plus(100, 200) #plus()함수 호출시 100과 200을 v1, v2에 대입,
                    #함수 처리후 반환값을 hap변수에 대입
print("plus(100,200) = ", hap) #반환 결과 출력
```

plus(100,200) = 300

# function

## • 함수 처리 순서

- (1) 함수호출(매개변수 전달) -> (2) 함수 실행 ->
- (3) 결과 반환 -> (4) 반환값 대입

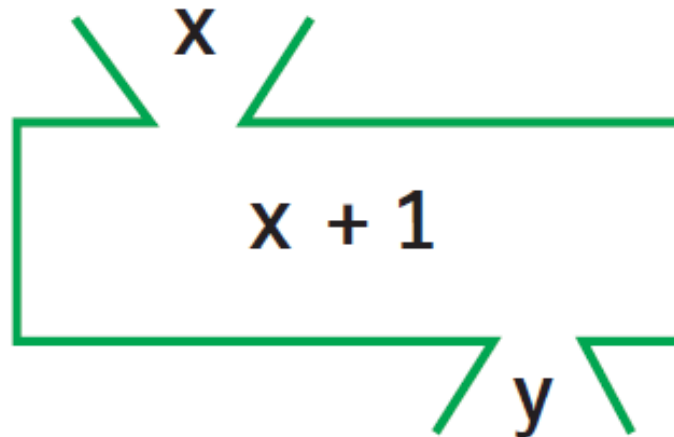


# function

- 수학 함수와 프로그래밍 함수

$$f(x) = x + 1$$

수학함수



```
def f(x)
    y = x + 1
    return y
```

파이썬 함수

# 함수 정의

- 함수의 정의

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

- def : 'definition'의 줄임말, 함수 정의 시작 의미
- 함수 이름 : 기능에 맞게 이름부여, 대부분 동사형으로 명명
- 매개변수(parameter) :
  - 함수 안으로 입력되는 값에 대한 변수
  - 여러 개를 콤마로 연결
- 수행문 :
  - 함수안에서 처리할 명령문들
  - 들여쓰기로 코드블럭 표현



# 함수 정의

- 함수정의 예

#(1)하나의 값을 입력하여 1을 더한 결과를 반환하는 함수 정의

# $f1(x) = x + 1$

#매개변수 1개, 반환값 1개인 함수

```
def f1(x):
    y = x + 1
    return y
```

#(2)두 값을 입력하여 더한 결과를 반환하는 함수 정의

# $f2(x1, x2) = x1 + x2$

#매개변수 2개, 반환값 1개인 함수

```
def f2(x1, x2):
    y = x1 + x2
    return y
```

#(3)다음 수식에 대한 함수 정의

# $f3(x) = x^2 + 2x + 3$

#매개변수 1개, 반환값 1개인 함수

```
def f3(x):
    y = x ** 2 + 2 * x + 3
    return y
```

#(4)키와 몸무게를 입력하여 BMI를 반환하는 수식에 대한 함수 정의

# $BMI = w / (h^2)$

#매개변수 2개, 반환값 1개인 함수

```
def getBMI(w, h):
    bmi = w / (h ** 2)
    return bmi
```

# 함수 정의

- 함수 정의 예

```
#(5)성명을 입력하여 "Hello 성명"을 출력하고 반환하는 함수
#매개변수 1개, 반환값 1개의 함수
def hello(name):
    hname = 'Hello ' + name + ' !!'
    print(hname)
    return hname

#(6)매개변수로 입력된 값이 짝수인지 홀수인지를 반환하는 함수
#매개변수 1개, 반환값 1개의 함수
def getOddEven(n):
    if (n % 2 == 0):
        r = '짝수'
    else:
        r = '홀수'
    return r

#(7)매개변수로 입력된 문자를 옆으로 10번 반복 출력하는 함수
#매개변수 1개의 함수
def printChar(ch):
    for i in range(10):
        print(ch, end='')

