**Python学习和开发环境的建立**

Python是一种面向对象、解释型[计算机程序设计语言](http://baike.so.com/doc/5912108-6125016.html" \t "http://baike.so.com/doc/_blank)，其语法简洁而清晰，具有丰富和强大的类库。它常被昵称为胶水语言，能够把用其他语言制作的各种模块(尤其是C/C++)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。

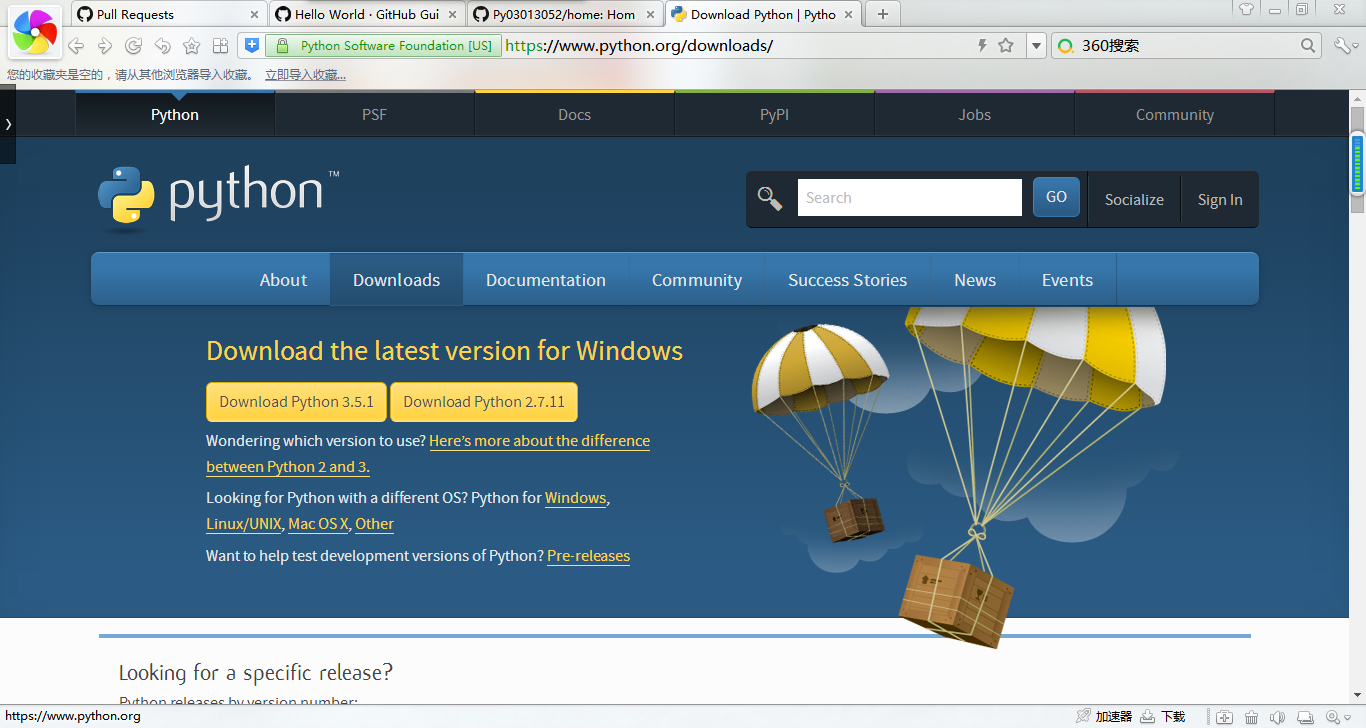
大型软件开发宜采用集成开发工具，常用的有: Eclipse，VS2015等，其中，Eclipse加PyDev插件的方式， 为较多Python程序开发人员使用。此外，还有使用编辑器(如Atom)构建开发环境的方法。

