2016-3-23

03013414 郑伟佳

东南大学能源与环境学院

课程实践作业一

目录

[**Python开发环境建立指导手册** 2](#_Toc446711885)

[**一、Python基本开发环境建立** 2](#_Toc446711886)

[**二、Python扩展包安装** 3](#_Toc446711887)

[**2.1 编码规范** 4](#_Toc446711888)

[**2.2 科学计算软件包** 4](#_Toc446711889)

[**三、基于Eclipse的Python开发环境** 9](#_Toc446711890)

[**3.1 安装Eclipse IDE** 9](#_Toc446711891)

[**3.2 安装Python开发插件PyDev** 11](#_Toc446711892)

[**3.3 联机指南** 13](#_Toc446711893)

[**四、配置 PyDev提高效率** 13](#_Toc446711894)

[**4.1 显示源码行号** 13](#_Toc446711895)

[**4.2 修改注释颜色提高可读行** 14](#_Toc446711896)

[**4.3 配置任务标签** 14](#_Toc446711897)

[**五、基于PyDev的语言规范静态检查** 16](#_Toc446711898)

[**5.1 PEP8检查和修改** 16](#_Toc446711899)

[**5.2 使用Pylint** 18](#_Toc446711900)

[**六、编写及运行程序** 19](#_Toc446711901)

[**七、安装配置使用过程中出现的问题、解决过程方法** 23](#_Toc446711902)

[**7.1安装Python后无法更新pip** 23](#_Toc446711903)

[**7.2无法用网页打开Notebook** 24](#_Toc446711904)

[**7.3无法正确安装MathJax** 24](#_Toc446711905)

[**7.4 Eclipse编程时无法输出汉字** 25](#_Toc446711906)

**Python开发环境建立指导手册**

Python是一种解释型、面向对象、动态数据类型的高级[程序设计语言](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=8645833)，也是一种功能强大而完善的通用型语言，具有丰富和强大的类库，被广泛应用于处理系统管理任务和[Web编程](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=66957537)。学习和软件开发需要建立Python的基本开发环境，本文就将Python开发环境如何建立进行指导。

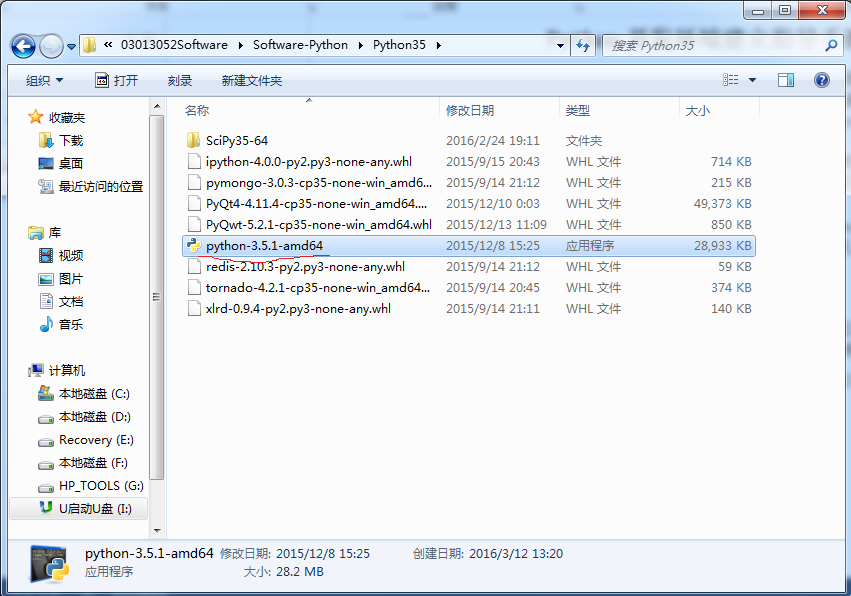
Python解释器、标准库及扩展包构成Python的基本开发环境，满足学习和软件开发的要求。Python解释器及其标准库可从Python官网下载；其他扩展包可用pip命令等方法安装。

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。

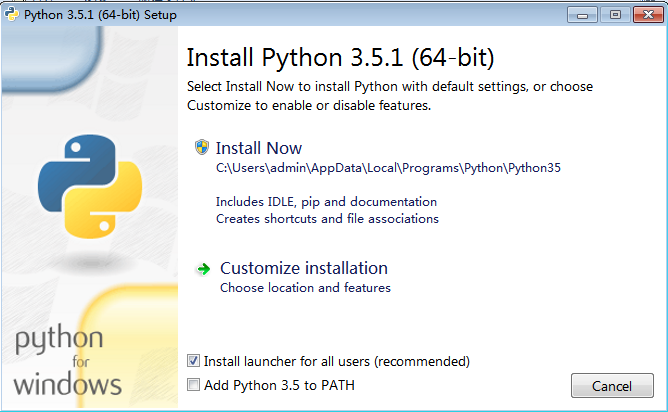
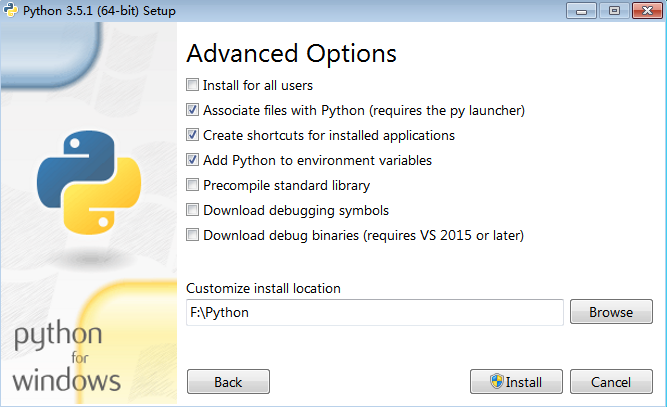
**一、Python基本开发环境建立**

Python主要[版本](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=221389)为2.x和3.x，目前暂时2.x使用居多。相对于Python的早期版本，Python 3.0版本是一个较大的[升级](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=64617477)。为了不带入过多的累赘，Python 3.0在设计的时候没有考虑[向下兼容](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=266268)。虽然许多针对早期Python版本设计的程序都无法在Python 3.0上正常运行，但是由于Python3.0版本有较大的升级，新装Python建议使用Python 3.0版本的语法。

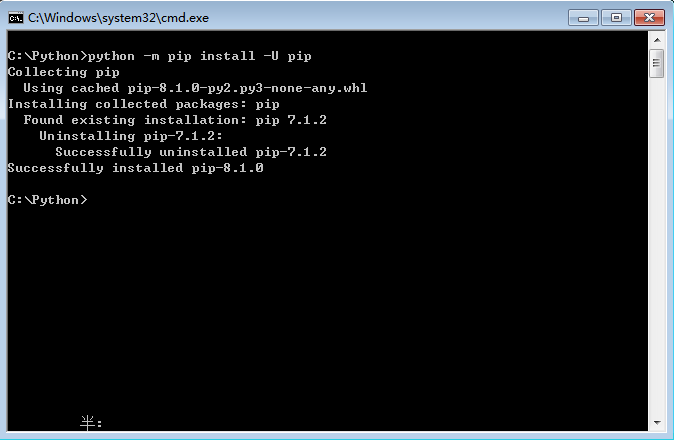
从官网：http://www.python.org下载和操作系统（32/64位）对应版本的Python3软件。下面以Python3.5.1版本安装为例做简单说明。



