

PYTHON

编程基础

from...import

语法格式

除了使用import将整个模块导入，也可以使用from import将模块中的标识符（变量名、函数名等）直接导入当前环境，这样我们在访问这些标识符时就不再需要指定模块名。

语法格式：

```
from 模块名 import 标识符1, 标识符2, ..., 标识符N
```

示例

例：直接导入模块中的标识符



testfibonacci.py脚本文件

```
1 from fibo import PrintFib, GetFib #导入fibo模块中的
  PrintFib和GetFib
2 PrintFib(5) #忽略fibo模块名直接调用PrintFib函数
3 ls=GetFib(5) #忽略fibo模块名直接调用GetFib函数
4 print(ls) #输出ls中保存的斐波那契数列前5项数据
```

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

示例

例：只导入一个模块中的部分标识符。



testfibonacci.py脚本文件

```
1 from fibo import PrintFib #只导入fibo模块中的PrintFib  
2 PrintFib(5) #忽略fibo模块名直接调用PrintFib函数
```

1 1 2 3 5

示例

提示：如果要导入一个模块中的所有标识符，也可以使用 “from 模块名 import * ” 的方式。例如，对于 “from fibo import PrintFib, GetFib” ，可以直接改为 “from fibo import * ” 。

如果一个模块定义了列表__all__，则from 模块名 import * 语句只能导入__all__列表中存在的标识符。例如，对于fibo.py脚本文件，如果在第一行加入__all__列表的定义__all__=['PrintFib']，则通过 “from fibo import * ” 只能导入fibo模块中的PrintFib、而不会导入GetFib。

示例

无论是利用import导入模块、还是用from import导入模块中的标识符，在导入的同时都可以使用as为模块或标识符起别名。

例如：



testfibonacci.py脚本文件

```
1 import fibo as f #导入fibo模块，并为fibo起了个别名f
2 f.PrintFib(5) #调用fibo模块中的PrintFib函数，输出斐波那契数列前5项
```

示例



testfibonacci.py脚本文件

- 1 `from fibo import PrintFib as pf` #导入fibo模块中的PrintFib, 并重命名为pf
- 2 `pf(5)` #调用fibo模块中的PrintFib函数, 输出斐波那契数列前5项