

**CSDN**博客 (<http://blog.csdn.net/?ref=toolbar>)学院 (<http://edu.csdn.net/?ref=toolbar>) 下载 (<http://download.csdn.net/?ref=toolbar>)GitChat (<http://gitbook.cn/?ref=csdn>) 更多 ▾

0



## python学习笔记之module && package

原创 2011年04月04日 21:27:00

标签 : [module \(http://so.csdn.net/so/search/s.do?q=module&t=blog\)](http://so.csdn.net/so/search/s.do?q=module&t=blog) /[python \(http://so.csdn.net/so/search/s.do?q=python&t=blog\)](http://so.csdn.net/so/search/s.do?q=python&t=blog) /[import \(http://so.csdn.net/so/search/s.do?q=import&t=blog\)](http://so.csdn.net/so/search/s.do?q=import&t=blog) /[attributes \(http://so.csdn.net/so/search/s.do?q=attributes&t=blog\)](http://so.csdn.net/so/search/s.do?q=attributes&t=blog) /[initialization \(http://so.csdn.net/so/search/s.do?q=initialization&t=blog\)](http://so.csdn.net/so/search/s.do?q=initialization&t=blog) /[reference \(http://so.csdn.net/so/search/s.do?q=reference&t=blog\)](http://so.csdn.net/so/search/s.do?q=reference&t=blog)

7725

# python学习笔记之module && package

## python的module

1. import只能导入模块，不能导入模块中的对象（类、函数、变量等）。如一个模块A（A.py）中有个函数getName，另一个模块不能通过import A.getName将getName导入到本模块，只能用import A。如果想只导入特定的类、函数、变量则用from A import getName即可。
2. import一个module时，会执行该module的所有方法，并且将该module添加到importing module的命名空间中。A module's body executes immediately the first time the module is imported in a given run



weixin\_3506... ▾

<http://my.csdn.net/?ref=toolbar><http://so.csdn.net/so/search/s.do?q=module&t=blog><http://blog.csdn.net/arganzheng><http://blog.csdn.net/arganzheng>

+ 关注

<http://blog.csdn.net/arganzheng>

码云

原创	粉丝	喜欢	未开通
34	7	2	( <a href="https://github.com/arganzheng">https://github.com/arganzheng</a> )

### 他的最新文章

[更多文章 \(http://blog.csdn.net/arganzheng\)](http://blog.csdn.net/arganzheng)[关于数据库冗余字段 \(http://blog.csdn.net/arganzheng/article/details/6869982\)](http://blog.csdn.net/arganzheng/article/details/6869982)[shell如何模块化和复用——shell深入学习 \(http://blog.csdn.net/arganzheng/article/details/6786926\)](http://blog.csdn.net/arganzheng/article/details/6786926)[敏捷之道——提高命令行编辑速度 \(http://blog.csdn.net/arganzheng/article/details/6292919\)](http://blog.csdn.net/arganzheng/article/details/6292919)[关于文件描述符和句柄 \(http://blog.csdn.net/arganzheng/article/details/6292457\)](http://blog.csdn.net/arganzheng/article/details/6292457)[NULL in Perl \(http://blog.csdn.net/arganzheng/article/details/6280688\)](http://blog.csdn.net/arganzheng/article/details/6280688)

of a program...An import statement creates a new namespace containing all the attributes of the module. 如：

### 3. fibo.py

```
# Fibonacci numbers module

def fib(n):    # write Fibonacci series up to n
    a, b = 0, 1
    while b < n:
        print b,
        a, b = b, a+b

def fib2(n): # return Fibonacci series up to n
    result = []
    a, b = 0, 1
    while b < n:
        result.append(b)
        a, b = b, a+b
    return result

print "EOF"
```



0



### 他的热门文章

sed实战之——删除空行（包括由空格组成的空行）(<http://blog.csdn.net/arganzheng/article/details/6261912>)

📖 31179

-exec和xargs的区别 (<http://blog.csdn.net/arganzheng/article/details/6260720>)

📖 9916

BootLoader & Grub详解 (<http://blog.csdn.net/arganzheng/article/details/6263517>)

📖 8291

python学习笔记之module && package (<http://blog.csdn.net/arganzheng/article/details/6302030>)

