|  |  |
| --- | --- |
| **Python环境建立指导手册**  **马智睿**  **03013418** |  |



[Python环境建立指导手册 3](#_Toc31778)

[一、 Python基本开发环境建立 4](#_Toc23668)

[二、 安装扩展包 5](#_Toc27113)

[2.1编码规范 5](#_Toc5435)

[2.2 科学计算软件包 7](#_Toc15206)

[三、 基于Eclipse的Python开发环境 14](#_Toc9964)

[3.1 安装Eclipse IDE 14](#_Toc1780)

[3.2 安装Python开发插件PyDev 18](#_Toc2304)

[3.3 联机指南 20](#_Toc16036)

[3.4 创建一个简单的Python工程 20](#_Toc17360)

[四、配置 PyDev提高效率 23](#_Toc30129)

[4.1 显示源码行号 23](#_Toc16592)

[4.2 修改注释颜色提高可读行 24](#_Toc3677)

[4.3 配置任务标签 24](#_Toc25965)

[五、基于PyDev的语言规范静态检查 26](#_Toc31687)

[5.1 PEP8检查和修改 26](#_Toc5640)

[5.2 使用Pylint 28](#_Toc24456)

[小结：软件工程这门课培养了我们的动手能力学习能力。只有自己跌跌撞撞去摸索，才能找到正确的方向。在开始配置软件的时候出了各种各样的问题，但最后都在同学的帮助下解决了，Python之旅才刚刚开始，用心领略编程的智慧。 29](#_Toc27049)

[参考文献： 30](#_Toc18709)

**Python环境建立指导手册**

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。

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> 下载与操作系统（Windows 32位/64位）对应版本的Python软件安装包。

安装Python软件时建议使用自定义安装方式

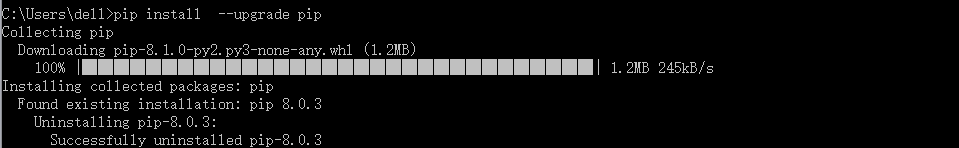


安装时需要联网。如果不接入互联网会在安装过程中停止在某一个阶段。

安装成功后，打开命令提示符，在命令行下输入

pip install --upgrade pip

更新pip至最新版本。



在更新过程中可能会失败，出现红字，此时是pip包bug，解决此问题的方法是卸载Python，删除一切相关文件，重新安装后，在命令行下输入

Python -m pip install -U pip

_9VA5]MU8AH]V1J8F~X_DHG

1. **安装扩展包**

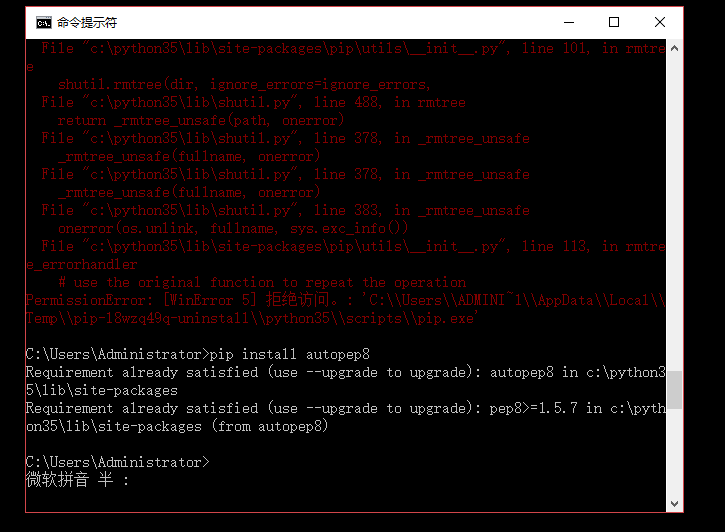
2.1编码规范

*2.1.1 autopepe8*

Autopep8是一个将Python代码自动排版为PEP8风格的小工具。它使用pep8工具来决定代码中的哪部分需要被排版。Autopep8可以修复大部分pep8工具中报告的排版问题。

