

Python中ImportError: No module named的解决

有时候我们会自己编写一些 Python 内置中没有的 module ，比如下面我自定义了一个名为 matrix 的 module ，然后直接在命令行中引入则会出现下面的错误：

```
[iteblog@www.iteblog.com ~]$ python
Python 2.7.3 (default, Aug 4 2016, 21:49:57)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import matrix
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named matrix
>>>
```

这是因为我们编写的 module 并没有加入到当前 Python 运行环节中，所以肯定会出现这种异常。其实我们只需要将自定义的 module 加入到当前环境中。这里列举出以下两种方法。

```
>>> import matrix
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named matrix
```

如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

通过 sys 加载自定义module

这个不需要在 Python 启动之前设置的，具体操作如下：

```
>>> import sys
>>> sys.path.append('/home/iteblog/mllib/')
>>>
```

然后我们可以通过 sys.path 来查看是否将自定义的 module 加入到环境变量中：

```
>>> sys.path
['', '/usr/local/lib/python2.7.zip', '/usr/local/lib/python2.7', '/usr/local/lib/python2.7/plat-linux2',
'/usr/local/lib/python2.7/lib-tk', '/usr/local/lib/python2.7/lib-old', '/usr/local/lib/python2.7/lib-
dynload', '/usr/local/lib/python2.7/site-packages', '/home/iteblog/mllib/']
>>> import matrix
```

从上面的输出可以看到 matrix 已经成功加入到当前 Python 运行环境变量中，所有我们现在运行 import matrix 不会有任何的错误。

通过PYTHONPATH添加

第二种方法就是通过 PYTHONPATH 环境变量添加，如下：

```
$export PYTHONPATH=$PYTHONPATH:/home/iteblog/mllib/
```

一样可以实现这个效果。

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接: [【】（）](#)