

**软件工程实践**

课程实践二：Python开发环境

**姓名：曾令超**

**学号：03014313**

**学院：能源与环境学院**

**专业：能源与动力工程**

**指导教师：程懋华**

**2017年3月19日**

**目录**

[**第一章 Python基本开发环境和程序示例** 1](#_Toc477727317)

[**1.1 Python语言介绍** 1](#_Toc477727318)

[**1.2 Python安装** 1](#_Toc477727319)

[**1.2.1 Python的下载** 1](#_Toc477727320)

[**1.2.2 Python的安装** 2](#_Toc477727321)

[**1.3 Python程序示例（使用IDLE编程）** 4](#_Toc477727322)

[**第二章 Python扩展软件包安装** 4](#_Toc477727323)

[**2.1 在线检查、更新pip到最新版本** 4](#_Toc477727324)

[**2.2 编码规范** 5](#_Toc477727325)

[**2.3 科学计算软件包** 7](#_Toc477727326)

[**2.4 IF97水和水蒸汽物性计算** 9](#_Toc477727327)

[**第三章 交互计算环境和程序示例** 10](#_Toc477727328)

[**3.1 安装Jupyter环境包** 10](#_Toc477727329)

[**3.2 创建Jupyter Notebook文件** 10](#_Toc477727330)

[**第四章 集成开发环境和程序示例** 12](#_Toc477727331)

[**4.1 安装Java SDK并配置环境变量** 12](#_Toc477727332)

[**4.2 Eclipse安装过程** 13](#_Toc477727333)

[**4.3 Python开发插件PyDev** 14](#_Toc477727334)

[**4.3.1安装PyDev插件** 14](#_Toc477727335)

[**4.3.2配置 Python解释器** 17](#_Toc477727336)

[**4.4 配置 PyDev提高效率** 22](#_Toc477727337)

[**4.4.1修改编辑器配色提高可读行** 22](#_Toc477727338)

[**4.4.2配置任务标签：** 22](#_Toc477727339)

[**4.5 基于PyDev的语言规范静态检查** 23](#_Toc477727340)

[**4.6 Eclipse配置** 23](#_Toc477727341)

[**4.6.1工作空间编码方式** 23](#_Toc477727342)

[**4.6.2 工程编码方式** 24](#_Toc477727343)

[**4.7 Eclipse使用示例** 24](#_Toc477727344)

[**第五章 建立开发环境等过程中出现的问题和解决方法** 26](#_Toc477727345)

[**5.1 安装文件下载后点击无反应** 26](#_Toc477727346)

[**5.3使用IDLE编程计算时出现问题** 26](#_Toc477727347)

[**第六章 小结** 27](#_Toc477727348)

[**参考文献** 28](#_Toc477727349)

**第一章 Python基本开发环境和程序示例**

**1.1 Python语言介绍**

Python，是一种面向对象、直译式的计算机程序设计语言，也是一种功能强大的通用型语言，已经具有近二十年的发展历史，成熟且稳定。它包含了一组完善而且容易理解的标准库，能够轻松完成很多常见的任务。Python的语法非常简捷和清晰，与其它计算机程序设计语言最大的不同在于，它采用缩进来定义语句块。

Python的官方介绍是：Python是一种简单易学，功能强大的编程语言，它有高效率的高层数据结构，能简单而有效地实现面向对象编程。Python简洁的语法和对动态输入的支持，再加上解释性语言的本质，使得它在大多数平台上的很多领域都是一个理想的脚本语言，特别适用于快速的应用程序开发。

Python支持命令式编程、面向对象程序设计、函数式编程、面向切面编程、泛型编程多种编程范式。与Scheme、Ruby、Perl、Tcl等动态语言一样，Python具备垃圾自动回收功能，能够自动管理内存使用。Python经常被当作脚本语言用于处理系统管理任务和Web编程，然而它也非常适合完成各种高阶任务。Python虚拟机本身几乎可以在所有的操作系统中运行。使用一些诸如py2exe、PyPy、PyInstaller之类的工具可以将Python源代码转换成可以脱离Python解释器执行的程序。

