

## 6.2 函数参数

- 函数定义时的参数称为形参，这些参数就像变量一样
- 参数在函数定义的圆括号内指定，用逗号分割。
- 当我们调用函数的时候，函数中的参数值称为实参
- 函数形参取得的值是你调用函数时提供的实参。

# 参数类型

- ◎ 位置参数
- ◎ 关键字参数
- ◎ 默认值参数
- ◎ 数量可变参数

# 位置参数

- ◎ Python处理参数的方式要比其他语言更加灵活。其中，最熟悉的参数类型是位置参数，传入参数的值是按照顺序依次赋值给形参。
- ◎ `from math import sqrt`
- ◎ `def dis(x1,y1,x2,y2): #求平面上两点距离`
- ◎  `print("x1={},y1={},x2={},y2={}".format(x1,y1,x2,y2))`
- ◎  `return sqrt((x1-x2)**2+(y1-y2)**2)`
- ◎ `print(dis(1,3,4,5))`

# 关键字参数

- 为了避免位置参数严格的位置要求，调用参数时可以指定对应形式参数的名字，这是关键字参数，它甚至可以采用与函数定义时不同的顺序。
- `from math import sqrt`
- `def dis(x1,y1,x2,y2): #求平 面上两点距离`
- `print("x1={},y1={},x2={},y2={}".format(x1,y1,x2,y2))`
- `return sqrt((x1-x2)**2+(y1-y2)**2)`
- `print(dis(x1=1,y2=5,y1=3,x2=4))`

# 位置参数和关键字参数混合

- ◎ 如果同时出现两种参数形式， 首先应该写的是位置参数， 然后是关键字参数

- ◎ `from math import sqrt`
- ◎ `def dis(x1,y1,x2,y2): #求平 面上两点距离`
- ◎  `print("x1={},y1={},x2={},y2={}".format(x1,y1,x2,y2))`
- ◎  `return sqrt((x1-x2)**2+(y1-y2)**2)`
- ◎ `print(dis(1,3,y2=5,x2=4))`

- ◎ 下面程序出错

- ◎ `from math import sqrt`
- ◎ `def dis(x1,y1,x2,y2): #求平面上两点距离`
- ◎  `print("x1={},y1={},x2={},y2={}".format(x1,y1,x2,y2))`
- ◎  `return sqrt((x1-x2)**2+(y1-y2)**2)`
- ◎ `print(dis(1,y1=3,4,5))`

# 默认值参数

- ◎ 当调用方没有提供对应形式参数的值时，你可以指定默认形式参数值。如果你提供实参，在调用时会代替默认值
- ◎ `from math import sqrt`
- ◎ `def dis(x1,y1,x2,y2=5): #求平面上两点距离`
- ◎  `print("x1={},y1={},x2={},y2={}".format(x1,y1,x2,y2))`
- ◎  `return sqrt((x1-x2)**2+(y1-y2)**2)`
- ◎ `print(dis(1,3,4))`

# 默认参数值在函数对象被创建时计算

- ⦿ `def init( arg, result=[]):`
- ⦿ `result.append(arg)`
- ⦿ `print( result)`
- ⦿ `init('a')`
- ⦿ `init('b')`
- ⦿ `init(1,[1])`

⦿ 程序输出：

- ⦿ `['a']`
- ⦿ `['a', 'b']`
- ⦿ `[1,1]`

# 数量可变参数

- 当函数参数数目不确定时，星号将一组可变数量的位置参数集成参数值的元组
- `def countnum(a,*b): #计算参数个数`  
    `print(b)`
- `print(len(b)+1)`
- `countnum(3,7,9)`
- `countnum(5,8,1,6,89)`
- 运行结果：
- `(7, 9)`     #以元组的形式存放
- `3`
- `(8, 1, 6, 89)`
- `5`



# 收集参数到字典中--\*\*

- ◎ `def countnum(a,**d): #计算参数个数`
- ◎ `print(d)`
- ◎ `print(len(d)+1)`
- ◎ `countnum(3,x1=9,x2=1,x3=6,x4=89)`
- ◎ 程序输出:
- ◎ `{'x1': 9, 'x2': 1, 'x3': 6, 'x4': 89}`
- ◎ 5

