**Python学习和开发环境建立指导手册**

Python（蟒蛇）是一种动态解释型的编程语言。Python功能强大、简单易学、支持面向对象、函数式编程。Python可以在Windows、UNIX等多种操作系统中使用，也可以在Java、NET开发平台上使用。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。

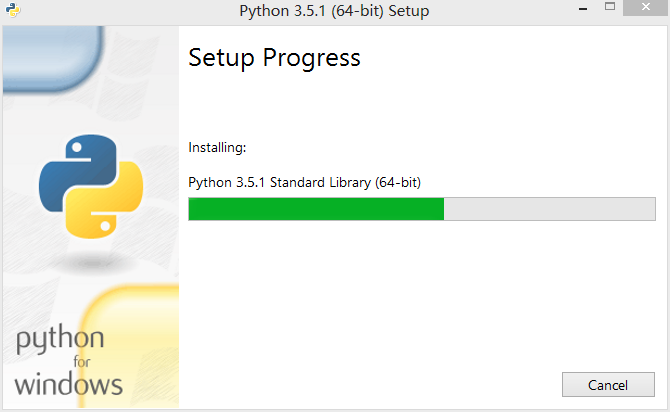
大型软件开发宜采用集成开发工具，常用的有: Eclipse，VS2015等，其中，Eclipse加PyDev插件的方式，为较多Python程序开发人员使用。此外，还有使用编辑器(如Atom)构建开发环境的方法。

1. **Python基本开发环境建立**

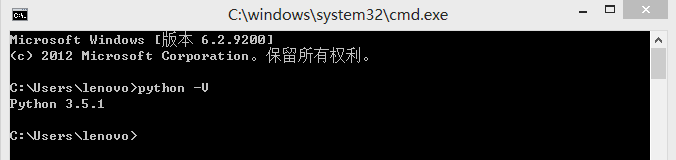
从官网<http://www.python.org>下载和操作系统（32/64位）对应版本的Python3软件。（如有安装包的话亦可以直接安装，本人采用的即为安装包安装方式）Windows系统建议使用定制方式安装: 1) 加环境变量PATH; 2) 自定义安装目录： C:\python35（默认装在C盘）



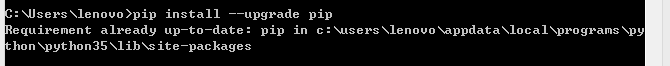
然后按下上图的install选项则进入下图的状态。



利用Windows + R 〉〉cmd 〉〉python -V；确认安装好python3.5.1如下图:



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



1. **Python扩展包安装**

pip命令是软件包的基本安装方式，不过，有些软件包使用pip安装时，需要本地编译，编译条件如果不满足安装就会失败。

这时可使用： 加州大学欧文分校提供的Windows32/64位Python扩展包编译版，网址为：<http://www.lfd.uci.edu/~gohlke/pythonlibs/>

加州大学欧文分校的编译版包，内容丰富、更新及时、安装方便。

**2.1 编码规范**

命令行下输入>pip install autopep8（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。）

命令行下输入> pip install pylint [Pylint 是一个 Python 代码分析工具，它分析 Python 代码中的错误，查找不符合代码风格标准（Pylint 默认使用的代码风格是 PEP 8，具体信息，请参阅参考资料）和有潜在问题的代码。]

**2.2 交互式计算jupyter**

**2.2.1安装jupyter**

命令行下输入>pip install jupyter 在线安装(The jupyter Notebook is a web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.)

**2.2.2安装Python语言内核**

命令行下输入>pip install ipython 支持Python语言（ipython 是一个 python 的交互式 shell，比默认的python shell 好用得多，支持变量自动补全，自动缩进，支持 bash shell 命令，内置了许多很有用的功能和函数。）

1. **基于Eclipse的Python开发环境**

Eclipse是Java开发的一个集成开发环境，而且是一个开源项目。Eclipse的扩展性非常强，它不仅可以作为Java的IDE使用，而且还可以开发大量的插件支持其他类型的语言，如C、C++、Python、PHP等。只需下载PyDev插件就可以在Eclipse平台上开发Python。Easyeclipse网站提供了eclipse的各种插件下载，而且可以获取单独运行的Eclipse for Python。

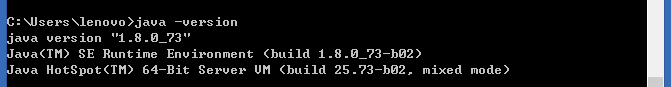
下载地址为：[www.easyeclipse.org](http://www.easyeclipse.org)。

