**Python学习和开发环境的建立**

Python语言是一种解释型、面向对象的计算机程序设计语言。Python语言广泛应用于计算机程序设计教学语言、科学计算等，特别适用于快速的应用程序开发。它属于高级语言，简单、明确；既支持面向过程的编程，也支持面向对象的编程；它的可扩展性良好，提供了丰富的API和工具，便于程序员能够轻松的使用C、C++语言来编写扩充模块；而且它是免费和开源的，允许自由地发布此软件的拷贝、阅读和修改其源代码，或者将其一部分用于新的自由软件中。下面简单介绍一下python环境的建立过程。

**一、Python基本开发环境建立**

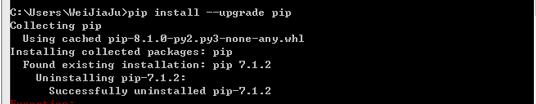
首先从python官网上下载与自己电脑操作系统相对应的python3软件。本人的电脑操作系统为64位，安装的是python3.5.1（目前最新版本），以下为简要安装说明：

文件下载完成后，打开它，出现安装程序的窗口时，勾选为所有用户安装+环境变量PATH，自定义安装目录C:\python35



完成之后，键盘输入win+r调出Windows，在命令行下更新pip到新版本：

>pip install --upgrade pip



**二、Python扩展包安装**

**2.1 科学计算软件包**

**2.1.1 SciPy**

SciPy官网<http://www.scipy.org/>，提供了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

**2.1.2 交互计算 Juoiter**

1） 安装Jupiter ：

>pip install jupyter 在线安装

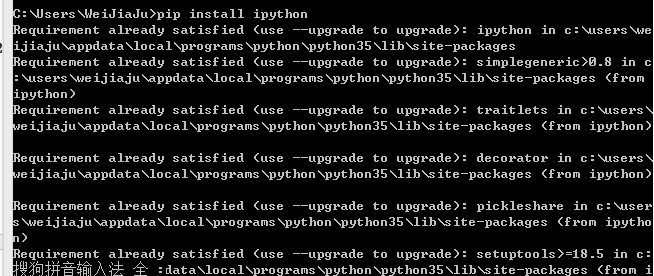
2） 安装Python语言内核

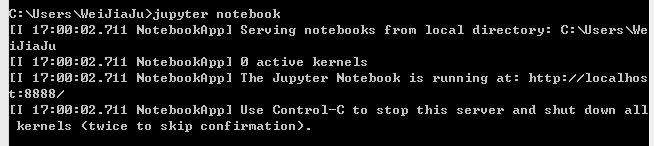
>pip install ipython ，支持Python语言

3） 运行notebook:

在iPython notebook文件所在目录下，打开命令行窗口：

>jupyter notebook





**2.1.3 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

**三、基于Eclipse的Python开发环境**

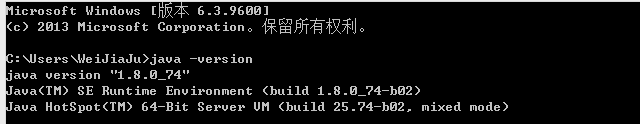
**3.1 安装Eclipse IDE**

**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**。**



本人电脑一开始未安装java软件包，通过从Oracle官网下载了Java JDK的安装包。

[**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.

**3.1.2 Eclipse IDE**

Eclipse IDE是插件型开发环境，根据本课程需要，推荐下载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32/64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。

为了以后打开eclipse运行方便，建议将eclipse.exe固定到任务栏中去（在eclipse.exe文件名上，点鼠标右键即可）。



C:\Users\WeiJiaJu\Documents\Tencent Files\906801734\FileRecv\MobileFile\Image\U0HDC{VD$7]SOUP~_QL}58X.png

同样也可以将ide固定到任务栏。

**3.2 安装Python开发插件PyDev**

**3.2.1 安装PyDev插件**

可采用离线下载安装，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.