Python的主要参考实现是CPython，它是一个由社区驱动的自由软件。目前由Python软件基金会管理。基于这种语言的相关技术正在飞速的发展，用户数量急剧扩大，相关的资源非常多。

**1.2 Python安装**

Python目前版本已非常丰富，各个版本的安装方法、步骤都大致相同。以下内容以个人电脑安装为例，进行介绍。由于个人电脑为windows7操作系统，32位。考虑到兼容性等问题，选择安装版本为Python3.5.0.以下安装步骤以Python3.5.0为例进行相关介绍。

**1.2.1 Python的下载**

进入Python的官方网站（<http://www.python.org>）。点击Downloads，进入下载页面，如下所示，在这里可以进行相关版本的选择：

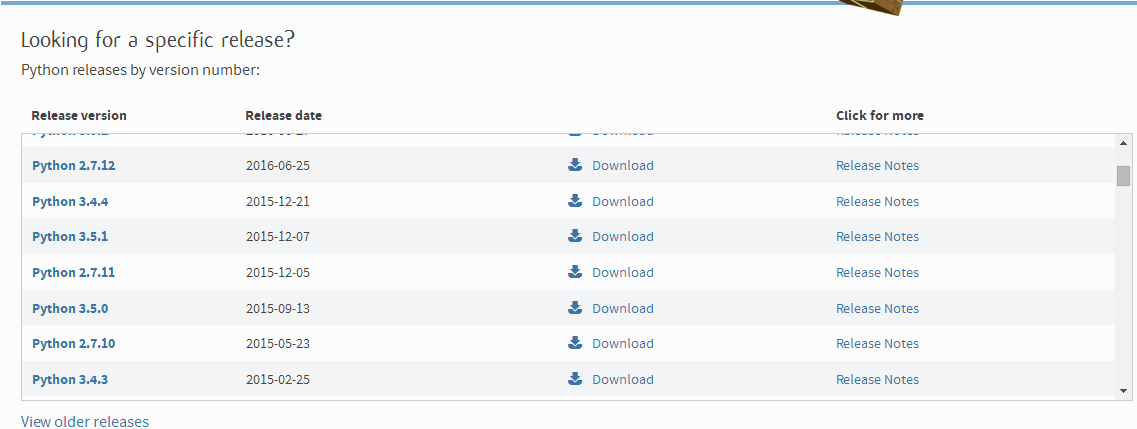


图1版本选择网页

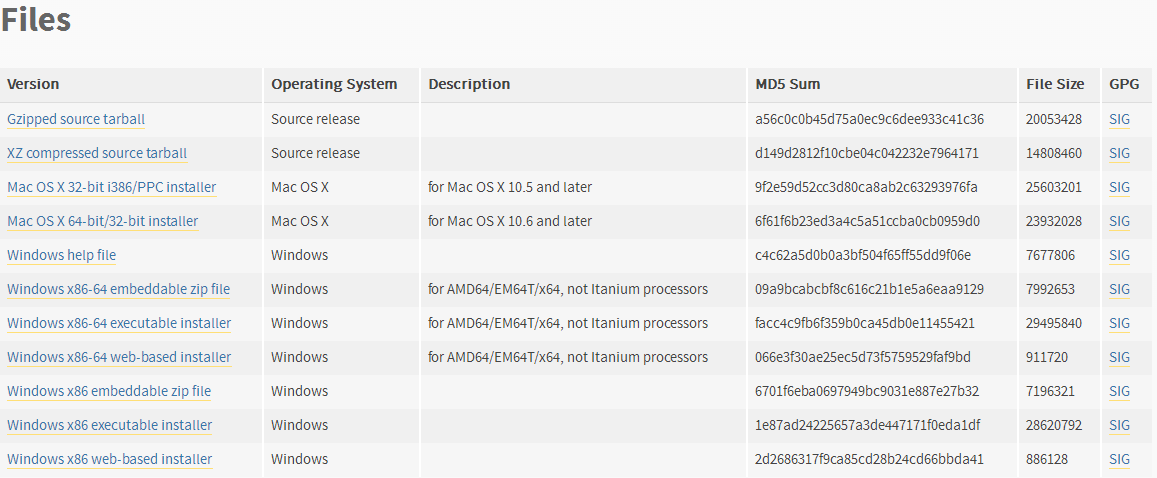
结合个人电脑环境，选择版本3.5.0，点击后可进入如下的下载页面：

