[**Python学习和开发环境的建立**]

**能源与环境学院**

**热能与动力工程**

**03013432**

**李建杰**

**2016.3.15**



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。

[一、 Python基本开发环境建立 3](#_Toc18782)

[二、 Python扩展包安装 3](#_Toc19793)

[2.1 编码规范 3](#_Toc16518)

[2.2 科学计算软件包 4](#_Toc32043)

[2.2.1 SciPy 4](#_Toc2588)

[2.2.2 交互计算 Jupiter 5](#_Toc18845)

[2.2.3 IF97 for Python 5](#_Toc2895)

[三、 基于Eclipse的Python开发环境 5](#_Toc27652)

[3.1 安装Eclipse IDE 5](#_Toc9524)

[3.1.1安装JavaSDK 6](#_Toc16811)

[3.1.2 Eclipse IDE 6](#_Toc18017)

[3.2 安装Python开发插件PyDev 7](#_Toc21340)

[3.2.1 安装PyDev插件 7](#_Toc7875)

[3.2.2 配置 Python解释器 7](#_Toc23158)

[四、 配置 PyDev提高效率 8](#_Toc11992)

[4.1配置任务标签 8](#_Toc14254)

[五、 给Eclipse提速的技巧 9](#_Toc30187)

[5.1调整Eclipse的Preferences 9](#_Toc12520)

[六、 创建一个简单的python工程 9](#_Toc16924)

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python-3.5.1-amd64安装为例

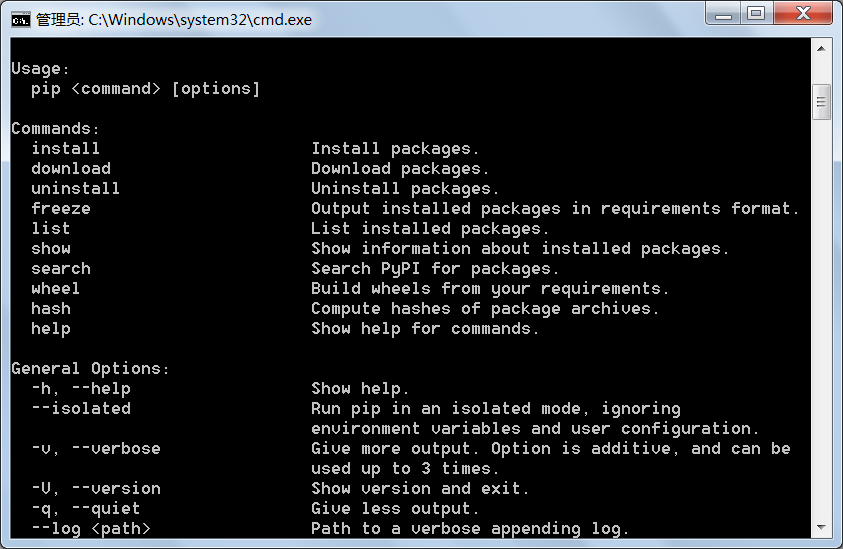
**Windows：** 建议使用定制方式安装: 1) 加环境变量PATH; 2) 自定义安装目录： C:\python35



