

**Python学习和开发环境的建立**

课程实践作业一



03013328章步鑫

目录

[**Python学习和开发环境的建立** 2](#_Toc453339445)

**一、Python基本环境开发建立**………………………………………………………………………………………………2

**[二、Python扩展包安装](#_Toc453339446)** [3](#_Toc453339446)

[**2.1 编码规范** 4](#_Toc453339447)

[**2.2 科学计算软件包** 4](#_Toc453339448)

[**三、基于Eclipse的Python开发环境** 8](#_Toc453339449)

[**3.1 安装Eclipse IDE** 8](#_Toc453339450)

[**3.2 安装Python开发插件PyDev** 10](#_Toc453339451)

[**3.3 联机指南** 11](#_Toc453339452)

[**四、配置 PyDev提高效率** 12](#_Toc453339453)

[**4.1 显示源码行号** 12](#_Toc453339454)

[**4.2 修改注释颜色提高可读行** 12](#_Toc453339455)

[**4.3 配置任务标签** 12](#_Toc453339456)

[**五、基于PyDev的语言规范静态检查** 14](#_Toc453339457)

[**5.1 PEP8检查和修改** 14](#_Toc453339458)

[**5.2 使用Pylint** 16](#_Toc453339459)

**六、编写及运行程序**…………………………………………………………………………………………………………...17

**七、安装配置使用过程中出现的问题、解决过程方法**……………………………………………………..20

**八、小结**....................................................................................................................................20

**九、参考文献**……………………………………………………………………………………………………………………….20

**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，由Guido van Rossum于1989年发明，第一个公开发行版发行于1991年。

Python是纯粹的自由软件， 源代码和解释器CPython遵循 GPL(GNU General Public License)协议 。

Python语法简洁清晰，特色之一是强制用空白符(white space)作为语句缩进。

Python具有丰富和强大的库。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。常见的一种应用情形是，使用Python快速生成程序的原型（有时甚至是程序的最终界面），然后对其中有特别要求的部分，用更合适的语言改写，比如3D游戏中的图形渲染模块，性能要求特别高，就可以用C/C++重写，而后封装为Python可以调用的扩展类库。需要注意的是在您使用扩展类库时可能需要考虑平台问题，某些可能不提供跨平台的实现。

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。

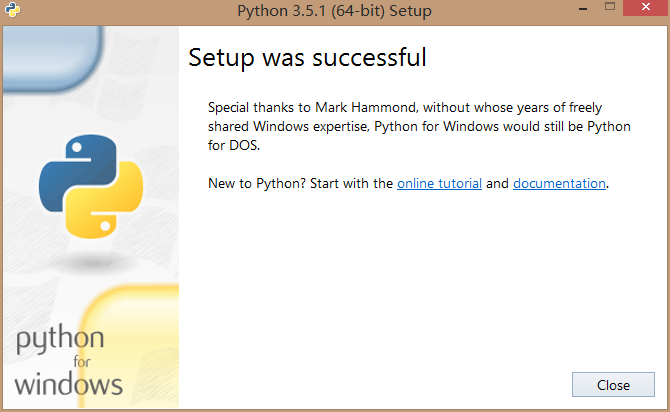
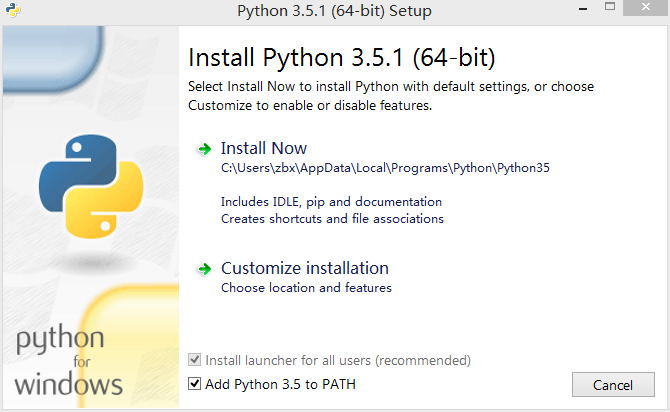
1. **Python基本开发环境建立**

