**Python学习和开发环境的建立**

Python具有丰富和强大的库。它常被昵称为[胶水语言](http://baike.baidu.com/view/2993364.htm)，能够把用其他语言制作的各种模块（尤其是C/[C++](http://baike.baidu.com/view/824.htm)）很轻松地联结在一起。

Python解释器、标准库及扩展包构成Python的基本开发环境，满足学习和软件开发的要求。Python解释器及其标准库可从Python官网下载；其他扩展包可用pip命令等方法安装。

大型软件开发宜采用集成开发工具，其中，Eclipse加PyDev插件的方式， 为较多Python程序开发人员使用。

1. **Python基本开发环境建立**

从官网：

<http://www.python.org>

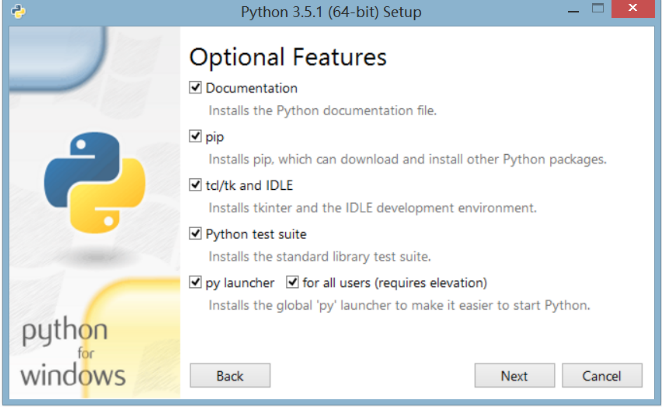
下载和操作系统（64位）对应版本的Python3.5.1软件。

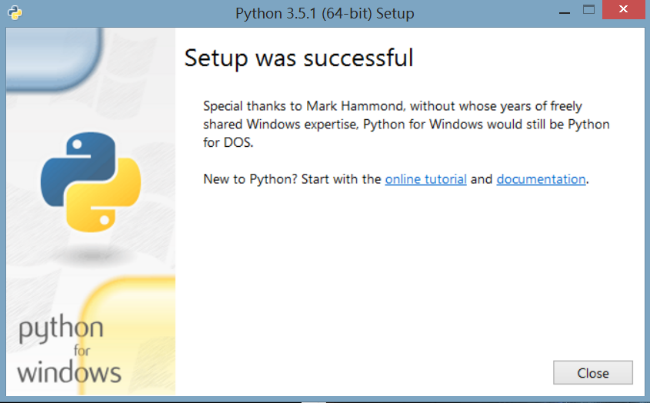
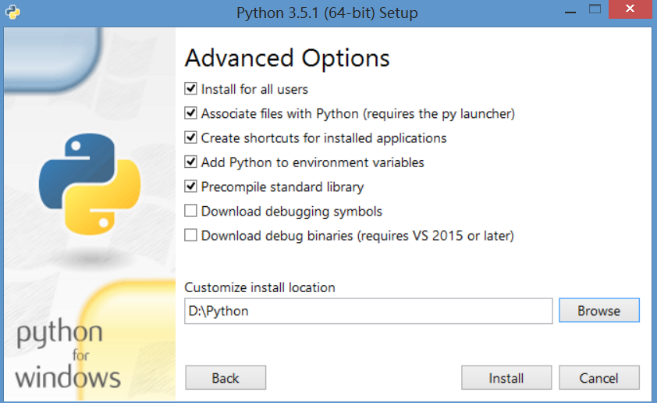


下面以Python3.5.1版本安装为例做简单说明

**Windows8.1：** 建议使用定制方式安装:

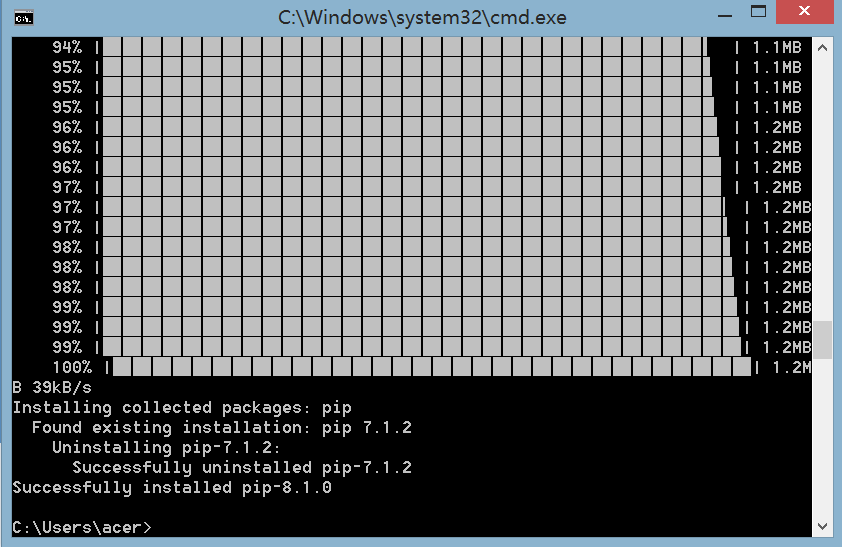
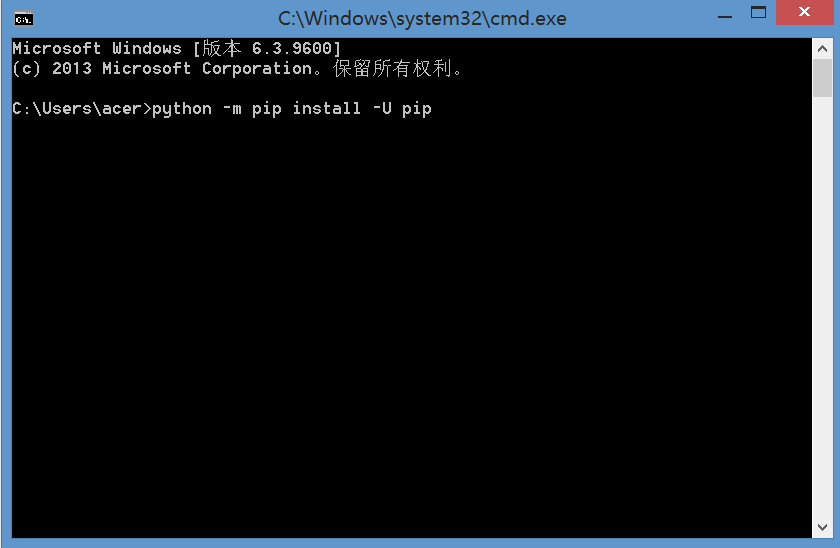
1) 加环境变量PATH; 2) 自定义安装目录： D:\Python





然后，命令行(cmd)下更新pip到新版本：

python -m pip install -U pip



1. **Python扩展包安装**

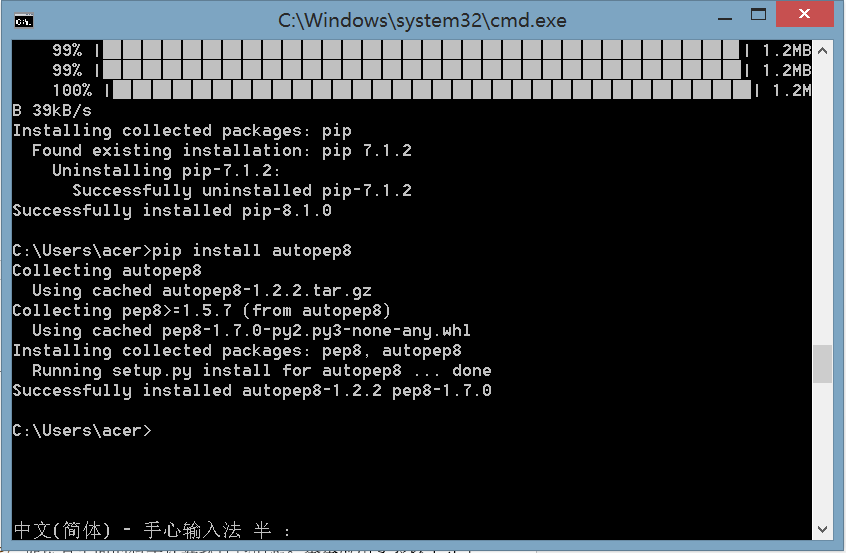
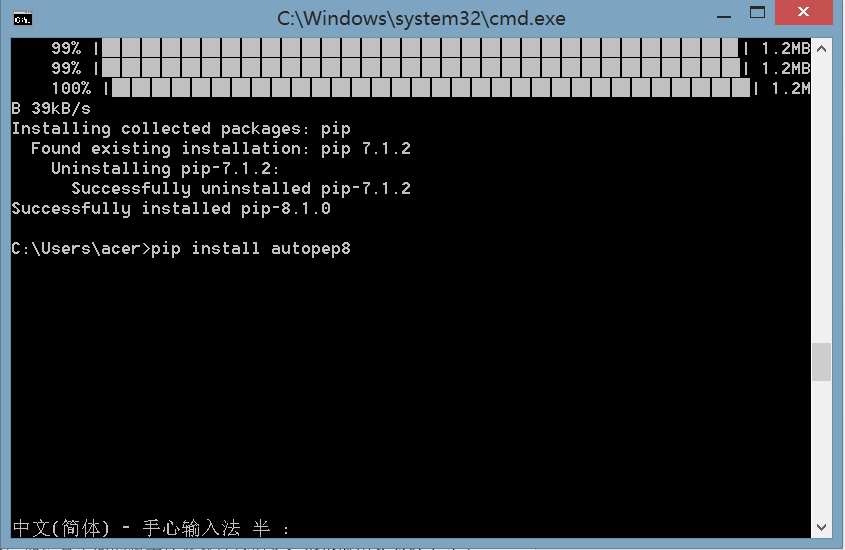
pip命令是软件包的基本安装方式，不过，有些软件包使用pip安装时，需要本地编译，编译条件如果不满足安装就会失败。