📖 7720

vi使用技巧——在VI中剪切或者拷贝任意文本区域 (<http://blog.csdn.net/arganzheng/article/details/6260690>)

📖 7601

```

In [1]: import fibo
EOF

In [2]: import fibo

In [3]: fibo.
fibo.__builtins__      fibo.__doc__          fibo.__hash__          fibo.__package__
fibo.__setattr__       fibo.fib              fibo.__init__          fibo.__reduce__
fibo.__class__         fibo.__file__         fibo.__init__          fibo.__reduce__
fibo.__sizeof__        fibo.fib2             fibo.__name__          fibo.__reduce_ex__
fibo.__delattr__       fibo.__format__       fibo.__name__          fibo.__reduce_ex__
fibo.__str__           fibo.py               fibo.__new__           fibo.__repr__
fibo.__dict__          fibo.__getattr__      fibo.__new__           fibo.__repr__
fibo.__subclasshook__  fibo.pyc

In [3]: fibo.__name__
Out[3]: 'fibo'

In [4]: fibo.fib(100)
1 1 2 3 5 8 13 21 34 55 89

In [5]: fibo.fib2(100)
Out[5]: [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

In [6]: from fibo import fib

In [7]: fib(100)
1 1 2 3 5 8 13 21 34 55 89

In [8]: fib2(100)
-----
NameError                                Traceback (most recent call last)

/home/forrest/study/python/<ipython console> in <module>()

NameError: name 'fib2' is not defined

In [9]: from fibo import *

In [10]: fib2(100)
Out[10]: [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

```

会将fibo添加在当前module的名字空间，并且执行fibo.py定义的函数（定义函数表示将函数名添加到module的命名空间）这样就可以通过fibo访问fibo中定义的方法。并且会执行module中的statement。上面只执行一次，说明python只加载了一次。



下面这段话道出了python module的本质，其实也是整个python语言的本质——邦定。1. 变量定义：赋值邦定，对一个 $x = y \Rightarrow$ 定义一个变量 $x$ ，他的值是 $y$ ，并且将这个变量邦定在其命名空间上（如果是全局变量，那么是该变量所在module）。如果是函数内部变量，运行时才会执行，并且是邦定在函数对象上。2. 函数定义：def functionName: 定义一个函数对象，并将其邦定在所在命名空间中。3. 类定义：class clsName: 定义一个类，并将该类对象邦定在其命名空间中。

Attributes of a module object are normally bound by statements in the module body. When a statement in the body binds a variable (a global variable), what gets bound is an attribute of the module object. The normal purpose of a module body is exactly that of creating the module's attributes: def statements create and bind functions, class statements create and bind classes, and assignment statements can bind attributes of any type. You can also bind and unbind module attributes outside the body (i.e., in other modules), generally using attribute reference syntax M.name (where M is any expression whose value is the module, and identifier name is the attribute name). For clarity, however, it's usually best to limit yourself to binding module attributes in the module's own body.

python的package

包通常总是一个目录，目录下为首的一个文件便是 `__init__.py`。然后是一些模块文件和子目录，假如子目录中也有 `__init__.py` 那么它就是这个包的子包了。差不多就像这样吧：

```
Package1/ __init__.py
         Module1.py
         Module2.py
Package2/ __init__.py
         Module1.py
         Module2.py
```

我们可以就这样导入一个包：

```
import Package1
```

或者调入一个子模块和子包：

```
from Package1 import Module1
from Package1 import Package2
import Package1.Module1
import Package1.Package2
```

可以深入好几层包结构：

```
from Package1.Package2 import Module1
import Package1.Package2.Module1
```

### \_\_init\_\_.py文件

The \_\_init\_\_.py files are required to make Python treat the directories as containing packages. In the simplest case, \_\_init\_\_.py can just be an empty file, but it can also execute initialization code for the package or set the \_\_all\_\_ variable, described later.