在命令行下输入

pip install autopep8



*2.1.2 pylint*

Pylint 是一个 Python 工具，除了平常代码分析工具的作用之外，它提供了更多的功能：如检查一行代码的长度，变量名是否符合命名标准，一个声明过的接口是否被真正实现等等。

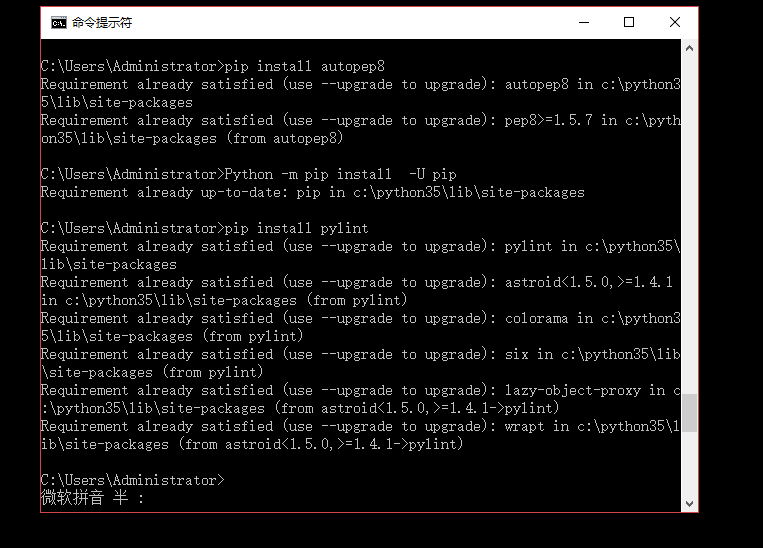
Pylint 的一个很大的好处是它的高可配置性，高可定制性，并且可以很容易写小插件来添加功能。