这时可使用： 加州大学欧文分校提供的Windows32/64位Python扩展包编译版，

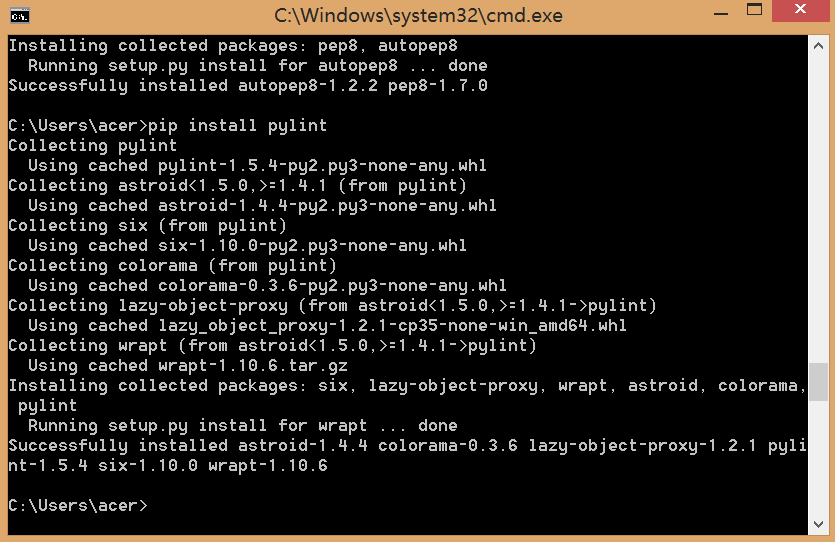
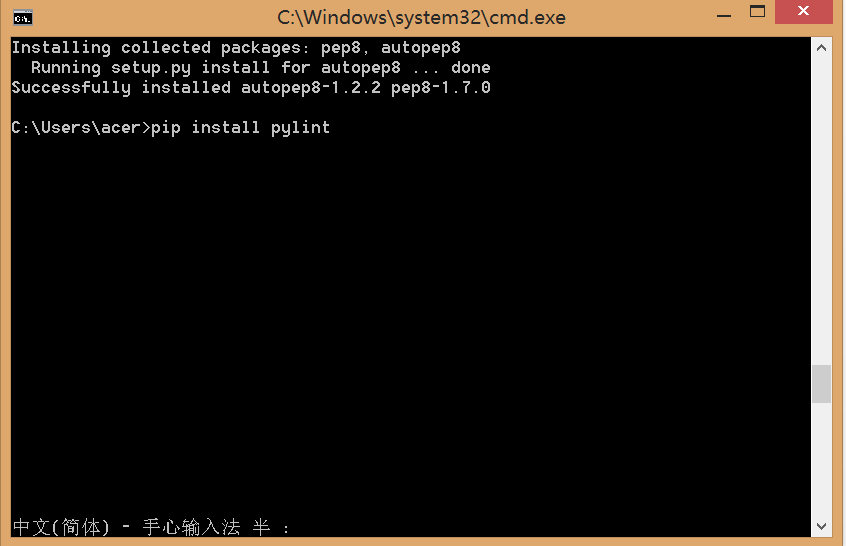
<http://www.lfd.uci.edu/~gohlke/pythonlibs/>

**2.1 编码规范**

pip install autopep8



pip install pylint



**2.2 科学计算软件包**

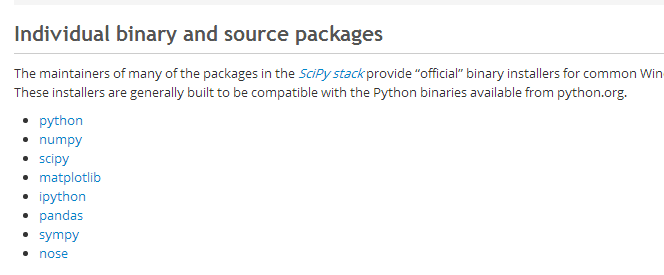
**2.2.1 SciPy （未使用）**

Scipy官网：<http://www.scipy.org/>,提供有全面的科学计算软件包信息。基本应用安装以下3个即可：

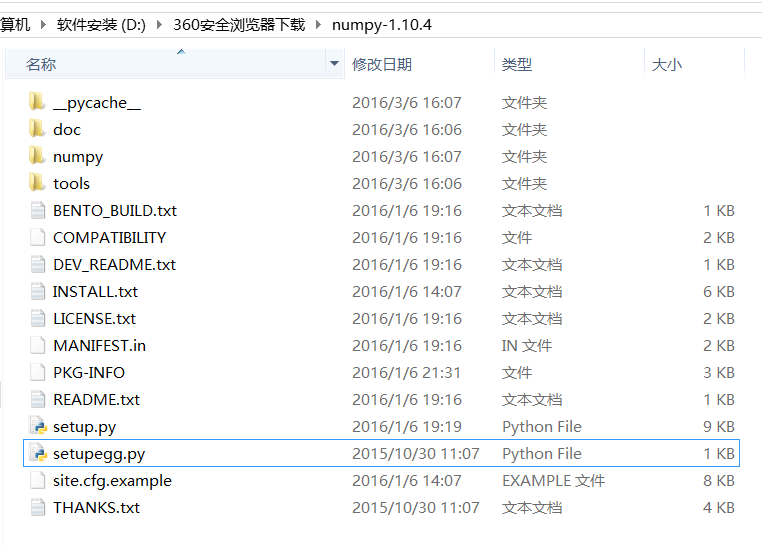
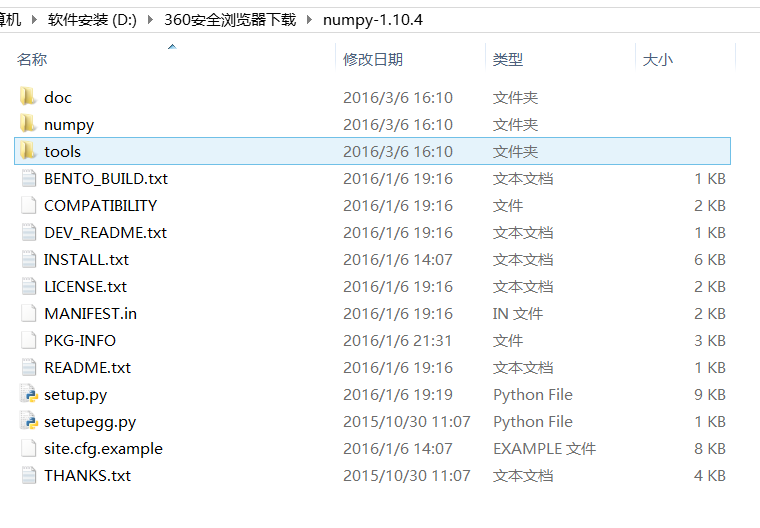
numpy