从官网：

http://www.python.org

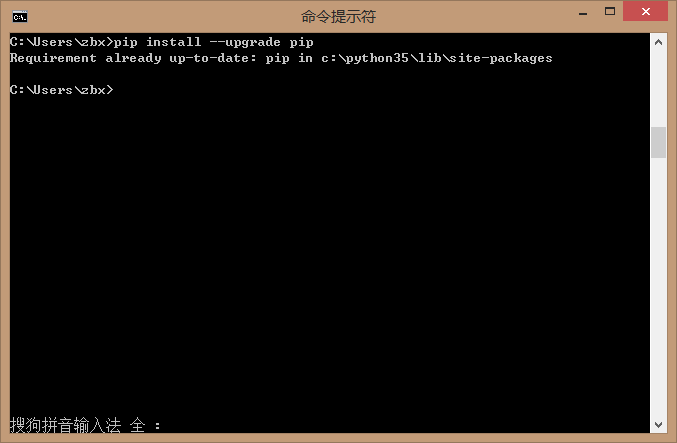
下载和操作系统（32/64位）对应版本的Python3软件。下面以Python3.4.3版本安装为例做简单说明

Windows： 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： C:\python35-32



然后，命令行下更新pip到新版本：

>pip install --upgrade pip



1. **Python扩展包安装**

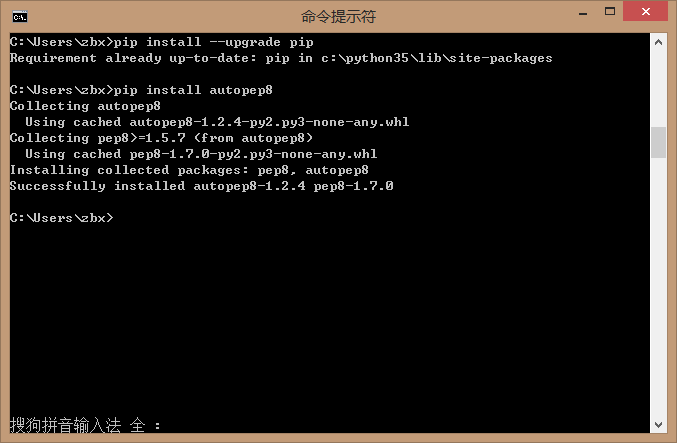
pip命令是软件包的基本安装方式，不过，有些软件包使用pip安装时，需要本地编译，编译条件如果不满足安装就会失败。