如果运行两次 Pylint，它会同时显示出当前和上次的运行结果，从而可以看出代码质量是否得到了改进。

目前在 eclipse 的 pydev 插件中也集成了 Pylint。

在命令行下输入

pip install pylint



2.2 科学计算软件包

*2.2.1 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。

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

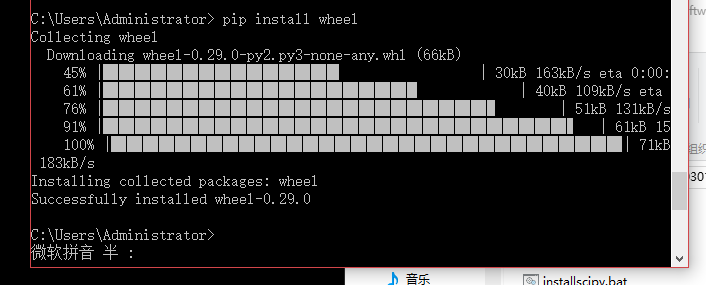
matplotlib

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文件。

官网地址如下

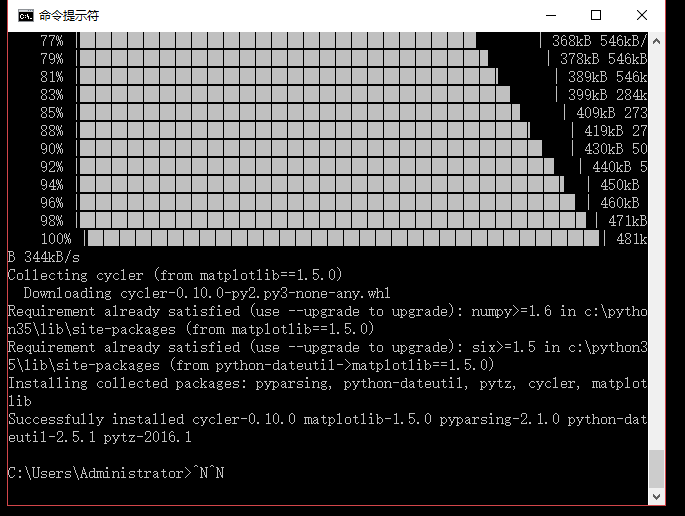
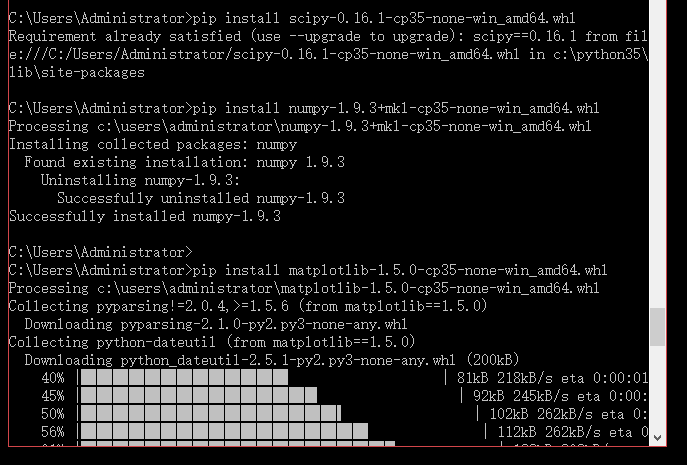
<http://www.lfd.uci.edu/~gohlke/pythonlibs/>