#(8) '-' 문자를 20번 반복 출력하는 함수
def printBar():
    for i in range(20):
        print('-', end='')

```

# 함수 호출

- 함수 호출(function call)

변수 = 함수명(매개변수값 #1, ...)

- 정의된 함수를 사용하기 위해 함수 명으로 호출
- 매개변수 값은 함수 호출 시 매개변수로 전달
- 함수의 반환 값은 변수에 저장
- 매개변수가 없는 경우는 함수 명 뒤에 빈 괄호만 사용
- 반환 값이 없는 경우는 변수에 대입하는 부분은 생략

# 함수 호출

- 함수 호출 예

*#(1)하나의 값을 입력하여 1을 더한 결과를 반환하는 함수 정의*

```
def f1(x):  
    y = x + 1  
    return y
```

*#함수 호출1,*

*#함수 f1을 호출할때 값 10이 매개변수 x에 대입*

*#함수안의 명령들이 처리되고 반환값을 y변수에 저장, 출력*

```
y = f1(10)  
print('f1(10)=', y)
```

*#함수 호출2,*

*#함수 f1을 호출할때 값 60이 매개변수 x에 대입*

*#함수안의 명령들이 처리되고 반환값을 y변수에 저장, 출력*

```
y = f1(60)  
print('f1(60)=', y)
```

f1(10)= 11

f1(60)= 61

# 함수 호출

- 함수 호출 예

*#(2)두 값을 입력하여 더한 결과를 반환하는 함수 정의*

```
def f2(x1, x2):
```

```
    y = x1 + x2
```

```
    return y
```

*#함수 호출1,*

*#함수 f2을 호출할때 값 10이 매개변수 x1에 대입, 20이 매개변수 x2에 대입*

*#함수안의 명령들이 처리되고 반환값을 y변수에 저장, 출력*

```
y = f2(10, 20)
```

```
print('f2(10, 20)=', y)
```

*#함수 호출2,*

*#x1,x2를 사용자 입력으로 저장하여*

*#함수 f2을 호출할때 변수값 x1이 매개변수 x1에 대입, x2가 매개변수 x2에 대입*

*#함수안의 명령들이 처리되고 반환값을 y변수에 저장, 출력*

```
x1 = int(input('x1?'))
```

```
x2 = int(input('x2?'))
```

```
y = f2(x1, x2)
```

```
print('f2(x1, x2)=', y)
```

```
f2(10, 20)= 30
```

```
x1?100
```

```
x2?200
```

```
f2(x1, x2)= 300
```

# 함수 호출

## • 함수 호출 예

```
#(4)키와 몸무게를 입력하여 BMI를 반환하는 수식에 대한 함수 정의
#BMI = w / (h ^ 2)
#매개변수 2개, 반환값 1개인 함수
def getBMI(w, h):
    bmi = w / (h ** 2)
    return bmi
#함수 호출1,
#함수 getBMI를 호출할때 80이 몸무게 w로, 1.78이 키 h에 대입
#함수안의 명령들이 처리되고 반환값을 bmi에 저장,출력
bmi = getBMI(80, 1.78)
print('getBMI(80, 1.78)= {0:.2f}'.format(bmi))
```

```
#함수 호출2,
#함수 getBMI를 호출할때 입력한 몸무게를 w로, 입력한 키를 h에 대입
#함수안의 명령들이 처리되고 반환값을 bmi에 저장,출력
while True:    #조건식이 True 이면 무한 반복
    w = float(input('몸무게입력(kg) ? '))
    if ( w == 0 ) :    # 0이면 반복 종료
        print ("반복 종료")
        break
    h = float(input('키입력(m) ? '))
    if (w < 0 or w > 200 or h <= 0 or h > 3):
        print ("몸무게 입력범위(1~200), 키 입력범위(1~3)를 확인하세요")
        continue
    bmi = getBMI(w, h)
    print('getBMI({0:.2f}, {1:.2f})= {2:.2f}'.format(w, h, bmi))
```

```
getBMI(80, 1.78)= 25.25
몸무게입력(kg) ? 55.5
키입력(m) ? 1.66
getBMI(55.50, 1.66)= 20.14
몸무게입력(kg) ? 80
키입력(m) ? 1.78
getBMI(80.00, 1.78)= 25.25
몸무게입력(kg) ? 300
키입력(m) ? 20
몸무게 입력범위(1~200), 키 입력범위(1~3)를 확인하세요
몸무게입력(kg) ? 0
반복 종료
```