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/>

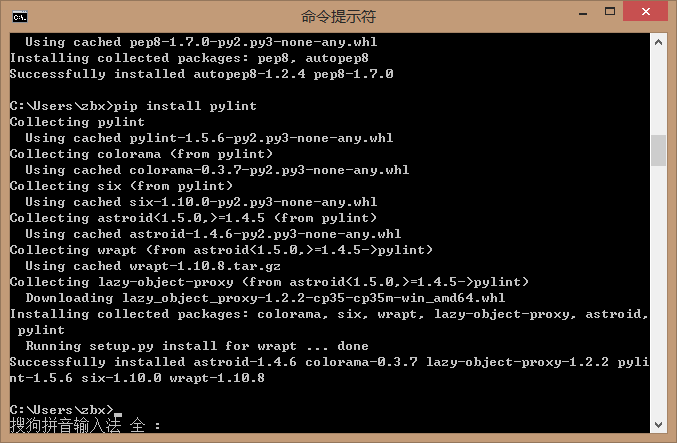
加州大学欧文分校的编译版包，内容丰富、更新及时、安装方便。

**2.1 编码规范**

pip install autopep8



同理，输入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：

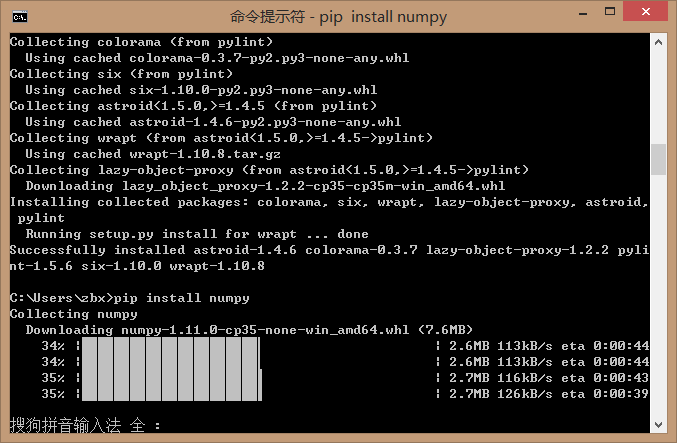
numpy

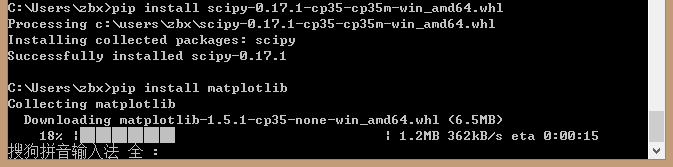
scipy

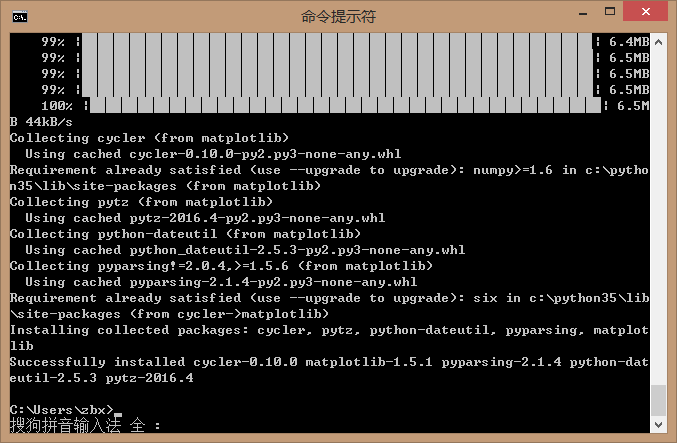
matplotlib

**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



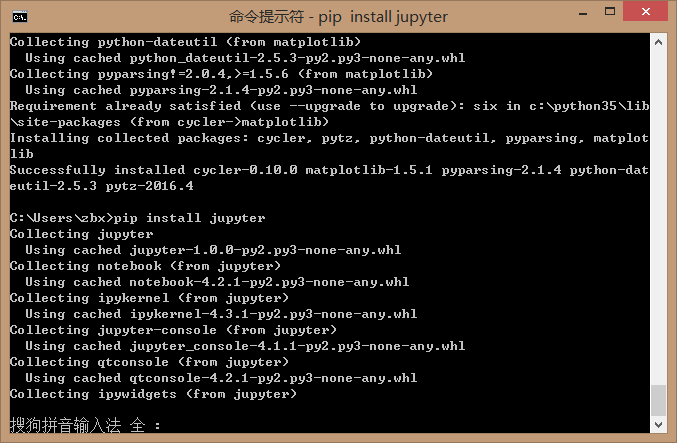


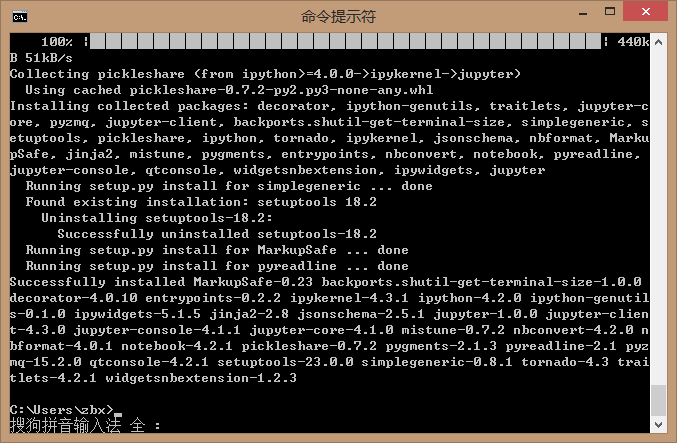


**2.2.2 交互计算 Jupiter**

1） 安装Jupiter ：

>pip install jupyter 在线安装



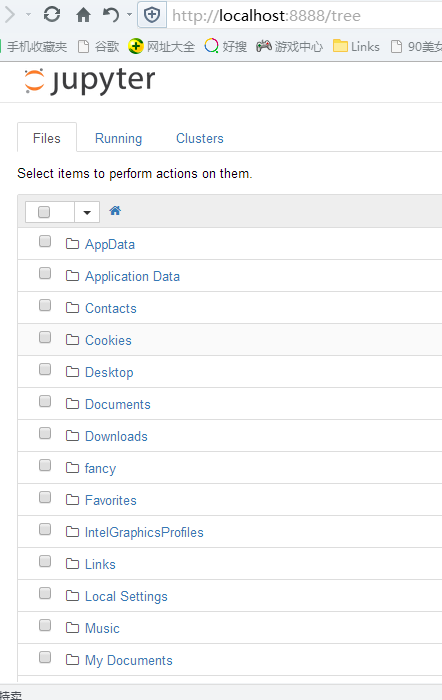


2） 运行notebook:

在iPython notebook文件所在目录下，打开命令行窗口：

>jupyter notebook





**2.2.3 IF97 for Python**

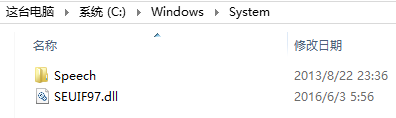
Windows 32/64位版: 从

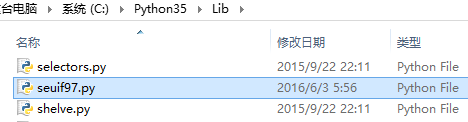
<https://github.com/Py03013052/SEUIF97>

下载：SEUIF97.dll和 seuif97.py，然后：

1) SEUIF97.dll拷贝到 c:\windows\system

2) seuif97.py拷贝到 c:\python35\Lib





1. **基于Eclipse的Python开发环境**

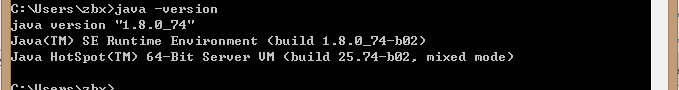
**3.1 安装Eclipse IDE**

**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：

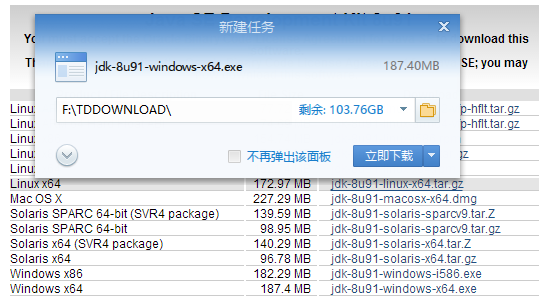
[**http://www.oracle.com/technetwork/java/javase/downloads/index.html**](http://www.oracle.com/technetwork/java/javase/downloads/index.html)

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。



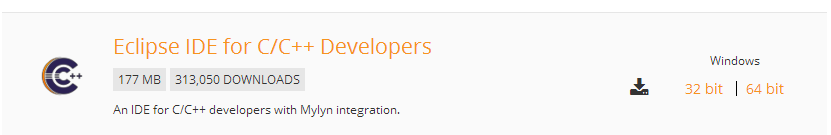
注：因从网站上下载的安装包无法正常运行（错误代码1335），实际安装软件是从老师所给文件中拷贝，所以下载和运行时的文件名不同。

**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.

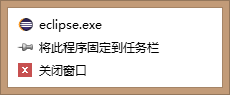
Eclipse CDT官方下载地址：

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。

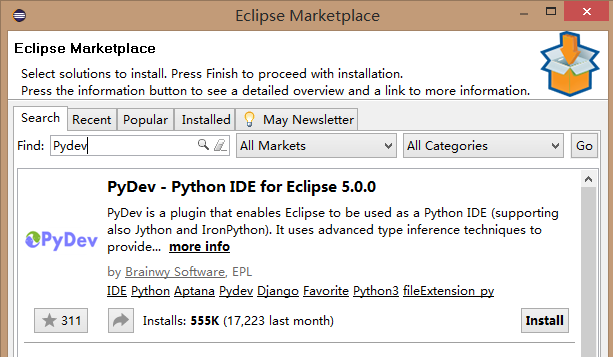


**3.2 安装Python开发插件PyDev**

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：

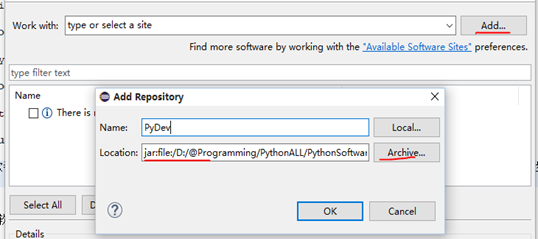


在线安装过程简单，但是，安装速度受网络环境影响。如果很慢，可从pydev官网（<http://www.pydev.org/>）提供的下载地址：

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

下载插件包，然后，通过：

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

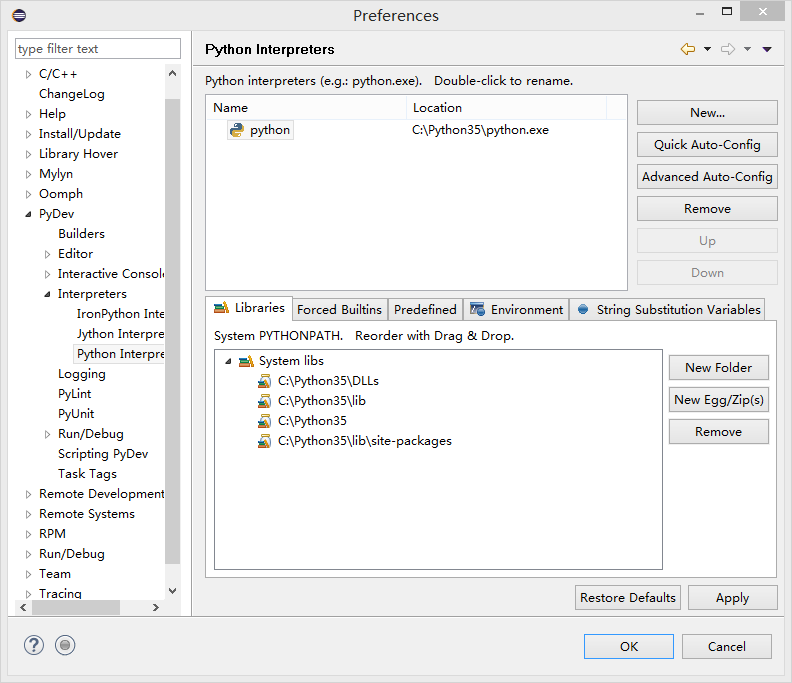
离线安装.

