# 一 模块

## 1简介

导入：import module1,mudule2...

调用：模块名.函数名

有时候我们只需要用到模块中的某个函数，只需要引入该函数即可，此时可以用下面方法实现：

from 模块名 import 函数名1,函数名2....

Python的from语句让你从模块中导入一个指定的部分到当前命名空间中：

from modname import name1[, name2[, ... nameN]]

把一个模块的所有内容全都导入到当前的命名空间也是可行的，只需使用如下声明：

from modname import \*

## 2 模块定位

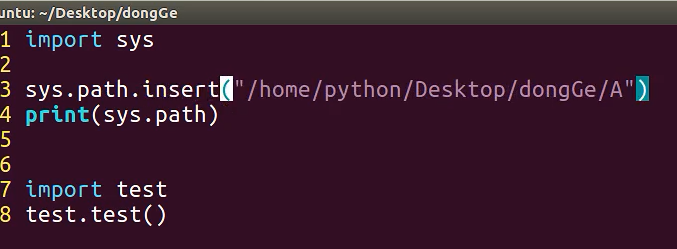
当你导入一个模块，Python解析器对模块位置的搜索顺序是：

当前目录

如果不在当前目录，Python则搜索在shell变量PYTHONPATH下的每个目录。

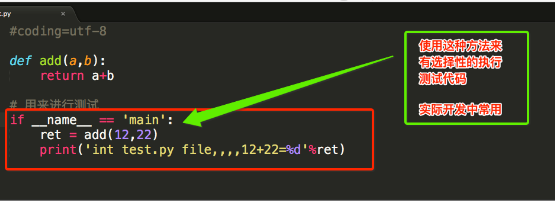
如果都找不到，Python会察看默认路径。UNIX下，默认路径一般为/usr/local/lib/python/

模块搜索路径存储在system模块的sys.path变量中。变量里包含当前目录，PYTHONPATH和由安装过程决定的默认目录。



## 3 测试模块

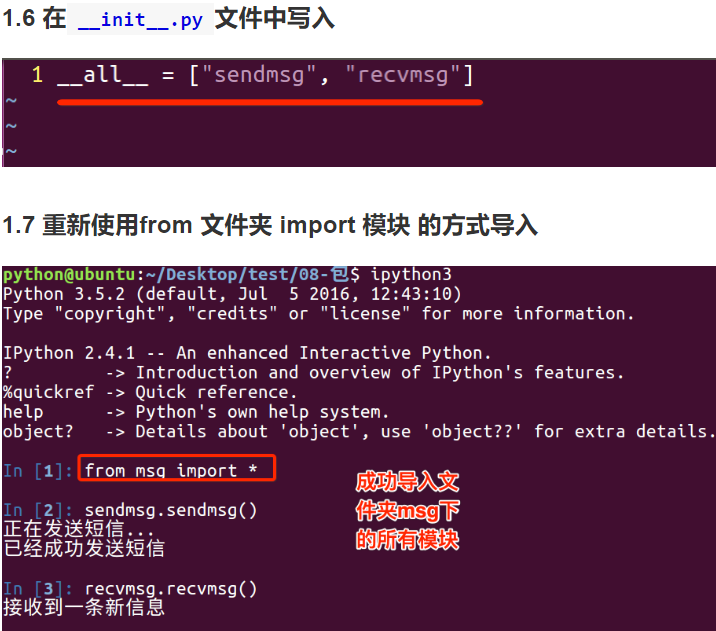
引入模块后，模块内的函数是可以被外界调用的，但是有些函数是模块制作者自己测试使用，这些函数不应该被外界调用，那么可以使用\_\_name\_\_：



# 二 包

有2个模块功能有些联系，所以将其放到同一个文件夹下，使用import 文件.模块 的方式导入。

包内需要有一个\_\_init\_\_.py文件控制导入：



\_\_init\_\_.py为空 仅仅是把这个包导入，不会导入包中的模块

在\_\_init\_\_.py文件中，定义一个\_\_all\_\_变量，它控制着 from 包名 import \*时导入的模块

假定我们的包的例子有如下的目录结构：

Phone/

\_\_init\_\_.py

common\_util.py

Voicedta/

\_\_init\_\_.py

Pots.py

Isdn.py

Fax/

\_\_init\_\_.py

G3.py

Mobile/

\_\_init\_\_.py

Analog.py

igital.py

Pager/

\_\_init\_\_.py

Numeric.py

Phone 是最顶层的包，Voicedta 等是它的子包。 我们可以这样导入子包：

import Phone.Mobile.Analog

Phone.Mobile.Analog.dial()

你也可使用 from-import 实现不同需求的导入

第一种方法是只导入顶层的子包，然后使用属性/点操作符向下引用子包树：

from Phone import Mobile

Mobile.Analog.dial('555-1212')

此外，我们可以还引用更多的子包：

from Phone.Mobile import Analog

Analog.dial('555-1212')

事实上，你可以一直沿子包的树状结构导入：

from Phone.Mobile.Analog import dial

dial('555-1212')

在我们上边的目录结构中，我们可以发现很多的 \_\_init\_\_.py 文件。这些是初始化模块，from-import 语句导入子包时需要用到它。 如果没有用到，他们可以是空文件。

包同样支持 from-import all 语句：

from package.module import \*

然而，这样的语句会导入哪些文件取决于操作系统的文件系统。所以我们在\_\_init\_\_.py 中加入 \_\_all\_\_ 变量。该变量包含执行这样的语句时应该导入的模块的名字。它由一个模块名字符串列表组成.。