

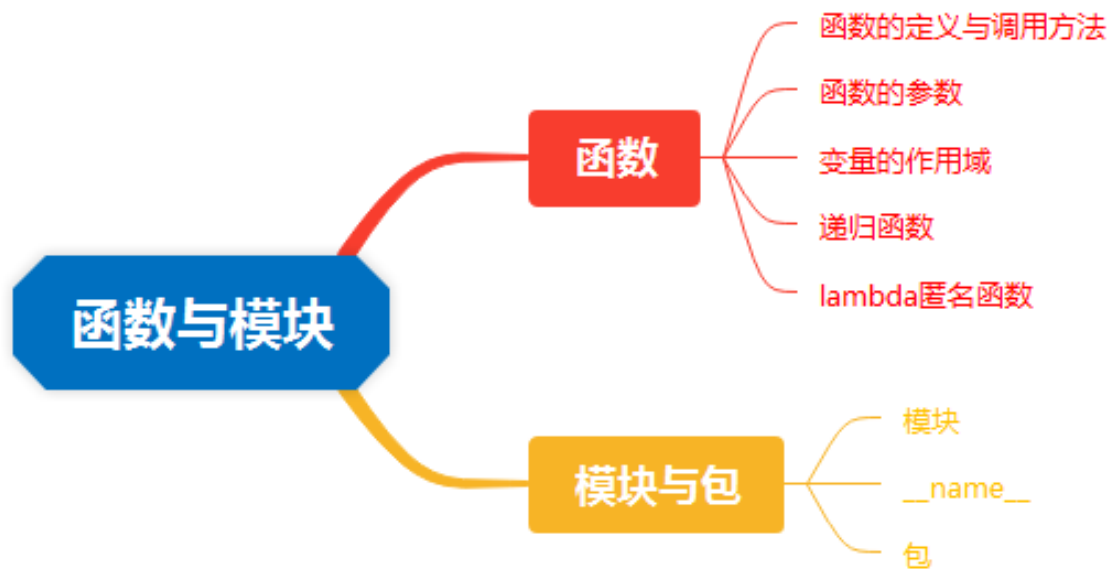
A faint, light-colored world map is visible in the background of the slide, centered behind the title and author text.

# 第4章 函数与模块

史秋实

# 目录

## Contents





01

函数

## 1.函数的定义与调用方法

函数（Function）即功能，是组织好的、可重复使用的、用来实现单一或某些相关联功能的代码段。Python中的函数分为系统函数和自定义函数两类。

系统自带的函数功能即前文介绍的内置函数，是系统已经写好的，比如输出函数`print()`、输入函数`input()`、最大值函数`max()`、计数函数`count()`、类型函数`type()`等，此处不再赘述本章主要学习自定义函数。

## 1.函数的定义与调用方法

自定义函数的语法格式:

```
def 函数名(a1, a2, .....):
```

```
    语句序列
```

```
    [return x]
```

其中，def是系统保留字，函数体相对于def关键字必须保持一定的缩进，语句序列是函数体，return x 是返回值。要注意的是：在定义函数时，参数可以有，也可以没有，return也可以缺失。

## 2.函数的参数

函数的参数分为形参和实参。

- 形参全称为“形式参数”是在定义函数名和函数体的时候使用的参数，目的是用来接收调用该函数时传递的参数。它仅仅是形式上的参数，表明一个函数里面哪个位置有哪个参数而已，不代表具体的值。
- 实参全称为“实际参数”，是一个实际存在的参数，可以是字符串或是数字等。一般出现在函数调用的时候，需要传递具体的值。

## 2.函数的参数

### (1) 位置参数

调用函数时，在函数的参数位置按顺序输入了相应的数据，即位置参数。

**例4-5：**编写函数，功能是将前 2 个参数相加，减去第 3 个参数，返回结果，并调用。

```
def calculate(a,b,c):  
    return a+b-c  
print(calculate(3, 4, 2))
```

## 2.函数的参数

### (2) 关键字参数

如果一个函数有多个参数，在调用时，若不想按位置顺序提供实参，那么可以通过按形参的变量名来传递参数。这种参数传递方式称为关键字参数。关键字参数必须跟在位置参数之后，并且它们的顺序并不重要。

**例4-6：**编写函数，功能是将前 2 个参数相加，减去第 3 个参数，返回结果，并调用。

```
def calculate(a,b,c):  
    return a+b-c  
print(calculate(3,c=2,b=4))
```