1. **Python基本开发环境建立**

从官网：

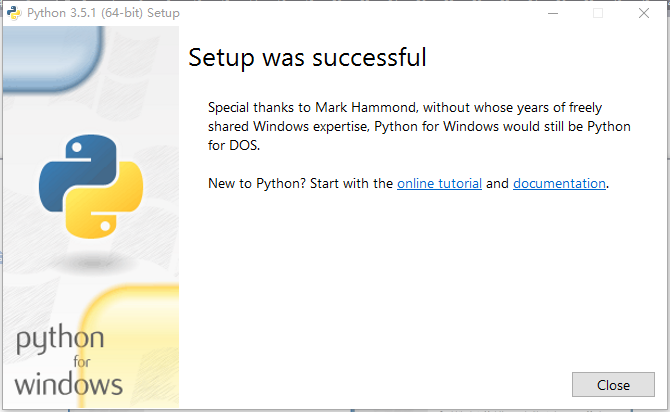
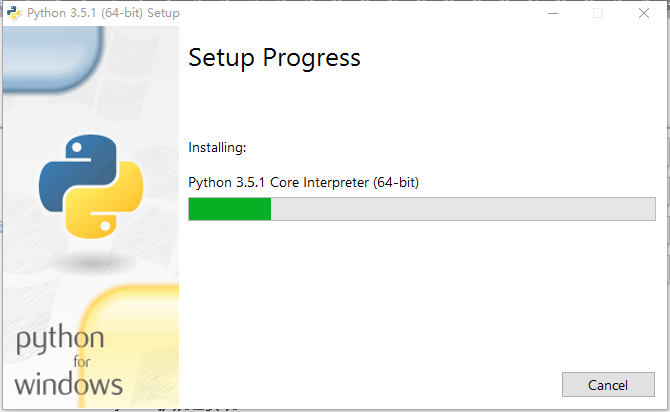
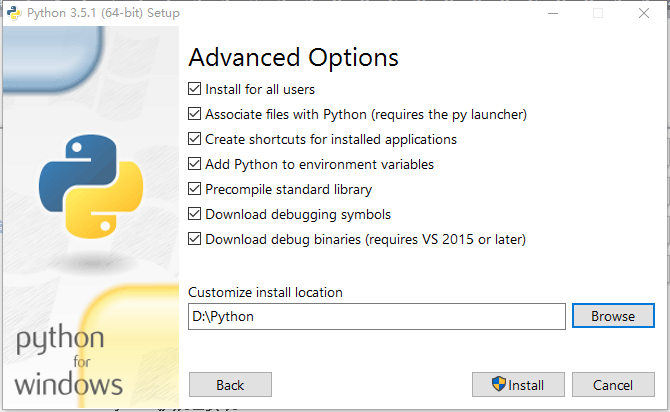
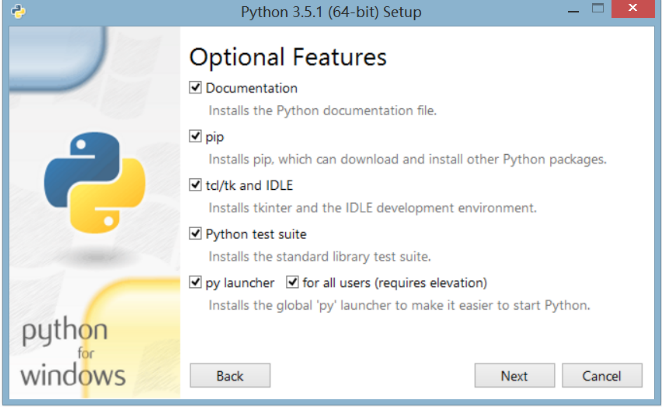
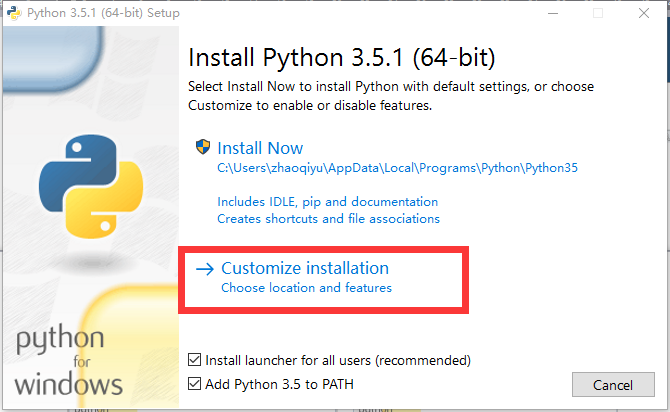
<http://www.python.org>