scipy

matplotlib



Numpy



Scipy

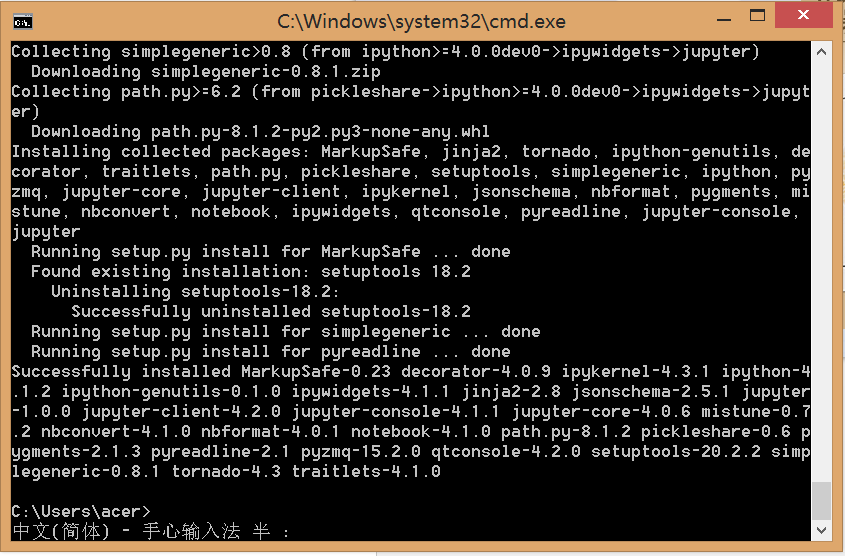
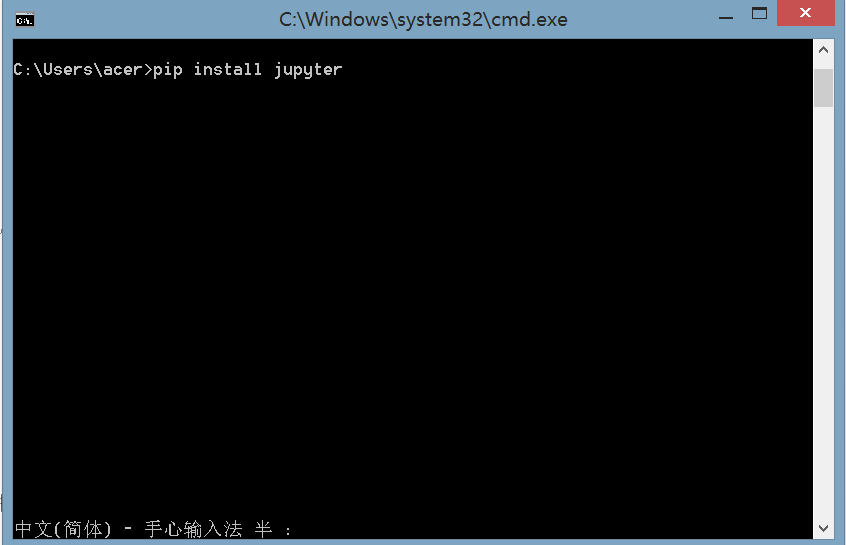
**Window**环境下Scipy 软件包的 安装，建议从加州大学欧文分校下载编译好的\*.whl，然后，用pip逐个本地安装：

>pip install \*.whl

**2.2.2 交互计算 Jupiter**

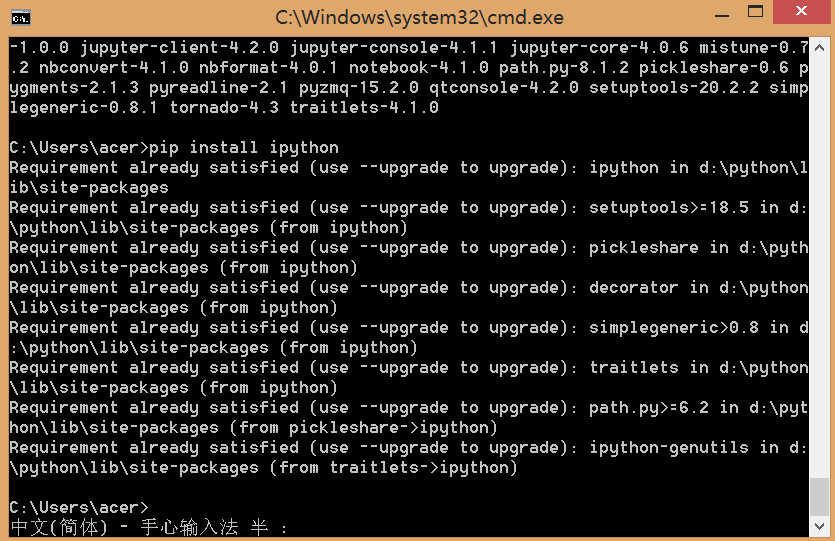
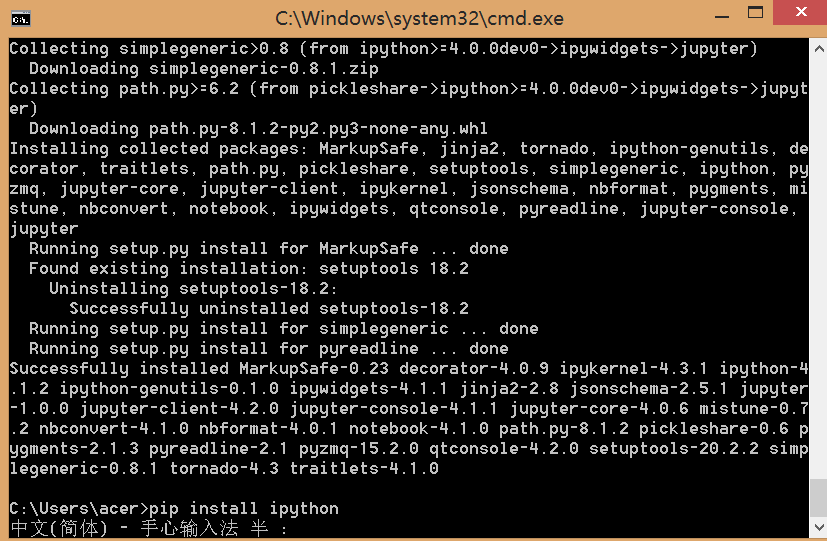
1） 安装Jupiter ：

>pip install jupyter 在线安装



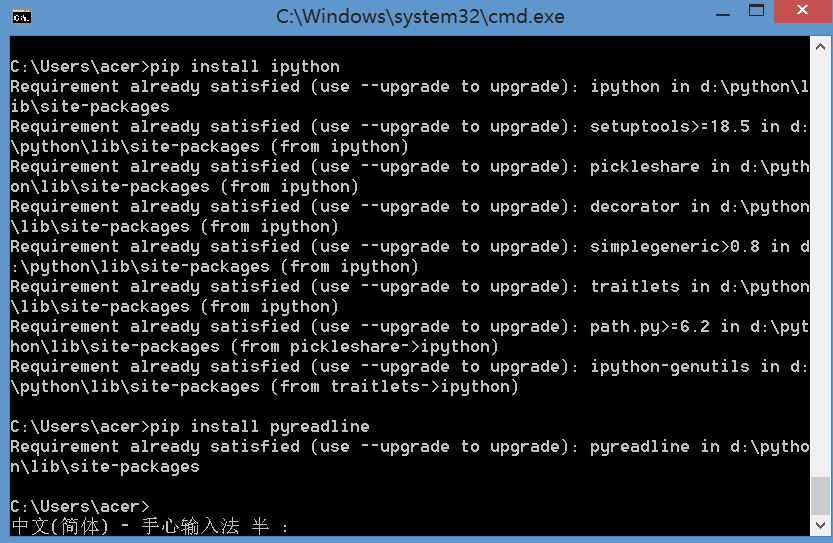
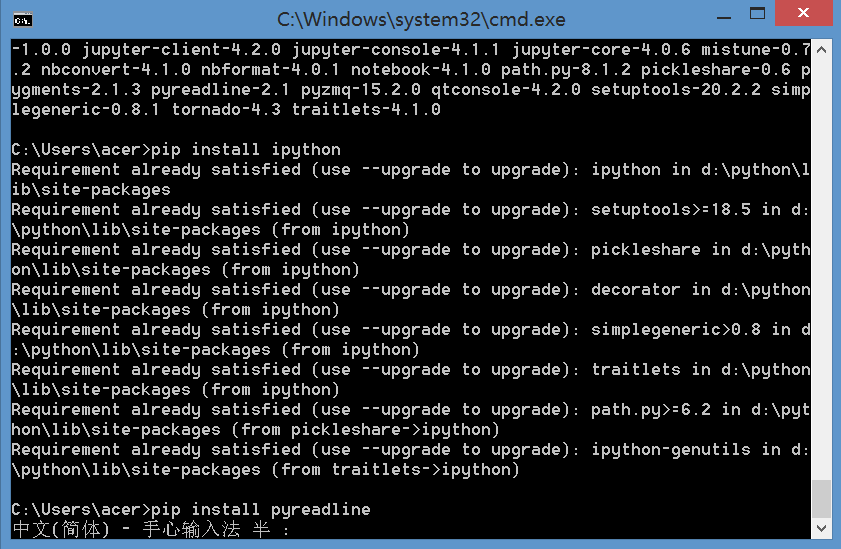
2） 安装Python语言内核