\_\_init\_\_.py 控制着包的导入行为。假如 \_\_init\_\_.py 为空，那么仅仅导入包是什么都做不了的。

```
>>> import Package1
>>> Package1.Module1
```

```
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in ?
    Package1.Module1
```

```
AttributeError: 'module' object has no attribute 'Module1'
```

我们需要在 \_\_init\_\_.py 里把 Module1 预先导入：

```
import Module1
```

测试：

```
>>> import Package1
>>> Package1.Module1
<module 'Package1.Module1' from
'Module.pyc'>
```

\_\_init\_\_.py 中还有一个重要的变量，叫做 \_\_all\_\_。我们有时会使出一招“全部导入”，也就是这样：

```
from PackageName import *
```

这时 import 就会把注册在包 \_\_init\_\_.py 文件中 \_\_all\_\_ 列表中的子模块和子包导入到当前作用域中来。比如：

```
__all__ = ['Module1', 'Module2', 'Package2']
```

测试：

```
>>> from Package1 import *
>>> Module2
<module 'Package1.Module2' from 'Module.pyc'>
```

\_\_init\_\_.py 其实就是一个普通的python文件，它会在package被导入时执行。



0




```
print ">>>in package1.__init__.py"

def say_hello():
    print "Hi, my name is Forrest!"
```

测试：

```
In [1]: import package1
>>>in package1.__init__.py

In [2]: package1.say_hello()
Hi, my name is Forrest!
```

 多级package——\_\_init\_\_.py依次被执行

```
In [1]: import package1.package2
<<<in package1.__init__.py>>>
<<<in package1.package2.__init__.py>>>

In [2]: package1.say_hello()
Hi, my name is Forrest!

In [3]: package2.foo_bar()
-----
NameError                                Traceback (most recent call last)

/home/forrest/study/python/<ipython console> in <module>()

NameError: name 'package2' is not defined

In [4]: package1.package2.foo_bar()
foobar!
```

注意到多级package的import，虽然该路径上的所有packages都被引入了，但是使用的时候仍然必须使用完整路径。

将package/\_\_init\_\_.py改成如下：

```
print "<<<in package1.__init__.py>>>"

import package2

def say_hello():
    print "Hi, my name is Forrest!"
```

```

In [1]: import package1
<<<in package1.__init__.py>>>
<<<in package1.package2.__init__.py>>>

In [2]: package1.package2.foo_bar()
foobar!

In [3]: package2.foo_bar()
-----
NameError                                Traceback (most recent call last)

/home/forrest/study/python/<ipython console> in <module>()

NameError: name 'package2' is not defined

```



0



可以看到也是一样的，必须全路径引用。这是因为在哪个module中import的module，是加入到importing module的名字空间，所以只有该importing module可以短路引用，其他的间接引用必须通过import module一路引用过去。

补记：关于Python的module

2011-04-03 星期天 阴雨

python的module特别像C++中的命名空间(namespace)，因此也就特别像java中的package。

例如，如果你定义了一个namespace mynamespace，那么你可以如下处理：

1> using namespace mynamespace;然后你就可以使用使用mynamespace中的每一个成员。类似与python中的from mynamespace import \*

2> using mynamespace::aFunc;在这个文件中可以并且只能使用mynamespace中的aFunc这个定义，直接aFunc()就可以了。类似于python中的from mynamespace import aFunc。

3> 你也可以在使用时直接使用mynamespace::aFunc();来调用mynamespace中的任何成员定义。在C++中你需要

但是python跟C++有个非常大的不同是，它的import不仅仅是引入命名空间，还引入了python文件（这点类似于C的#import头文件），就是说它的import语法共用了（呃，你可以说它职责不单一）。事实上，python的import一定是引入一个module，也就是一个python文件，如果你想要只引入该module的某个部分，那么可以使用from module import xxx。而C++中，你要引入一个namespace中的某一个定义，你必须用两个语句：

首先，引入该定义的头文件：

```
#include "xxx.h"
```

然后，引入该头文件的某个命名空间：

```
#using namespace xxx或者using xxx::yyy
```

所以，其实python的module，更像是java的package，但是又不像java的package一定是目录，python的module一般是python文件。

其实，作这种对比，目的在于说明，任何语言原理都是差不多的，不需要太纠结于语法细节。应该关注更本质的东西——算法与数据结构。这才是王道。



0



发表你的评论

(http://my.csdn.net/weixin\_35068028)

## Python学习笔记——有用的module (http://blog.csdn.net/Tony\_Wong/article/details/399954...)

Webbrowser：可以使用webbrowser控制打开一个网页



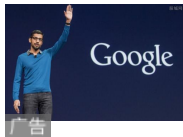
Tony\_Wong (http://blog.csdn.net/Tony\_Wong) 2014年10月11日 15:46 372