下载和操作系统（64位）对应版本的Python3软件。



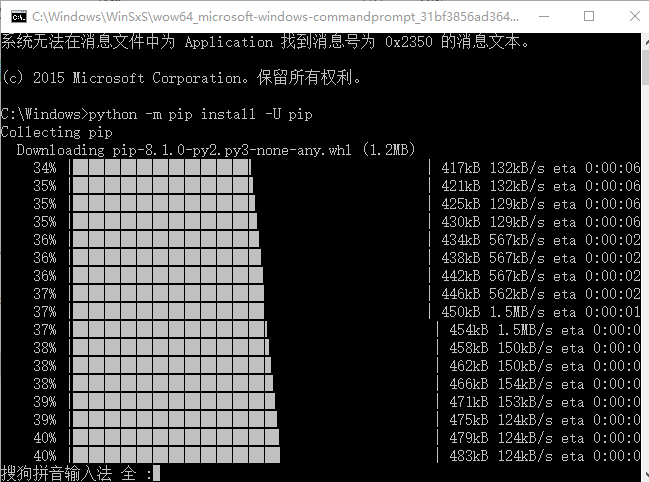
下面以Python3.5.1版本安装为例做简单说明

**Windows 10：** 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，在cmd中输入以下命令进行pip的更新：

python -m pip install -U pip



二、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。

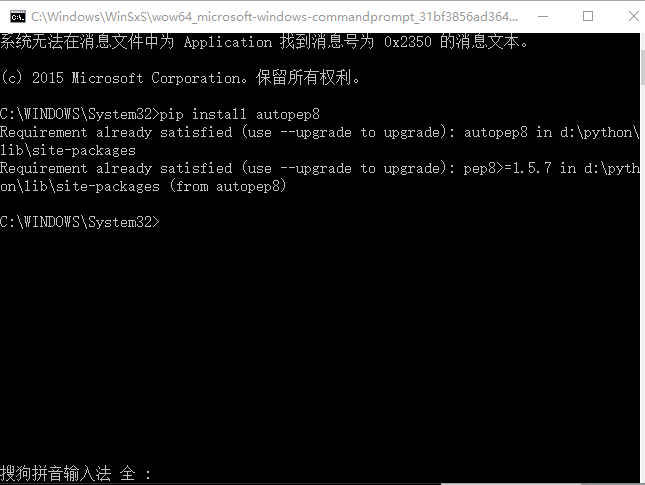
这时可使用： 加州大学欧文分校提供的Windows32/64位Python扩展包编译版，