之后在命令行下输入

pip install wheel

然后，用pip逐个本地安装，在命令行下输入

pip install \*.whl（\*为下载下来的文件名）

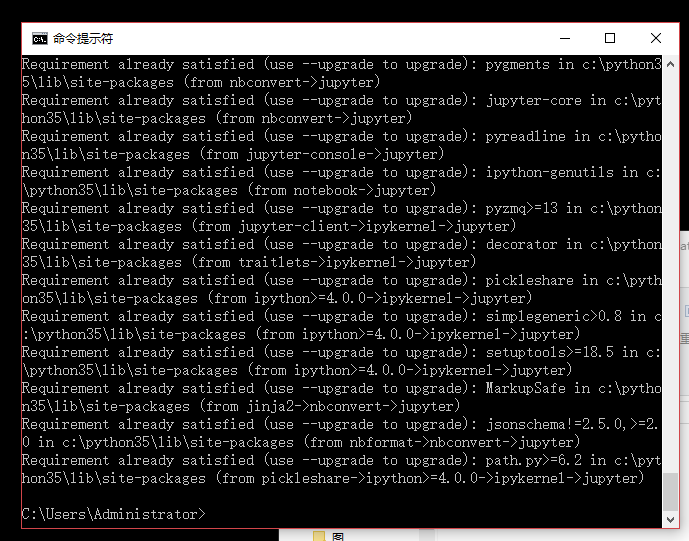


*2.2.2 交互计算 Jupiter*

1. 安装Jupiter

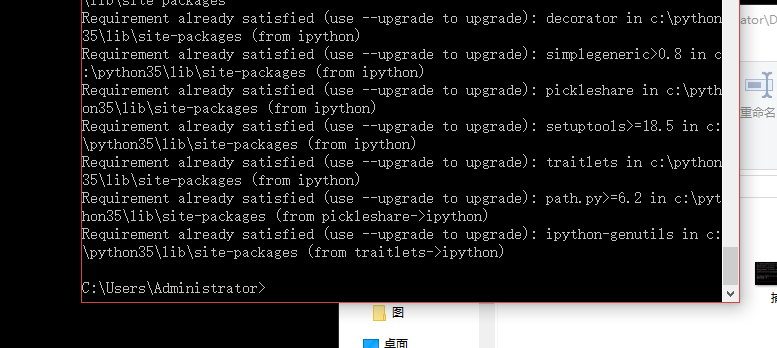
在命令行输入

pip install jupyter



1. 安装Python语言内核

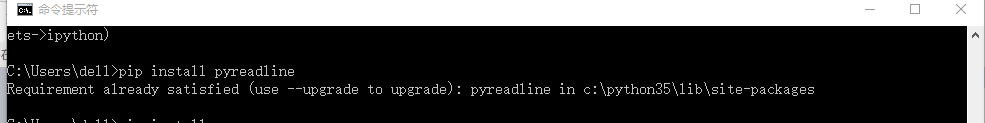
在命令行输入

pipinstallipython

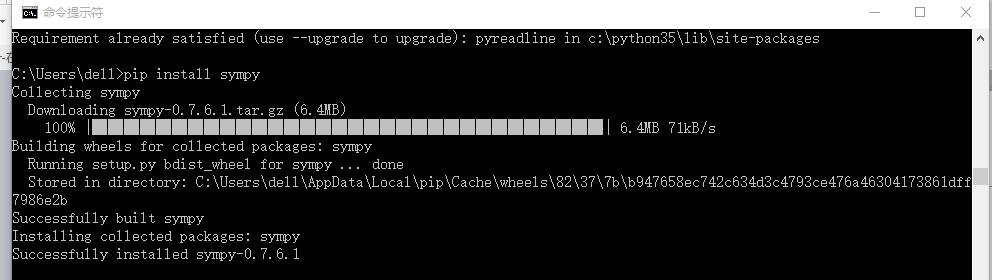
1. 安装依赖包

在命令行输入

pip install pyreadline



pip install sympy



4）支持显示数学符号、公式，安装MathJax

在命令行输入

IPython打开一个IPython的shell，然后，在其中键入如下代码

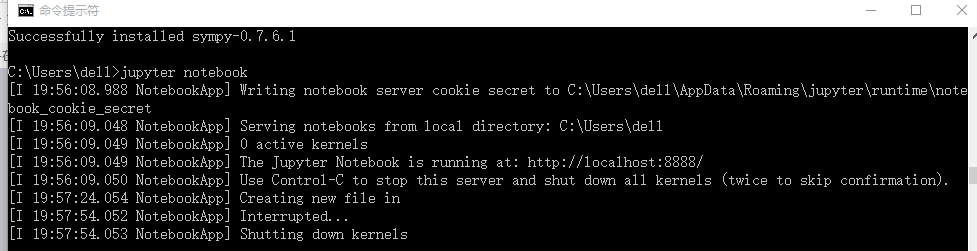
from IPython.external.mathjax import install\_mathjax

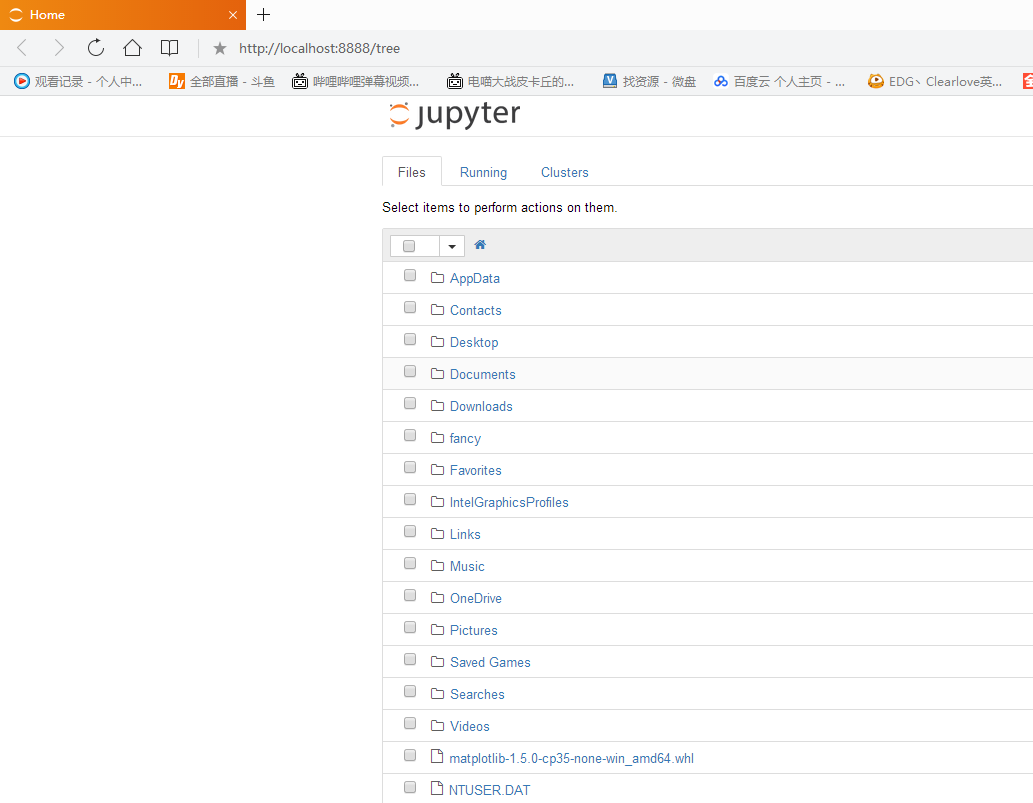
install\_mathjax()

