# 开发环境建立手册

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解释器及其标准库来自班级分享；其他扩展包可用pip命令等方法安装。

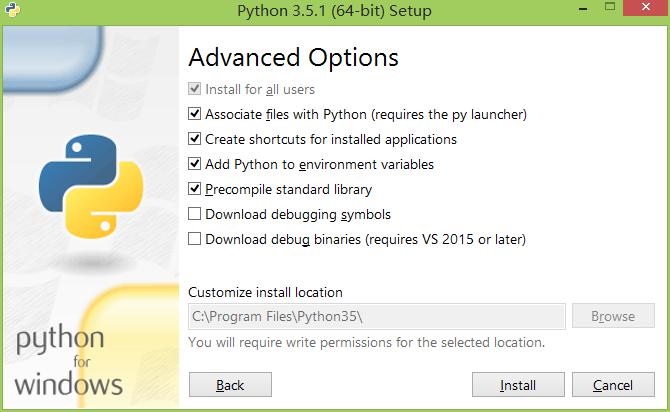
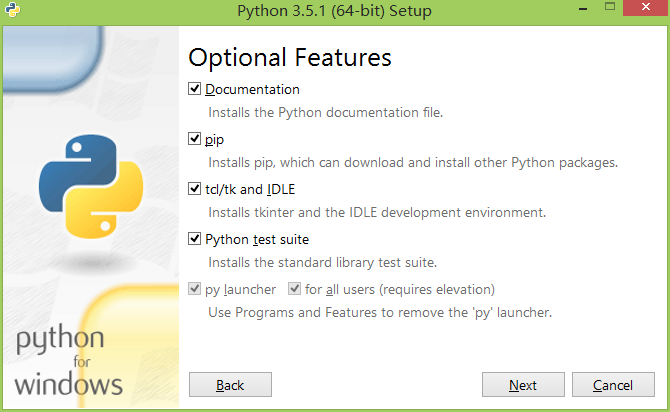
大型软件开发宜采用集成开发工具，常用的有: Eclipse，VS2015等，其中，Eclipse加PyDev插件的方式， 为较多Python程序开发人员使用。此外，还有使用编辑器(如Atom)构建开发环境的方法。

## Python**基本开发环境建立**

用Github中分享的安装文件安装Python3.5。

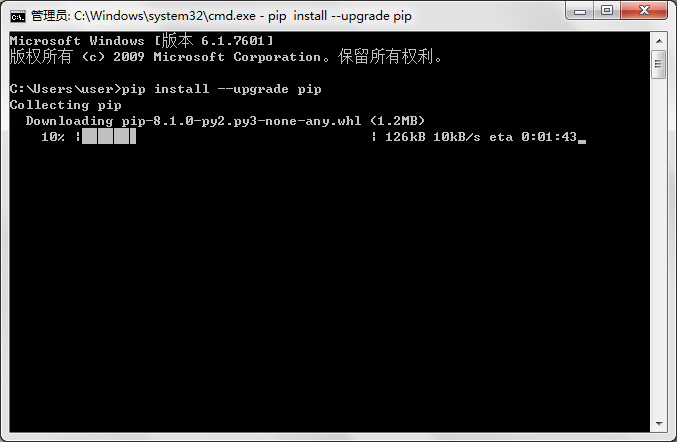
Windows系统安装： 1) 加环境变量PATH;

