**环境建立指导手册**

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



（安装Python时忘记截图了，所以就用教学版上的原图了）

然后，命令行下更新pip到新版本：>pip install --upgrade pip



1. **Python扩展包安装**

**2.1 SciPy**

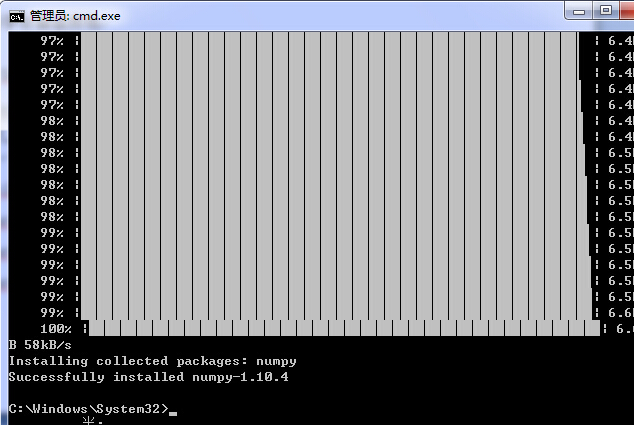
SciPy是一款方便、易于使用、专为科学和工程设计的Python工具包.它包括统计,优化,整合,线性代数模块,傅里叶变换,信号和图像处理,常微分方程求解器等等。