**Windows：**使用定制方式安装: 1) 加环境变量PATH; 2) 自定义安装目录： C:\python

然后，命令行下更新pip到新版本：

>pip install --upgrade 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。

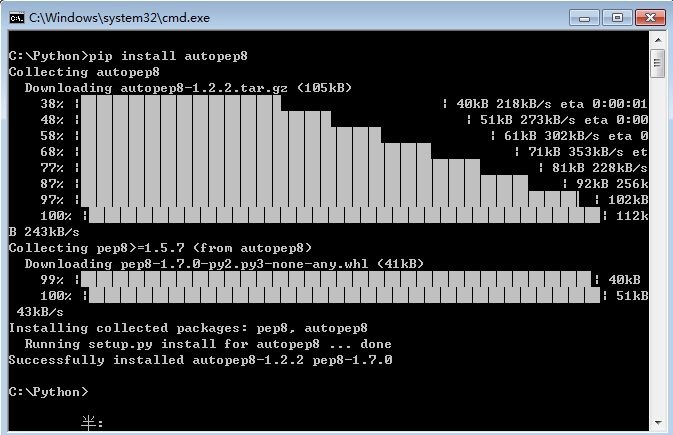
这时可使用： 加州大学欧文分校提供的Windows32/64位Python扩展包编译版，

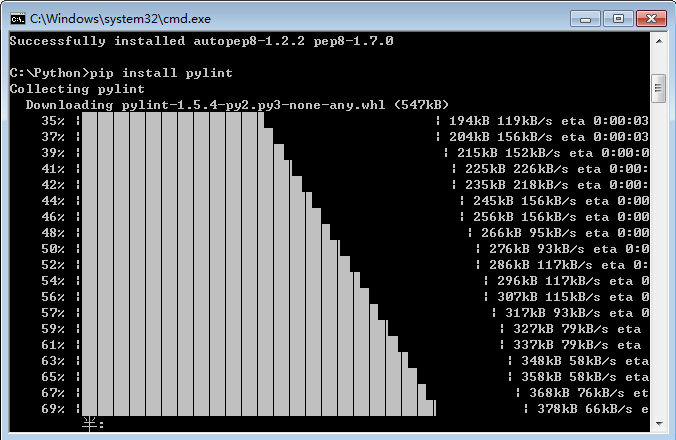
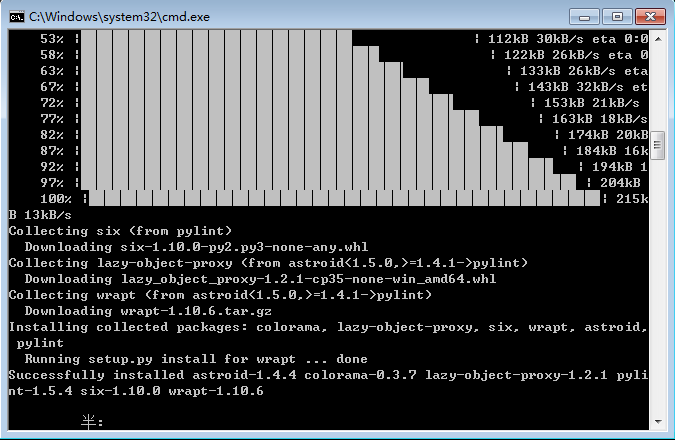
<http://www.lfd.uci.edu/~gohlke/pythonlibs/>

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

**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

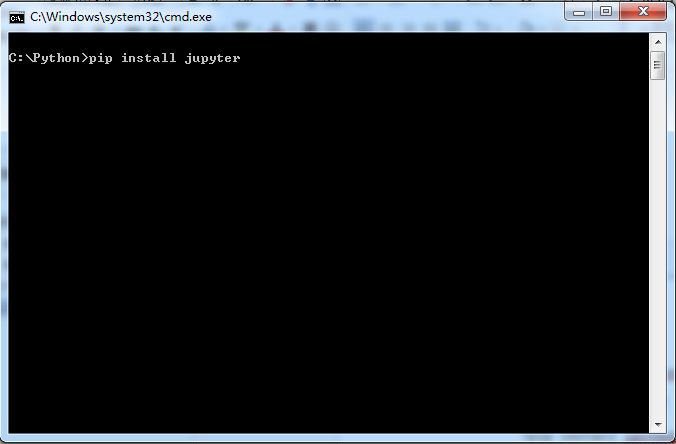
**Window**环境下Scipy 软件包的 安装，建议从加州大学欧文分校下载编译好的\*.whl，然后，用pip逐个本地安装：

>pip install \*.whl

**2.2.2 交互计算 Jupiter**

1）安装Jupiter ：

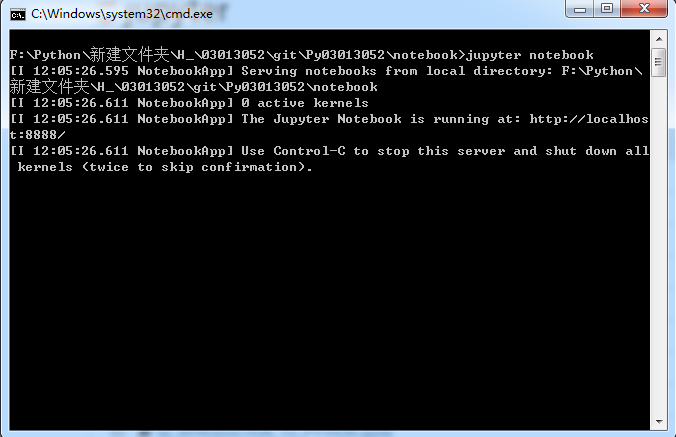
>pip install jupyter 在线安装



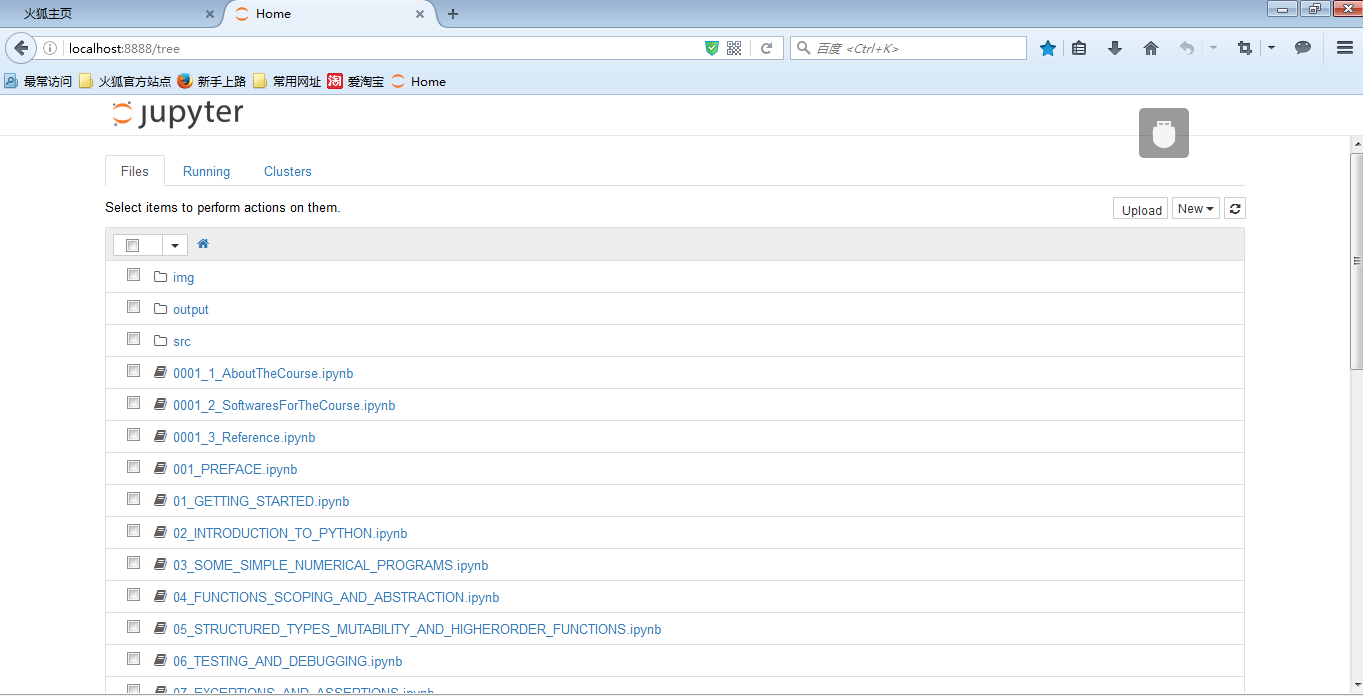
2）运行notebook:

在iPython notebook文件所在目录下，打开命令行窗口：

>jupyter notebook



打开的界面如下：



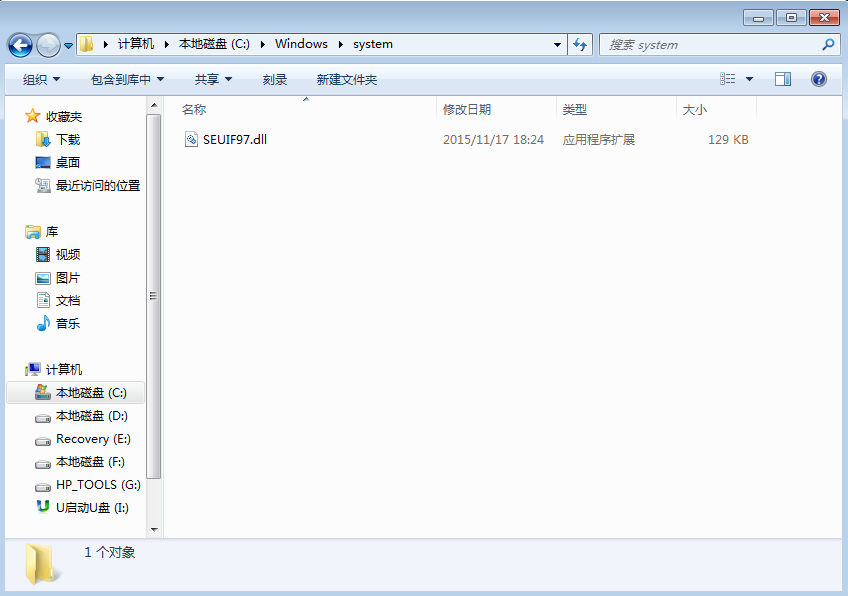
**2.2.3 IF97 for Python**

Windows 32/64位版: 从

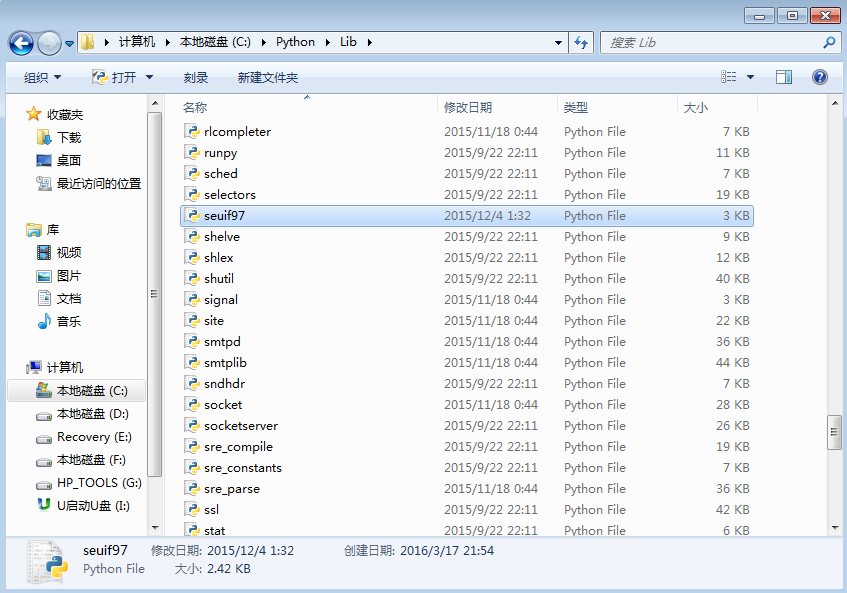
<https://github.com/Py03013052/SEUIF97>

下载：SEUIF97.dll和 seuif97.py，然后：

1)SEUIF97.dll拷贝到 c:\windows\system



1. seuif97.py拷贝到 c:\python35\Lib



**三、基于Eclipse的Python开发环境**

**3.1 安装Eclipse IDE**

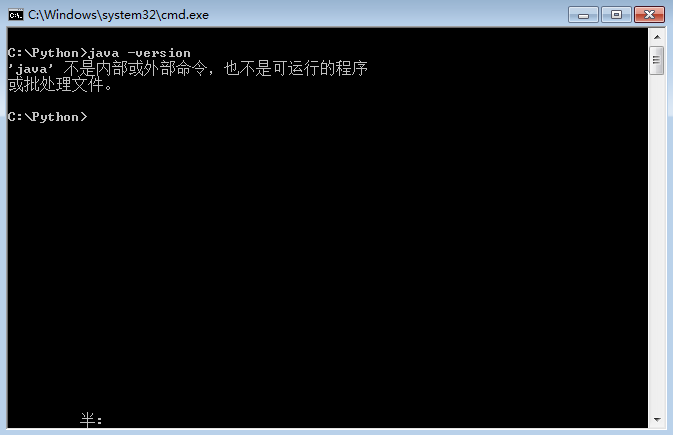
Eclipse 是一个开放[源代码](http://baike.baidu.com/subview/60376/5122159.htm)的、基于[Java](http://baike.baidu.com/subview/29/12654100.htm)的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，包括Java[开发工具](http://baike.baidu.com/view/1355803.htm)（Java Development Kit，[JDK](http://baike.baidu.com/subview/25214/5047948.htm)）。尽管 Eclipse 是使用Java语言开发的，但它的用途并不限于 Java 语言；例如，支持诸如[C/C++](http://baike.baidu.com/view/1107274.htm)、[COBOL](http://baike.baidu.com/view/90933.htm)、[PHP](http://baike.baidu.com/view/99.htm)、Android等编程语言的插件已经可用，或预计将会推出。Eclipse的本身只是一个框架平台，但是众多插件的支持使得Eclipse拥有其他功能相对固定的IDE软件很难具有的灵活性。许多软件开发商以Eclipse为框架开发自己的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。