注：因为我是使用在线安装，所以离线安装方法没有去尝试，截屏图片来自老师所给word文档。

**3.2.2 配置 Python解释器**

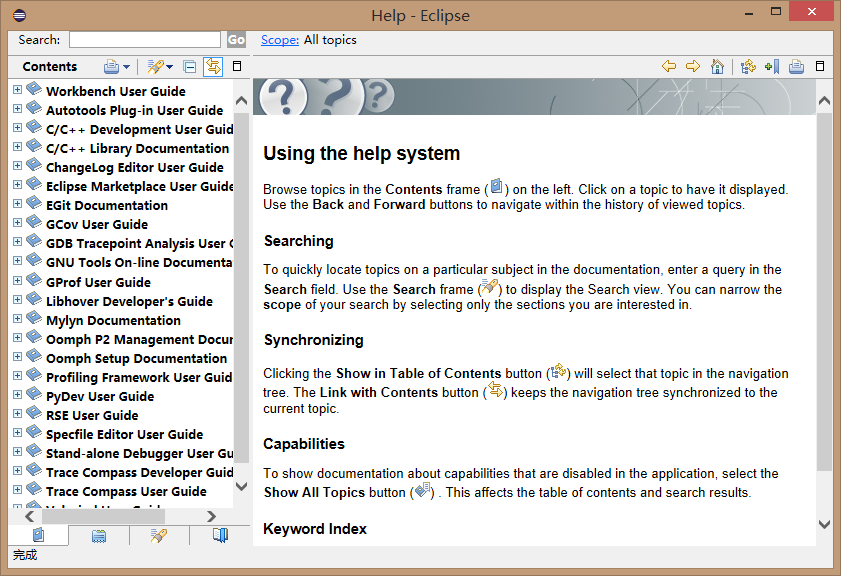
安装好后，重新启动。通过：

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:



**3.3 联机指南**

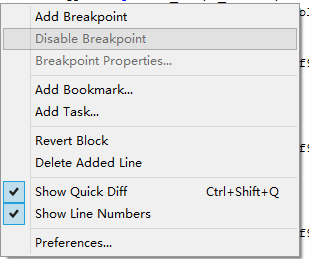
**Help** -> **Help Contents**



1. **配置 PyDev提高效率**

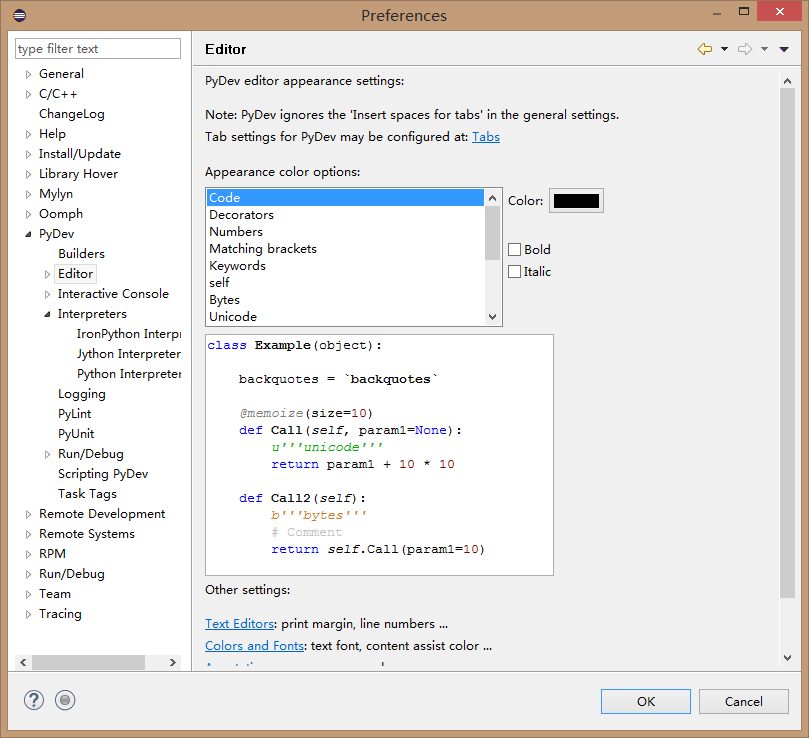
**4.1 显示源码行号**

右键源码的左边缘，选中“Show Line Number”