然后打开控制台（windows键+R）,输入

>pip install --upgrade pip （更新pip到最新版本）

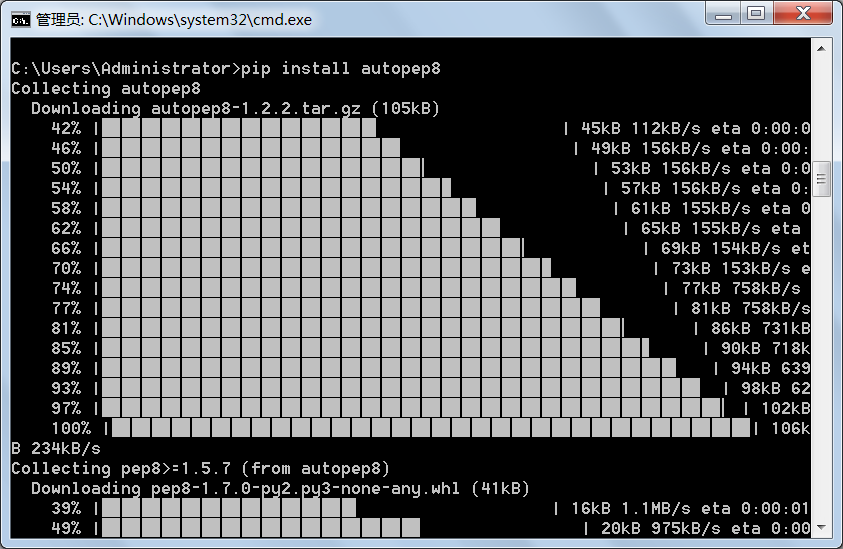
直至输入pip出现如下界面



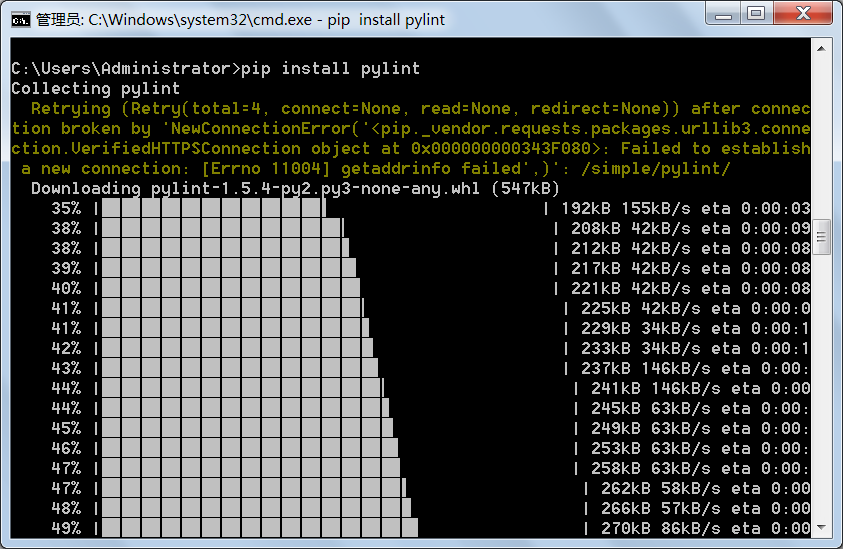
1. **Python扩展包安装**

**2.1 编码规范**

pip install autopep8



pip install pylint（安装代码检查工具，让python更规范）



**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） 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：

from IPython.external.mathjax import install\_mathjax

install\_mathjax()