注意：离线下载好的**PyDev**插件包必须解压之后放到eclipse文件所在目录之下，否则在Add->Archive添加**PyDev**的路径位置时将会出错。

**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**

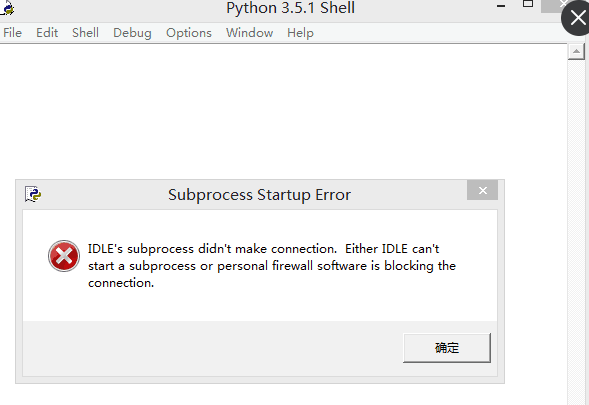
**Help** -> **Help Contents**



**四、Python安装过程遇到的问题及解决方案**

**4.1安装IDLE**

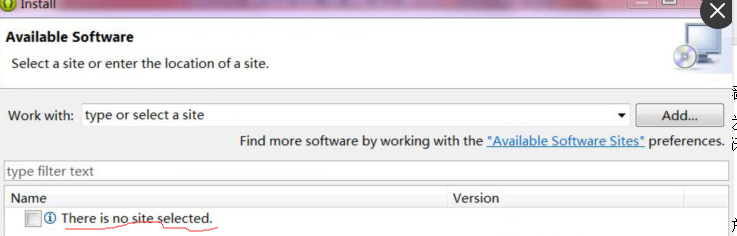
当完成Python的初步安装之后，初次打开出现错误如下：

****

然后按照上面的提示，将计算机的防火墙关闭了，重新打开ｉｄｌｅ，依然出现上面的错误提示对话框。多次尝试之后，问题始终没能解决，整个人濒临崩溃，最终，只好卸载Python，再重新安装，问题解决。

**4.2PyDev插件包的安装**

本人采用的是PyDev插件包的离线安装方法，先从官网上下载完安装包，然后按照教程进行。在添加PyDev，点击ＯＫ后，出现了下面的错误：

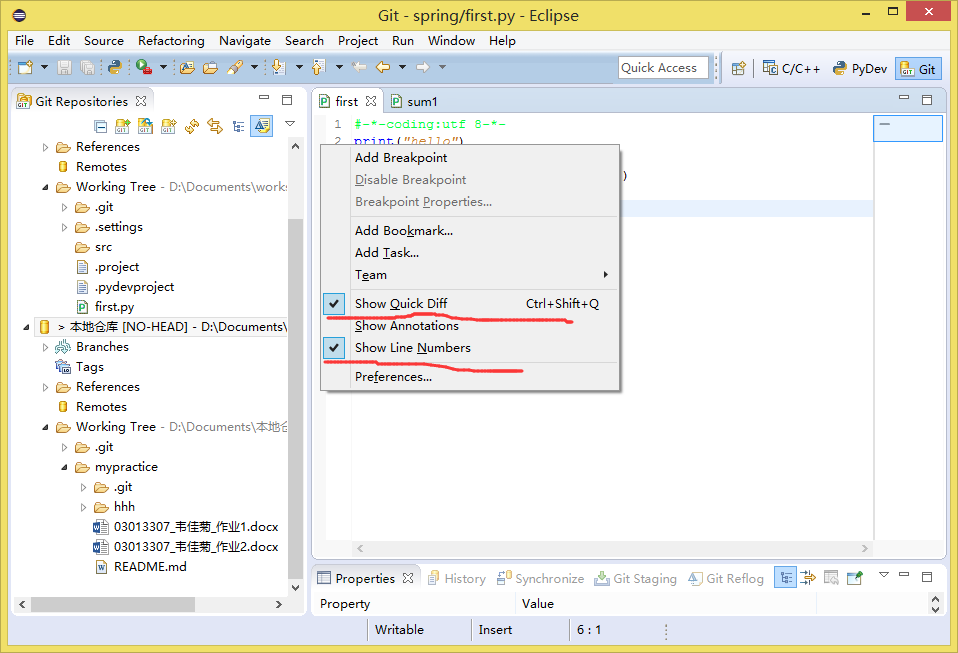


初步怀疑未解压PyDev插件包造成错误，于是退出后进行解压，问题还是没有解决。最后，百度发现离线下载插件包之后，必须将压缩包解压到eclipse安装目录下才能被识别，问题解决。