**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。如下图：



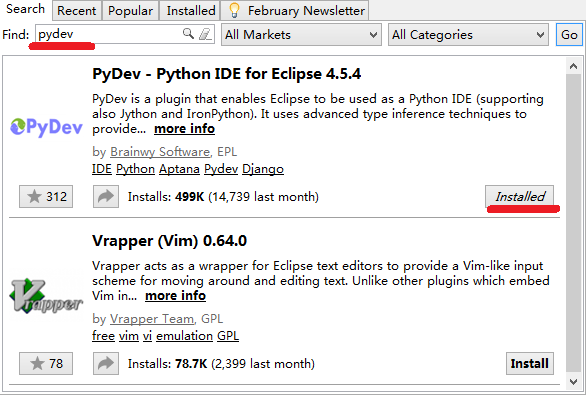
如果没有安装java 或者版本不能满足Eclipse IDE最低要求，需要从Oracle下载Java包。本人是从Java官网下载的。下载时需要注意操作系统32/64位，选择相应版本下载。

**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Python3.5。

**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：



在线安装过程简单，但是，安装速度受网络环境影响。如果很慢，可从pydev官网（<http://www.pydev.org/>）提供的下载地址：<http://sourceforge.net/projects/pydev/files/>

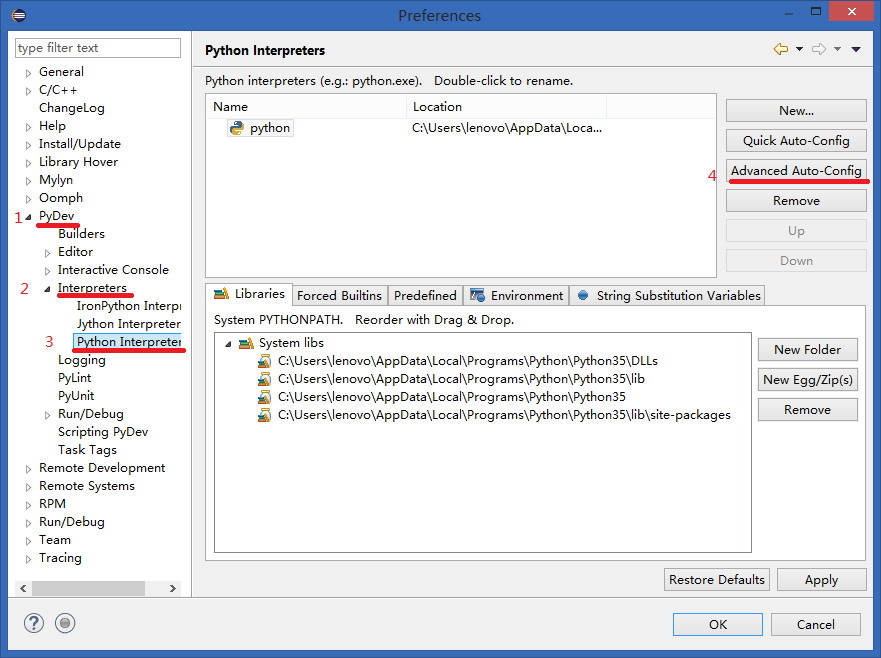
下载插件包，然后通过：Help->Install->New Software->Add->Archive

离线安装。(此过程是在线安装故采用老师给的参考资料)