5）运行notebook

在iPython notebook文件所在目录下，在命令行输入

jupyter notebook





*2.2.3 IF97 for Python*

Windows 32/64位版: 从

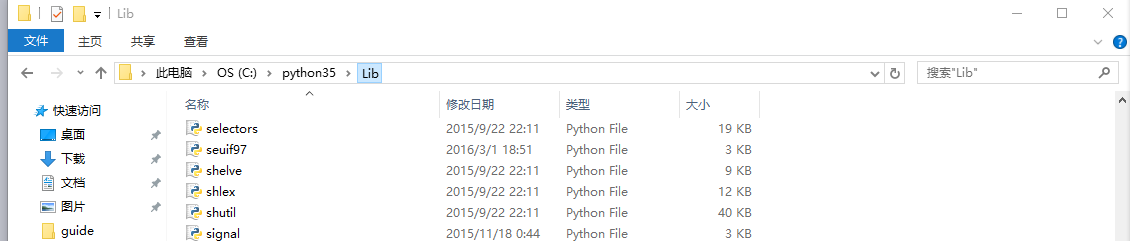
<https://github.com/Py03013052/SEUIF97>

下载：SEUIF97.dll和seuif97.py，然后

1) SEUIF97.dll拷贝到 c:\windows\system

![WTW9U9P](RSS39]{BK@O~G7](data:image/png;base64,)

2) seuif97.py拷贝到 c:\python35\Lib（如果安装时安装在了其他文件夹就拷贝到安装文件夹的lib文件夹中）



1. **基于Eclipse的Python开发环境**

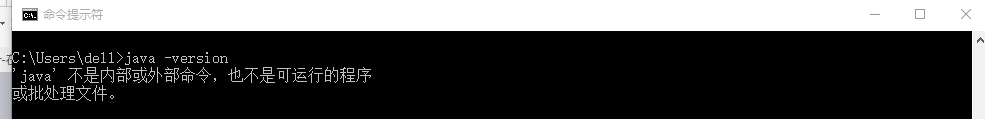
Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台，专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台。Eclipse只是一个框架和一组服务，用于通过插件组件构建开发环境。Eclipse 附带了一个标准的插件集，包括 Java 开发工具（Java Development Tools，JDT）。Eclipse 开源，而且一般Java开发更倾向于Eclipse，使用安装免费，操作简单，且包含很多插件，使用方便，但程序启动较慢。

