



## 6.2 函数的使用及调用过程（函数的参数传递、函数的返回值）



# 函数的调用

调用是运行函数代码的方式

```
def fact(n) :  
    s = 1  
    for i in range(1, n+1):  
        s *= i  
    return s
```

函数的定义

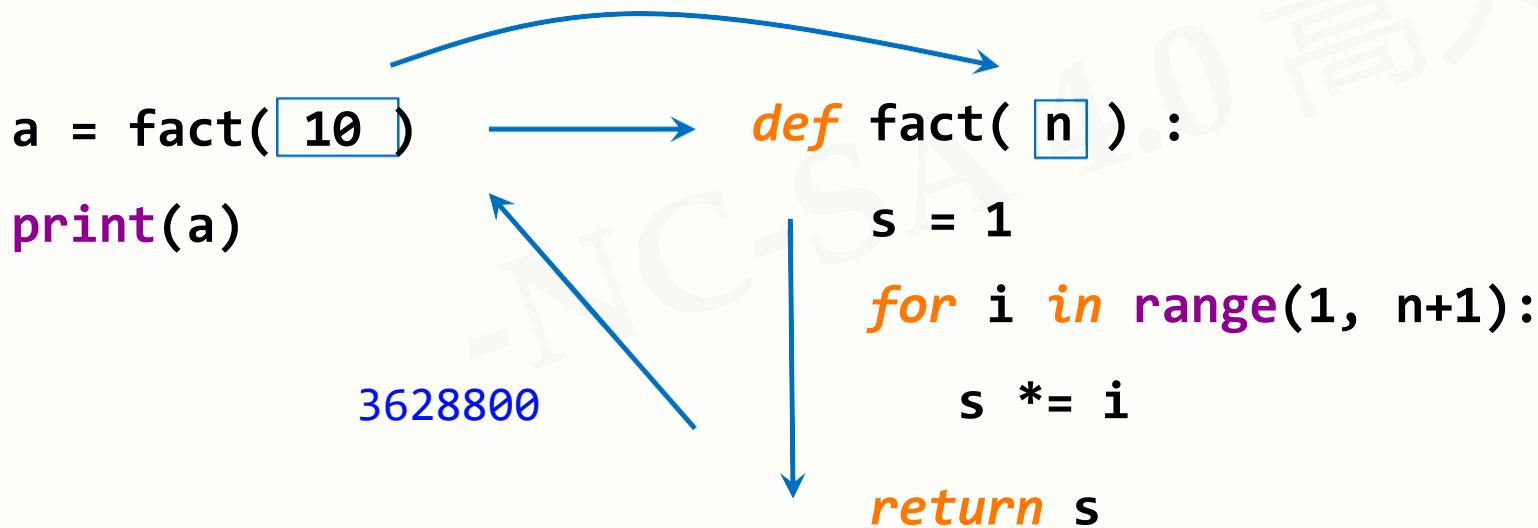
- 调用时要给出实际参数
- 实际参数替换定义中的参数
- 函数调用后得到返回值

fact(10)

函数的调用



# 函数的调用过程





# 函数的参数传递





# 参数个数

函数可以有参数，也可以没有，但必须保留括号

*def* <函数名>() :

<函数体>

*return* <返回值>

*def* fact() :

*print*("我也是函数")



# 可选参数传递

函数定义时可以为某些参数指定默认值，构成可选参数

```
def    <函数名>(<非可选参数>, <可选参数>) :  
  
    <函数体>  
    return    <返回值>
```



# 可选参数传递

可选参数



```
def fact(n, m=1) :
```

```
    s = 1
```

```
    for i in range(1, n+1):
```

```
        s *= i
```

```
    return s//m
```

计算  $n!//m$

```
>>> fact(10)
```

```
3628800
```

```
>>> fact(10,5)
```

```
725760
```



# 可变参数传递

函数定义时可以设计可变数量参数，既不确定参数总数量

*def* <函数名>(<参数>, \*b ) :

<函数体>

*return* <返回值>





# 可变参数传递

可变参数

*def* fact(n, \*b) :

s = 1

*for* i *in* range(1, n+1):

s \*= i

*for* item *in* b:

s \*= item

*return* s

>>> fact(10,3)

10886400

>>> fact(10,3,5,8)

435456000

计算 n!乘数



# 参数传递的两种方式

函数调用时，参数可以按照位置或名称方式传递

```
def fact(n, m=1) :  
    s = 1  
    for i in range(1, n+1):  
        s *= i  
    return s//m
```

>>> fact( 10, 5 )

725760

位置传递

>>> fact( m=5, n=10 )

725760

名称传递



# 函数的返回值





# 函数的返回值

函数可以返回0个或多个结果



PY01B21 名正

- **return**保留字用来传递返回值
- 函数可以有返回值，也可以没有，可以有**return**，也可以没有
- **return**可以传递0个返回值，也可以传递任意多个返回值



# 函数的返回值

函数调用时，参数可以按照位置或名称方式传递

```
def fact(n, m=1) :  
    s = 1  
    for i in range(1, n+1):  
        s *= i  
    return s//m, n, m
```

```
>>> fact( 10, 5 )
```

```
(725760, 10, 5)
```

元组类型

```
>>> a,b,c = fact(10,5)
```

```
>>> print(a,b,c)
```

```
725760 10 5
```