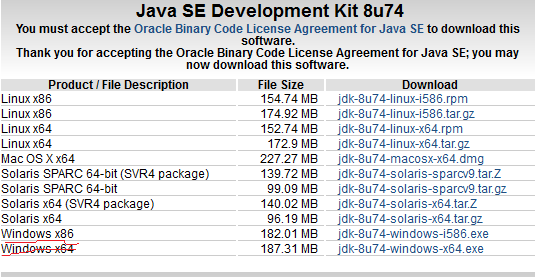
Java JDK的Oracle官方下载地址如下：

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

****

****

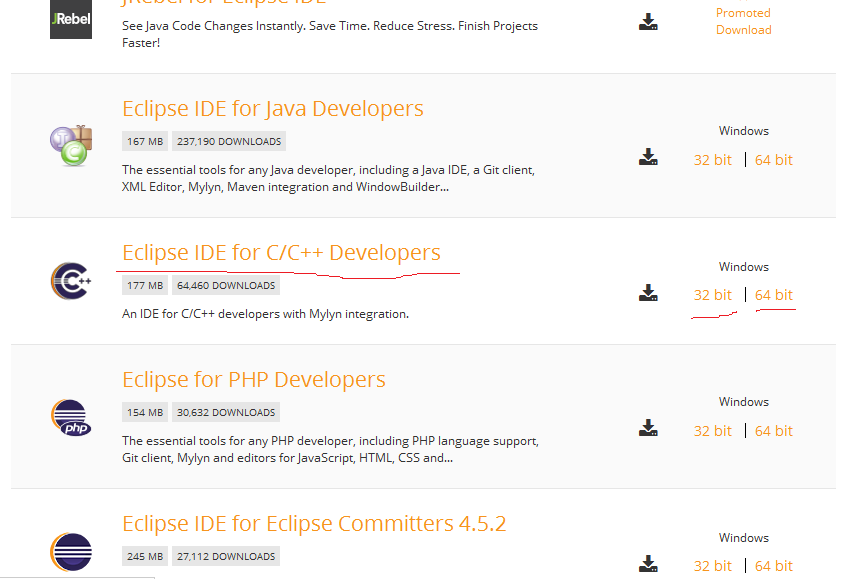
运行下载Oracle的Java安装包（安装时注意配置环境变量），即可安装好Java环境。

**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.

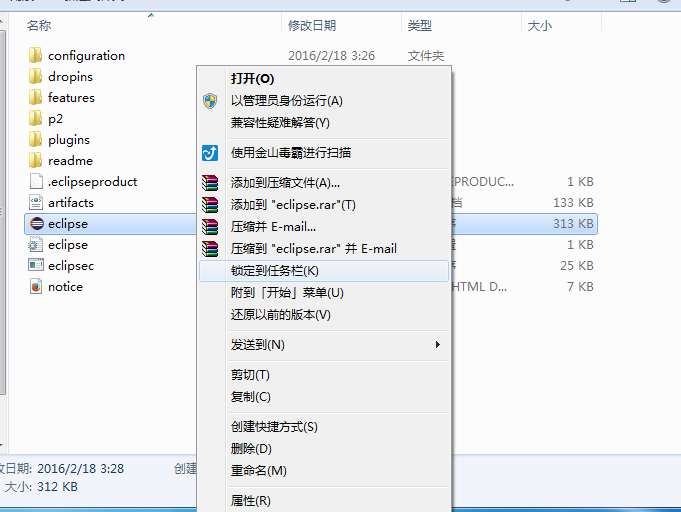
Eclipse CDT官方下载地址：

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）





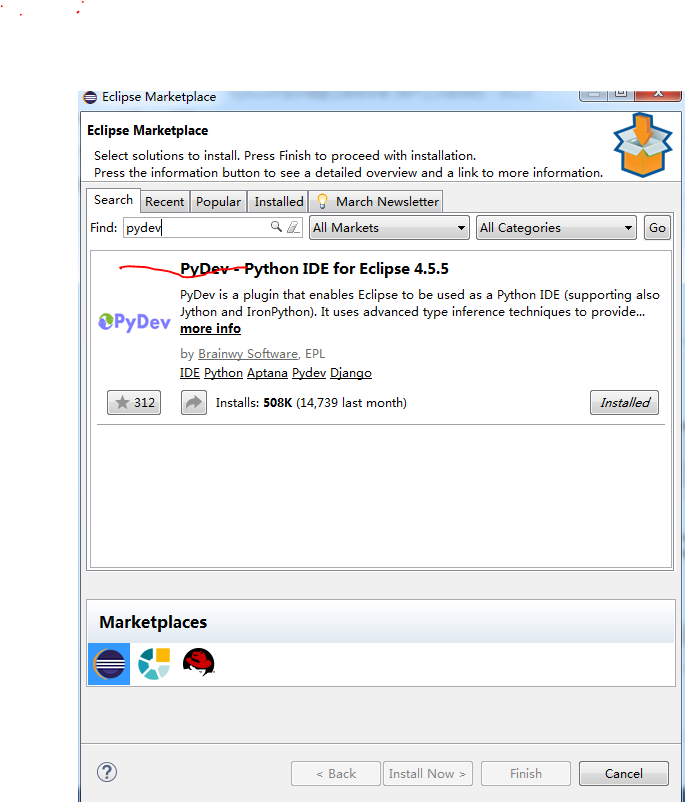
以后，从任务栏启动Eclipse方便很多（同样也可以将IDLE固定到任务栏）。

**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Python；

**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：

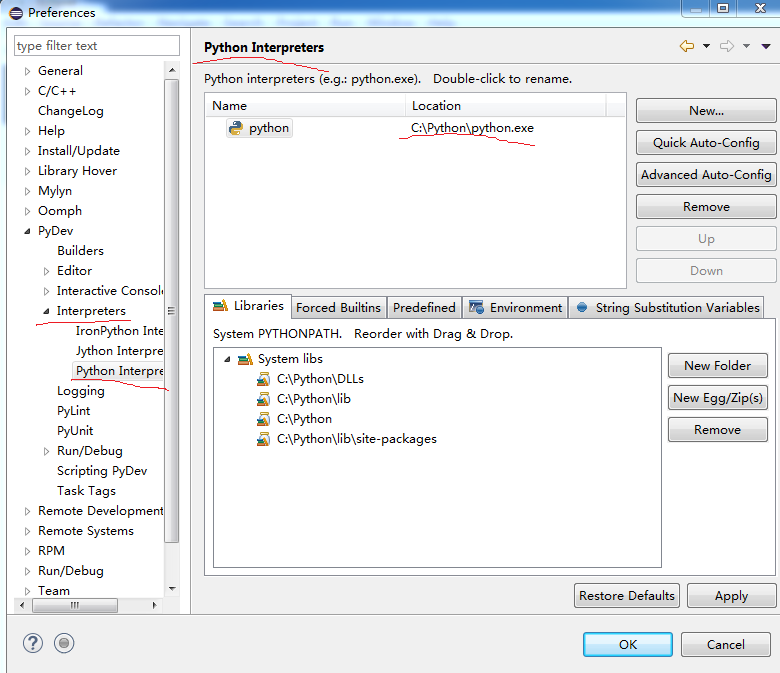
Help->Install->New Software->Add->Archive

离线安装. 