3.1 安装Eclipse IDE

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

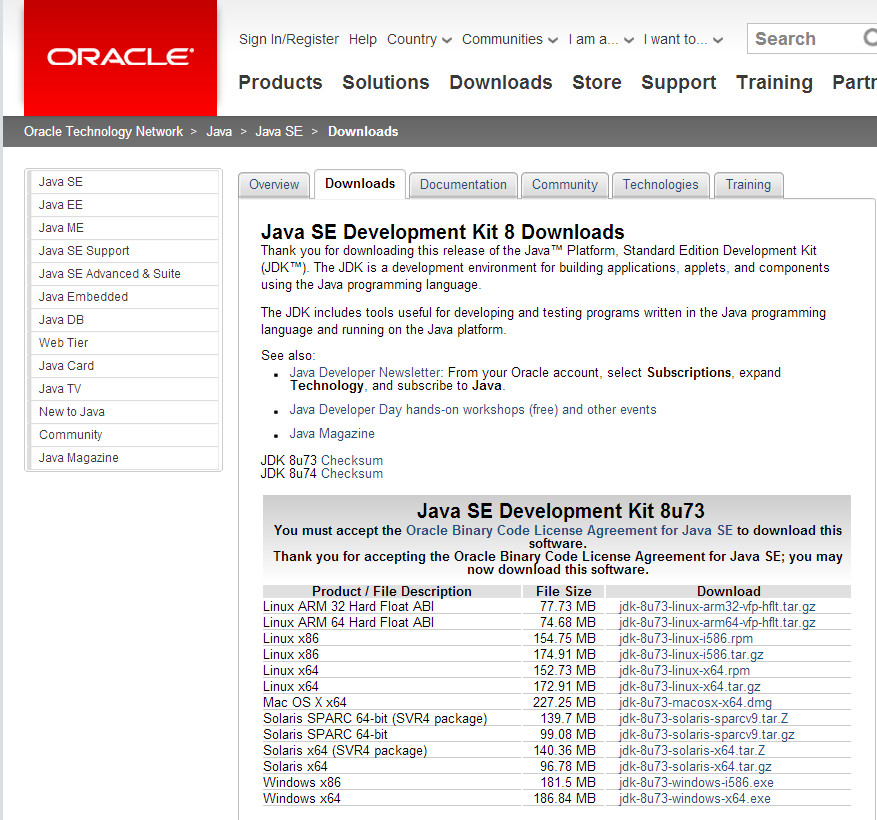
检查是已经安装了Java软件包。如果电脑中已经安装了Java ,会显示有关版本。



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的官网

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>



按照电脑配置下载相应的安装包。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。

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32/64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。

