

# PYTHON

## 编程基础

# 模块概述和import语句

# 概述

- Python提供了交互式和脚本式两种运行方式。当要执行的代码比较长、且需要重复使用时，我们通常将代码放在扩展名为.py的Python脚本文件中。
- 当我们要编写一个规模比较大的程序时，如果将所有代码都放在一个脚本文件中，则不方便维护和多人协同开发。
- 对于可以在多个程序中重用的功能，我们也最好将其放在单独的脚本文件中，以方便多个程序通过引用该脚本文件、共享这些功能。
- 此时，我们需要按照代码功能的不同，将代码分门别类地放在不同的脚本文件中，这些脚本文件就称为是模块（Module）。

# import语句

- 当要使用一个模块中的某些功能时，我们可以通过import方式将该模块导入。
- 例如，假设模块A中定义了一些变量和函数，如果希望在模块B中使用，则可以在模块B中通过import将模块A导入，此时在模块B中就可以使用这些变量并调用模块A的所有函数。
- 语法格式：

```
import module1  
import module2  
...  
import moduleN  
或  
import module1, module2, ..., moduleN
```

## 例：模块使用方法示例



fibonacci.py

```
1 def PrintFib(n): #定义函数PrintFib, 输出斐波那契数列的前n项
2     a, b = 1, 1 #将a和b都赋为1
3     for i in range(1,n+1): #i的取值依次为1,2,...,n
4         print(a, end=' ') #输出斐波那契数列的第i项
5         a, b = b, a+b #更新斐波那契数列第i+1项的值, 并计算第
        i+2项的值
6     print() #输出一个换行
```

## 例：模块使用方法示例



fibonacci.py

```
7 def GetFib(n): #定义函数GetFib, 返回斐波那契数列的前n项
8     fib=[] #定义一个空列表fib
9     a, b = 1, 1 #将a和b都赋为1
10    for i in range(1,n+1): #i的取值依次为1,2,...,n
11        fib.append(a) #将斐波那契数列的第i项存入列表fib中
12        a, b = b, a+b #更新斐波那契数列第i+1项的值, 并计算第
        i+2项的值
13    return fib #将列表fib返回
14    PrintFib(10) #调用PrintFib输出斐波那契数列前10项
15    ls=GetFib(10) #调用GetFib函数获取斐波那契数列前10项组成
        的列表
16    print(ls) #输出列表ls中的元素
```

## 例：模块使用方法示例



testfibonacci.py

- 1 import fibo #导入fibo模块
- 2 fibo.PrintFib(5) #调用fibo模块中的PrintFib函数，输出斐波那契数列前5项
- 3 ls=fibo.GetFib(5) #调用fibo模块中的GetFib函数，得到斐波那契数列前5项的列表
- 4 print(ls) #输出ls中保存的斐波那契数列前5项

1 1 2 3 5 8 13 21 34 55

[1, 1, 2, 3, 5, 8, 13, 21, 34, 55]

1 1 2 3 5

[1, 1, 2, 3, 5]

提示：导入模块后，如果要使用该模块中定义的标识符，则需要通过“模块名.标识符名”的方式。