2) 自定义安装目录： C:\Program Files\python35。



然后，命令行下更新pip到新版本：

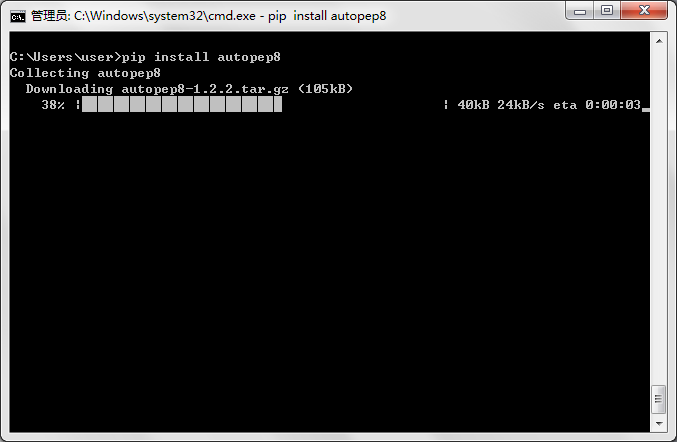
>pip install --upgrade pip



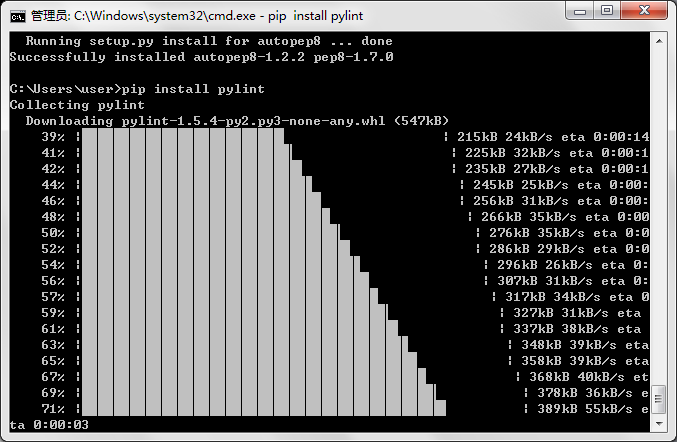
## 二、 Python拓展包安装

**2.1 编码规范**

pip install autopep8



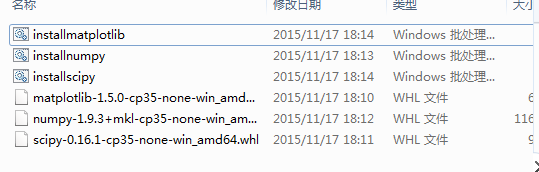
pip install pylint



**2.2 科学计算软件包**

**2.2.1 SciPy**

从共享文件中找到安装文件,依次安装。



**2.2.2 交互计算 Jupiter**

1） 安装Jupiter ：

>pip install jupyter 在线安装

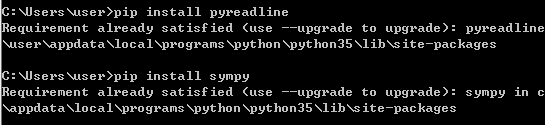
2） 安装Python语言内核

>pip install ipython ，支持Python语言

3） 安装依赖包

>pip install pyreadline

>pip install sympy



4）支持显示数学符号、公式，安装MathJax：

命令行下，

>IPython

打开一个IPython的shell，然后，在其中键入如下代码：in

from IPython.external.mathjax import install\_mathjax

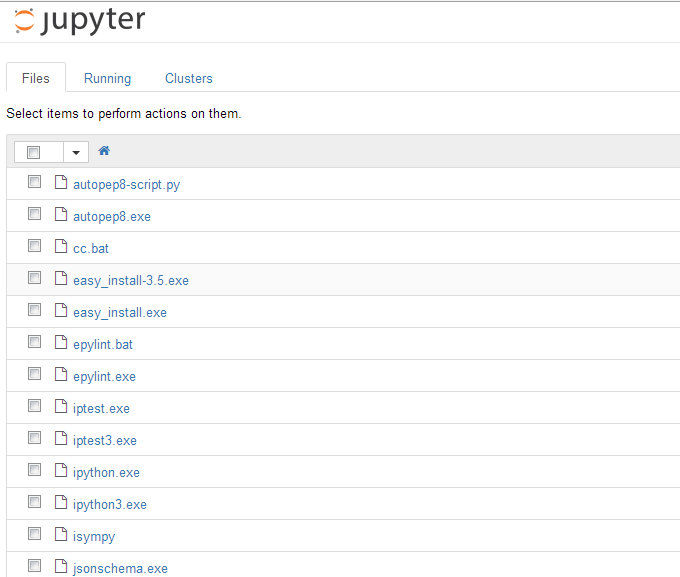
install\_mathjax()

（此步骤发生错误，待解决）

5） 运行notebook:

在记事本写入 jupyter notebook,保存为cc.bat。

在ipython notebook所在目录打开。



**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:\program file\python35\Lib

## 三、 基于Eclipse的Python开发环境

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台，专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台。Eclipse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。Eclipse 附带了一个标准的插件集，包括 Java 开发工具（Java Development Tools，JDT）。Eclipse 开源，而且一般Java开发更倾向于Eclipse，使用安装免费，操作简单，且包含很多插件，使用方便，但程序启动较慢。

