

PYTHON

编程基础

全局变量__name__ 和系统模块

全局变量__name__

- 每个模块中都有的一个全局变量__name__。
- __name__的作用是获取当前模块的名称，如果当前模块是单独执行的，则其__name__的值就是__main__；否则，如果是作为模块导入，则其__name__的值就是模块的名字。

全局变量__name__



例如：

module.py脚本文件

1 print(__name__) #输出全局
变量__name__的值

testmodule.py脚本文件

1 import module #导入module
模块

提示：当我们执行module.py时，会在屏幕上输出：__main__；而当我们执行testmodule.py时，则会在屏幕上输出：module。

即module.py单独运行和作为模块导入时其__name__的值是不同的。

全局变量__name__



修改后的fibonacci.py脚本文件

```
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
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
```

全局变量__name__



修改后的fibonacci.py脚本文件

```
14     if __name__ == '__main__': #只有单独执行fibonacci.py时该条件才成立
15         PrintFib(10) #调用PrintFib输出斐波那契数列前10项
16         ls=GetFib(10) #调用GetFib函数获取斐波那契数列前10项组成的列表
17         print(ls) #输出列表ls中的元素
```



当执行fibonacci.py时，将在屏幕上输出如下结果：

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



当执行testfibonacci.py时，将在屏幕上输出如下结果：

```
1 1 2 3 5
[1, 1, 2, 3, 5]
```

系统模块

可以直接导入系统提供的模块，使用其中的功能。

例：通过sys模块获取运行Python脚本时传入的参数。



修改后的testfibonacci.py脚本文件

```
1 import fibo #导入fibo模块
2 import sys #导入系统提供的sys模块
3 n=int(sys.argv[1]) #通过sys模块的argv获取执行脚本时传
   #入的参数
4 fibo.PrintFib(n) #调用fibo模块中的PrintFib函数，输出斐
   #波那契数列前n项
5 ls=fibo.GetFib(n) #调用fibo模块中的GetFib函数，得到斐
   #波那契数列前n项的列表
6 print(ls) #输出ls中保存的斐波那契数列前n项
```

系统模块

执行 “python testfib.py 5”

1 1 2 3 5
[1, 1, 2, 3, 5]

执行 “python testfib.py 10”

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