5） 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**

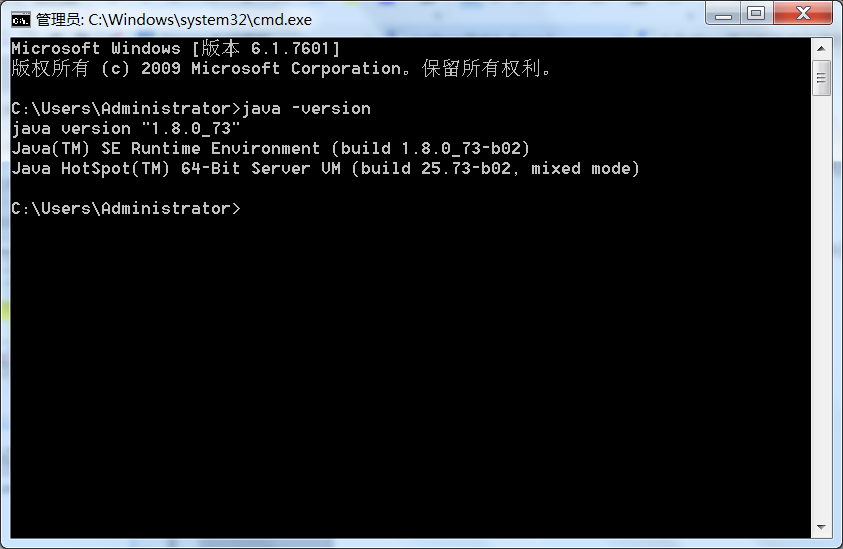
**3.1 安装Eclipse IDE**

**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>**

**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。

**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:



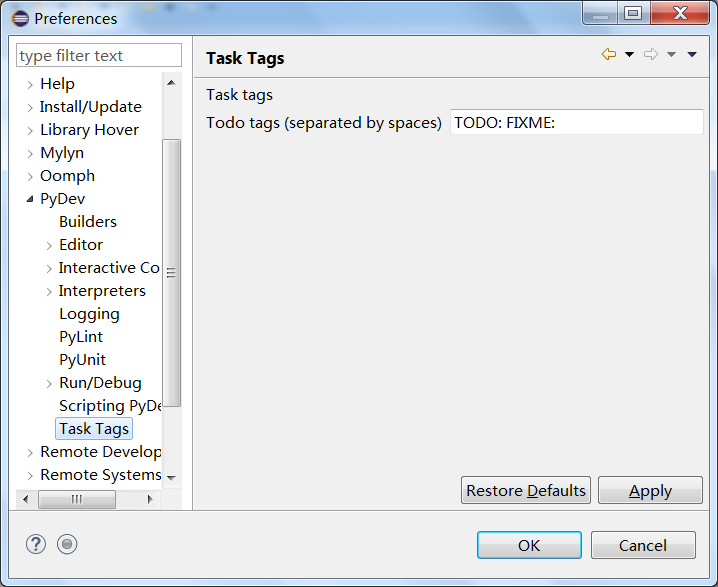
1. **配置 PyDev提高效率**

**4.1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:

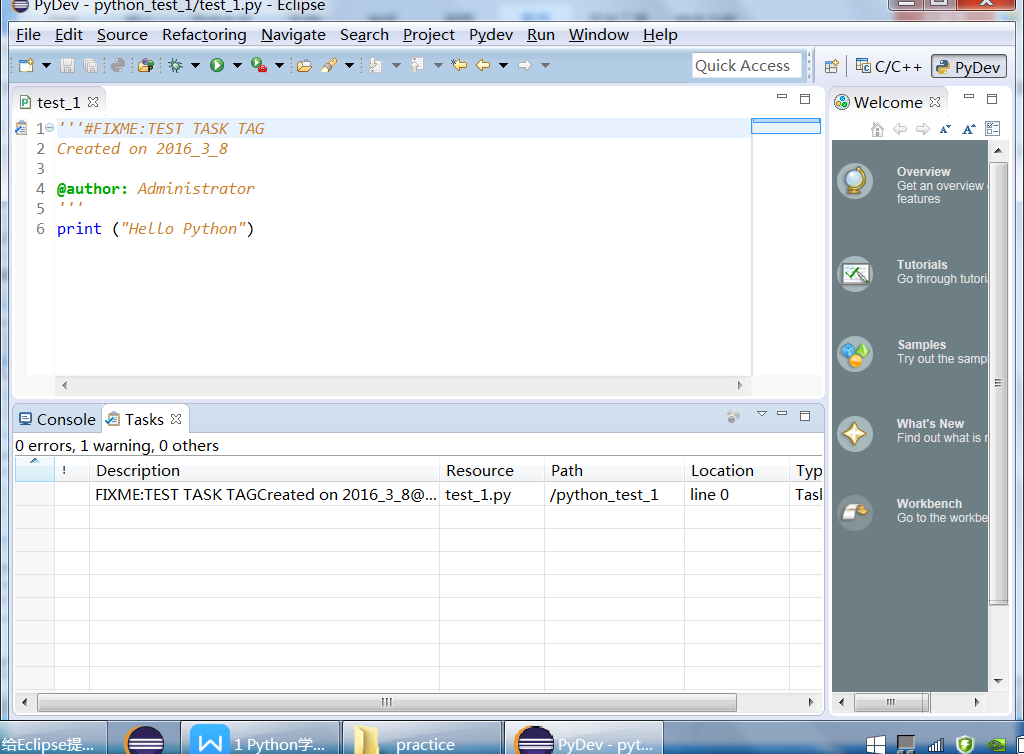


**2）添加任务标签注释**

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，



1. **给Eclipse提速的技巧**

**5.1调整Eclipse的Preferences**

General > Startup and Shutdown : 移除所有在启动时加载的插件。

General > Editors > Text Editors > Spelling : 关闭拼写检查。

General > Validation > 勾选“Suspend all validator”。

Window > Customize Perspective > 移除所有用不到或不想用的内容（尽量使用快捷键），菜单栏也是如此（你用过几次菜单栏的打印按钮？）。

