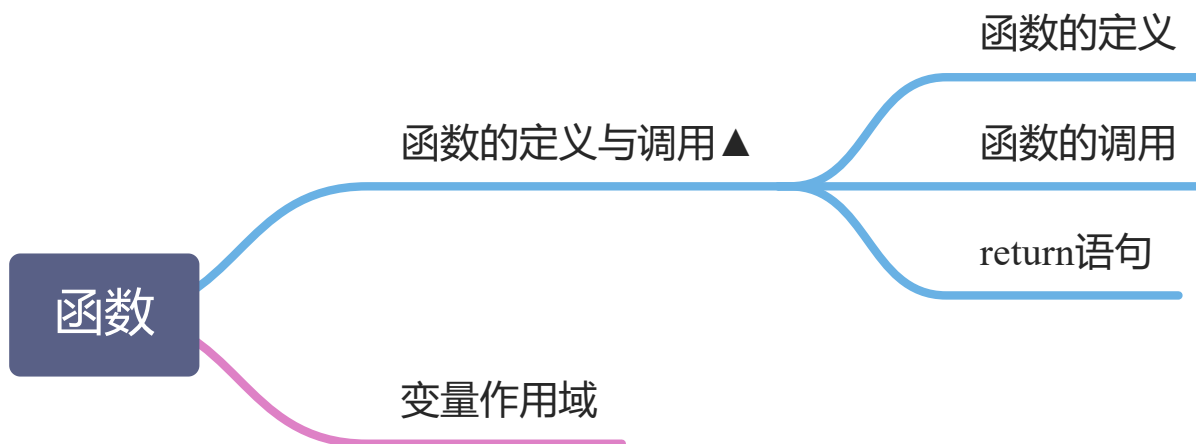


第9课 函数

一、课程结构导图



注：▲为重点知识点。

二、知识点讲解

2.1 函数定义与调用

2.1.1 函数的定义

概念：函数是组织好的、可以重复使用的、用来实现**单一功能**的代码。函数的定义包含函数名，参数，函数体，return语句。

定义函数的语法

```
1  def 函数名 (参数1,参数2.....参数n):  
2      函数体  
3      return 语句
```

by 风变编程

用法1： 自定义函数。

示例1：

```
1 # 定义函数  
2 def greet():  
3     print('早上好! ')  
4     return # 函数返回值为None
```

注意：

1. 函数在定义时，代码末尾要加**冒号**，函数体要整体**缩进**（包括return语句）。
2. 在定义函数时，若不添加return语句或return语句不添加任何返回值时，会默认函数返回的值为None值。

2.1.2 函数的调用

概念： 函数的调用包含函数执行和参数传递这两个操作。一般在定义函数后，会先进行函数调用的操作，运行函数，在运行代码的同时，进行参数传递操作。

用法1：直接输入函数名，调用函数。

示例1：

```
1 # 定义函数
2 def greet():
3     print('早上好! ')
4     return
5
6 # 调用greet()函数
7 greet()
```

注意：

1. 函数定义后是不会自动执行，需要使用函数名调用后才会执行。
2. 调用函数的时候，调用格式为**函数名(参数名)**，()不能漏掉。
3. 若在定义函数时，带有参数，则在调用函数时，**必须要给参数进行赋值**。例如定义函数改为def greet(name)，则在调用函数时，调用语句就变为greet(name)，参数name必须要赋值，如name='陈明'，greet(name)。

用法2：四种参数类型（位置参数，默认参数，不定长参数，关键参数）的传递方式，参数数量可以为0个或多个，多个参数间需用逗号隔开即可。

示例2：

```
1 # 代码一（位置参数）
2 def greet(name):
3     print(name + '早上好! ')
4
5 greet('小明')
6
7 # 代码二（默认参数）
8 def greet(name='小明'):
9     print(name + '早上好! ')
10
11 greet()
12
13 # 代码三（不定长参数）
14 def greet(*name):
15     print(name)
```

```
16
17 greet('小明', '小红')
18
19 #代码四（关键参数）
20 def greet(name1, name2):
21     print(name1, name2)
22
23 greet(name2='小明', name1='小红')
24
25 # 打印结果对比；
26 # 代码一和代码二，打印结果为小明，早上好！
27 # 代码三，打印结果为（'小明', '小红'）
28 # 代码四，打印结果为小红，小明
```

注意：

1. 代码一：位置参数，可以简单理解为‘对号入座’，函数定义的**参数数量**必须与传递的**参数值数量**一致。
2. 代码二：默认参数，**必须放在位置参数的后面**，而且在调用函数时，无须与对应的参数名传递值。
3. 代码三：不定长参数，格式为星号（*）+参数名，以**元组**的形式将参数值打包，一起传递给函数。
4. 代码四：关键字参数，可以简单理解为‘带号入座’，与位置参数不同的是，关键字参数在传递时，会根据参数名，将对应的进行参数值传递。而位置参数则只会根据参数顺序——传递。

2.1.3 return语句

概念： return语句是函数返回值的载体。return是返回的意思，当函数执行完后，计算机会给函数返回一个值，返回的值由return语句决定，例如return 1，意思就是给函数返回的值为1，任何函数都有返回值。

用法： return+返回值。

示例：

```
1 # 代码一（单个返回值）
2 def greet(name):
3     print(name + '早上好! ')
4     return 1
5
6 print(greet('小明'))
```