**3.2.2 配置 Python解释器**

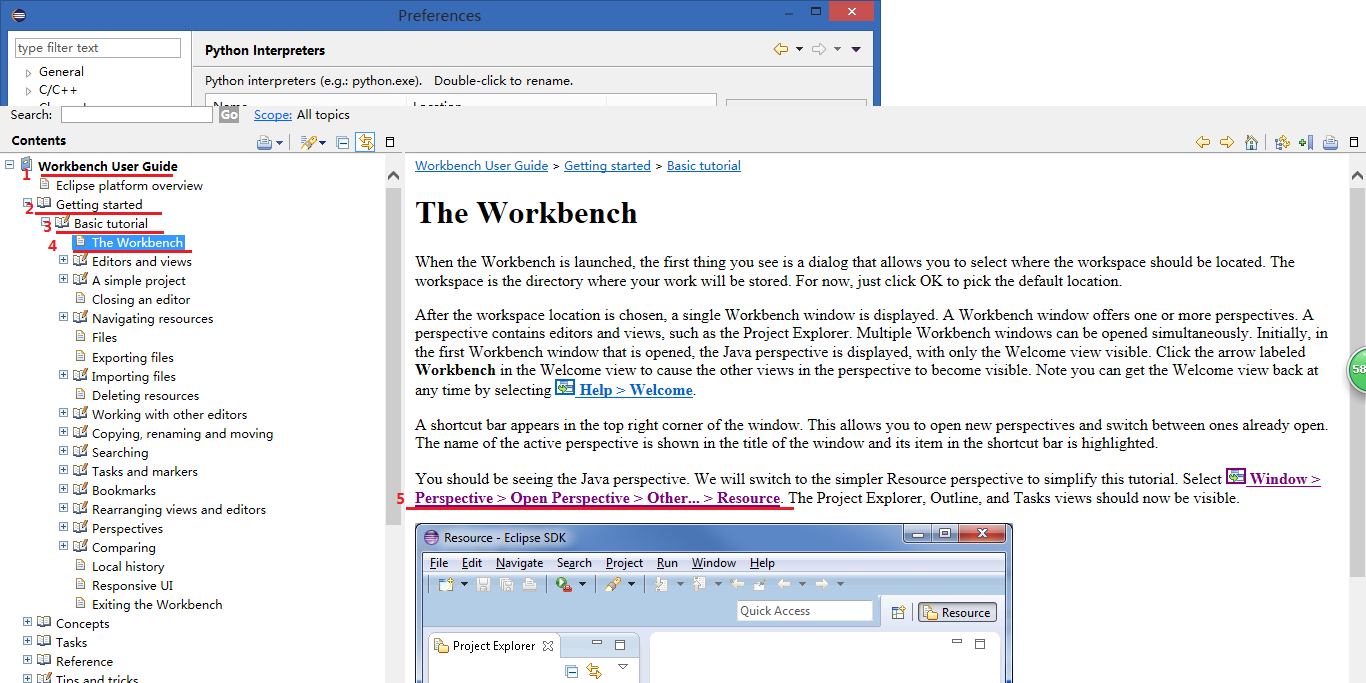
安装好后，重新启动。通过：Windows->Preference->Pydev->Interperters->Python Interperter点其中的：Advanced Auto-config配置开发使用的 Python解释器版本:



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**

****

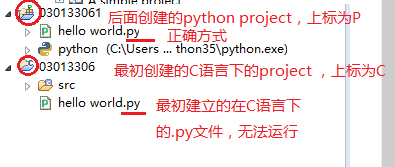
1. **总结**

**4.1 心得体会**

一开始对python真的是一无所知，也很惶恐学不好这门课（虽然现在也是很惶恐），就像热锅上的蚂蚁扎进图书馆找关于python的资料，所以对python有了初步的了解。安装python时我并不急，而是先通过书籍对这门课有个初步的认识。我知道我什么都不知道装了估计也是一堆错误还要各种重装什么的**。**因为觉得老师的安装包太多所以就干脆直接从python官网下载了python3.5.1，直到上周末才开始安装，额，果然还是卸载了一次，主要是然后就觉得自己安装的怎么长的那么奇怪和别的同学的为什么不一样，情急之下就将其全都卸载了。咨询了大神，然后再按着老师给的步骤一步一步下来，安装过程中一开始下载的Java不使用于eclipse，所以又从Java官网直接下载了适合64位的，eclipse才可以用得起来。

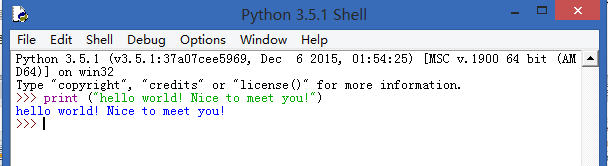
打开eclipse时干了一件很蠢的事情，启动界面是welcome，然后一直在惶恐自己是不是装错，又问了一个很蠢的问题，我的eclipse要怎样打开啊？最后大神提醒一句你把welcome界面关掉啊，呃，好吧，忽略刚才那个问题。在使用eclipse时一下子忘记了老师说过的要在文件名后加入.py所以一直没运行起来，加入.py后又发现自己创建的project是基于C语言的，总之真的回头再看会有自己真是蠢的感觉。惶恐都是不了解不自信才会产生的情感，所以克服惶恐的最好方式就是努力掌握这门课，一步一步慢慢来。加油！加油！加油！

错误详情见下图：



此指导书中的部分当时安装过程未截取的图参考了老师的，而且参考指导书中有一部分未按照步骤全部实现，故不说明。

Python 3.5.1的编辑结果，如下图：



虽然只是很简单的一句hello world！Nice to meet you!是一个开始，一个初次接触python语言的开始。

重新安装后的Python安装目录：