图2下载与电脑操作环境相符的文件

选择，即可下载安装版本到电脑本地。

**1.2.2 Python的安装**

在电脑本地找到下载好的安装文件，使用管理员身份运行安装软件，进入如下图所示的安装界面：



图3安装开始界面

此步中要注意勾选“AddPython3.5to Path” 将Python3.5的安装目录添加到Windows系统的环境变量PATH中，接下来，就可以点击“Customize installation”进入安装过程（若选择“Install Now”，则会在默认路径安装，为指定安装路径，选择“Customize installation”）进入下一步。

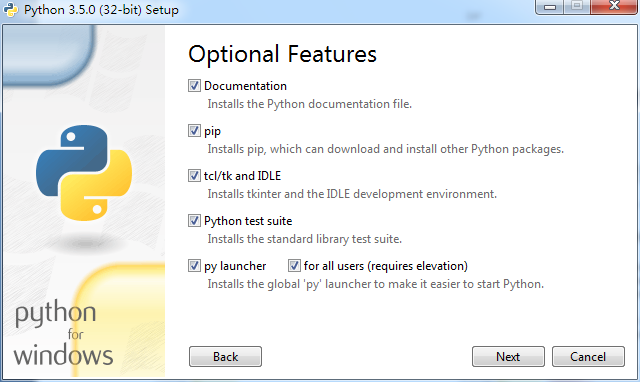


图4安装过程界面

此步操作保持默认选项，按下“Next”键，进入下一步。

此步中，勾选“Advanced options”项目中除“Install debug binaries”以外的所有项目，并更改安装路径为：C:\Python 35