>pip install ipython ，支持Python语言

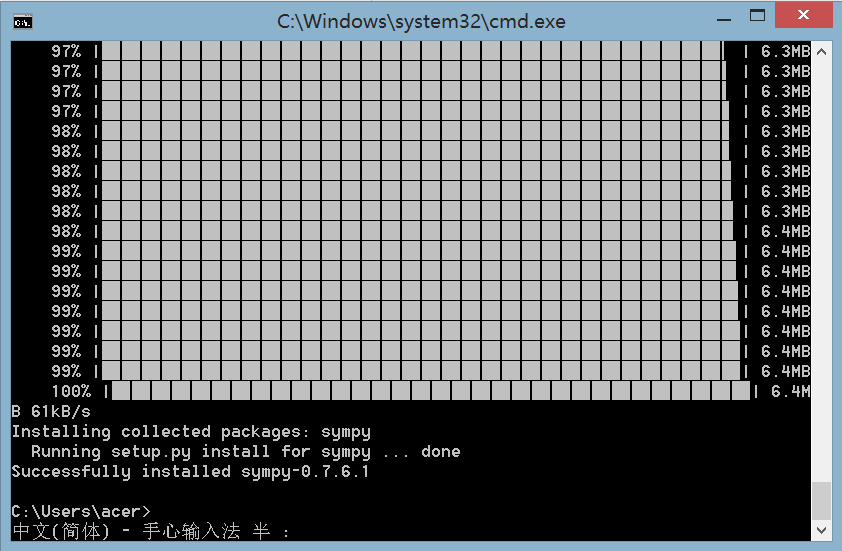
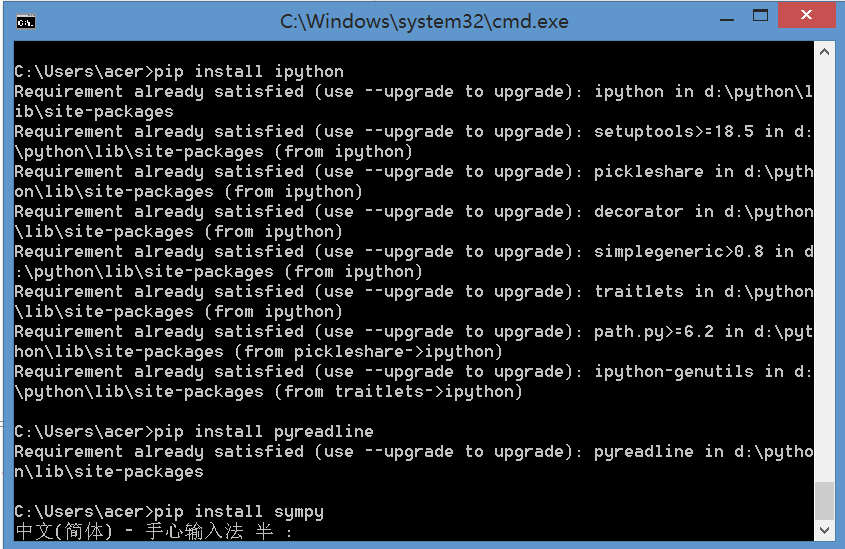


3） 安装依赖包

>pip install pyreadline



>pip install sympy



4）支持显示数学符号、公式，安装MathJax：

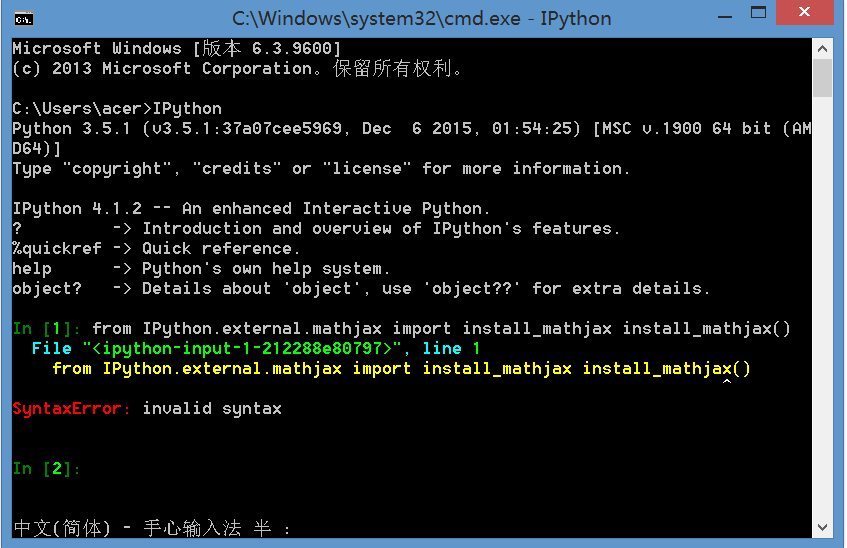
命令行下，

>IPython

打开一个IPython的shell，然后，在其中键入如下代码：

from IPython.external.mathjax import install\_mathjax

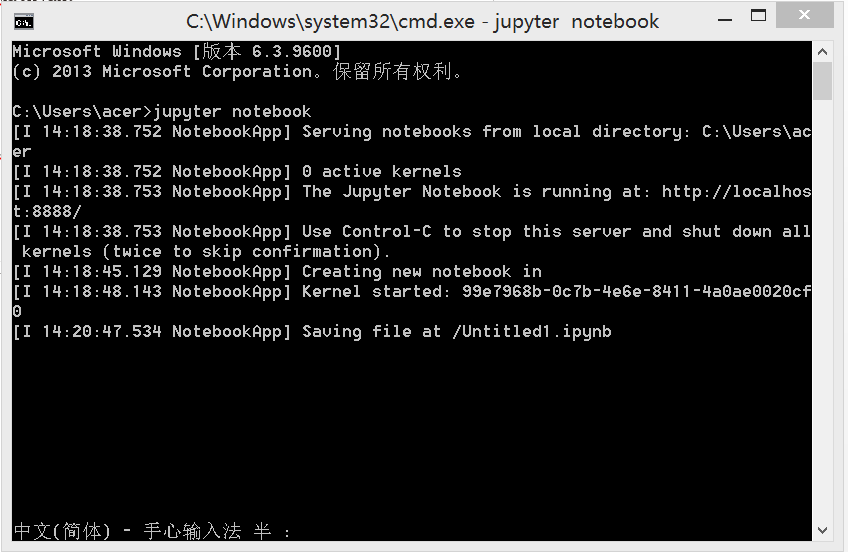
install\_mathjax()



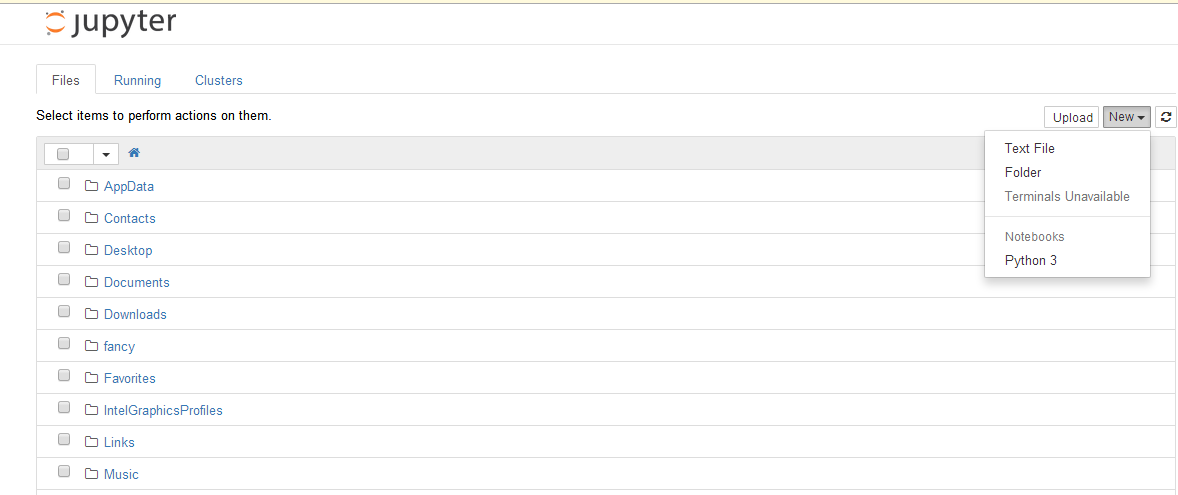
试了好多次，也尝试分段等，均未实现。

5） 运行notebook:

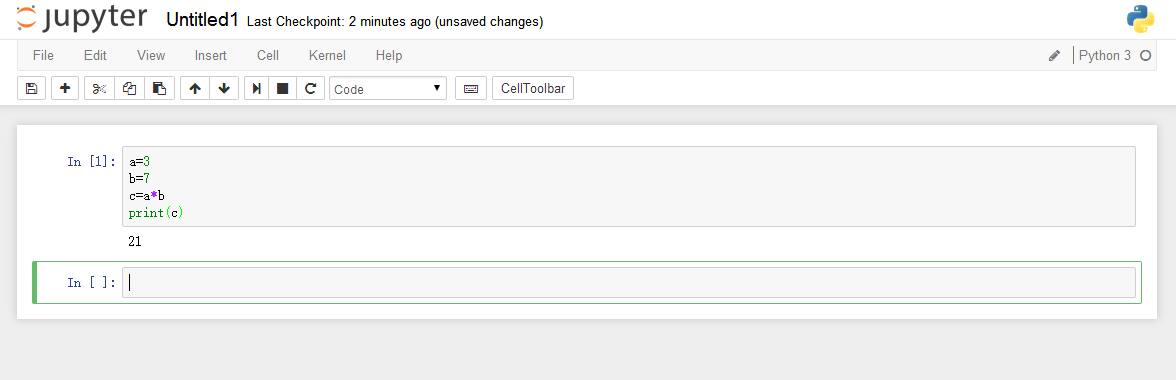
在cmd命令下键入：jupyter notebook，回车



显示图示窗口。新建一个python3文件



实现简单运算



建议：多试试不同的运算，挺好玩的。

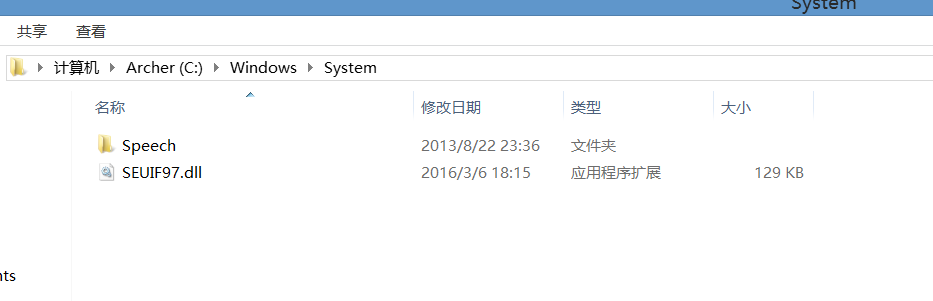
**2.2.3 IF97 for Python**

Windows 32/64位版: 从

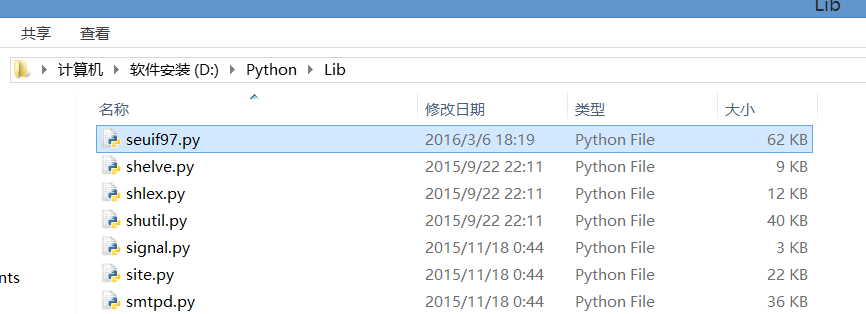
<https://github.com/Py03013052/SEUIF97>

下载：SEUIF97.dll和 seuif97.py，然后：

1. SEUIF97.dll拷贝到 C:\windows\system



1. seuif97.py拷贝到 D:\python\Lib



1. **基于Eclipse的Python开发环境**

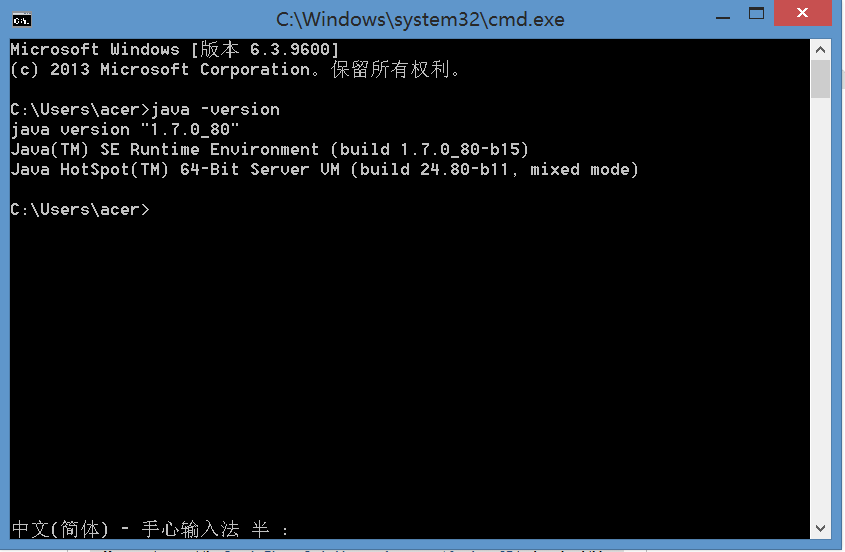
Eclipse 是一个开放[源代码](http://baike.baidu.com/subview/60376/5122159.htm)的、基于[Java](http://baike.baidu.com/subview/29/12654100.htm)的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，包括Java[开发工具](http://baike.baidu.com/view/1355803.htm)（Java Development Kit，[JDK](http://baike.baidu.com/subview/25214/5047948.htm" \t "_blank)）。功能强大、专业性强。

**3.1 安装Eclipse IDE**

**3.1.1安装JavaSDK**

Eclipse IDE是使用Java开发的，电脑中需要预先安装好Java JRE/JDK软件包，因此，安装前，在命名行下，输入：

>java –version



检查是否已经安装了Java软件包。 如果电脑中已经安装了Java ,会显示有关版本**。**

建议：更新吧

此版本非最新版，下载更新：

Java JDK的Oracle官方下载地址如下：

[**http://www.oracle.com/technetwork/java/javase/downloads/index.html**](http://www.oracle.com/technetwork/java/javase/downloads/index.html)

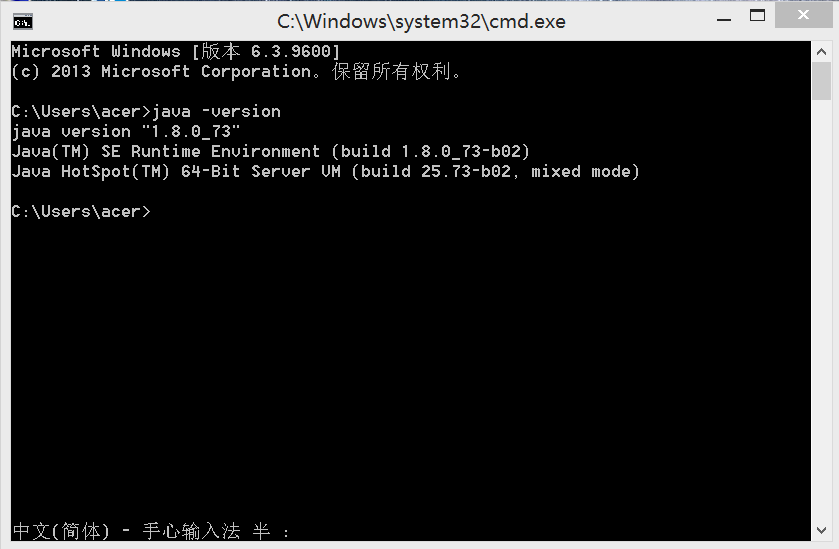
下载时需要注意操作系统64位，选择相应版本下载（\*是Java版本号）：

Windows 64位：\*-windows-i64.exe.