**4.2 修改注释颜色提高可读行**

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

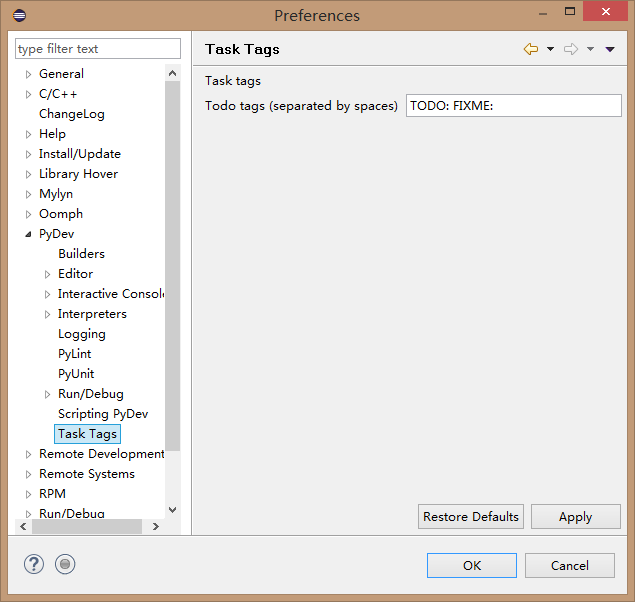


**4.3 配置任务标签**

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：**

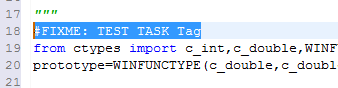
通过PyDev->Task Tags中配置任务标签:



可以根据不同的任务类型，自己增加自定义标签。

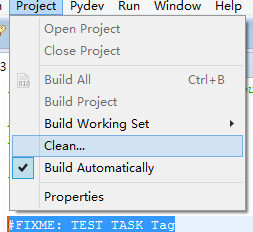
**2）添加任务标签注释**

在程序源码中加入使用任务标签的注释：

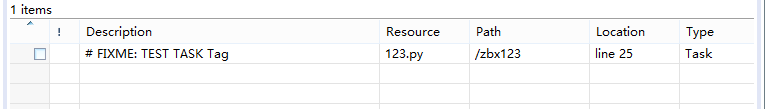


**3）识别任务**

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。

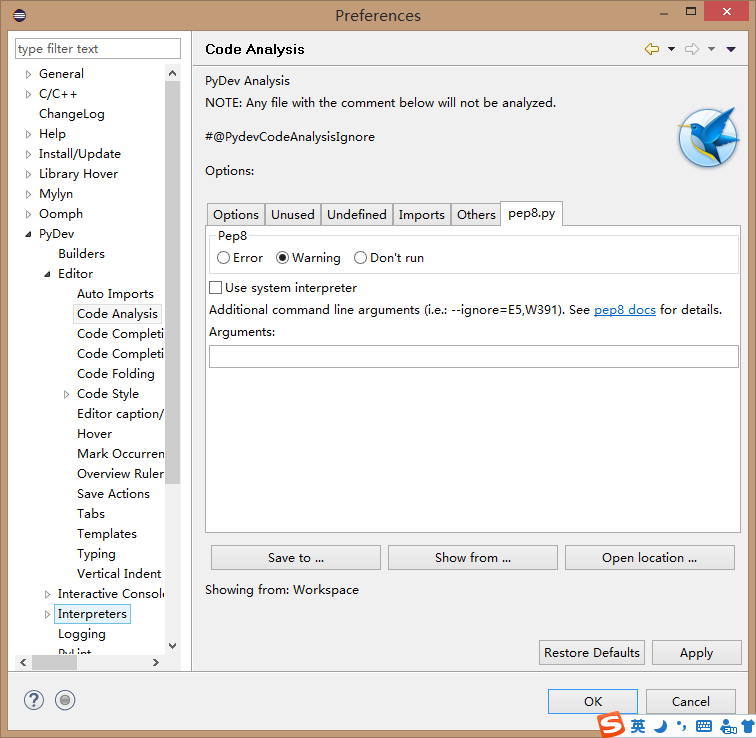
**5.1 PEP8检查和修改**

1）启动 pep8 检查：

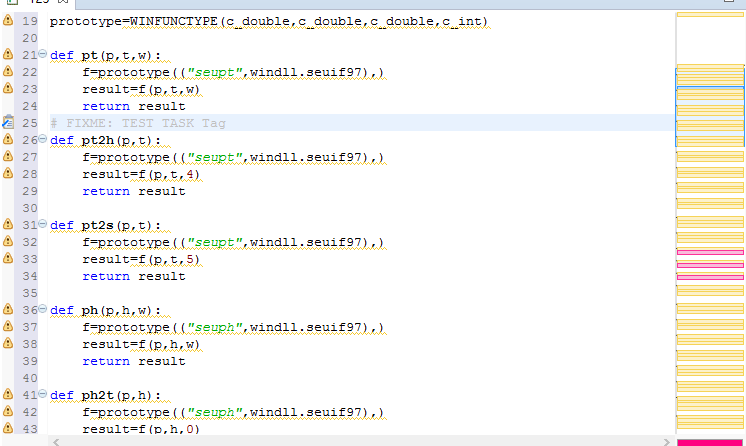
Window > Preferences

PyDev > Editor > Code Analysis > pep8.py

选择Errors/Warnings其中之一..

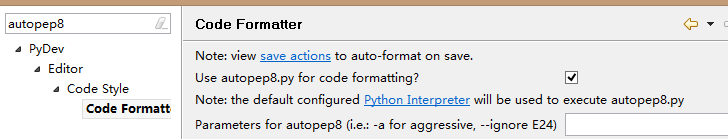


右键Python工程，选择 PyDev，点 "code analysis"，即可对工程中所有Python源码进行PEP8检查：



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?

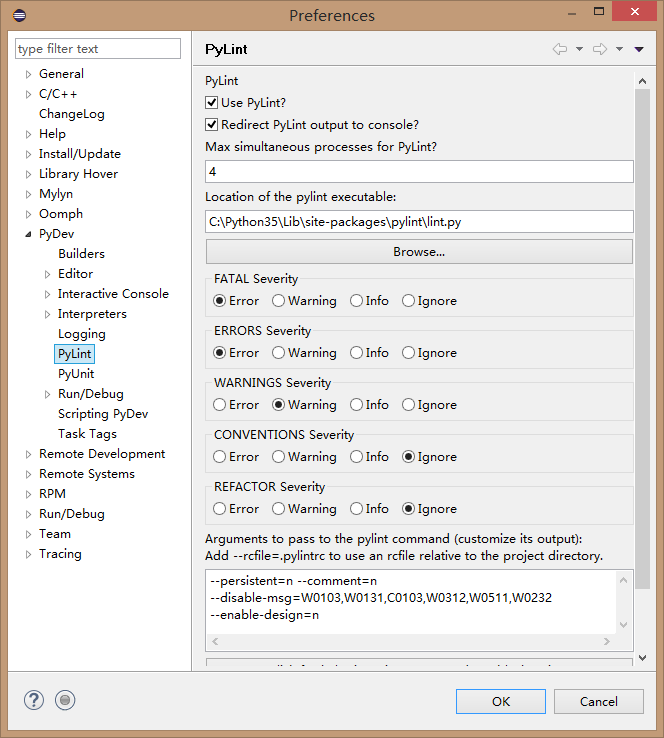


在Python源码窗口，按 CTRL-SHIFT-F 就可以自动修改代码

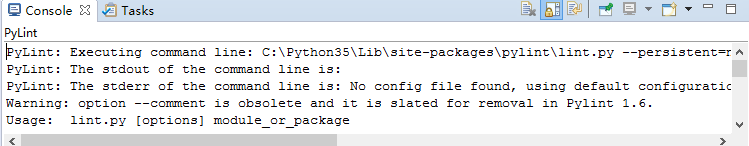
**5.2 使用Pylint**

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:\Python35\Lib\site-packages\pylint\lint.py"配置参数，限制Pylint的输出

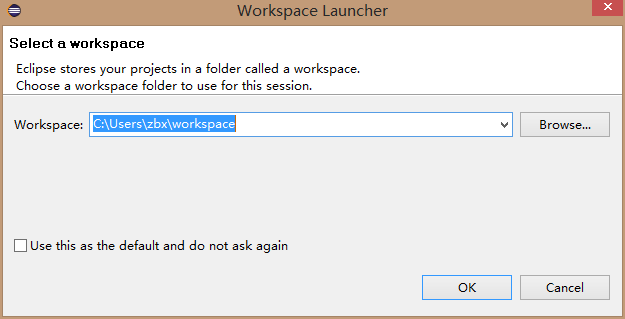


选中Project->Build Automatically,这样程序修改，保存时pylint就会自动检查项目中的代码,也可用Ctrl+B手动build触发pylint。