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

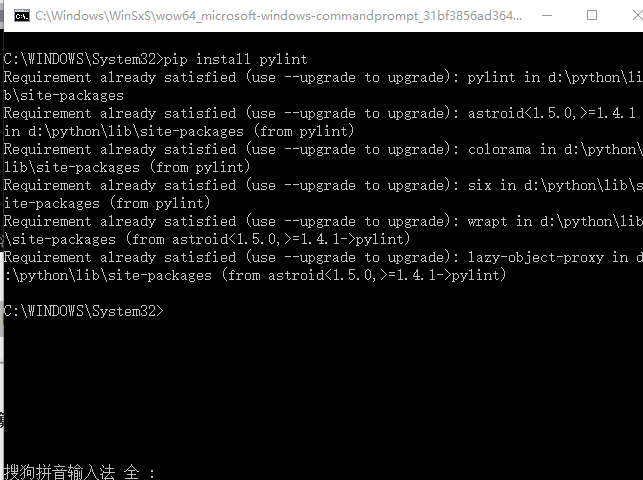
加州大学欧文分校的编译版包，内容丰富、更新及时、安装方便。

2.1 编码规范

pip install autopep8



pip install pylint（方法如上）



2.2 科学计算软件包

**2.2.1 SciPy**

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

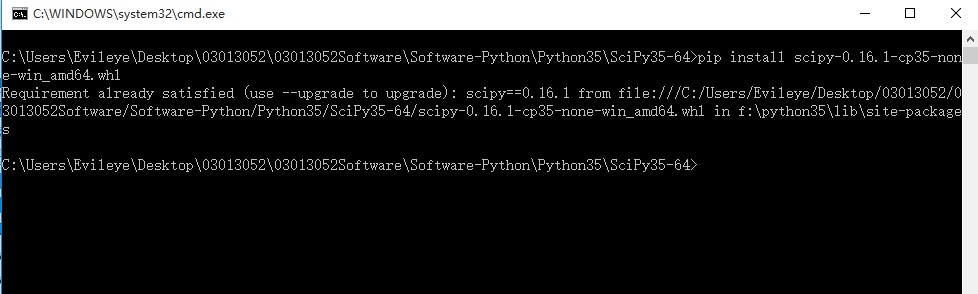
scipy

matplotlib

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

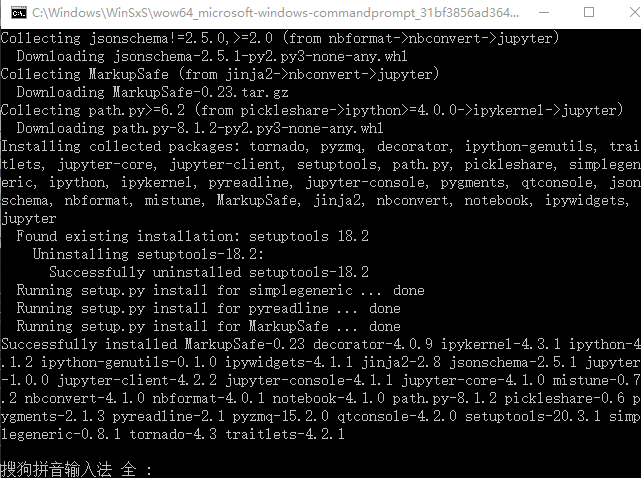
示例安装scipy：找到下载whl文件所在位置，然后shift+鼠标右键在此处打开命令窗口，键入>pip install scipy-0.16.1-cp35-none-win\_amd64.whl



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（方法如上）

**3） 安装依赖包**

>pip install pyreadline

>pip install sympy （方法如上）

**4）支持显示数学符号、公式，安装MathJax：**

命令行下，

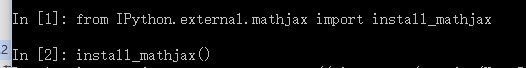
>IPython

打开IPythonshell，然后，在In[1]:后键入：

from IPython.external.mathjax import install\_mathjax

在In[2]:后键入代码：

install\_mathjax()