**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。



**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，下载Eclipse CDT(Eclipse IDE　for　C/C++)版。

**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；

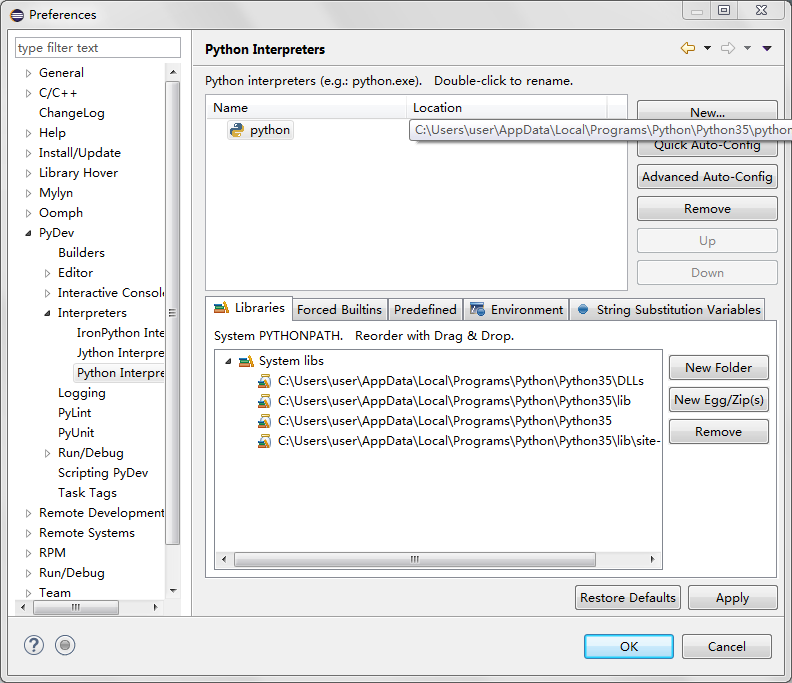
**3.2.1 安装PyDev插件**

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。

**3.2.2 配置 Python解释器**

安装好后，重新启动。通过：

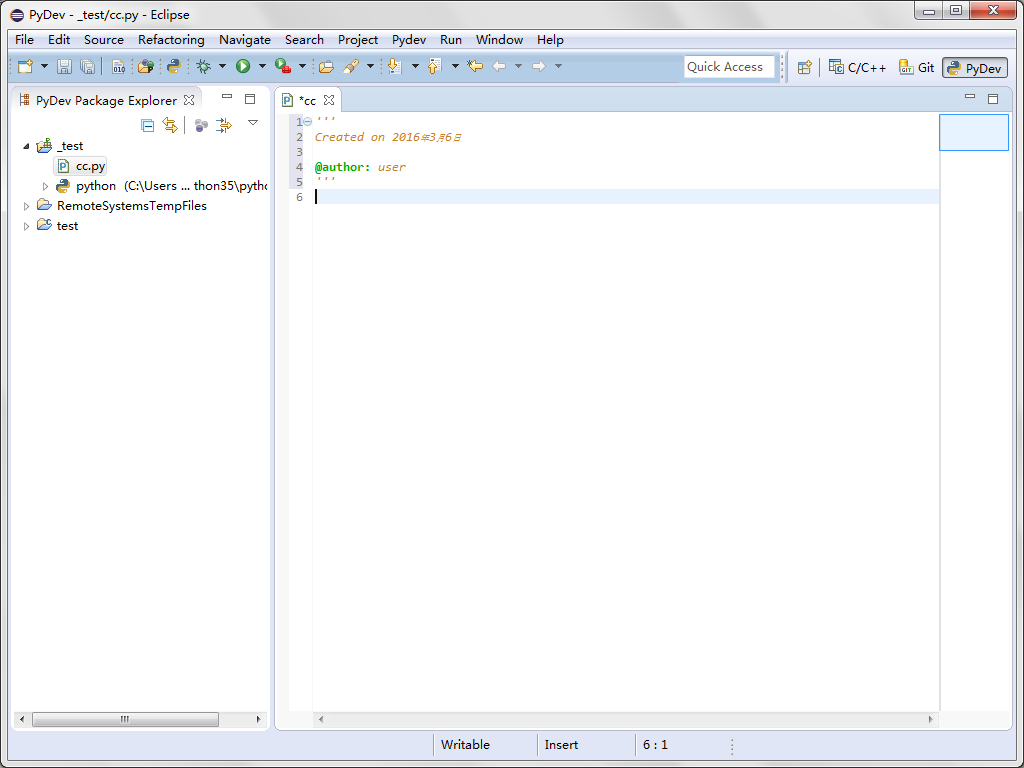
Windows->Preference->Pydev->Interperters->Python Interperter点其中的：Advanced Auto-config配置开发使用的 Python解释器版本:



配置好后切换到Python场景，就可以进行Python开发了。

1. **配置 PyDev提高效率**

新建一个\_test工程。

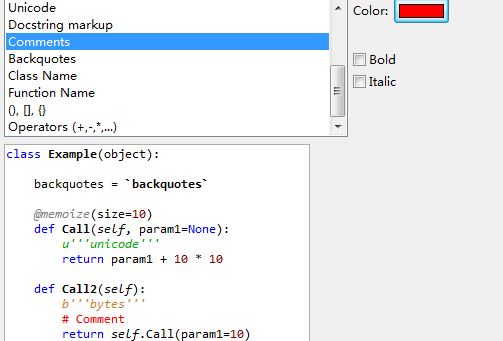


**4.1 显示源码行号**

右键源码的左边缘，选中Show Line Number（已经默认选中）。

**4.2 修改注释颜色提高可读行**

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



**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。

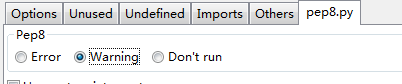
**5.1 PEP8检查和修改**

1）启动 pep8 检查：

Window > Preferences

PyDev > Editor > Code Analysis > pep8.py

选择Errors/Warnings其中之一..



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?",

## 小结

Python的开发环境多种多样，通过这次的实践，我大致了解了环境建立的过程，但还是有很多不懂的地方需要通过日后的学习实践深入了解。