## Python Tutorial 学习笔记4 ——Module (http://blog.csdn.net/niceSimon7/article/details/51...)

1.A module is a file containing Python definitions and statements. The file name is the module name ...



niceSimon7 (http://blog.csdn.net/niceSimon7) 2016年07月08日 11:54 226



### 2017年！全球AI人才薪酬报告出炉了！中国区AI人才最贵？

给大家分享下2017年AI程序员的人才报告，服了！薪酬最高的竟然不是谷歌！~

(http://www.baidu.com/cb.php?c=lgF\_pyfqHmknjnvPjn0IZ0qnfK9ujYzP1f4PjDs0Aw-

5Hc3rHnYnHb0TAq15HfLPWRznjb0T1YsmWmvPhF-



P161uy7hnynY0AwY5HDdnHfLPWT1P1f0IgF\_5y9YIZ0IQzq-

uZR8mLPbUB48ugfEIqspynEmybz5LNYUNq1ULNzmvRqmhkEu1Ds0ZFb5HD0mhYqn0KsTWYs0ZNGujYkPHTYn1mk0AqGujYknWb3rjDY0APGujYLnWm4n1c0ULI85H00TZbqnWC

## python学习笔记2——python 中安装package注意事项 (http://blog.csdn.net/u011127242/art...)



python学习笔记2——python 中安装package注意事项 使用python时候，经常需要安装相关的包，安装方法包括两种：  
(1) 直接使用easy\_install或者p...

 u011127242 (<http://blog.csdn.net/u011127242>) 2017年02月09日 18:55  209

### 【总结】Python中第三方的库(library)、模块(module)，包(package)的安装方法 (<http://blog....>)

【教程】Python中的内置的模块 和第三方的模块 2012 年 11 月 15 日 上午 11:11crifan已有3684人围观1个评论 1.Python内置模块和第三方模块 ...

 franktan2010 (<http://blog.csdn.net/franktan2010>) 2014年04月14日 15:35  2257

### Python Module和Package辨析 (<http://blog.csdn.net/afandaafandaafanda/article/details/6...>)



Python 基础学习说明 这不是最基础的入门教程，如需了解Python的数据类型、变量等基础内容，请移步:<https://docs.python.org/2/tutorial/index.html> ...

 afandaafandaafanda (<http://blog.csdn.net/afandaafandaafanda>) 2017年03月24日 16:45  1917

 45.00/只 ODF，ODF单元箱， 12芯ODF箱，机架式	 400.00/套 一级供应摩托罗拉a8 对讲机 Mag OneA8无	 140.00/台 厂家直销对讲机民用 10W大功率无线自驾
--	---	--


### Python中的module,library,package之间的区别 (<http://blog.csdn.net/u011331383/article/de...>)

背景Python中有一些基本的名词，很多人，尤其是一些初学者，可能听着就很晕。此处，简单总结一下，module，library，package之间的大概区别。Python中的module的简介modu...

 u011331383 (<http://blog.csdn.net/u011331383>) 2015年04月04日 15:46  779

### 【总结】Python中第三方的库(library)、模块(module)，包(package)的安装方法 (<http://blog....>)

提示：【教程】Python中的内置的模块 和第三方的模块 Python中，想要安装第三方安装包，即third library，package等，对于熟悉的人来说，很简单。但是对于新手，至...

 mcumsj1 (<http://blog.csdn.net/mcumsj1>) 2015年01月28日 10:36 1330

## Python , 从 package name 得到 module name (<http://blog.csdn.net/shellpaul/article/detai...>)

探索怎样从包名得到模块名？可以把 pypi 上的所有包爬下来，然后解析出对应的模块就可以了。其中解析模块显然是一个难点。因为我要的不仅仅是模块，还包括子模块。这会涉及到文件结构和安装逻辑，工作...

 shellpaul (<http://blog.csdn.net/shellpaul>) 2016年06月03日 14:05 165

## python package module import 完全总结 (<http://blog.csdn.net/weixingstudio/article/detai...>)

总结这个python中 package以及module的导入，是遵循下面的文档结构进行的： sound/ Top-level package ...

