# [python有三种导入模块的方法](https://www.cnblogs.com/allenblogs/archive/2011/05/24/2055149.html)

**python有三种导入模块的方法**

其一,

import modname ： 模块是指一个可以交互使用，或者从另一Python 程序访问的代码段。只要导入了一个模块，就可以引用它的任何公共的函数、类或属性。模块可以通过这种方法来使用其它模块的功能。

用import语句导入模块，就在当前的名称空间(namespace)建立了一个到该模块的引用.这种引用必须使用全称，也就是说，当使用在被导入模块中定义的函数时，必须包含模块的名字。所以不能只使用 funcname，而应该使用 modname.funcname

其二,   
from modname import funcname   
from modname import fa, fb, fc   
或者  from modname import \*   
与第1种方法的区别：funcname 被直接导入到本地名字空间去了，所以它可以直接使用，而不需要加上模块名的限定   
\* 表示，该模块的所有公共对象(public objects)都被导入到 当前的名称空间，也就是任何只要不是以”\_”开始的东西都会被导入。   
modname没有被定义，所以modname.funcname这种方式不起作用。并且，如果funcname如果已经被定义，它会被新版本（该导入模块中的版本）所替代。如果funcname被改成指向其他对象，modname不能不会觉察到。   
建议:

1)如果你要经常访问模块的属性和方法，且不想一遍又一遍地敲入模块名，使用 from module import   
2)如果你想要有选择地导入某些属性和方法，而不想要其它的，使用 from module import   
3)如果模块包含的属性和方法与你的某个模块同名，你必须使用import module来避免名字冲突   
4)尽量少用 from module import \* ，因为判定一个特殊的函数或属性是从哪来的有些困难，并且会造成调试和重构都更困难。   
其三   
内建函数\_\_import\_\_()   
除了前面两种使用import关键字的方法以外，我们还可以使用内建函数 \_\_import\_\_() 来导入 module。两者的区别是，import 后面跟的必须是一个类型(type)，而\_\_import\_\_() 的参数是一个字符串，这个字符串可能来自配置文件，也可能是某个表达式计算结果。例如   
mymodule = \_\_import\_\_ (’module\_name’)   
附注:   
1)模块的内容都放在一个模块文件中，如 mymodule 的内容应该放在PYTHONPATH 目录下的一个mymodule.py中，C实现的除外   
2)包可以将几个模块名称空间组织起来, 如A.b 就表示在包A中的一个子模块b   
可以单独导入某一个子模块，如Python文档中给出的例子   
import sound.effects.echo   
这样必须使用全称对里面的对象进行引用，如   
sound.effects.echo.echofilter(input, output, delay=0.7, atten=4)   
还可以使用下面的语句来加载echo子模块   
from Sound.Effects import echo   
它在没有包前缀的情况下也可以使用， 所以它可以如下方式调用：   
echo.echofilter(input, output, delay=0.7, atten=4)   
不主张从一个包或模块中用import \* 导入所有模块，因为这样的通常会导致可读性很差。  
from Package import specific\_submodule的用法并没有错，实际上这还是推荐的用法，除非导入的模块需要使用其它包中的同名子模块(the importing module needs to use submodules with the same name from different packages).   
综上所述，一般情况应该使用import , 但有几个例外   
1)module文档告诉你要用from-import的   
2)导入一个包组件。需要一个包里面的某个子模块，一般用from A.b import c比import A.b.c 更方便 且不会冒混淆的危险.

Work for fun,Live for love!