**5） 运行notebook:**

在iPython notebook文件所在目录下，打开命令行窗口：

>jupyter notebook

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

2) seuif97.py拷贝到 D:\python35\Lib

三、基于Eclipse的Python开发环境

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：

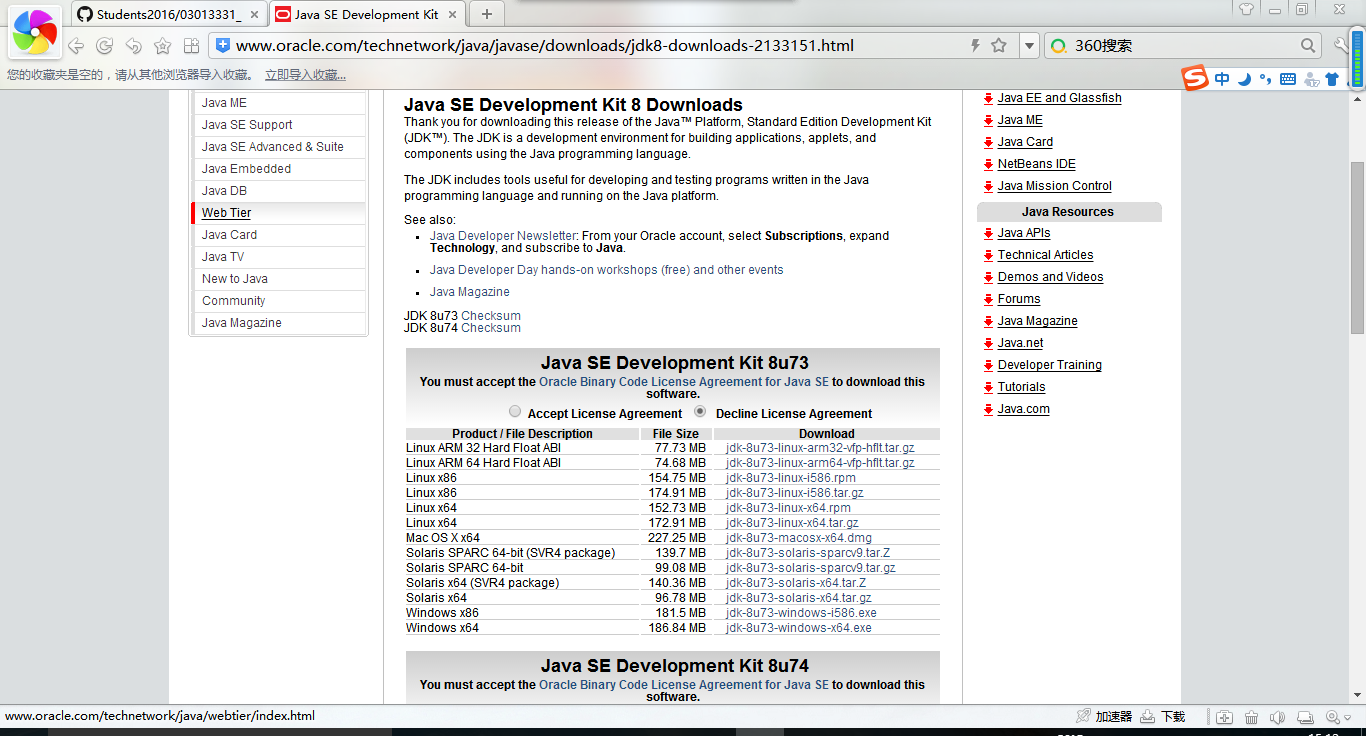
>java –version

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。

如果没有安装java 或者版本不能满足Eclipse IDE最低要求，需要从Oracle下载Java包。

Java JDK的Oracle官方下载地址如下：

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>



下载时需要注意操作系统32/64位，选择相应版本下载（\*是Java版本号）：

Windows 32位：\*-windows-i586.exe

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。

**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32/63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。

如果使用Windows7 以上版本操作系统，建议将运行eclipse.exe，固定到任务栏。（在eclipse.exe文件名上，点鼠标右键即可）