 weixingstudio (<http://blog.csdn.net/weixingstudio>) 2015年03月27日 09:27 2465


## Python中的模块(module)和包(package) (<http://blog.csdn.net/liuguiyangnwpu/article/detai...>)

创建一个模块(module) 一个module不过是一些函数，class放在一个文件中 """ this only is a very simple test module """ ...

 liuguiyangnwpu (<http://blog.csdn.net/liuguiyangnwpu>) 2013年07月21日 11:01 4869


## python语法31[module/package+import] (<http://blog.csdn.net/u010680471/article/details/5...>)

作者：iTech 出处：<http://itech.cnblogs.com/> — module 通常模块为一个文件，直接使用import来导入就好了。可以作为module的文件类型有"....

 u010680471 (<http://blog.csdn.net/u010680471>) 2016年03月26日 21:58 140


## python module, package (<http://blog.csdn.net/iCode0410/article/details/39433111>)

任何Python程序都可以作为模块导入；在导入自己的模块时，需要添加路径：`import sys sys.path.append('absolute-path');`  
( \_...

 iCode0410 (<http://blog.csdn.net/iCode0410>) 2014年09月20日 16:56 853

## python语法[module/package+import] (<http://blog.csdn.net/jasonzzj/article/details/539442...>)

一 . module 通常模块为一个文件，直接使用import来导入就好了。可以作为module的文件类型有".py"、".pyo"、".pyc"、".pyd"、".so"、".dll"。 二...

 jasonzzj (<http://blog.csdn.net/jasonzzj>) 2016年12月30日 16:12 160

## Delphi Package 學習筆記（含討論信件精華）(<http://blog.csdn.net/diligentcat/article/detai...>)

在實作 package 時，我遇到了一些問題，承蒙DelphiChat多位熱心朋友的幫忙解決小弟許多疑惑，因此當傳兄(James Fu)提議將討論信件整理出來並配合範例供大家參考時，我也很贊同，並且將...

 diligentcat (<http://blog.csdn.net/diligentcat>) 2012年06月09日 07:18 1731


## java学习笔记三——package,final和static (<http://blog.csdn.net/BeforeEasy/article/details/...>)

PackageJava里的每一个类都属于一个包，包是管理类的容器，在默认情况下，java会为每一个类生成一个默认包，同一个包下的类可以相互引用非private的方法，但是不同的包之间不可以。 Jav...

 BeforeEasy (<http://blog.csdn.net/BeforeEasy>) 2016年11月21日 21:16 214


## ROS基础学习笔记——工作空间、CMakeList、package.xml (<http://blog.csdn.net/u0143030...>)

ROS基础学习与配置；ros入门

 u014303046 (<http://blog.csdn.net/u014303046>) 2016年03月13日 15:18 1345



## Lua学习笔记六——package (<http://blog.csdn.net/boshuzhang/article/details/74294504>)

Lua学习笔记六——package 标签：luapathn2modulefunction语言 2009-05-09 15:15 23280人阅读 评论(1) 收藏 举报 ...

 boshuzhang (<http://blog.csdn.net/boshuzhang>) 2017年07月04日 10:25 56

## JAVA学习笔记6——package+API文档+Scanner类 (<http://blog.csdn.net/u014440351/article...>

最近在看JAVA教学的视频，觉得老师讲的很好，同时借用源代码还有笔记来撰写本系列博客，记录自己的学习内容，同时也供看到的人学习。 第六篇介绍的内容稍微少了些，因为这一小部分知识和前、后均没有太大的联...

 u014440351 (<http://blog.csdn.net/u014440351>) 2015年01月19日 18:26  459





0





## PackageManagerService学习笔记四-构造方法(扫描Package) (<http://blog.csdn.net/u013378953>...

构造方法代码片段：{ /\*-----省略部分代码-----\*/ /\*优化扫描，check是否有需要字节码优化\*/ // Set flag to ...

 u013378953 (<http://blog.csdn.net/u013378953>) 2014年02月06日 11:44  651

## windows下python如何安装模块或包？ How to install package or module in windows OS ...

摘要：本文介绍了在windows下利用cmd安装第三方模块module或包package的方法。

 chennudt (<http://blog.csdn.net/chennudt>) 2017年05月26日 20:41  902