**五、配置 PyDev提高效率**

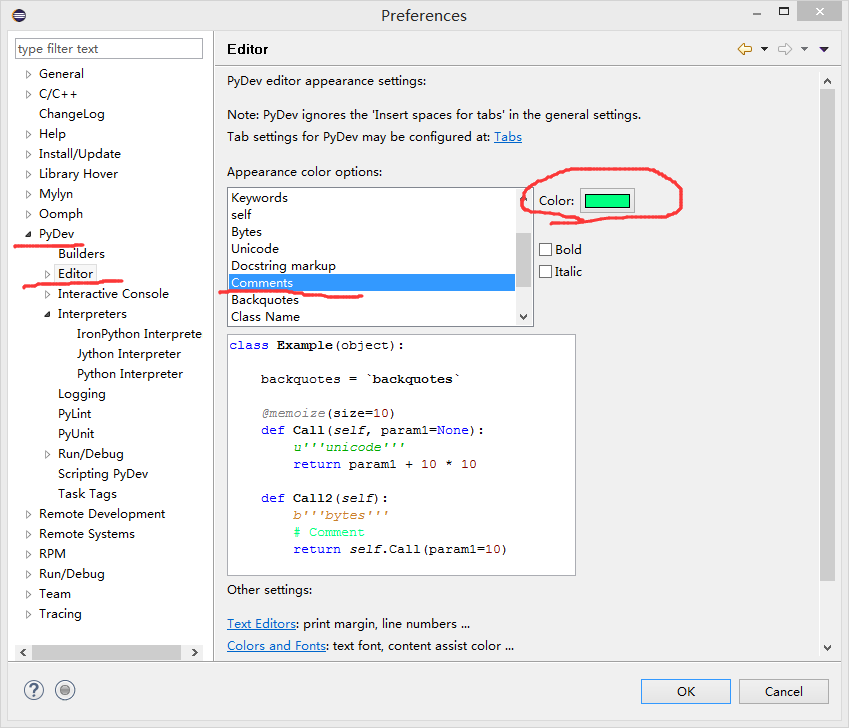
**5.1显示源码行号**

右键源码的左边缘，选中“Show Line Number（新版本已默认选中）。

****

**5.2 修改注释颜色提高可读行**

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

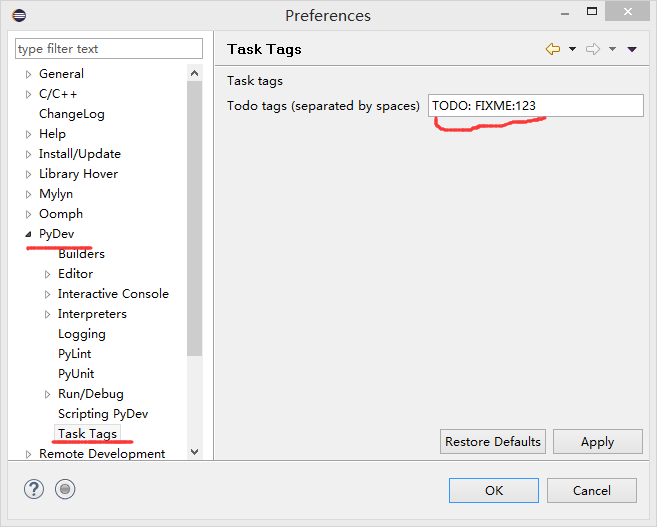


**5.3配置任务标签**

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，让开发环境将其识别。加入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:



**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：

C:\Users\WeiJiaJu\Documents\Tencent Files\906801734\FileRecv\MobileFile\Image\EAR@BOP$[0QBLDS6ZJO1J_B.png

**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。

**六、基于PyDev的语言规范静态检查**

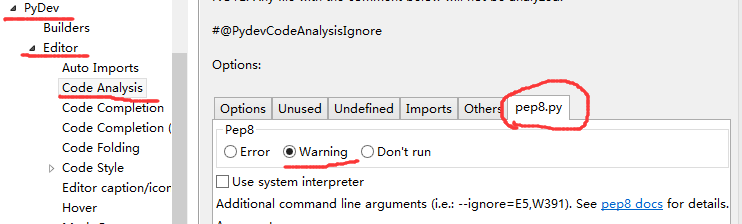
**6.1 PEP8检查和修改**

1）启动 pep8 检查：

Window > Preferences

PyDev > Editor > Code Analysis > pep8.py

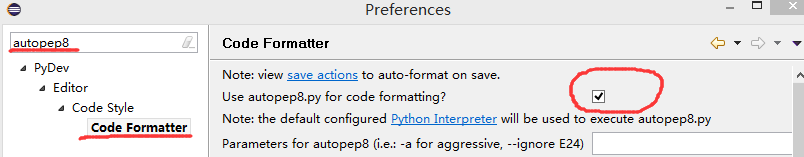
选择Errors/Warnings其中之一..



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。

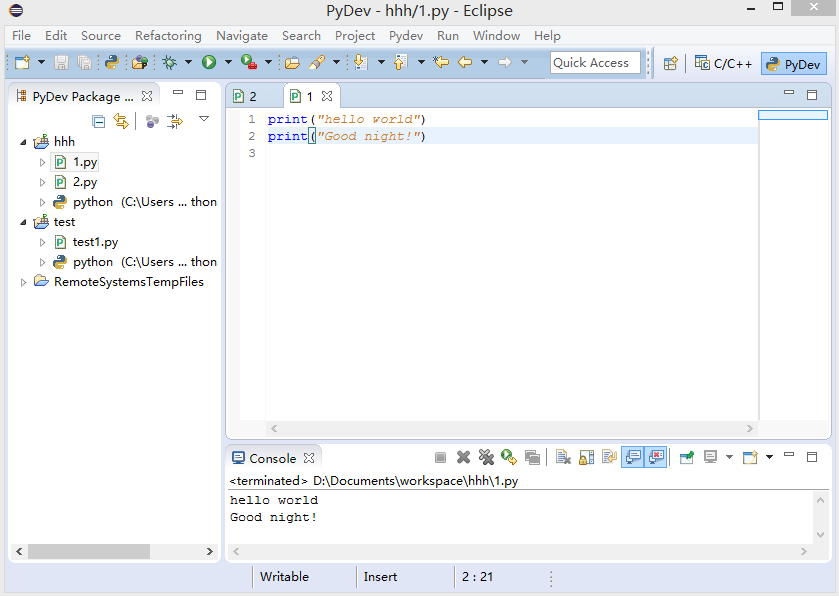
**6.2使用Pylint**

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"

个人电脑搜索不到lint.py的安装路径，在此省略配置过程。

**七、安装完毕，程序的实现与运行**

**八、小结**

总之，本人在Python的安装过程中碰到了各种各样的问题，难的问题多次尝试解决无果之后，只好选择卸载再重新安装；简单的百度一下或者自己琢磨或与同学交流之后就可以得到解决，上面只是列举其中两个较具代表的两个问题。通过这个安装的过程，我对Python的了解得到更进一步。了解掌握一款软件的应用，需要一个循环渐进的过程，只有不断地尝试摸索，不断地犯错误，再寻找解决方案，才能提高自己对它的认知水平。作为一个零基础的新手，初次接触学习Python，肯定会存在很多问题，必须不断的积累经验，才能加深了解。