以后，从任务栏启动Eclipse方便很多（同样也可以将IDLE固定到任务栏）。

**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。

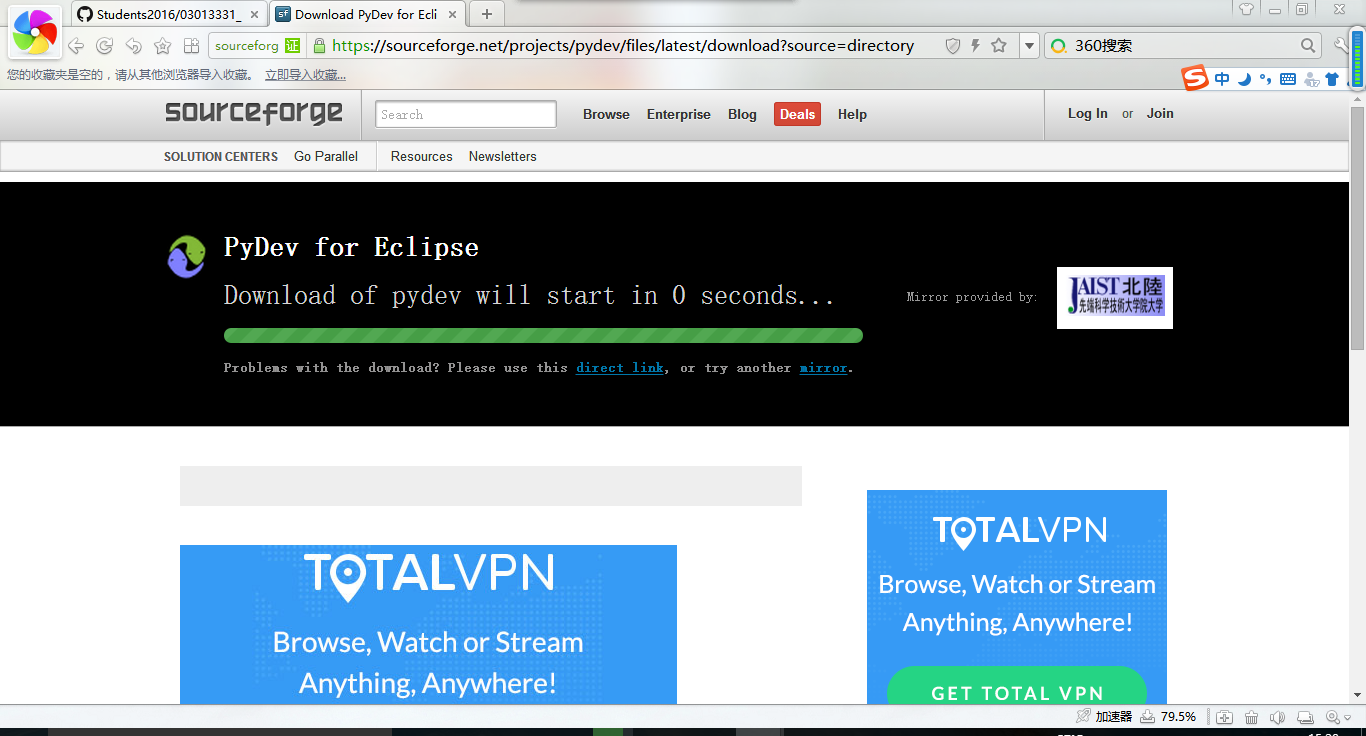
如果很慢，可从pydev官网（<http://www.pydev.org/>）提供的下载地址：

http://sourceforge.net/projects/pydev/files/

下载插件包，然后，通过：

Help->Install->New Software->Add->Archive

离线安装.



**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。

**四、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



**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。

五、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"



配置参数，限制Pylint的输出



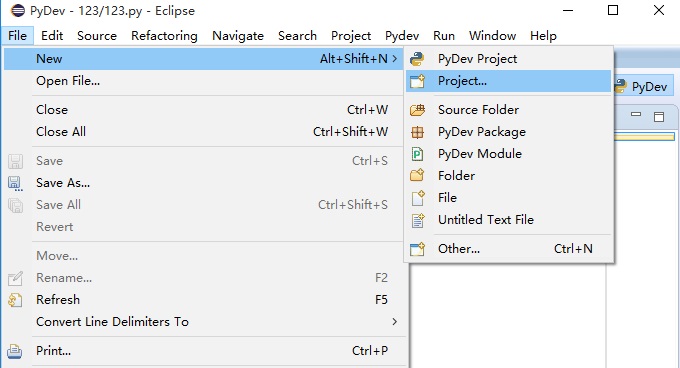
选中Project->Build Automatically,这样程序修改，保存时pylint就会自动检查项目中的代码,也可用Ctrl+B手动build触发pylint。