运行下载Oracle的Java安装包（安装时注意配置环境变量），即可安装好Java环境。



再次检查版本，可以了。

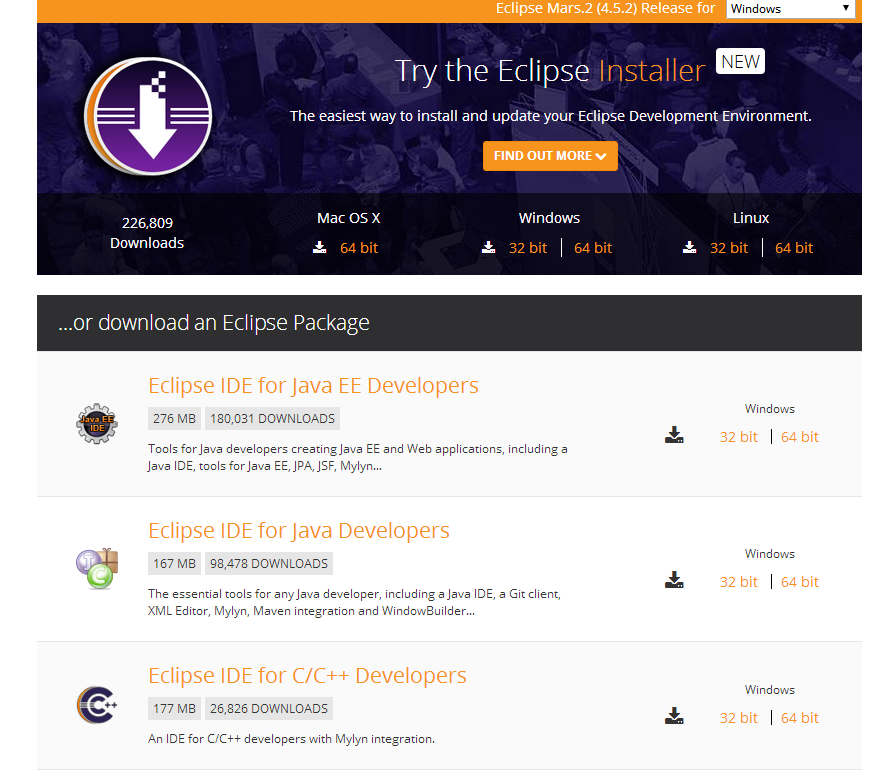


**3.1.2 Eclipse IDE**

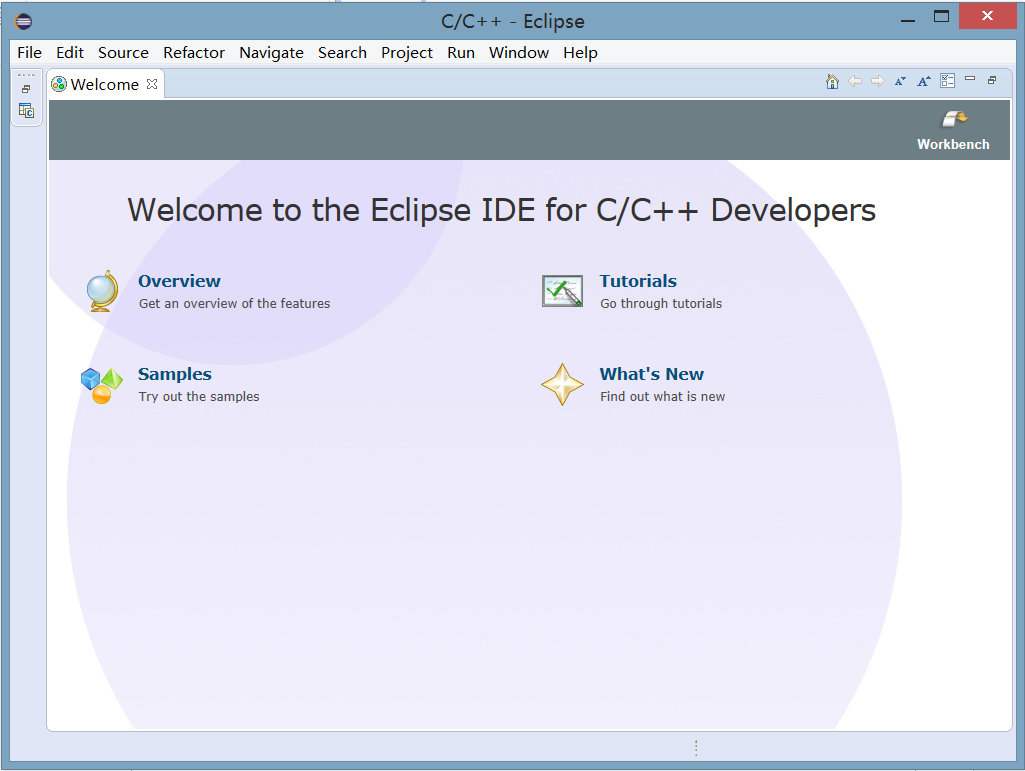
　　Eclipse IDE是插件型开发环境，有很多版本可以下载。本课程主要使用Python语言，会涉及C/C++开发，所以，建议下载Eclipse CDT(Eclipse IDE　for　C/C++)版.

Eclipse CDT官方下载地址：

<http://www.eclipse.org/downloads/>



根据操作系统64位，下载相应的版本，然后将下载的Eclipse CDT解压到指定目录下，运行解压目录下的：eclipse.exe即可。



注意：Eclipes非安装性文件，打开即用，只是打开时需要选择文件保存目录而非安装目录。之前找了半天Eclipes放在哪里，其实，它就在那里，一直都在。

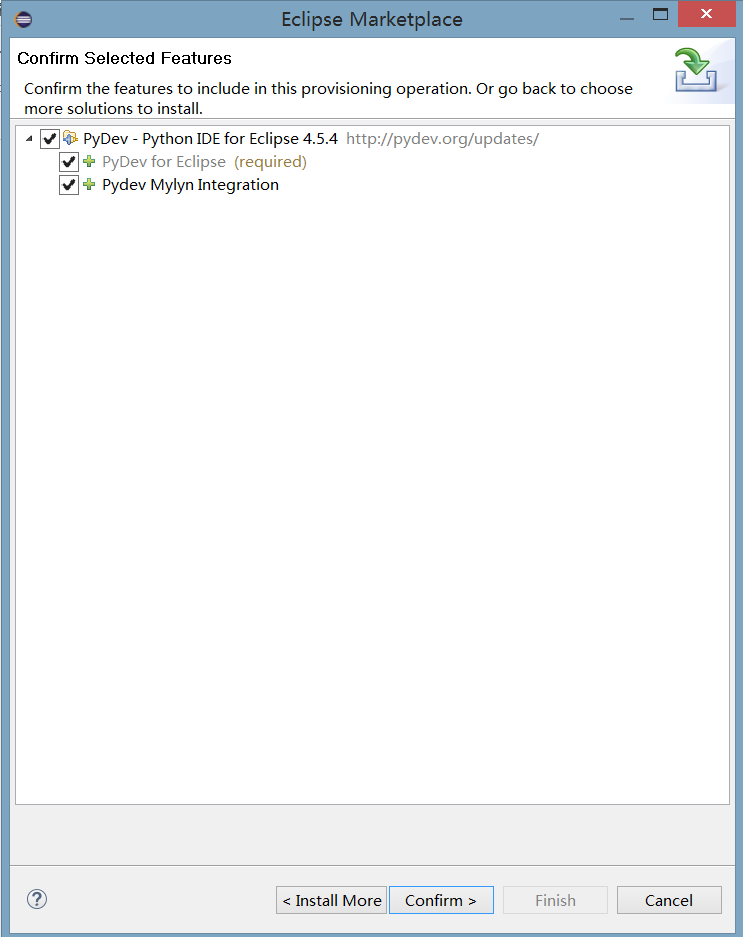
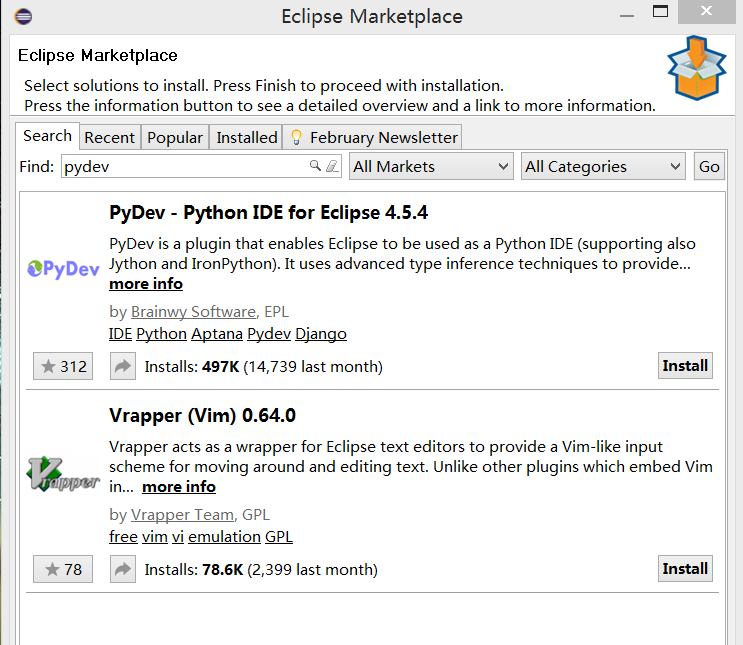
**3.2 安装Python开发插件PyDev**