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：

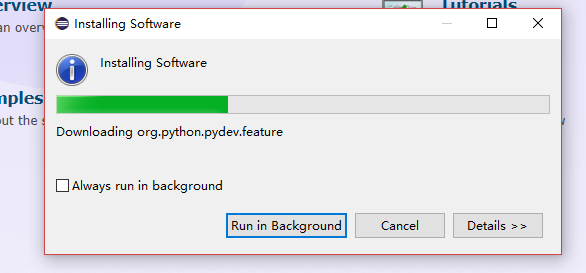
1. 安装PyDev插件；

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：





*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



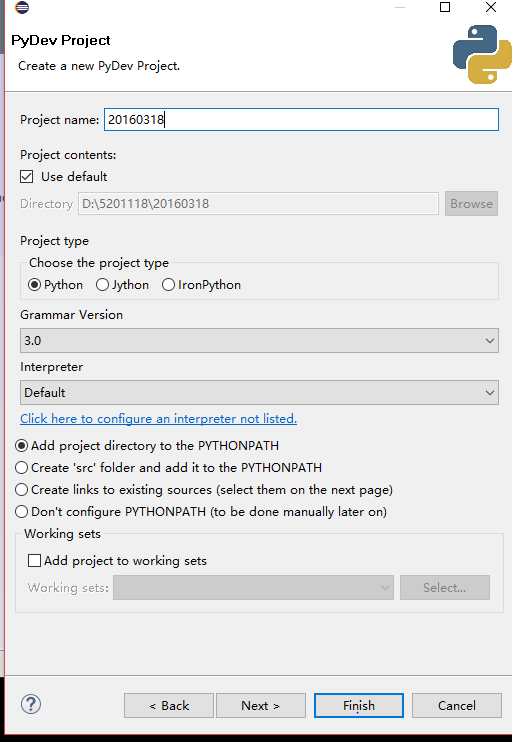
配置好后切换到Python场景，就可以进行Python开发了。

3.3 联机指南

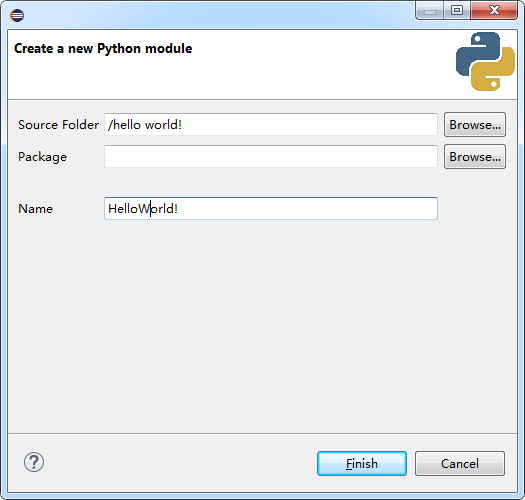
**Help** -> **Help Contents**



**3.4 创建一个简单的Python工程**

通过File->New->Project，在New Project中选择Pydev->Pydev Project，建立新的工程。

在左侧workplace中选择新建的工程，File->New->Pydev Module以建立可编写窗口。（注：文件名Name中不能有空格）

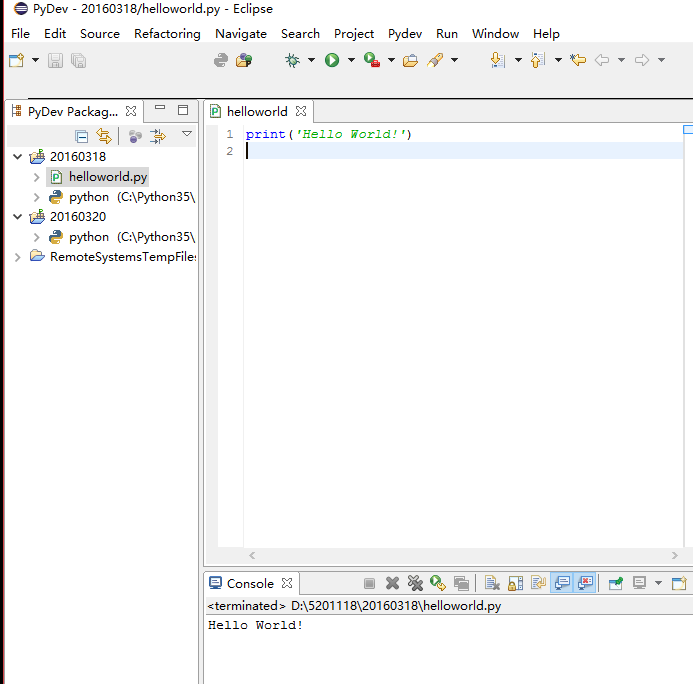
****

将已有的模版内容删除即可做为空工程使用。编写最简单的工程示例Hello World!作为示范

代码内容：

print(*'Hello World!'*)

于右上角选择PyDev，并点击左侧的运行按钮运行代码，运行结果显示于下方的console中。（注：命令必须区分大小写，若输入大写Print则无法识别）



**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：

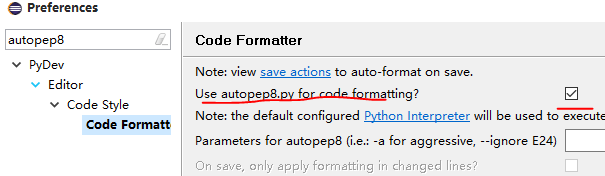


*2）启动autopep8自动修改：*

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:\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：软件工程这门课培养了我们的动手能力学习能力。只有自己跌跌撞撞去摸索，才能找到正确的方向。在开始配置软件的时候出了各种各样的问题，但最后都在同学的帮助下解决了，Python之旅才刚刚开始，用心领略编程的智慧。**

**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