安装Scipy中的3个扩展包：

pip install numpy

pip install scipy

pip install matplotlib

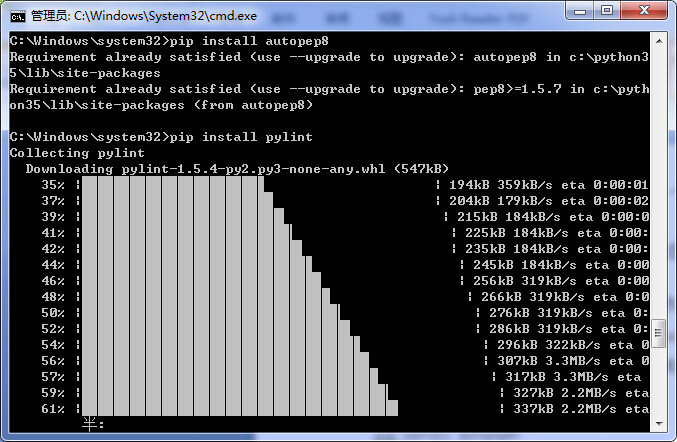


Scipy和matplotlib的安装与上图类似，故不再截图。

**2.2 安装编码规范扩展包**

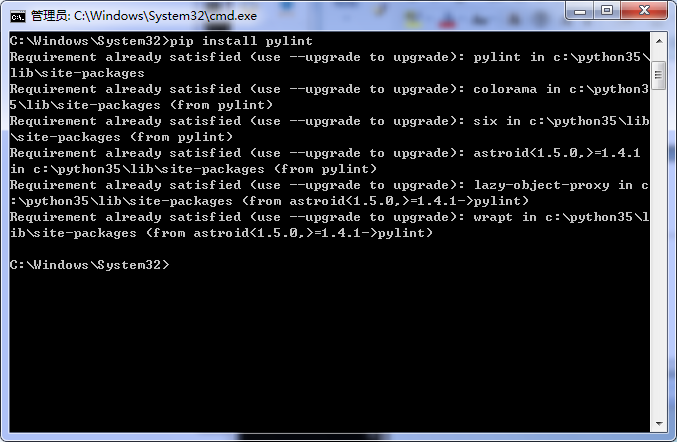
**2.2.1 安装autopep8**

安装命令：pip install autopep8

****

**2.2.2 安装pylint**

安装命令：pip install pylint

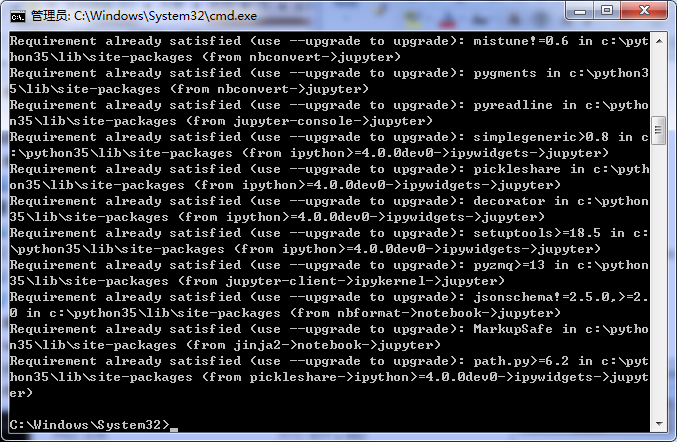
****

**2.3 科学计算软件包**

**2.3.1 交互计算Jupiter**

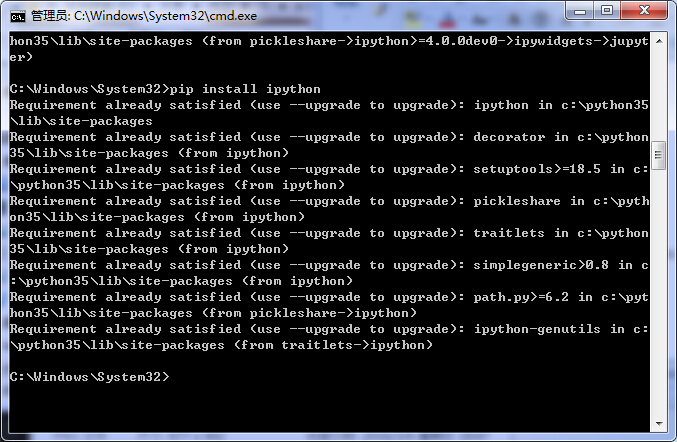
1) 安装Jupiter

安装命令：pip install Jupiter



2) 安装Python语言内核

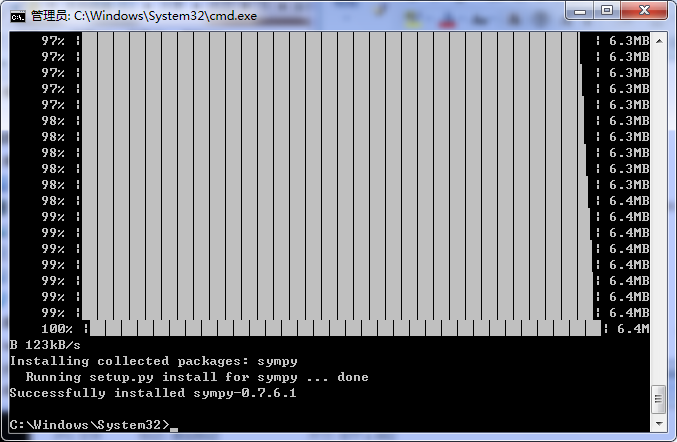
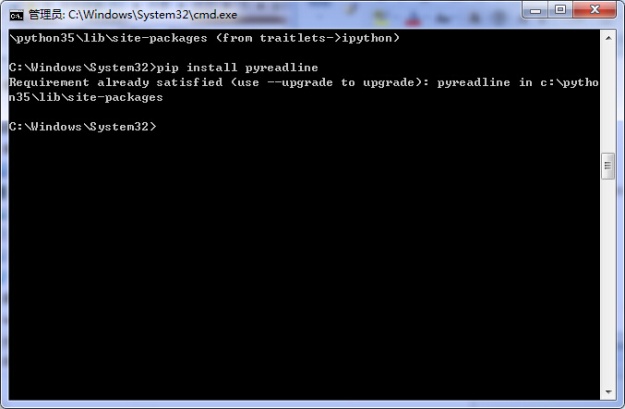
安装命令：pip install ipython



3) 安装依赖包

安装命令：pip install pyreadline

Pip install sympy



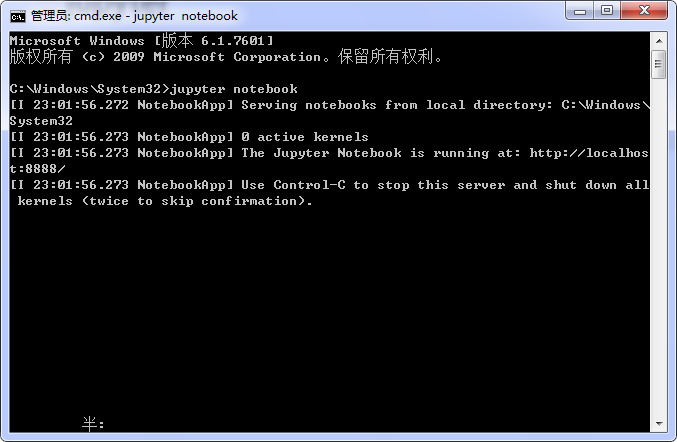
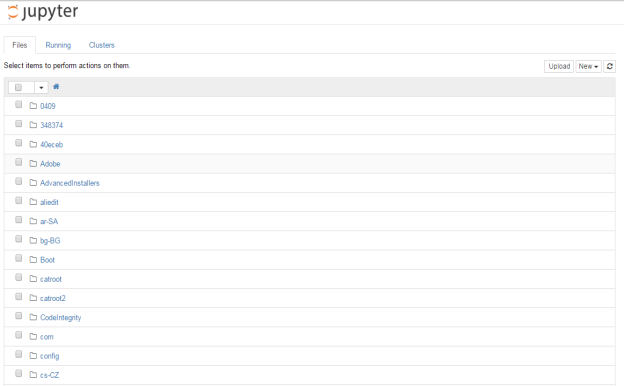
4) 支持显示数学符号、公式，安装MathJax

