

马哥教育



主讲：马永亮(马哥)

QQ: 1661815153

<http://www.magedu.com>

马哥教育

www.magedu.com



马哥教育

# Python执行环境

主讲：马永亮(马哥)

QQ:113228115

客服QQ: 2813150558, 1661815153

<http://www.magedu.com>

<http://mageedu.blog.51cto.com>

# 解释器环境与选项

## ❖ Python解释器启动

➔ `python [options] [-c cmd| filename | - ] [args]`

选 项	描 述
-3	启用将从Python 3中删除或更改某些功能的警告
-B	阻止在导入时创建.pyc或.pyo文件
-E	忽略环境变量
-h	打印所有可用命令行选项的列表
-i	在程序执行后进入交互模式
-m <i>module</i>	以脚本的形式运行库模块 <i>module</i>
-O	优化模式
-OO	优化模式，在创建.pyo文件时删除文档字符串
-Qarg	指定Python 2中除法运算符的行为，值为-Qold（默认值）、-Qnew、-Qwarn或-Qwarnall之一
-s	阻止将用户站点目录添加到sys.path
-S	阻止包含site初始化模块
-t	报告关于不一致的标签使用警告
-tt	由于不一致的标签使用而导致TabError异常
-u	未缓冲的二进制stdout和stdin
-U	Unicode字面量。所有字符串字面量都以Unicode形式处理（仅在Python 2中使用）
-v	详细模式。跟踪导入语句
-V	打印版本号并退出
-x	跳过源程序的第一行
-c <i>cmd</i>	以字符串形式执行 <i>cmd</i>



环境变量	描 述
PYTHONPATH	以冒号分隔的模块搜索路径
PYTHONSTARTUP	在以交互方式启动时执行的文件
PYTHONHOME	Python安装的位置
PYTHONINSPECT	相当于-i选项
PYTHONUNBUFFERED	相当于-u选项
PYTHONIOENCODING	针对stdin、stdout和stderr的编码和错误处理。这是一个 <code>encoding[:errors]</code> 形式的字符串，如utf-8或utf-8:ignore
PYTHONDONTWRITEBYTECODE	相当于-B选项
PYTHONOPTIMIZE	相当于-O选项
PYTHONNOUSERSITE	相当于-s选项
PYTHONVERBOSE	相当于-v选项
PYTHONUSERBASE	每个用户站点包的根目录
PYTHONCASEOK	指示为导入所使用的模块名称使用不区分大小写的匹配方式

www.magedu.com



Reference: 《Python参考手册(第4版)》



马  
哥  
教  
育

# Python代码的测试、调试 与探查

主讲：马永亮(马哥)

QQ:113228115

客服QQ: 2813150558, 1661815153

<http://www.magedu.com>

<http://mageedu.blog.51cto.com>

- ❖ 如果函数、类或模块的第一行是一个字符串，这个字符串就称为文档字符串(docstrings)
- ❖ 内置函数`help()`或对象的默认方法`__doc__`可以显示这些文档字符串

```
>>> def Sum(num1, num2):  
    """the sumary of num1 and num2.  
    >>> Sum(2,5)  
    7  
    >>> Sum(12, 77)  
    89  
    """  
    return num1 + num2
```

```
>>> help(Sum)  
Help on function Sum in module __main__:
```

```
Sum(num1, num2)  
    the sumary of num1 and num2.  
    >>> Sum(2,5)  
    7  
    >>> Sum(12, 77)  
    89
```

```
>>> Sum.__doc__  
'the sumary of num1 and num2.\n\t>>> Sum(2,5)\n\t7\n\t>>> Sum(12, 77)\n\t89\n\t'
```



- ❖ **doctest**模块允许在文档字符串内嵌入注释以显示各种语句的期望行为，尤其是函数和方法的结果
- ➡ 此处的文档字符串看起来如同一个交互式**shell**会话
- ➡ 可用于测试文档是否与程序主体保持同步，或基于文档对程序本身做测试

自定义模块test:

```
#!/usr/bin/python
#
def add(num1, num2):
    """
    >>> add(12, 43)
    55
    """
    return num1 + num2
```

类似于python会话的文档串

使用doctest模块进行测试

```
>>> import test
>>> import doctest
>>> doctest.testmod(test)
TestResults(failed=0, attempted=1)
>>> doctest.testmod(test, verbose=True)
Trying:
    add(12, 43)
Expecting:
    55
ok
1 items had no tests:
    test
1 items passed all tests:
   1 tests in test.add
1 tests in 2 items.
1 passed and 0 failed.
Test passed.
TestResults(failed=0, attempted=1)
```





- ❖ 如果文档字符串中的结果与预期结果不一致，测试会显示出错的结果信息
- ❖ 例如，将前述例子文档中的结果修改为错误的结果，其测试结果如图所示

```
>>> import test
>>> import doctest
>>> doctest.testmod(test)
*****
File "test.py", line 5, in test.add
Failed example:
    add(12,43)
Expected:
    35
Got:
    55
*****
1 items had failures:
  1 of  1 in test.add
***Test Failed*** 1 failures.
TestResults(failed=1, attempted=1)
```



## ❖ 创建可自测试的模块

➡ 在模块的尾部添加如下代码即可

```
➡ if __name__ == '__main__':  
    import doctest  
    doctest.testmod()
```

➡ 此类模块在python解释器中直接运行时即能进行自我测试

```
[root@www.magedu.com python]# python test.py  
*****  
File "test.py", line 5, in __main__.add  
Failed example:  
    add(12,43)  
Expected:  
    35  
Got:  
    55  
*****  
1 items had failures:  
    1 of 1 in __main__.add  
***Test Failed*** 1 failures.
```



- ❖ 博客: <http://magedu.blog.51cto.com>
- ❖ 主页: <http://www.magedu.com>
- ❖ QQ: 1661815153, 113228115
- ❖ QQ群: 203585050, 279599283



马哥教育

Thank You!