

함수 호출

- 함수 호출 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

#함수 호출1,
#함수 hello를 호출할때 'kim'이 매개변수 name에 대입
#함수안의 명령들이 처리되고 반환값을 hname변수에 저장,출력
hname = hello('kim')
print("hello('kim')=", hname)

#함수 호출2,
#함수 hello를 호출할때 '홍길동'이 매개변수 name에 대입
#함수안의 명령들이 처리되고 반환값을 hname변수에 저장,출력
hname = hello('홍길동')
print("hello('홍길동')=", hname)
```

```
Hello kim !!
hello('kim')= Hello kim !!
Hello 홍길동 !!
hello('홍길동')= Hello 홍길동 !!
```

함수 호출

- 함수 호출 예

*#(6)매개변수로 입력된 값이 짝수인지 홀수인지를 반환하는 함수
#매개변수 1개, 반환값 1개인 함수*

```
def getOddEven(n):
    if (n % 2 == 0):
        r = '짝수'
    else:
        r = '홀수'
    return r
```

*#함수 호출1,
#함수 getOddEven을 호출할때 값 5가 매개변수 x에 대입
#함수안의 명령들이 처리되고 반환값을 r변수에 저장, 출력*

```
r = getOddEven(5)
print('getOddEven(5)=', r)
```

*#함수 호출2,
#함수 getOddEven을 호출할때 값 10이 매개변수 x에 대입
#함수안의 명령들이 처리되고 반환값을 r변수에 저장, 출력*

```
r = getOddEven(10)
print('getOddEven(10)=', r)
```

getOddEven(5)= 홀수
getOddEven(10)= 짝수

함수 호출

• 함수 호출 예

```
#(7)매개변수로 입력된 문자를 옆으로 10번 반복 출력하는 함수
#매개변수 1개인 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 호출1,
#함수 printChar 호출할때 '*'이 매개변수 ch에 대입
#함수안의 명령들이 처리
print("printChar('*')=", end = '')
printChar('*')

#함수 호출2,
#함수 printChar 호출할때 '#'이 매개변수 ch에 대입
#함수안의 명령들이 처리
print("\nprintChar('#')=", end = '')
printChar('#')

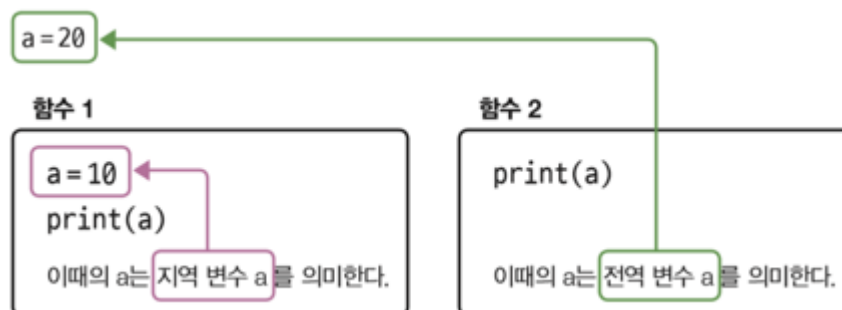
#함수 호출3,
#함수 printBar 호출로 함수안의 명령들이 처리
print("\nprintBar()=' ', end = '')
printBar()
```

```
printChar('*')=*****
printChar('#')=#####
printBar()=-----
```



변수 범위

- 변수 범위(scope)
 - 변수가 생성된 범위 내에서만 사용
- 지역 변수(local variable)
 - 함수 내에서 생성된 변수
 - 해당 함수 내에서만 사용 가능
- 전역 변수(global variable)
 - 함수 외부 본문에서 생성된 변수
 - 프로그램 전체에서 사용 가능
 - 함수 내에서 전역변수를 지정하기 위해 ‘global’ 키워드 사용



변수 범위

```
#변수 범위(scope) 예1
#함수 정의
def f1():
    a = 10    #지역변수 선언
    print ("f1:a=", a)

def f2():
    print ("f2:a=", a)