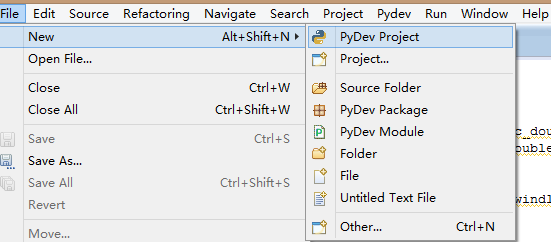


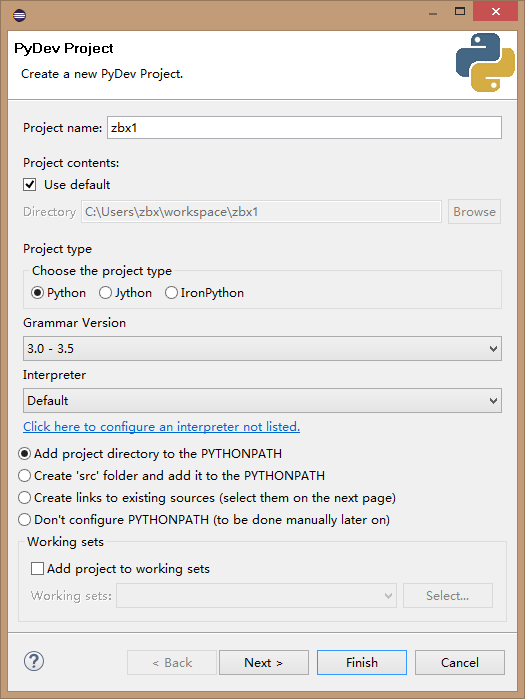
**六、编写及运行程序**

1）打开Eclipse，选择存储空间

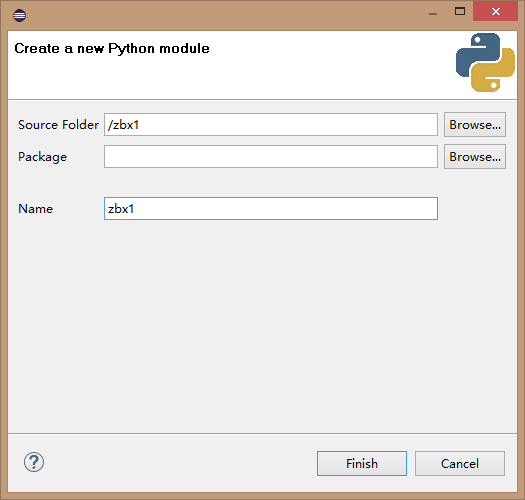


2）通过File->New->PyDev Project建立新的工程。

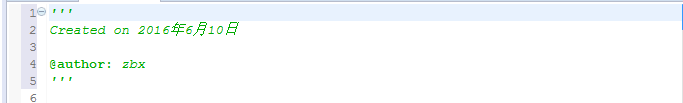


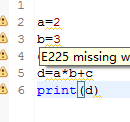


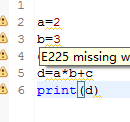
3）在左侧workplace中选择新建的工程，File->New->Pydev Module建立可编写窗口。



4）输入代码并运行程序







源代码：

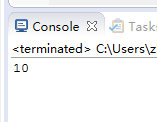
a=2

b=3

c=4

d=a\*b+c

print(d)



**七、安装配置使用过程中出现的问题、解决过程方法**

1.误将Python64位下成32位，导致无法正常运行，之后重新下载。

2.pip install scipy 无法正常下载，之后从加州大学网站下载scipy-0.17.1-cp35-cp35m-win\_amd64.whl文件，并在同学的帮助下正常运行。

3.从官网下载的Java安装包无法正常安装（错误代码1335），在网上搜寻解决方案并尝试后无果，之后拷贝原课堂文件中的Java安装包进行安装，在几次调试之后终于可以正常运行。

**八、小结**

Python 学习和开发环境多种多样，因自己在计算机方面知识薄弱，单是这个比较简单的作业也花费了相当长的时间。但与此同时我也收到了思考之后恍然大悟的乐趣，可谓苦乐参半。

**九、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

注：除参考文献外，还利用了网络上的百科知识，并参考了部分优秀作业，同时还借助了同学的帮助。