## 2.函数的参数

### (3) 默认值参数

在自定义函数时，形参可以指定默认值。若形参有默认值，要将其放置于关键字参数之后。

**例4-7：**编写函数，功能是将前 2 个参数相加，减去第 3 个参数，返回结果，并调用。要求默认第2个参数为5。

```
def calculate(a,c,b=5,):  
    return a+b-c  
print(calculate(3,2))
```

## 2.函数的参数

### (4) 可变长参数

Python中可变长度参数有两种形式，分别是\*args和\*\*kwargs。\*args将接收到的多个实参放在一个元组中，\*\*kwargs将接收到的键值对实参存入字典中。

**例4-8：** 定义函数：包含3个参数和不定长参数\*args，并分别输出3个参数和\*args。

```
def demo(a, b, c, *p):  
    print(a, b, c)  
    print(p)  
demo(1, 2, 3, 4, 5, 6)
```

### 3.变量的作用域

变量的作用域就是变量起作用的范围，或者说是可以被哪部分程序访问，也可称为命名空间。一个变量在函数内部和外部定义的，其作用范围不同，根据程序中变量所在的位置和作用范围，变量分为局部变量和全局变量。

局部变量可以说就是指函数内部定义的变量，仅在函数内部作用，当函数退出时，变量将不存在。全局变量指在函数外定义的变量，在整个程序执行过程中有效。

全局变量如若要在函数内部使用，须在函数内部先用关键字global声明，格式为：

```
global <全局变量>
```

## 4.递归函数

在函数的定义中，函数内部的语句调用函数本身，为递归函数。

**例4-12：**使用递归函数来定义计算阶乘的函数。

```
def jiecheng(n):  
    if n==0:  
        return 1  
    else:  
        return n * jiecheng(n - 1)  
  
#调用  
print(jiecheng(8))
```

## 5. lambda匿名函数

Python 使用 lambda语句来创建匿名函数。lambda只是一个表达式，函数体比def简单很多。

lambda函数的语法只包含一个语句：

<函数名>=lambda <参数列表>:<表达式>

**例4-13：**定义f=lambda x:x+1，并调用。

```
f=lambda x:x+1
```

```
print(f(3))
```



02

# 模块与包

## 1.模块

为了编写代码复用，只有函数还不够，因为函数的作用范围只在当前的python文件内，在另一个文件则无法使用之前定义好的函数功能。这时，建立专门的python文件来保存自己定义好的函数，再通过调用此文件来实现函数功能，这个python文件就是模块。

模块的创建，就是新建一个python文件，并在文件中自定义多个函数，保存好文件。

# 1.模块

模块文件创建好后，有3种导入与调用模块的方式：

- 第1种：import 模块名[as 模块别名]

调用模块：模块名.函数名，或模块别名.函数名

- 第2种：from 模块名 import 函数名

调用模块：直接使用模块中导入的函数

- 第3种：from 模块名 import \*

调用模块：直接使用模块中的所有函数

## 2. \_\_name\_\_

Python中为了区分代码块是单独运行，还是作为模块导入到另一个代码中进行运行，通过对模块的\_\_name\_\_属性值的判断来进行识别。

模块作为单独的程序运行时，\_\_name\_\_的属性值是“\_\_main\_\_”，而作为模块导入时，\_\_name\_\_属性的值就是该模块的名字了。

因此，在编写模块的py程序文件时，都需要在最后增加if \_\_name\_\_ == '\_\_main\_\_' 的判断，在if语句块内，通常会写入执行测试的代码，或示例用法等内容。

## 3.包

包 (Packages) 可以理解为一组模块的容器，并用“包.模块”的方式来构建命名空间。以文件系统来类比的话，可以将包视为文件系统上的目录，而将模块视为目录中的文件，包中必须包含一个名为“\_\_init\_\_.py”的文件，

“\_\_init\_\_.py”文件的内容可以是空，仅用于表示该目录是一个包，也可以写入一些初始化代码。包也可以嵌套，即把子包放在某个包内。

如何能在其他项目文件中使用我们自定好的包中的模块文件呢？将包的文件夹拷贝到python安装目录下的Lib文件夹，即可在其他项目文件中使用。

引用包 (Package) 中的模块：

方法一：`from 包.模块 import 函数`

方法二：`from 包 import 模块`