安装命令：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()

5) 运行notebook

命令：jupyter notebook

**2.3.2 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**

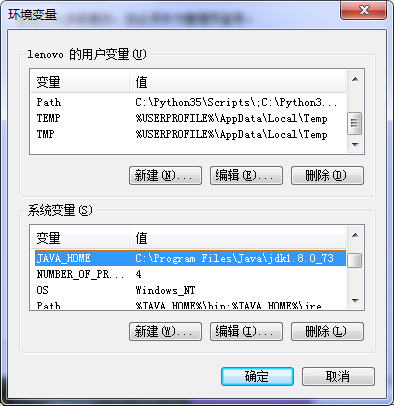
**3.1.1 安装JavaSDK**

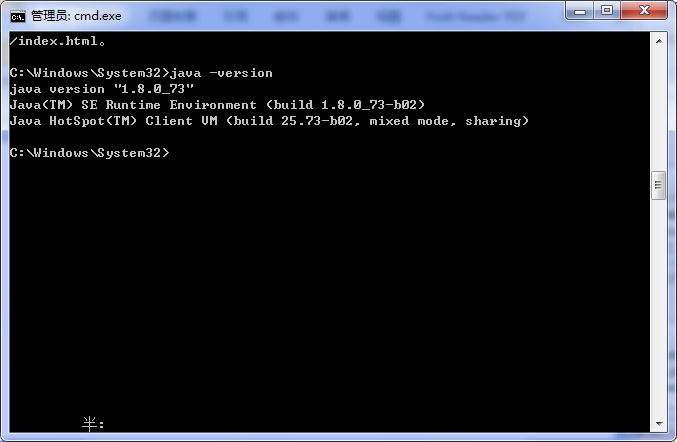
Eclipse IDE是使用Java开发的，电脑中需要预先安装好Java JRE/JDK软件包，因此，安装前，在命名行下，输入：>java –version

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**。**我的电脑没有安装Java软件包，所以从Oracle下载Java包，并进行重新安装。如下图（显示未安装）：



所以进行了重新安装,下图分别为配置环境变量和安装好Java后的显示图：





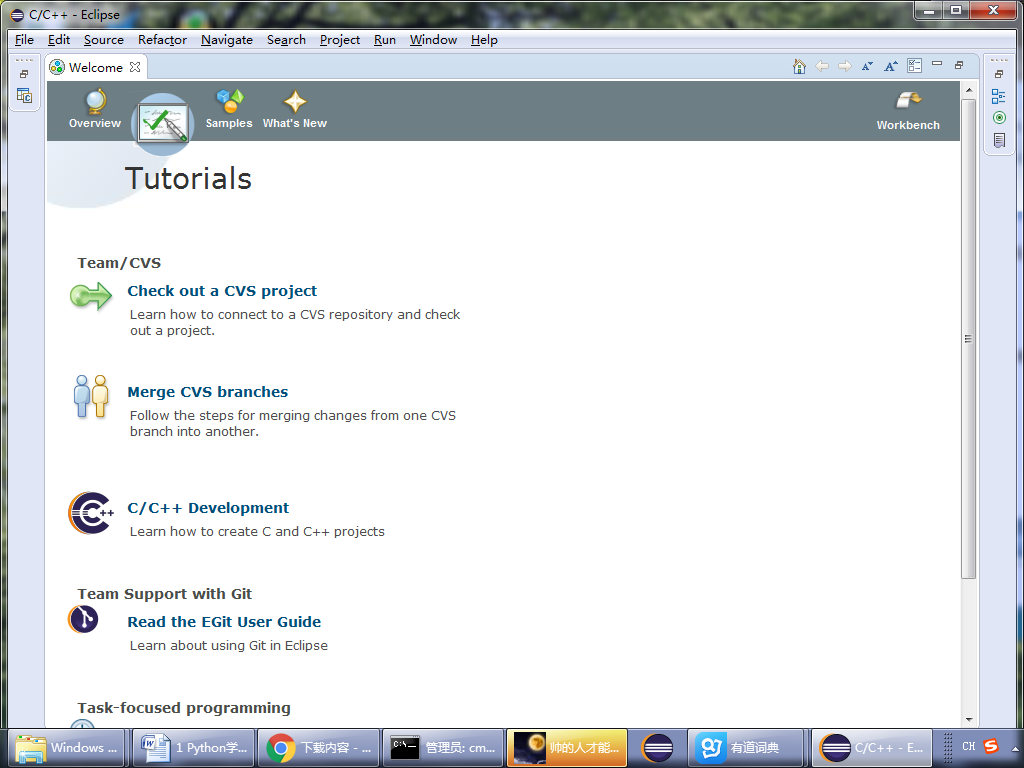
**3.1.2 安装Eclipse IDE**

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）



(安装Eclipse时忘记截图，所以就用教学版上的图片)