**3.2.2 配置 Python解释器**

安装好后，重新启动。通过：

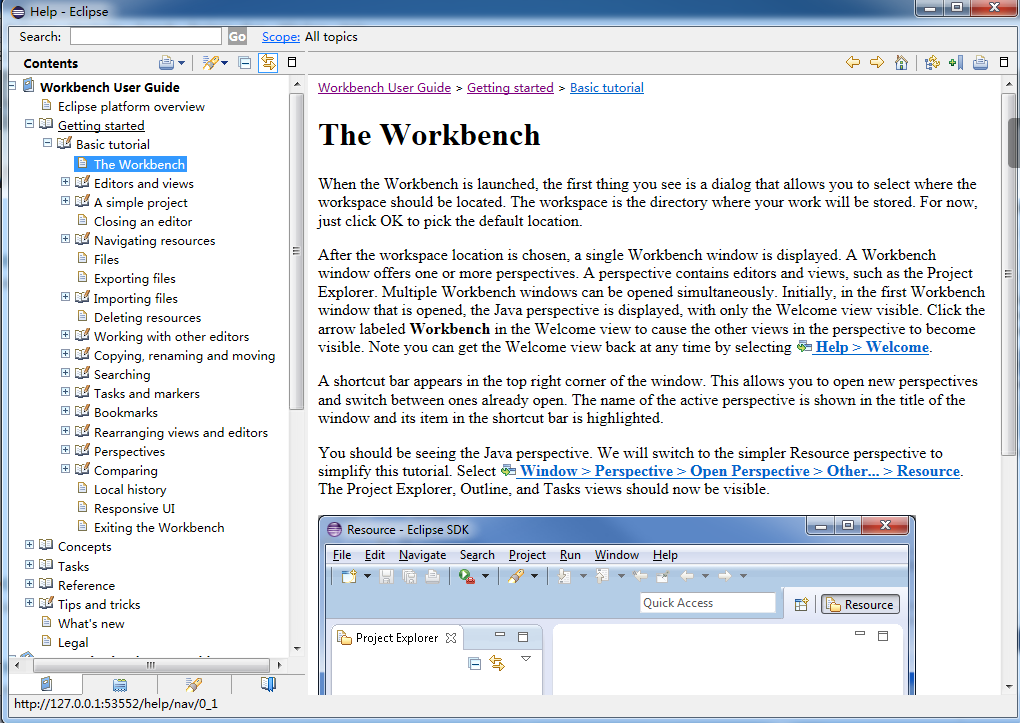
Windows->Preference->Pydev->Interperters->Python Interperter点其中的：Advanced Auto-config配置开发使用的 Python解释器版本:



配置好后切换到Python场景，就可以进行Python开发了。

**3.3 联机指南**

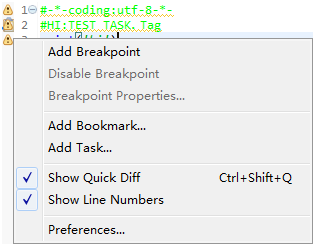
**Help** -> **Help Contents**



**四、配置 PyDev提高效率**

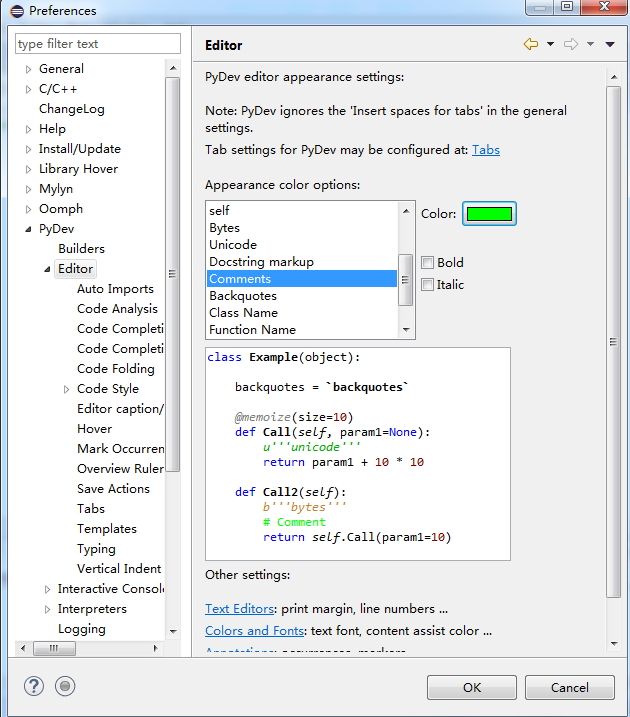
**4.1 显示源码行号**

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：

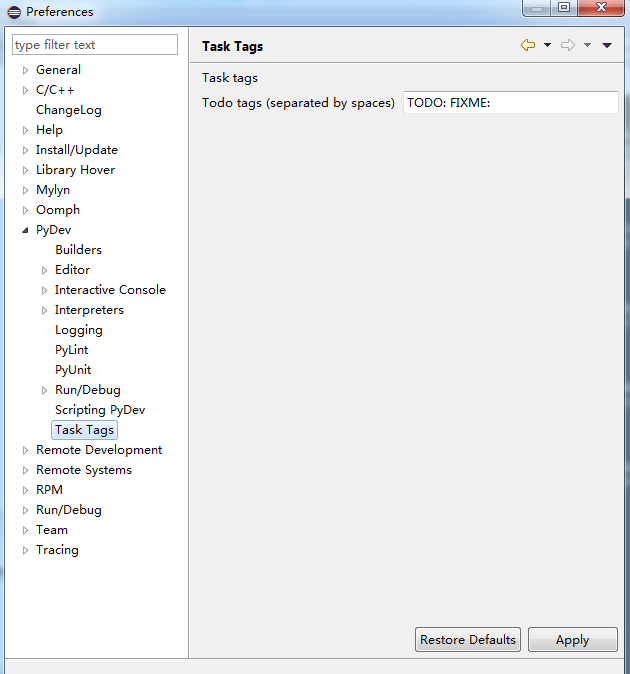


**4.3 配置任务标签**

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：**

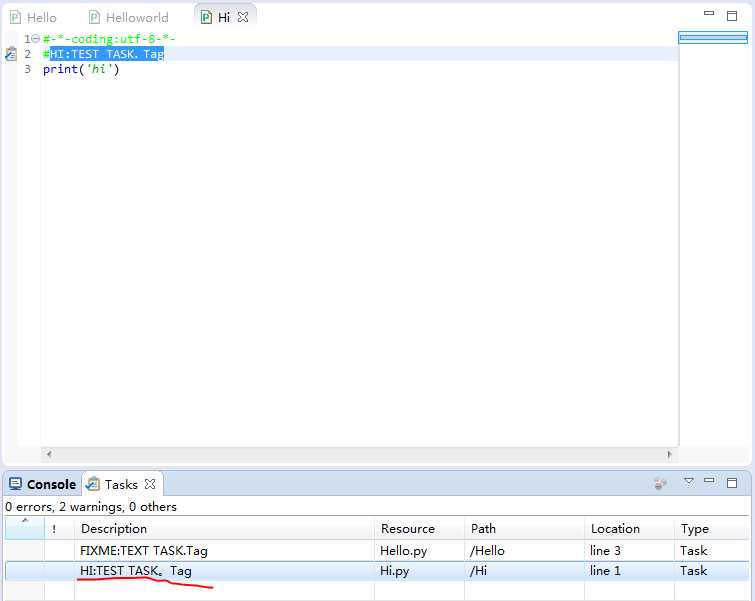
通过PyDev->Task Tags中配置任务标签:



可以根据不同的任务类型，自己增加自定义标签

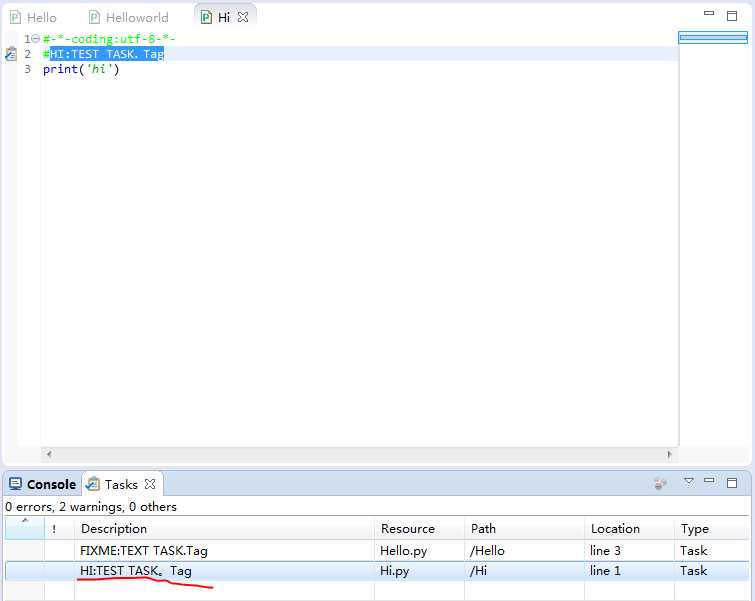
**2）添加任务标签注释**

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。

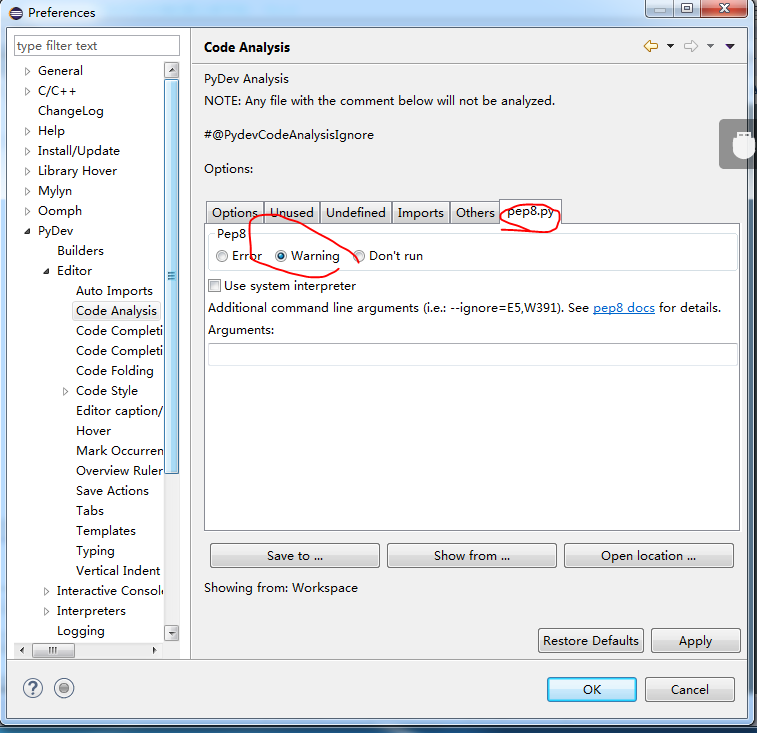
**5.1 PEP8检查和修改**

1）启动 pep8 检查：

Window > Preferences

PyDev > Editor > Code Analysis > pep8.py

选择Errors/Warnings其中之一

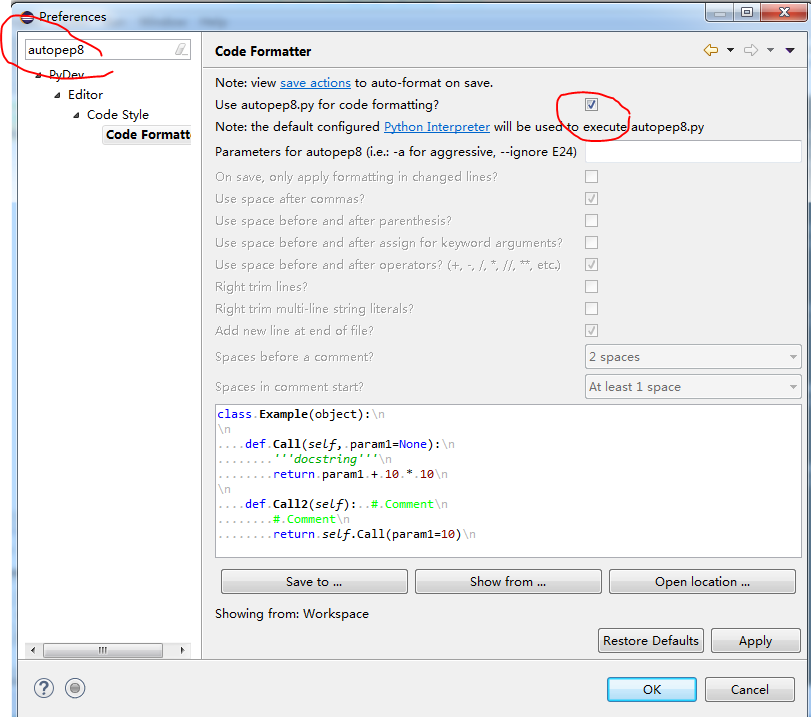
..****

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： .



2）启动autopep8自动修改：

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

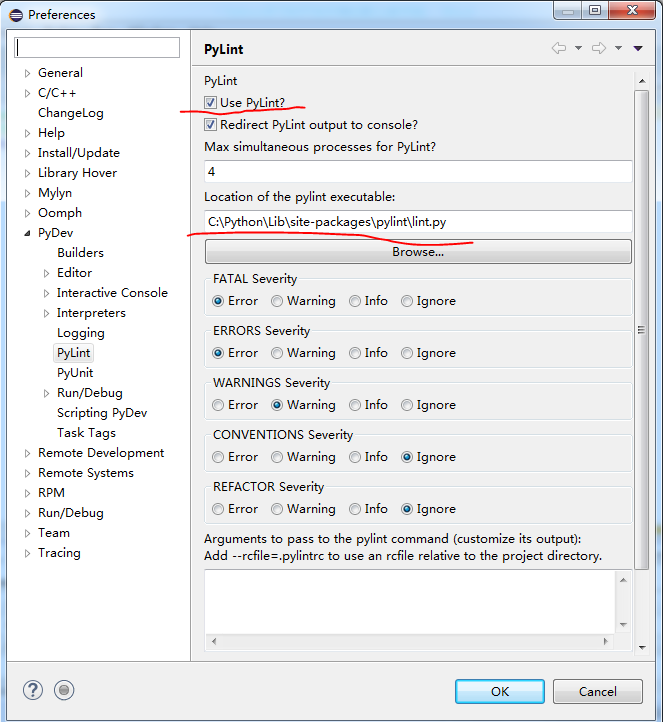


选择（Check）: Use autopep8.py for code formatting?

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?",

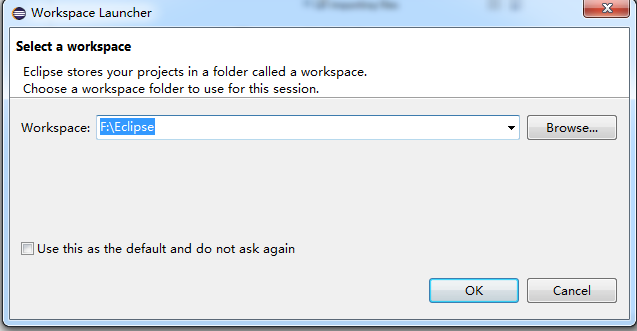
找到安装好的lint.py的地址,例如"C:\Python34\Lib\site-packages\pylint\lint.py"

