# 查看python的模块和函数帮助文档方法

**python的一个优势是有着大量自带和在线的模块(module)资源，可以提供丰富的功能，在使用这些模块的时候，如果每次都去网站找在线文档会过于耗费时间，结果也不一定准确。因此这里介绍下python自带的查看帮助功能，可以在编程时不中断地迅速找到所需模块和函数的使用方法**

**通用帮助函数help()**

在python命令行中键入help(),可以看到：

>>> help()

Welcome to Python 3.5's help utility!

If this is your first time using Python, you should definitely check out

the tutorial on the Internet at http://docs.python.org/3.5/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing

Python programs and using Python modules. To quit this help utility and

return to the interpreter, just type "quit".

To get a list of available modules, keywords, symbols, or topics, type

"modules", "keywords", "symbols", or "topics". Each module also comes

with a one-line summary of what it does; to list the modules whose name

or summary contain a given string such as "spam", type "modules spam".

help>

进入help帮助文档界面，根据屏幕提示可以继续键入相应关键词进行查询，继续键入modules可以列出当前所有安装的模块：

help> modules

Please wait a moment while I gather a list of all available modules...

AutoComplete \_pyio filecmp pyscreeze

AutoCompleteWindow \_random fileinput pytweening

......

Enter any module name to get more help. Or, type "modules spam" to search

for modules whose name or summary contain the string "spam".

可以继续键入相应的模块名称得到该模块的帮助信息。   
这是python的通用的查询帮助，可以查到几乎所有的帮助文档，但我们很多时候不需要这样层级式地向下查询，接下来会介绍如何直接查询特定的模块和函数帮助信息。

**模块帮助查询**

***查看.py结尾的普通模块help(module\_name)***

例如要查询math模块的使用方法，可以如下操作：

>>> import math

>>> help(math)

Help on built-in module math:

NAME

math

DESCRIPTION

This module is always available. It provides access to the

mathematical functions defined by the C standard.

FUNCTIONS

acos(...)

acos(x)

Return the arc cosine (measured in radians) of x.

...

>>>

使用help(module\_name)时首先需要import该模块，有些教程中不进行导入而在模块名中加入引号help('module\_name')，这种方法可能会带来问题，大家可以用math模块测试，建议使用先导入再使用help()函数查询

***查看内建模块sys.bultin\_modulenames***

>>> import sys

>>> sys.builtin\_module\_names

('\_ast', '\_bisect', '\_codecs', '\_codecs\_cn', '\_codecs\_hk', ... 'zlib')

>>>

需要导入sys模块。这里列举的一般是自带的使用C/C++编译链接的模块

**查询函数信息**

***查看模块下所有函数dir(module\_name)***

如我们需要列举出math模块下所有的函数名称

>>> dir(math)

['\_\_doc\_\_', '\_\_loader\_\_', '\_\_name\_\_',...]

>>>

同样需要首先导入该模块

***查看模块下特定函数信息help(module\_name.func\_name)***

如查看math下的sin()函数

>>> help(math.sin)

Help on built-in function sin in module math:

sin(...)

sin(x)

Return the sine of x (measured in radians).

>>>

***查看函数信息的另一种方法print(func\_name.\_\_doc\_\_)***

如查看内建函数print用法

>>> print(print.\_\_doc\_\_)

print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

Prints the values to a stream, or to sys.stdout by default.

...

>>>

\_\_doc\_\_前后是两个短下划线，在python中会合并为长下划线

python中的help()类似unix中的man指令，熟悉后会对我们的编程带来很大帮助

随着使用Python的时间越来越长，安装的python模块也越来越多，时间久了都不记得自己之前到底对自己的电脑做过些什么了，于是乎就想要查看一下自己安装的python模块，现将查看方法总结如下

import sys

sys.path 查看python目录

#### 一、命令行下使用pydoc命令

在命令行下运行$ pydoc modules即可查看

#### 二、在python交互解释器中使用help()查看

在交互式解释器中输入>>> help("modules")即可，效果跟在命令行下输入$ pydoc modules是一样的

#### 三、在python交互是解释器下导入sys模块查看

python的sys模块也是可以用来查看模块信息的

>>> import sys

>>> sys.modules.keys()

#### 四、命令行下使用pip查看

如果你使用的是pip来作为你的python包管理器的话，可以在命令行下直接运行$ pip freeze或者$ pip list来查看安装包的信息，当然其它的包管理器也有类似的功能，同时，你也可以在python交互式解释器中导入pip模块来查看包信息

import pip

installed\_packages = pip.get\_installed\_distributions()

installed\_packages\_list = sorted(["%s==%s" % (i.key, i.version)

for i in installed\_packages])

print(installed\_packages\_list)

#### 五、使用yolk来查看

如果你安装过yolk的话可以使用yolk命令来查看包信息，你可以使用$ pip install yolk来安装它   
yolk使用简单，只需在命令行下操作即可

$ yolk -l #列出所有安装模块

$ yolk -a #列出激活的模块

$ yolk -n #列出非激活模块

$ yolk -U [packagename] # 通过查询pypi来查看（该）模块是否有新版本

**查看python安装路径**

1. **import** sys
2. **print** (sys.path)