使用Eclipse IDE作为Python开发环境，需要：

1） 安装PyDev插件；2）配置解释器为Python3.4；

**3.2.1 安装PyDev插件**

通过Windows->Eclipse Marketplaces进入市场，输入Pydev，找到Pydev安装/更新项目，在线安装即可：

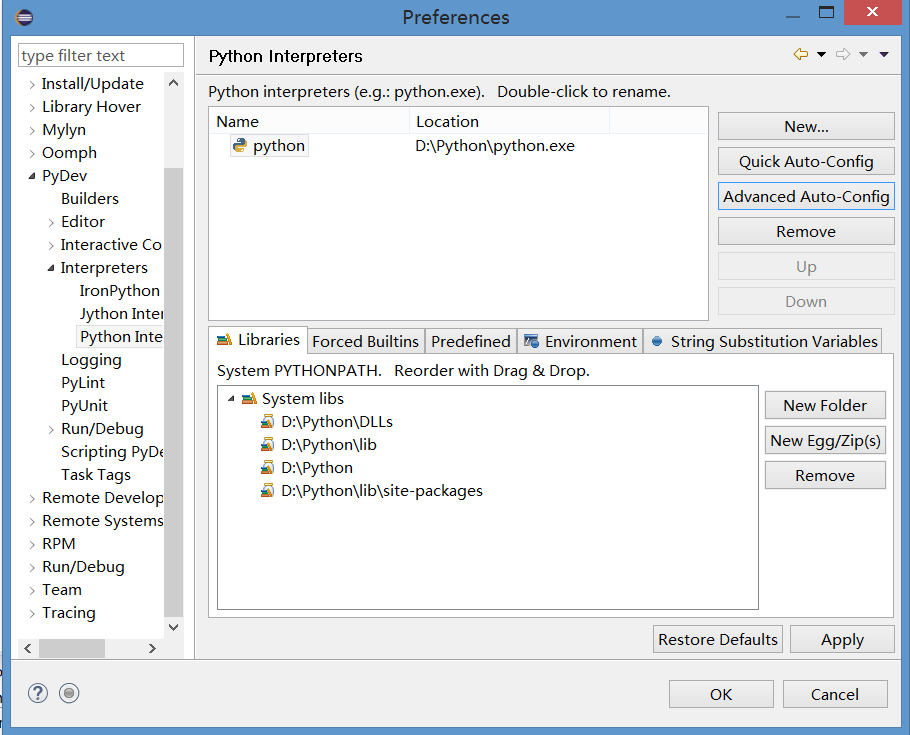


在线安装过程简单，但是，安装速度受网络环境影响。的确很慢！

**3.2.2 配置 Python解释器**

安装好后，重新启动。通过：

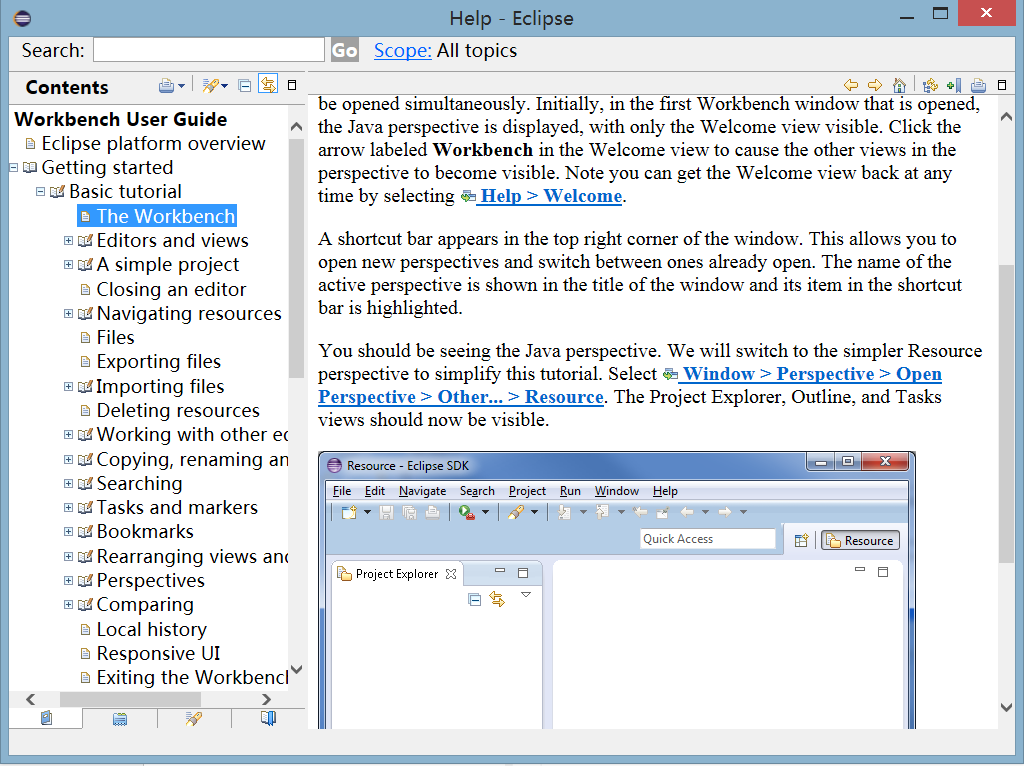
Windows->Preference->Pydev->Interperters->Python Interperter点其中的：Advanced Auto-config配置开发使用的 Python解释器版本:



配置好后切换到Python场景，就可以进行Python开发了。

**3.3 联机指南**

**Help** -> **Help Contents**



1. **配置 PyDev提高效率**

**4.1 显示源码行号**

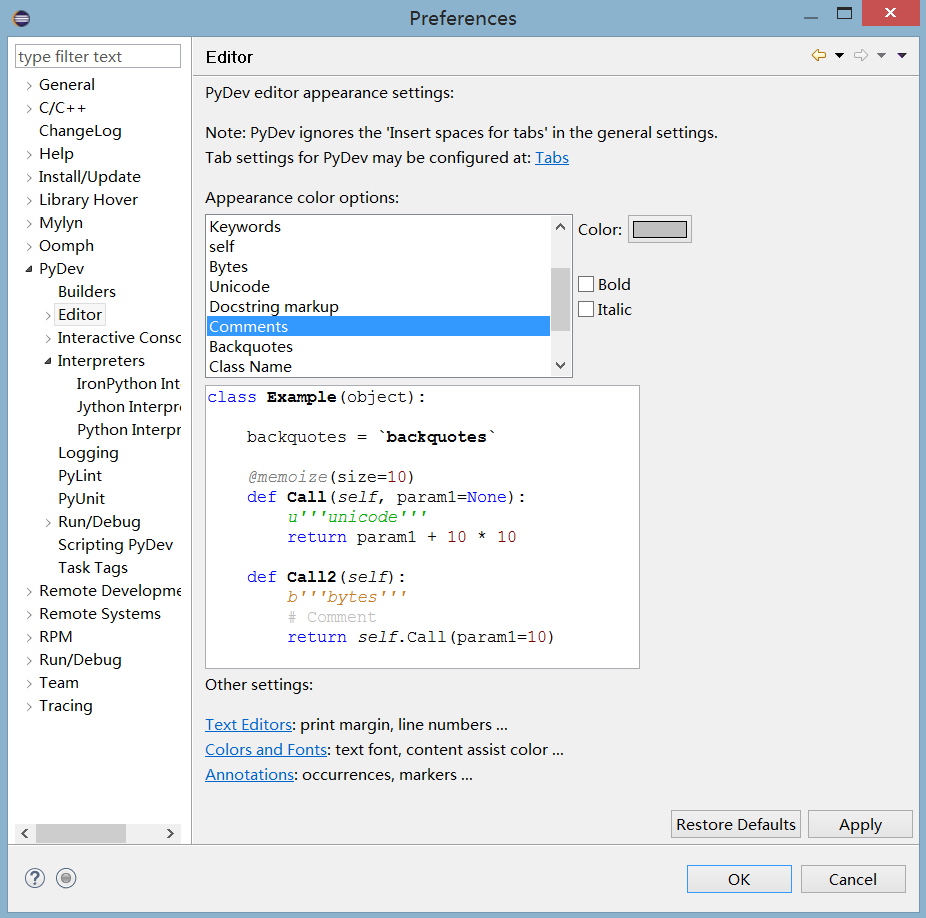
右键源码的左边缘，选中Show Line Number

P.S使用时显示行数方便检查



**4.2 修改注释颜色提高可读行**

从Window → Preferences→PyDev→Editor,进入配置界面：

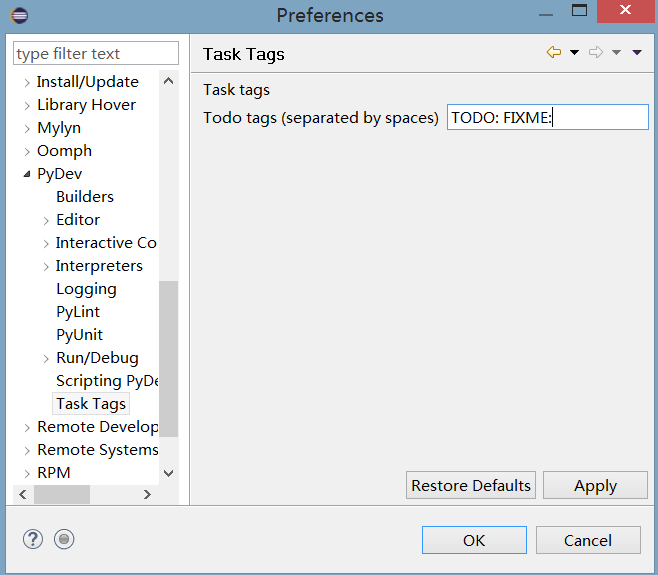
****

**4.3 配置任务标签**

在程序开发过程中可以在代码中标识，当前任务状态，计划开发工作。在代码中标识任务，可以使用任务标签，然后，让开发环境识别到工作空间的任务列表中。

1. **配置任务标签：**

通过PyDev->Task Tags中配置任务标签:



