

安装第三方模块

1986次阅读

在Python中，安装第三方模块，是通过setuptools这个工具完成的。  
如果你正在使用Mac或Linux，安装setuptools本身这个步骤就可以跳过了。  
如果你正在使用Windows，请首先从这个地址下载ez\_setup.py：

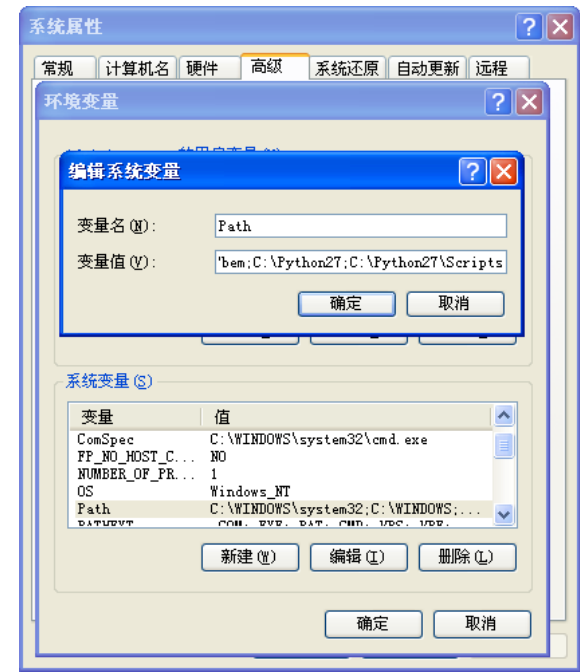
<https://pypi.python.org/pypi/setuptools#windows>



下载后，随便放到一个目录下，然后运行以下命令来安装setuptools：

```
python ez_setup.py
```

在命令提示符窗口下尝试运行easy\_install，Windows会提示未找到命令，原因是easy\_install.exe所在路径还没有被添加到环境变量Path中。请添加C:\Python27\Scripts到环境变量Path：



重新打开命令提示符窗口，就可以运行easy\_install了：

现在，让我们来安装一个第三方库——Python Imaging Library，这是Python下非常强大的处理图像的工具库。一般来说，第三方库都会在Python官方的pypi.python.org网站注册，要安装一个第三方库，必须先知道该库的名称，可以在官网或者pypi上搜索，比如Python Imaging Library的名称叫PIL，因此，安装Python Imaging Library的命令就是：

```
easy_install PIL
```

耐心等待下载并安装后，就可以使用PIL了。

有了PIL，处理图片易如反掌。随便找个图片生成缩略图：

```
>>> import Image
>>> im = Image.open('test.png')
>>> print im.format, im.size, im.mode
PNG (400, 300) RGB
>>> im.thumbnail((200, 100))
```

```
>>> im.save('thumb.jpg', 'JPEG')
```

其他常用的第三方库还有MySQL的驱动：MySQL-python，用于科学计算的NumPy库：numpy，用于生成文本的模板工具Jinja2，等等。

## 模块搜索路径

当我们试图加载一个模块时，Python会在指定的路径下搜索对应的.py文件，如果找不到，就会报错：

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

默认情况下，Python解释器会搜索当前目录、所有已安装的内置模块和第三方模块，搜索路径存放在sys模块的path变量中：

```
>>> import sys
>>> sys.path
['', '/Library/Python/2.7/site-packages/pycrypto-2.6.1-py2.7-macosx-10.9-intel.egg', '/Library/Python/2.7/site-packages/PIL-1.1.7-py2.7-macosx-10.9-intel.egg', ...]
```

如果我们要添加自己的搜索目录，有两种方法：

一是直接修改sys.path，添加要搜索的目录：

```
>>> import sys
>>> sys.path.append('/Users/michael/my_py_scripts')
```

这种方法是在运行时修改，运行结束后失效。

第二种方法是设置环境变量PYTHONPATH，该环境变量的内容会被自动添加到模块搜索路径中。设置方式与设置Path环境变量类似。注意只需要添加你自己的搜索路径，Python自己本身的搜索路径不受影响。

---