**3.2 安装Python开发软件插件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Python3.5；

**3.2.1 安装PyDev插件**

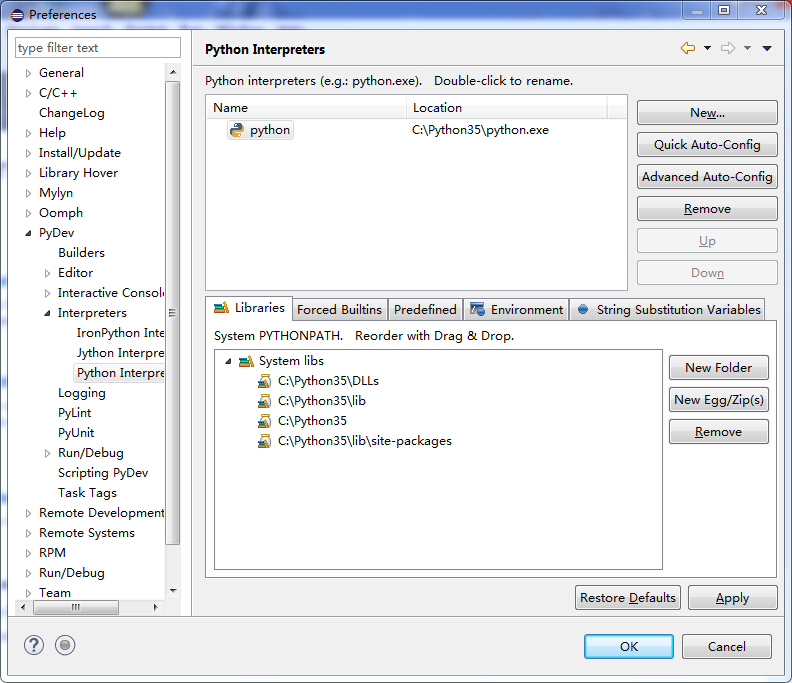
通过Windows->Eclipse Marketplaces进入市场，输入Pydev，找到Pydev安装/更新项目，在线安装即可：



安装好后，重新启动Eclipse。

**3.2.2 配置Python解释器**

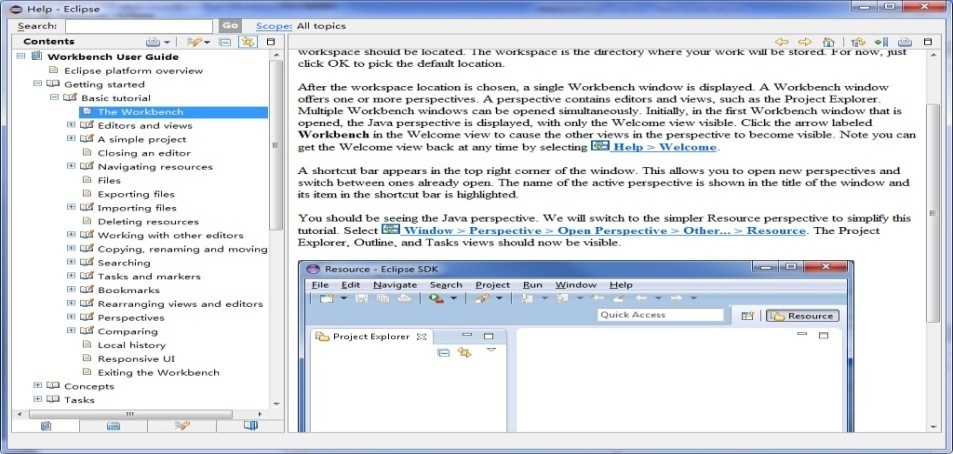
通过：Windows->Preference->Pydev->Interperters->Python Interperter点其中的：Advanced Auto-config配置开发使用的 Python解释器版本:



配置好后切换到Python场景，就可以进行Python开发了。

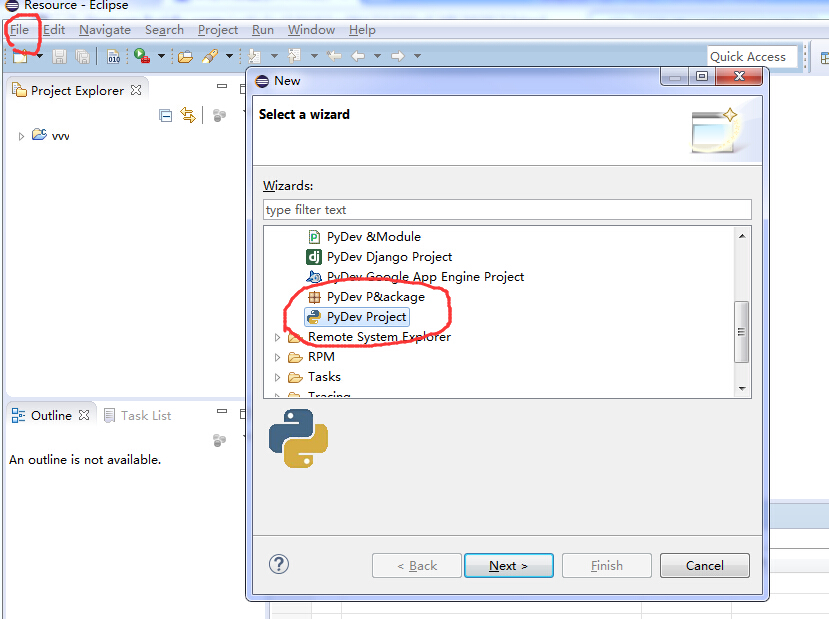
**3.3 联机指南**

通过：Help->Help Contents

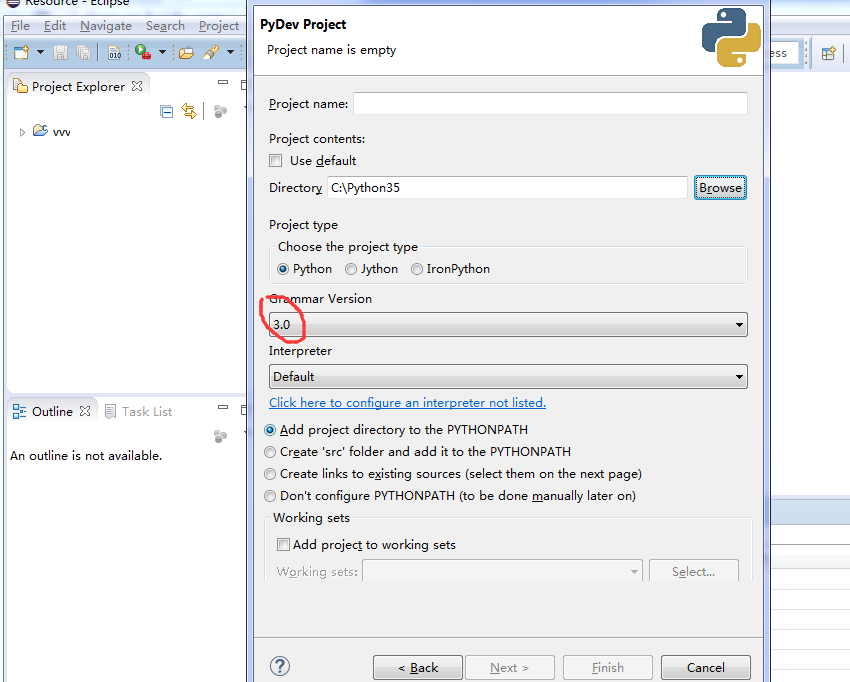


**3.4 创建一个Python工程**

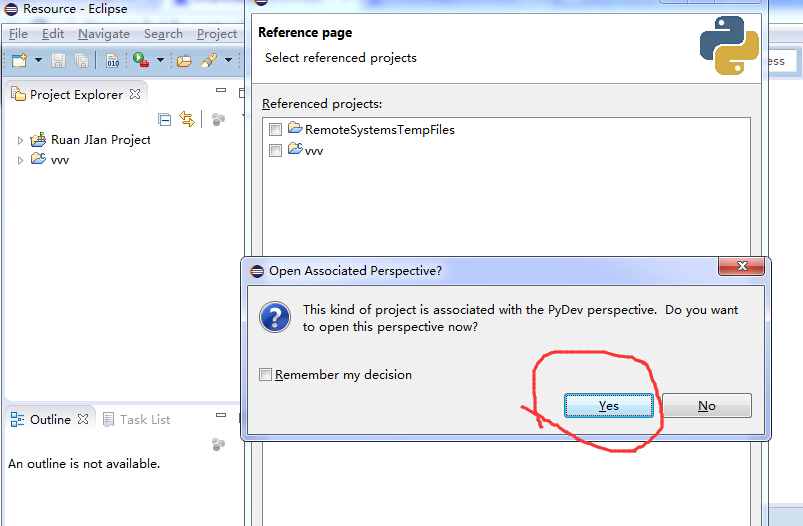
通过：打开Eclipse，选择菜单中file->new->other->PyDev->PyDev Project