#메인 코드
# 전역변수 선언
a = 20
f1()    # f1() 함수 호출
f2()    # f2() 함수 호출
print ("a=", a)
```

```
f1:a= 10
f2:a= 20
a= 20
```

```
#변수 범위(scope) 예2
#함수 정의
def f1():
    global a    #전역변수 사용
    a = 10      #전역변수 변경
    print ("f1:a=", a)

#메인 코드
# 전역변수 선언
a = 20
f1()    # f1() 함수 호출
print ("a=", a)
```

```
f1:a= 10
a= 10
```

매개변수 전달방식

• 함수 매개변수 전달방식

종류	설명
값에 의한 호출 (call by value)	<ul style="list-style-type: none"> 함수에 인수를 넘길 때 값만 넘김 함수 안의 인수값 변경 시, 호출된 변수에 영향을 주지 않음
참조 호출 (call by reference)	<ul style="list-style-type: none"> 함수에 인수를 넘길 때 메모리 주소를 넘김 함수 안의 인수값 변경 시, 호출된 변수값도 변경됨

```
#call by value & call by reference
def f(v, r):
    print("f:v=", v) # 매개변수 v 출력
    print("f:r=", r) # 매개변수 r 출력
    v = 100          #값 변경
    r[0] = 200       #리스트의 0번째 요소의 값 변경
    print("f:v=", v) #변경된 값 출력
    print("f:r=", r)

x = 10
y = [10, 20, 30]
f(x, y)             #함수호출,
                    #변수 x는 v변수로 값만 대입 : call by value
                    #리스트 y는 r로 주소를 대입 : call by reference

print("x=", x)      #함수 처리 후 x 출력
print("y=", y)      #함수 처리 후 y 출력
```

```
f:v= 10
f:r= [10, 20, 30]
f:v= 100
f:r= [200, 20, 30]
x= 10
y= [200, 20, 30]
```

기본 매개변수 값

- 기본 매개변수 값 (default parameter value)
 - 함수 정의 시 매개변수에 값을 대입하면 기본 매개변수 값으로 설정

```
# hello 함수, 매개변수 2개(name, upper), upper 매개변수의 디폴트값 False
def hello(name, upper=False):
    print('Hello', name) #전달된 name 그대로 출력
    if upper:
        print('HELLO ', name.upper()) #upper 가 True이면 대문자로 변환하여 출력
    else:
        print('hello ' + name.lower()) #upper 가 False이면 소문자로 변환하여 출력

hello('Jini') #함수호출
hello('GiIDong', True) #함수호출
```

```
Hello Jini
hello jini
Hello GiIDong
HELLO GILDONG
```

키워드 인수

- 키워드 인수(keyword arguments)
 - 함수 호출 시 매개변수에 직접 값을 지정하는 방식

```
# 반지름을 매개변수로 넣어 원의 넓이를 구하는 함수
def circleArea(radius):
    a = 3.14 * radius ** 2
    return a

#반지름 r 변수선언
r = 20.5
#원의 면적 구하는 함수 호출
a = circleArea(r)
#원의 면적 반환 값 출력
out = 'radius={0:.2F} area={1:.2f}'
print(out.format(r, a))

#키워드 인수 사용
#원의 면적 구하는 함수 호출시 매개변수에 직접 값 설정
a = circleArea(radius=10.7)
#원의 면적 반환 값 출력
print(out.format(10.7, a))
```

```
radius=20.50 area=1319.59
radius=10.70 area=359.50
```


키워드 인수

```
# 사각형의 폭, 높이를 매개변수로 넣어 사각형의 면적을 구하는 함수
def recArea(w, h):
    a = w * h
    return a
```

#사각형의 면적 구하는 함수 호출

```
a = recArea(10, 20)
print('recArea(10, 20) =', a)
```

#키워드 인수 사용

#사각형의 면적 구하는 함수 호출시 매개변수에 직접 값 설정

```
a = recArea(h=20, w=10)
print('recArea(h=20, w=10) =', a)
```

recArea(10, 20) = 200

recArea(h=20, w=10) = 200