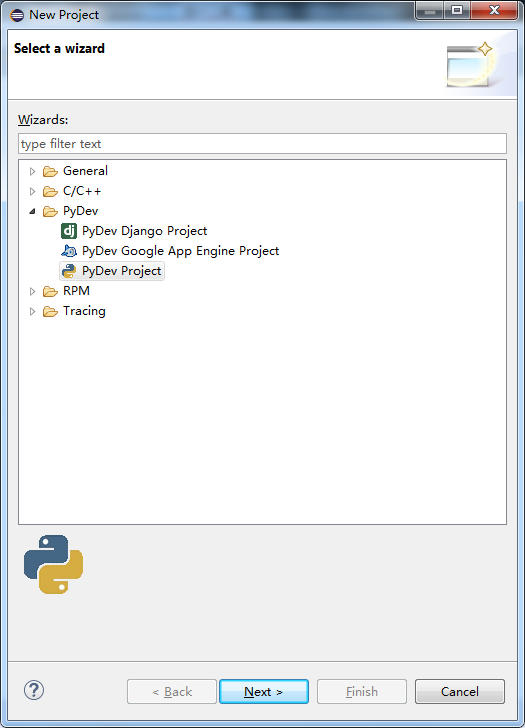
Install/Update > Automatic Updates > 取消勾选“Automatically find new updates and notify me”。

General > Appearance > 取消勾选“Enable Animations”。

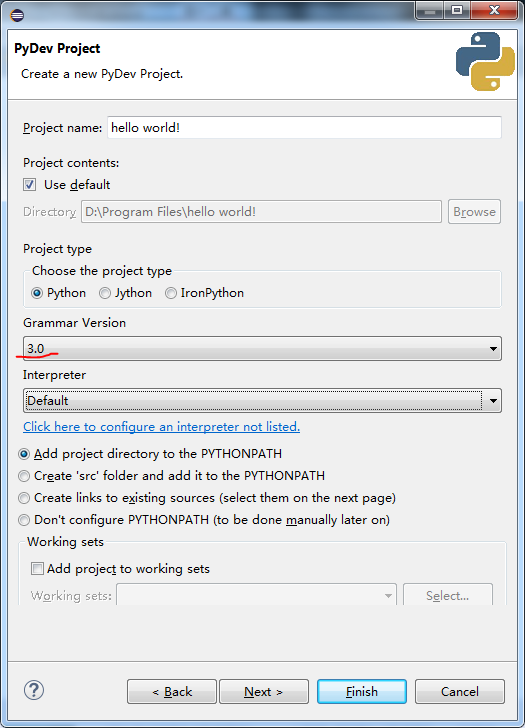
使用默认的主题。其他主题可能会降低运行速度。

1. **创建一个简单的python工程**

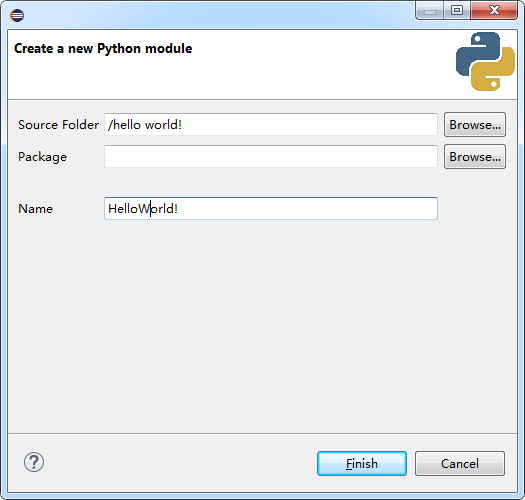
通过File->New->Project，在New Project中选择Pydev->Pydev Project，建立新的工程。

****

选择语法版本为3.0

****

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 Python!作为示范

代码内容：

print(*'Hello Python!'*)

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）