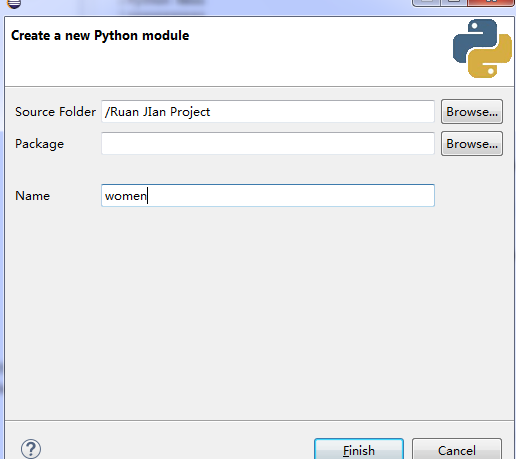
修改必要参数以及储存位置：



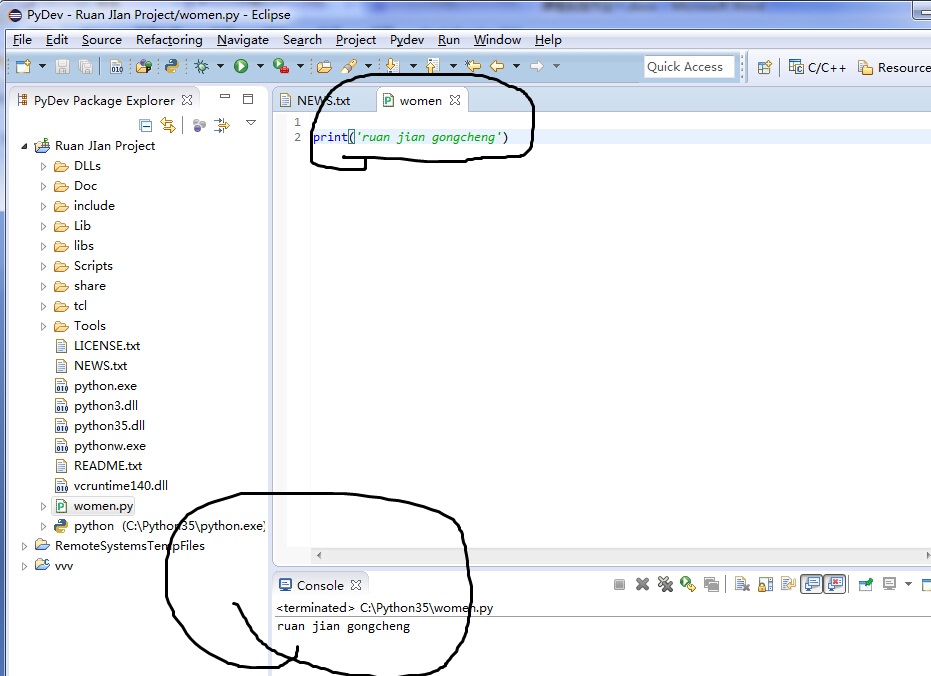
点击finish就完成创建了：



接下来创建新的工程，用于编写代码，File->New->PyDev module,出现下面窗口：



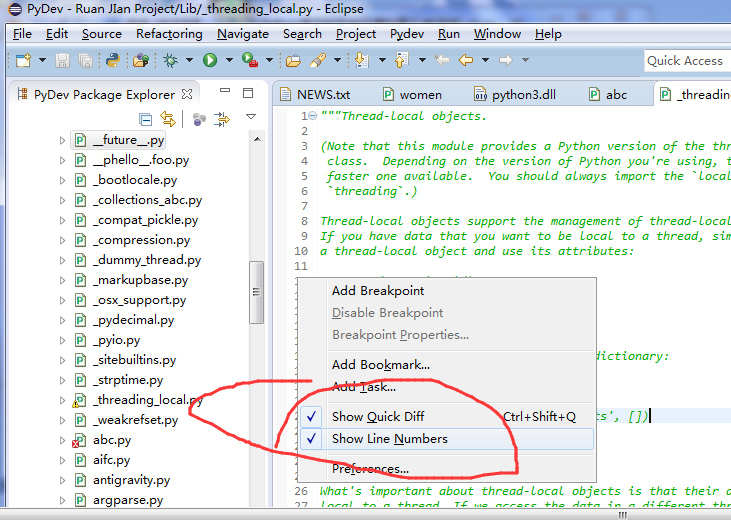
进而就可以写代码编程了,如下图：程序代码为ruan jian gongcheng，写完后点击工具栏中的Run按钮，在下方就会出现相应的结果。