# print函数完整表示

- ④ `print(*object,sep=" ",end="\n",file=sys.stdout)`
- ④ `object`: 输出参数, 可变数量
- ④ 缺省值参数
- ④ `sep=" "`: 输出分割符
- ④ `end="\n"`: 输出函数结束换行
- ④ `file=sys.stdout`: 输出到屏幕缺省

# 实参拆包

- `>>>l=[2,7,5]`
- `>>>print(l)` #参数个数是一个
- `[2, 7, 5]`
- 
- `>>>print(*l)` #实参拆包
- `#print(2,7,5)`
- `2 7 5`
- \*或\*\*都是加在形参的前面，表示不定长参数，分别用来接收不带变量名的多余参数和带有变量名的多余参数，分别将它们以元组和字典的形式接收进函数。
- 当在实参的前面加上\*，就意味着拆包。\*表示将序列拆成一个个单独的实参。

# 可变对象和不可变对象当参数

- ① 可变对象和不可变对象当参数，效果可能是不同的
- ② 当实参是不可变对象时，形参值改变不会影响实参！
- ③ 当实参是可变对象时，形参值改变可能会影响实参！

# 实参和形参（1）

Write code in Python 3.6 (drag lower right corner to resize code editor)

```
1 def change(a,b):
2     a=3
3     b=b+a
4     x,y=7,9
5     change(x,y)
6     print(x,y)
7
8
9 def change1(a):
10     a[0]=3
11     a[1]=11+a[0]
12     l=[7,9]
13     change1(l)
14     print(l)
```

Print output (drag lower right corner to resize)

Frames

Global frame

|        |   |
|--------|---|
| change |   |
| x      | 7 |
| y      | 9 |

Objects

function

change(a, b)

change

|   |   |
|---|---|
| a | 7 |
| b | 9 |

→ line that has just executed  
→ next line to execute

<< First < Back Step 4 of 16 Forward > Last >>

# 实参和形参（2）

Write code in Python 3.6 (drag lower right corner to resize code editor)

```
1 def change(a,b):
2     a=3
3     b=b+a
4 x,y=7,9
5 change(x,y)
6 print(x,y)
7
8
9 def change1(a):
10     a[0]=3
11     a[1]=11+a[0]
12 l=[7,9]
13 change1(l)
14 print(l)
```

Print output (drag lower right corner to resize)

Frames

Global frame

|        |   |
|--------|---|
| change |   |
| x      | 7 |
| y      | 9 |

Objects

function change(a, b)

|              |      |
|--------------|------|
| change       |      |
| a            | 3    |
| b            | 12   |
| Return value | None |

→ line that has just executed  
→ next line to execute

<< First < Back Step 7 of 16 Forward > Last >>

# 实参和形参 (3)

Write code in Python 3.6 (drag lower right corner to resize code editor)

```
1 def change(a,b):
2     a=3
3     b=b+a
4     x,y=7,9
5     change(x,y)
6     print(x,y)
7
8
9 def change1(a):
10     a[0]=3
11     a[1]=11+a[0]
12     l=[7,9]
13     change1(l)
14     print(l)
```

Print output (drag lower right corner to resize)

7 9

Frames

Global frame

change

x 7

y 9

change1

l

change1

a

Objects

function change(a, b)

function change1(a)

list

| 0 | 1 |
|---|---|
| 7 | 9 |

→ line that has just executed

→ next line to execute

<< First < Back Step 12 of 16 Forward > Last >>

# 实参和形参（4）

Write code in Python 3.6 (drag lower right corner to resize code editor)

```
1 def change(a,b):
2     a=3
3     b=b+a
4 x,y=7,9
5 change(x,y)
6 print(x,y)
7
8
9 def change1(a):
10     a[0]=3
11     a[1]=11+a[0]
12 l=[7,9]
13 change1(l)
14 print(l)
```

Print output (drag lower right corner to resize)

7 9

Frames

Global frame

change

x 7

y 9

change1

l

Objects

function change(a, b)

function change1(a)

list

| 0 | 1  |
|---|----|
| 3 | 14 |

change1

a

Return value None

→ line that has just executed

→ next line to execute

<< First < Back Step 15 of 16 Forward > Last >>