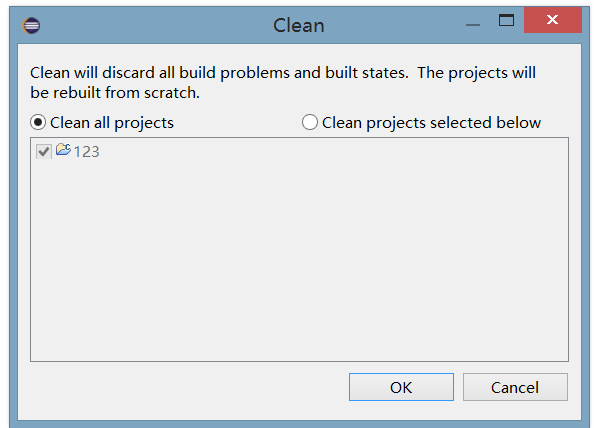
可以根据不同的任务类型，自己增加自定义标签

**2）添加任务标签注释**

在程序源码中加入使用任务标签的注释：

**3）识别任务**

保存新修改、运行程序一次或者选择 Project → Clean，



就可将使用任务标签注释加入任务列表。如果当前任务窗口可视，刚加的任务，就会立刻显示在任务窗口中。

如任务窗口没有打开，，可以Window->Show View->Tasks开启任务窗口。

1. **基于PyDev的语言规范静态检查**

PyDev中集成了PEP8,AutoPEP8 和Pylint代码检查功能，这些功能默认状态都是关闭的。

程序开发过程中，要有规范意识，但不可能有很高的规范性，过分注意规范会影响开发进程。这时如果一直开启代码规范检查，经常提示不规范，会对开发形成负面影响，所以，默认关闭是合适的。在程序开发一个阶段结果出来时，进行规范性检查更好。

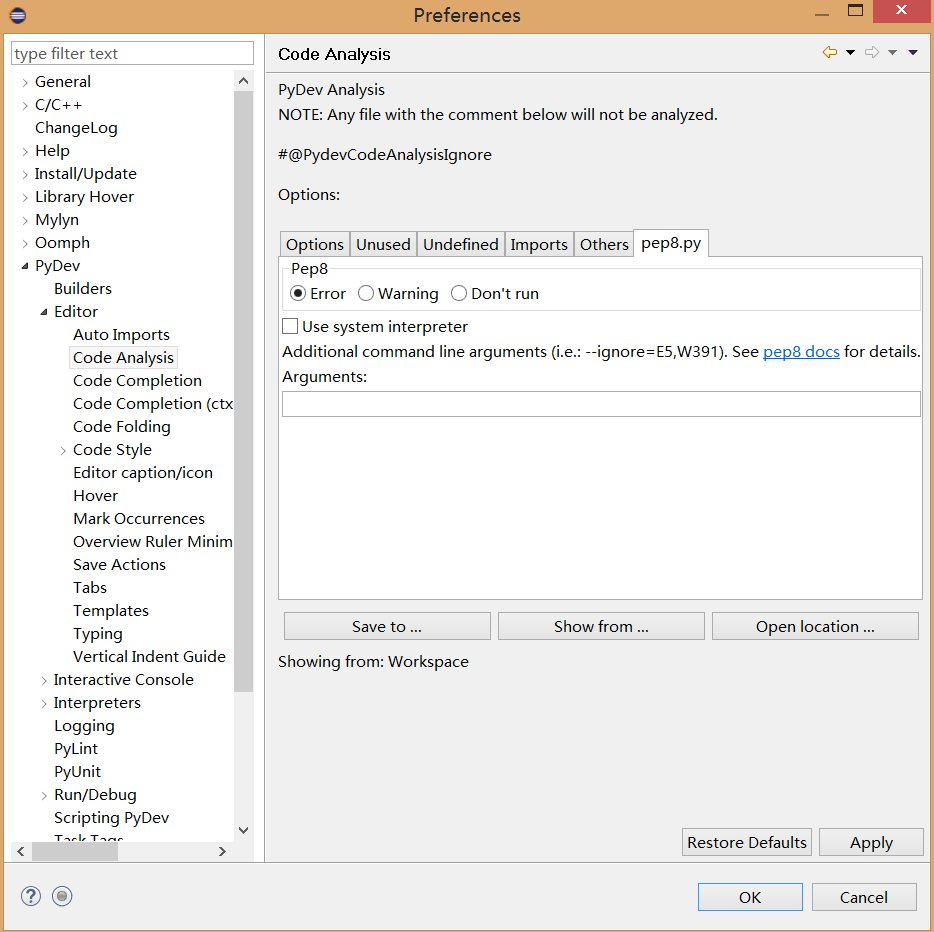
**5.1 PEP8检查和修改**

1）启动 pep8 检查：

Window > Preferences

PyDev > Editor > Code Analysis > pep8.py

选择Errors/Warnings其中之一..

****

右键Python工程，选择 PyDev，点 "code analysis"，即可对工程中所有Python源码进行PEP8检查:



1）启动autopep8自动修改：

点Windows -> Preferences ->输入 'autopep8' 作为搜索串.

选择（Check）: Use autopep8.py for code formatting?

|  |
| --- |
|  |

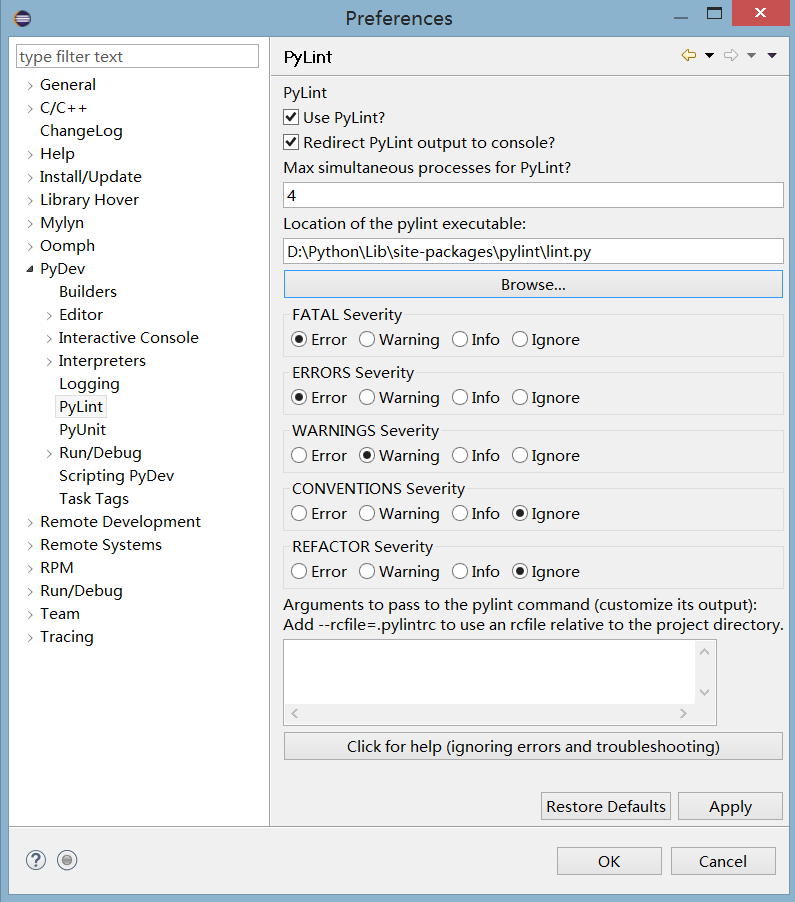
在Python源码窗口，按 CTRL-SHIFT-F 就可以自动修改代码

**5.2 使用Pylint**

PyDev 默认不开启Pylint。通过

     Window -> preferences -> Pydev -> Pylint,选中"Use pylint?",

找到安装好的lint.py的地址,例如"C:\Python34\Lib\site-packages\pylint\lint.py"



**小结**

刚接触Python，就安装而言，在这一过程中我逐渐对python的组成和基本功能有了一个初步的了解：pip安装、python开发环境与扩展模块、在网页上用python进行简单的编程，但仍有很多不懂的地方，比如whl文件的安装总是显示失败，对Eclipse加PyDev不够了解等等。

仅是这次的python与其开发环境、扩展模块的建立，我了解了很多相关知识，能够利用互联网解决操作上的难题，但更多是问题，相信在今后的学习中会找到相关答案！