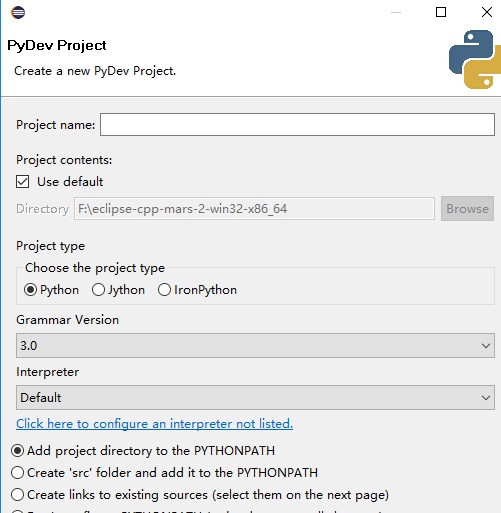


六、程序实现及运行过程

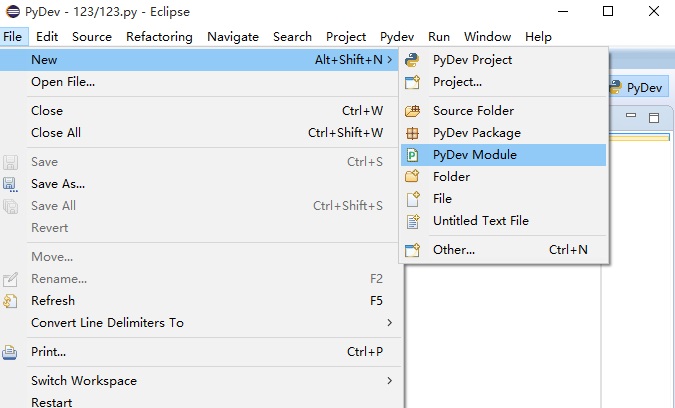
1）打开Eclipse并进入PyDev开发环境

2）进入File<New<PyDev Project,新建新工程

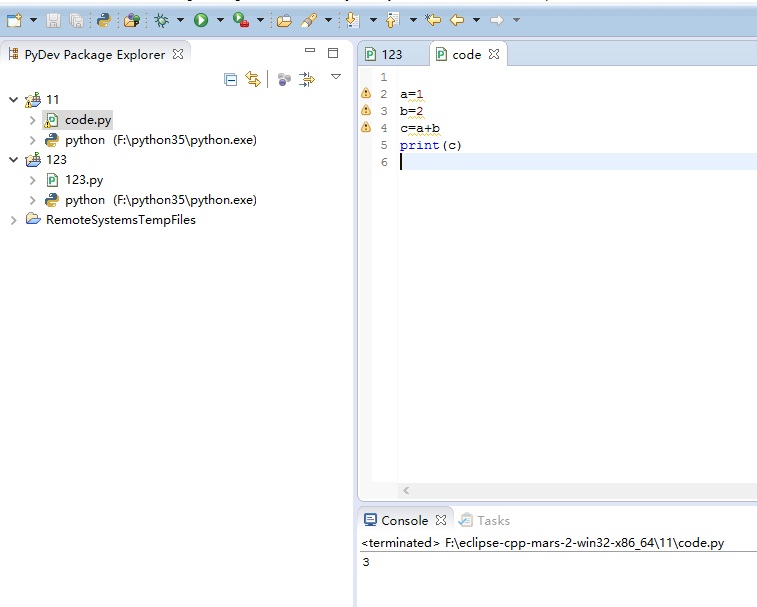


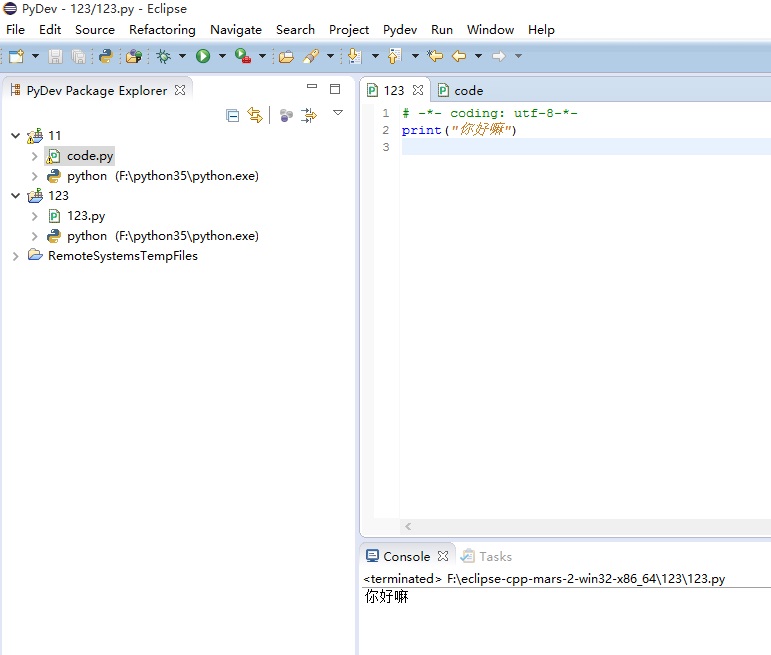


3）进入进入File<New<PyDev Module,新建Module



4）键入代码，运行程序

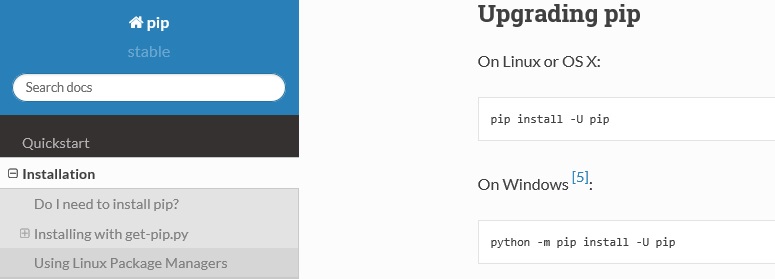




七、安装过程中出现的问题及解决办法

**7.1 用pip install --upgrade pip 安装时提示失败**

解决方案：用下列命令行：python -m pip install -U pip



参考pypi网站，遇到问题可以从python社区寻找答案。

**7.2 安装Maxjax出现问题**

解决方法命令行>IPython

打开IPythonshell，然后，在In[1]:后键入：

from IPython.external.mathjax import install\_mathjax

在In[2]:后键入代码：install\_mathjax()

**7.3 Eclipse编程时无法输出汉字**

解决办法加上 #coding=gbk 或者 #-\*-conding:utf-8-\*-；