1. **配置PyDev提高效率**

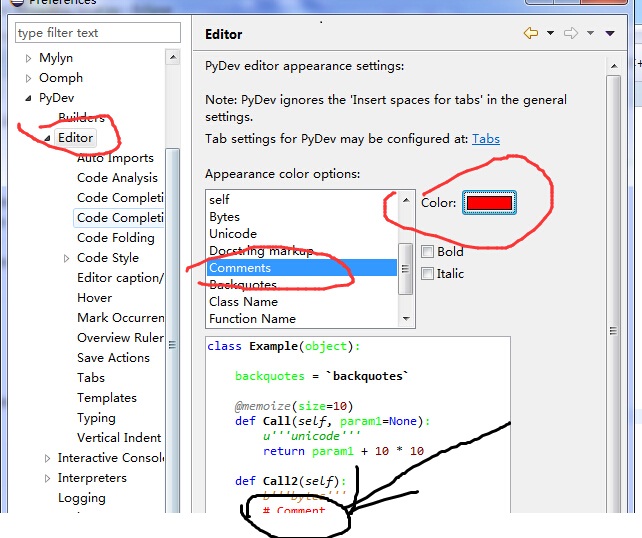
**4.1 显示源码行号**

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：

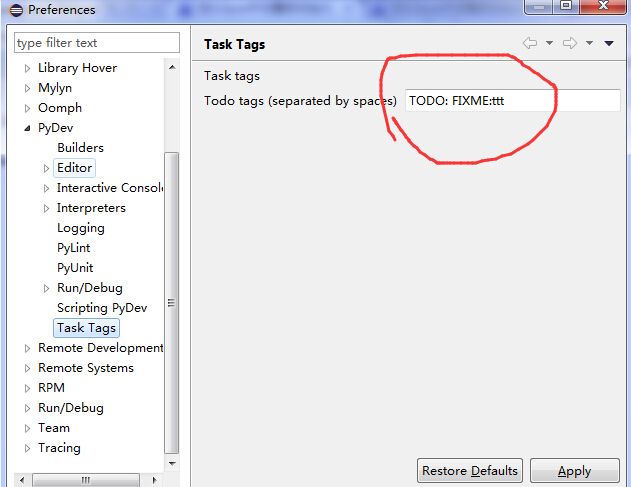


**4.3 配置任务标签**

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。

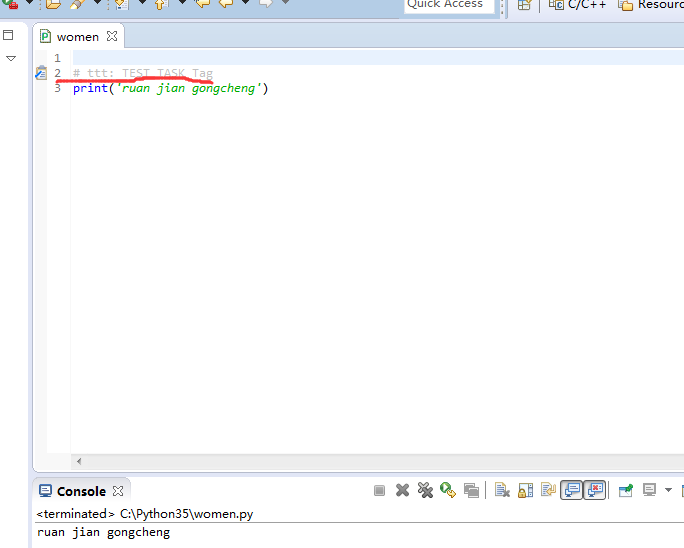
1. **配置任务标签**

通过PyDev->Task Tags中配置任务标签(自定义一个标签ttt):



1. **添加任务标签注释**

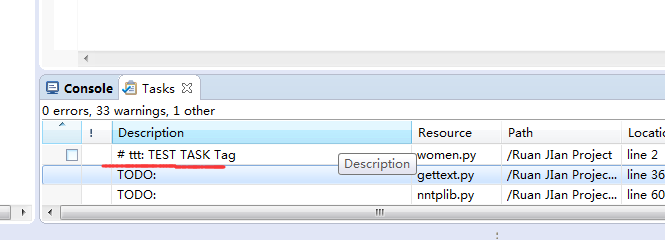
在程序源码中加入使用任务标签的注释：

****

1. **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。

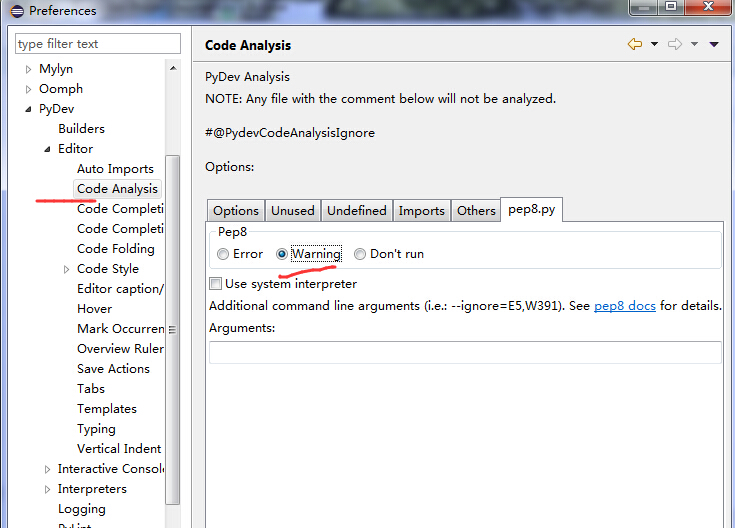
程序开发过程中，要有规范意识，但不可能有很高的规范性，过分注意规范会影响开发进程。这时如果一直开启代码规范检查，经常提示不规范，会对开发形成负面影响，所以，默认关闭是合适的。在程序开发一个阶段结果出来时，进行规范性检查更好。