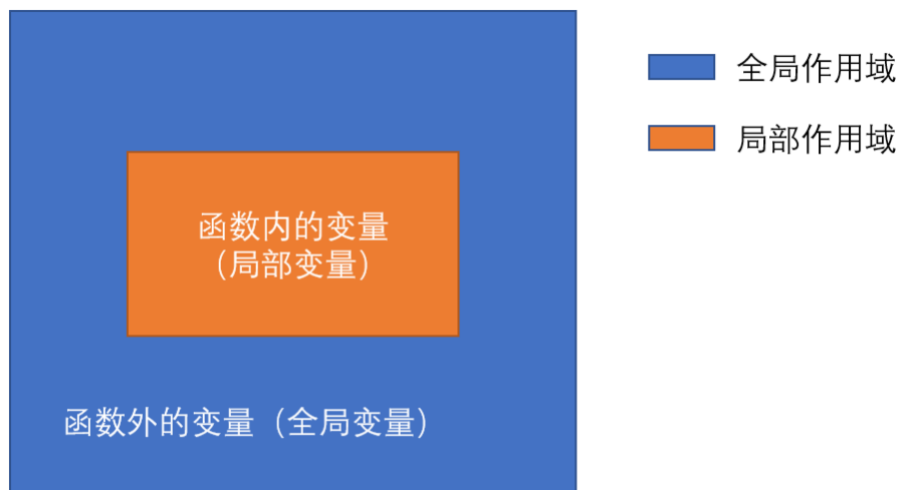
```
7
8 # 代码二 (多个返回值)
9 def greet(name):
10     print(name + '早上好! ')
11     return 1,2
12
13 print(greet('小明'))
14
15 # 代码三 (返回值为None)
16 def greet(name):
17     print(name + '早上好! ')
18     return
19
20 print(greet('小明'))
21
22 # 打印结果对比:
23 # 代码一, 打印结果为1
24 # 代码二, 打印结果为(1,2)
25 # 代码三, 打印结果为None
```

注意:

1. 第8行代码, **当多个返回值时 (两个或两个以上) , 返回函数的值, 以元组的形式返回。**
2. return语句只用于返回相应的值, 不具有打印功能, 如需打印函数返回值, **格式为: print(函数名(参数名))。**
3. **当执行完return语句后, 函数会自动结束, 不会再往下继续执行。**如return 1, print(123), 当执行完return 1后, print语句则不会被执行, 此时函数已经结束。

2.3 变量的作用域

概念: 在函数内部定义的变量, 称为局部变量, 只能作用于函数内部 (相当于是私人物品) 。在函数外部定义的变量, 称为全局变量 (相当于是公共物品) 。



注：不能在全局作用域调用局部变量

用法1：局部变量和全局变量的打印。

示例1：

```
1 # 代码一
2 a = 200
3 def car():
4     a = 100
5     print(a) # 打印局部变量a
6
7 car()
8 print(a) # 打印全局变量a
9
10 # 打印结果是100, 200
```

注意：若将代码中的a=200去掉，则会报错。由于a=200是全局作用域的全局变量，去掉之后，全局作用域就没有全局变量a，只有局部变量a，但由于全局作用域不能调用局部变量，报错信息显示变量a没有被定义。

用法2：用global关键字定义全局变量或者修改全局变量。

示例2：

```
1 # 代码一（声明全局变量）
2 def B():
3     global number # 定义全局变量number
4     number = 100
5
6 B()
7 print(number) # 打印全局变量number
8
9
10 # 代码二（修改全局变量）
11 number = 150
12 def B():
13     global number # 修改全局变量number
14     number = 200
15
16 B()
17 print(number) # 打印全局变量number
18
19 # 打印结果对比：
20 # 代码一，打印结果为100
21 # 代码二，打印结果为200
```

注意：

1. 如全局变量在没有被定义的情况下，global的作用是定义全局变量（代码一），如全局变量在已经有被定义的情况下，global的作用是修改全局变量（代码二）。
2. **global无论是在定义全局变量还是修改全局变量，都不能在定义全局变量或者修改全局变量时，对变量进行赋值操作。**例如示例2的代码一，如将global = number，改写为global number = 100，将会报语法错误。

三、巩固练习

1. （单项选择题）以下代码运行结果正确的选项是（）？

```
1 name = '易小川'
2 def A():
3     global name
```



```
4     print('穿越前胡歌的名字是: '+name)
5     name = '蒙毅'
6     print('穿越后胡歌的名字是: '+name)
7 A()
```

- A、 穿越前胡歌的名字是：易小川
穿越后胡歌的名字是：蒙毅
- B、 穿越前胡歌的名字是：易小川
穿越后胡歌的名字是：易小川
- C、 穿越前胡歌的名字是：蒙毅
穿越后胡歌的名字是：蒙毅
- D、 穿越前胡歌的名字是：蒙毅
穿越后胡歌的名字是：易小川

2. (单项选择题) 以下代码运行结果正确的选项是 () ?

```
1 def math(x):
2     y = x ** 2 + x
3     return y
4 a = math(10)
5 print(a)
```

- A、 110 B、 30 C、 None D、 140

3. (实操题)

题目要求：小明是一位开便利店的老板，小明现在需要进货商品，想编写一个函数代码，来计算出进货商品的总价（商品单价×进货数量），并显示出进货的商品名称，商品单价，商品数量和商品总价这四个关键信息。

风变科技

风变科技