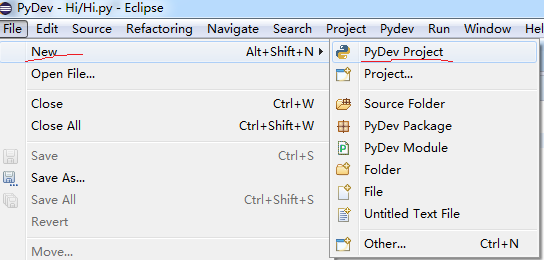
配置参数，限制Pylint的输出



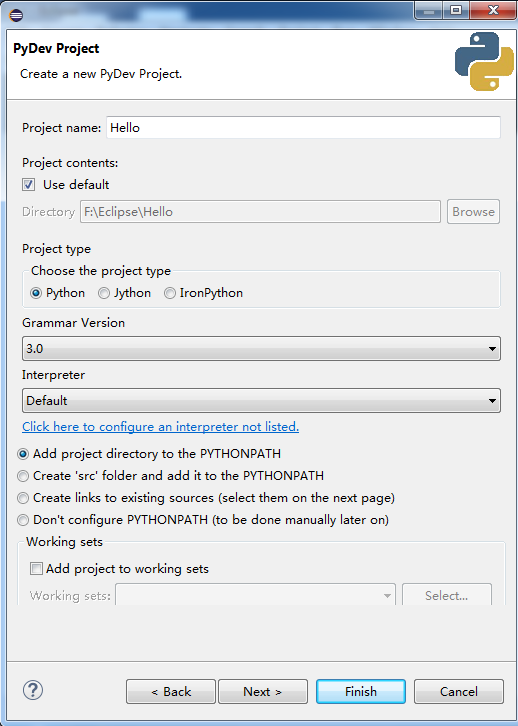
选中Project->Build Automatically,这样程序修改，保存时pylint就会自动检查项目中的代码,也可用Ctrl+B手动build触发pylint。