两者之间的差别是：字符均使用双字节来表示，只不过为区分中文，将其最高位都定成1。至于UTF－8编码则是用以解决国际上字符的一种多字节编码，它对英文使用8位（即一个字节），中文使用24位（三个字节）来编码。对于英文字符较多的论坛则用UTF－8节省空间。GBK包含全部中文字符；UTF-8则包含全世界所有国家需要用到的字符。GBK是在国家标准GB2312基础上扩容后兼容GB2312的标准（好像还不是国家标准）UTF-8编码的文字可以在各国各种支持UTF8字符集的浏览器上显示。比如，如果是UTF8编码，则在外国人的英文IE上也能显示中文，而无需他们下载IE的中文语言支持包。 所以，对于英文比较多的论坛 ，使用GBK则每个字符占用2个字节，而使用UTF－8英文却只占一个字节。UTF8是国际编码，它的通用性比较好，外国人也可以浏览论坛，GBK是国家编码，通用性比UTF8差，不过UTF8占用的数据库比GBK大。

**小结**

对于基础接触Python的我来说，这个确实是有些许的难度，不过在对Python与其开发环境、扩展模块的建立过程中，我也解到了很多相关知识，并且能够利用互联网解决操作上的难题。虽说刚开始这个过程是很难，不过我也一直在努力，相信在今后的学习中我会慢慢跟上学习的进度，慢慢喜欢上这样一门实用性高的学科。

**参考文献**

1．Brainwy Software Ltd. PyDev Manual. http://www.pydev.org/manual\_101\_root.html

2．郑伟芳. PyDev for Eclipse简介. <http://www.ibm.com/developerworks/cn/opensource/os-cn-ecl-pydev/> 2008.11

3．张颖. Python 代码调试技巧. <http://www.ibm.com/developerworks/cn/linux/l-cn-pythondebugger/> 2012.05