**5.1 PEP8检查和修改**

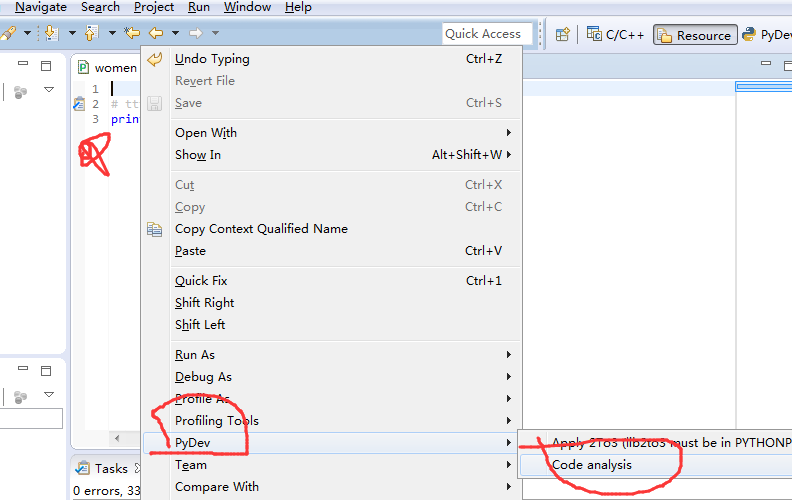
1) 启动pep8检查：

Window > Preferences PyDev> Editor > Code Analysis > pep8.py

选择Errors/Warnings其中之一..

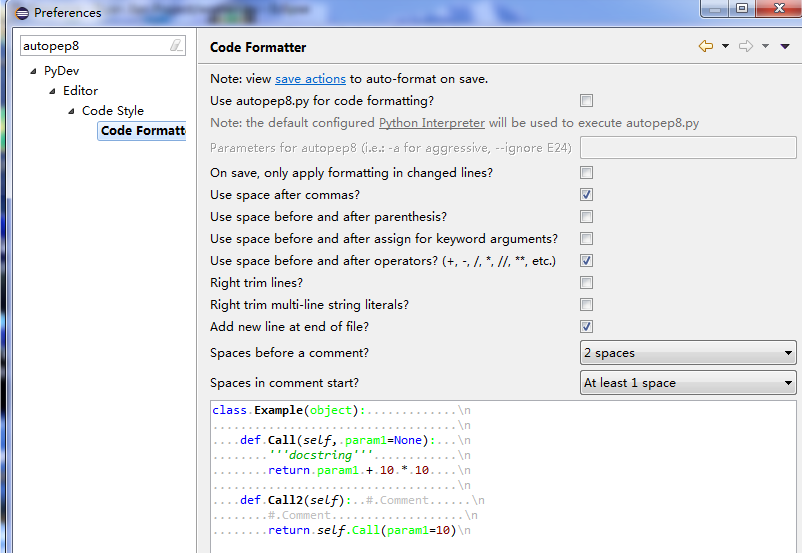


右键Python工程，选择PyDev，点 "code analysis"，即可对工程中所有Python源码进行PEP8检查：



2）启动autopep8自动修改：

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



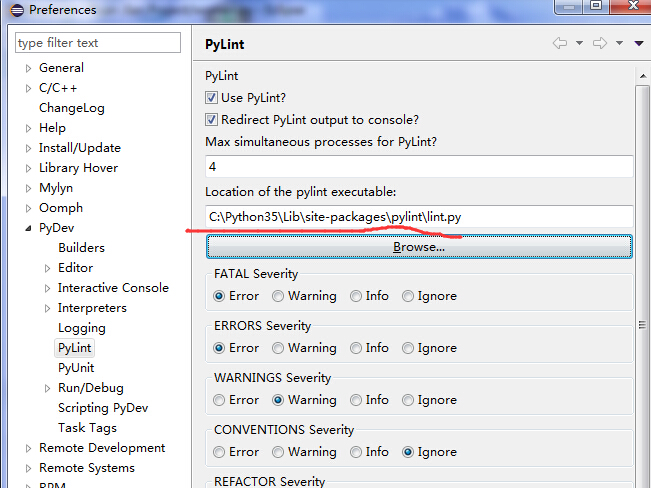
在Python源码窗口，按 CTRL-SHIFT-F 就可以自动修改代码

**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。

**小结**

Python学习和开发环境多种多样，现在正逐步了解IDLE，然后再去掌握其他。对于Eclipse和PyDev，是需要长时间的摸索的。在这次Python环境建立中，遇到了一些安装问题，从百度上查阅资料及询问同学，已经基本解决。

**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