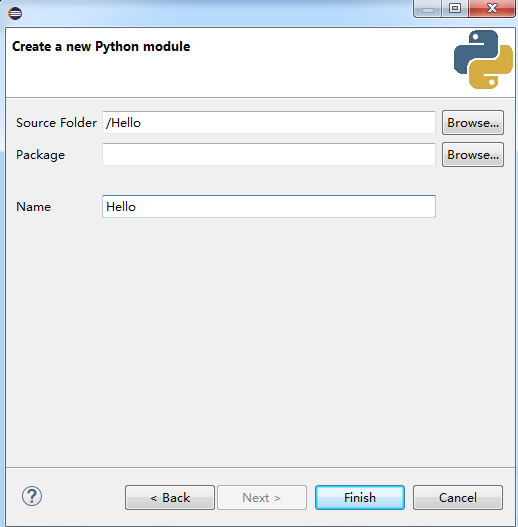
六、编写及运行程序

1）打开Eclipse，选择存储空间2）通过File->New->PyDev Project建立新的工程。

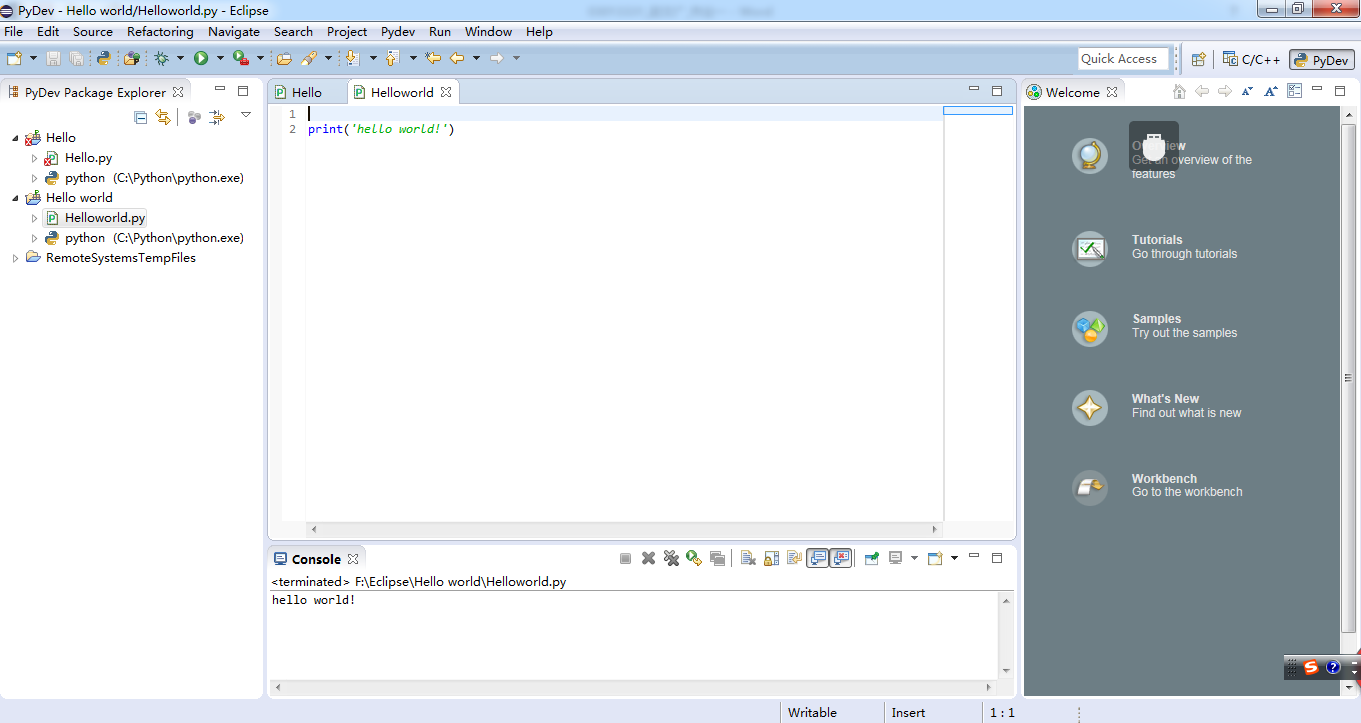


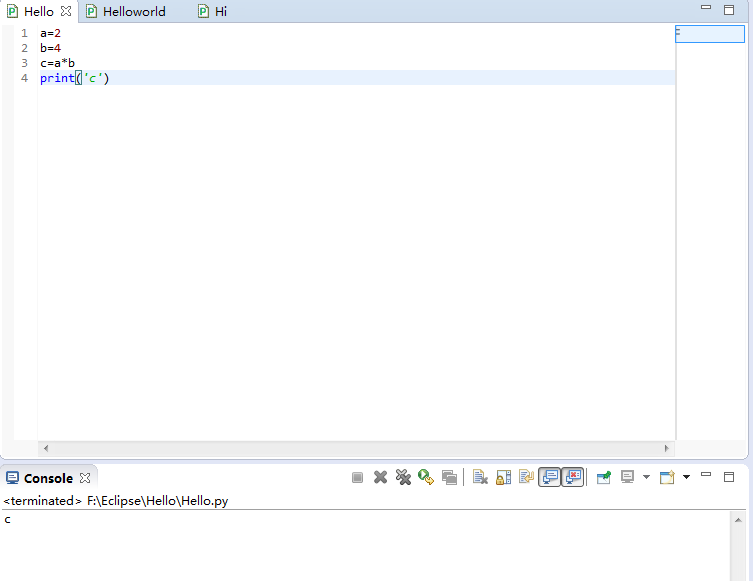
选择语法版本为3.0

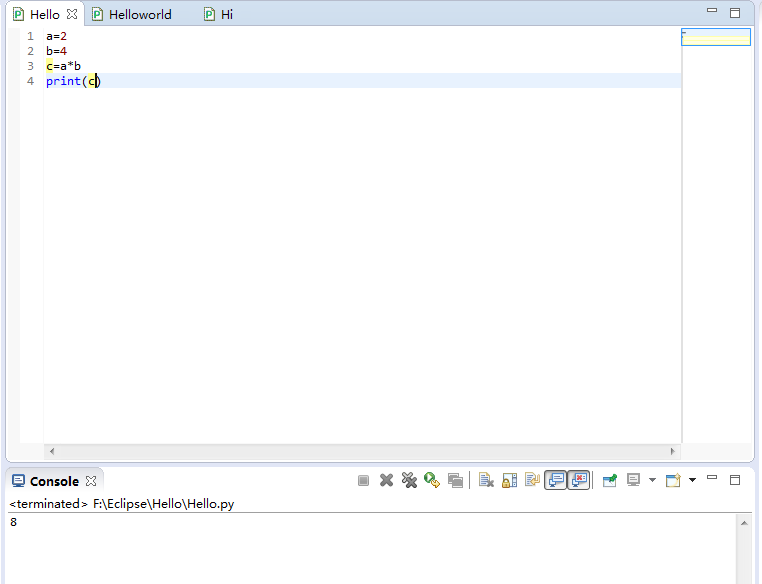


3）在左侧workplace中选择新建的工程，File->New->Pydev Module建立可编写窗口。

4）键入代码，运行程序





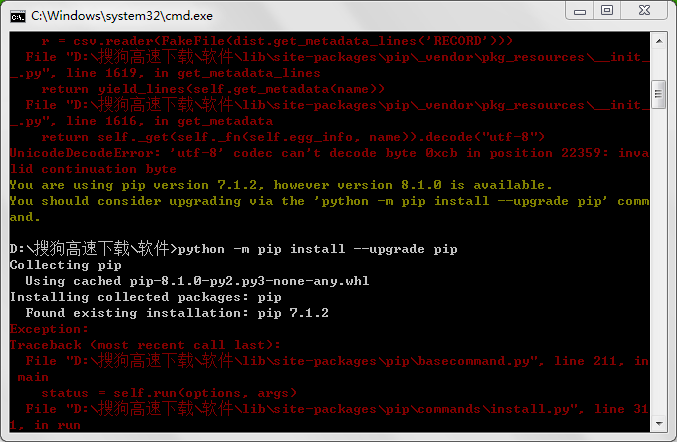


七、安装配置使用过程中出现的问题、解决过程方法

**7.1安装Python后无法更新pip**

问题：正常安装Python3.5.1版本时，安装后命令行里输入>pip install --upgrade pip 后无法更新将pip从7.1.2版本更新到8.1.0，无法进行后续操作。

解决：认为是本地编译条件不满足，重新敲入命令>python –m pip install –upgrade pip，结果如下，仍未解决问题。



从图中意识到可能为安装路径出了问题，认为应该卸载Python后重装在系统盘。

新问题：意识到问题后发现无法卸载Python，求助老师，认为是原先的windows系统缺少了补丁。

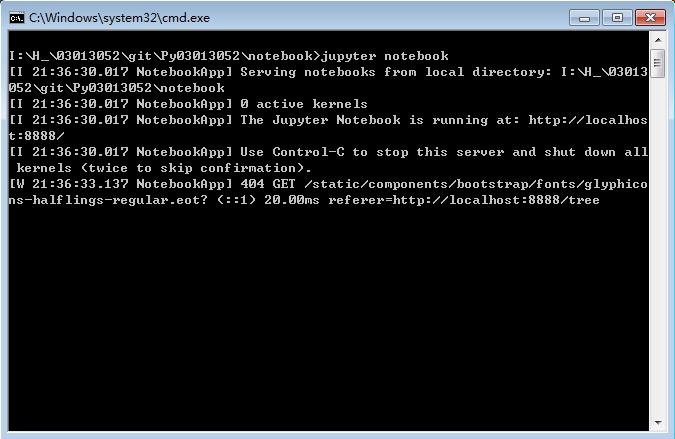
解决：安装了补丁，问题没有得到解决。

决定重装系统，所有步骤从头开始，问题得到解决。

**7.2无法用网页打开Notebook**

问题：在iPython notebook文件所在目录下，打开命令行窗口：

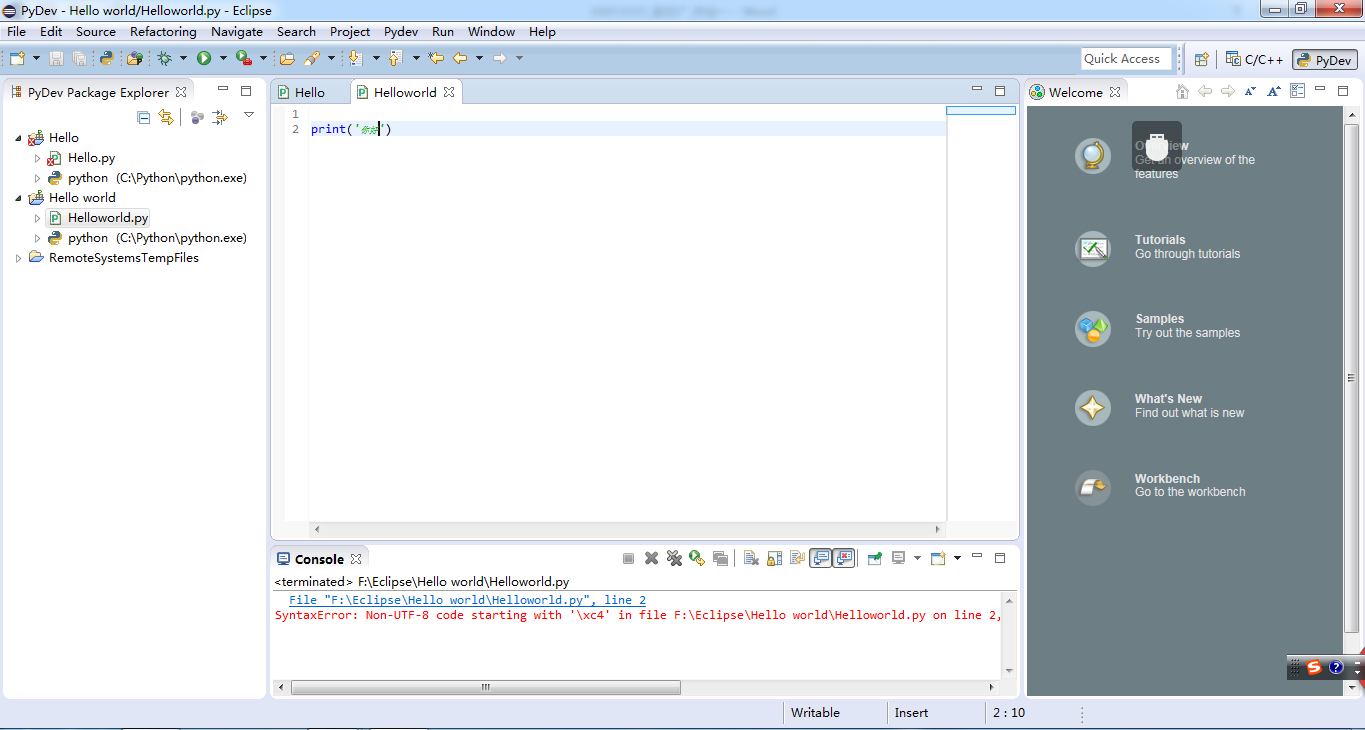
>jupyter notebook，打开网页显示为blank，无法正确打开notebook。



解决：使用搜索引擎搜索出现的类似问题以及解决方案，几乎没有帮助，尝试卸载使用的默认浏览器—猎豹浏览器，并使用火狐浏览器，成功打开notebook。

**7.4 Eclipse编程时无法输出汉字**

问题：源码中含有汉字，运行时报错，无法输出汉字



解决：加上 #coding=gbk 或者 #-\*-conding:utf-8-\*-。