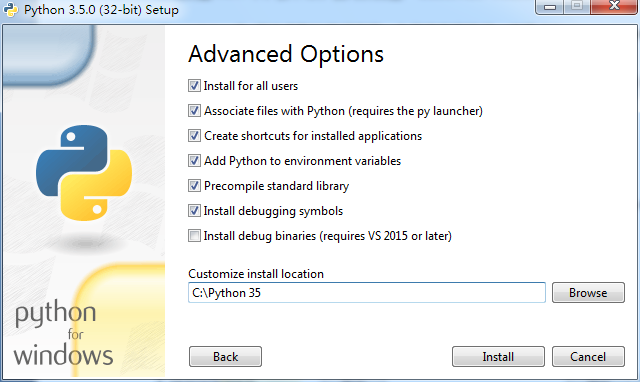


图5安装过程界面

点击“Install”进入安装过程：

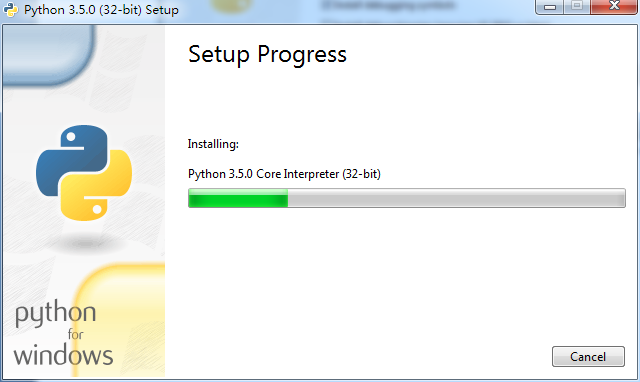


图6安装过程界面

在安装过程中可能会出现进度条卡顿的现象，在保证网络环境良好的条件下，耐心等待即可，最终可成功安装：

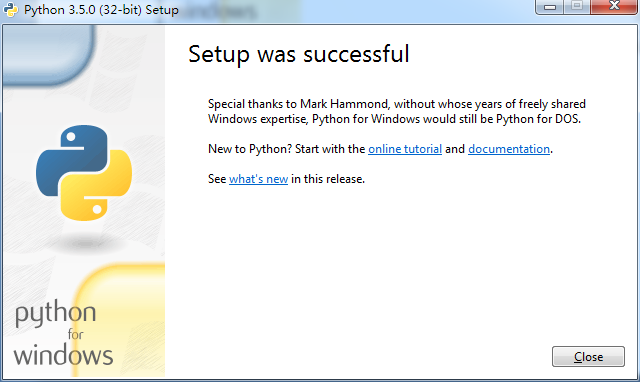


图7安装成功提示界面

在安装路径中，选择如下图所选文件固定到任务栏：

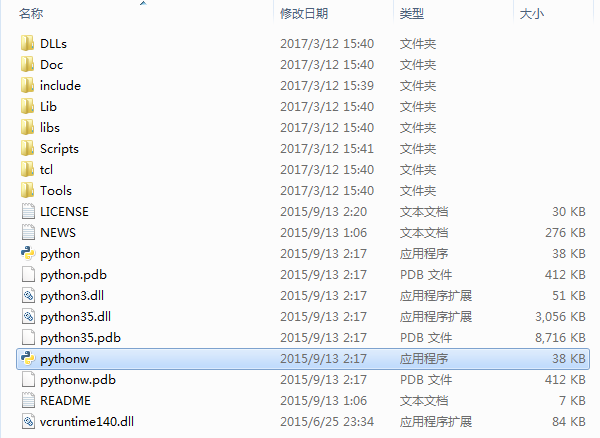


图8固定到任务栏

至此，完成了Python安装的全部过程。

**1.3 Python程序示例（使用IDLE编程）**

IDLE是Python软件包提供的轻量级集成编程环境，通过IDLE可进行简单程序的编写，举例如下：

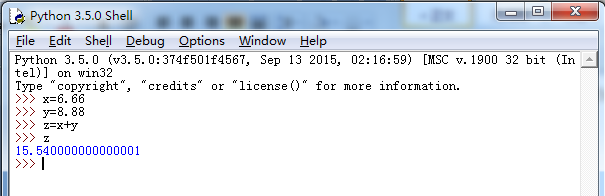


图9使用IDLE编程

结果中显示的不是15.54，而是十分接近15.54.这是由于计算机内部通常采用二进制的原因。在大多数现代编程语言中，除了整数，其他数都表示为浮点数。可以用一对整数来表示数字：有效数字和指数。有效数字的位数决定了可以表示的数字的精度。由于有效位数总是有限的，所以用有限位的二进制数表示十进制小数，总会存在一定的差别，不过当有效位数较多时，由此带来的差别对计算结果几乎不会带来影响，不影响正常的使用。

同时，命令提示符中也可以直接运行Python，这种输入命令的模式叫交互模式，只能直接执行输入的程序，结果也直接输出。也有另一种命令行模式，就是编写一个后缀为“.py”的文件，然后运行，只有当文件里有print才有输出。在Python的交互式命令行写程序，好处是一下就能得到结果，缺点是没法保存，所以，实际开发的时候，人们总是使用一个文本编辑器来写代码，写完保存为一个文件，程序就可以反复运行了。

**第二章 Python扩展软件包安装**

**2.1 在线检查、更新pip到最新版本**

pip是在线安装Python软件包的软件。安装好Python后，DOS命令行下,在线检查、更新pip到最新版本（电脑联网）

打开命令提示符，输入指令 >python –m pip install –U pip。检查是否安装成功以及更新pip到最新版本。pip是在线安装Python软件包的软件，发现是否有可更新版本，如有，则会进一步安装，如无，则会给出相应的提示信息。

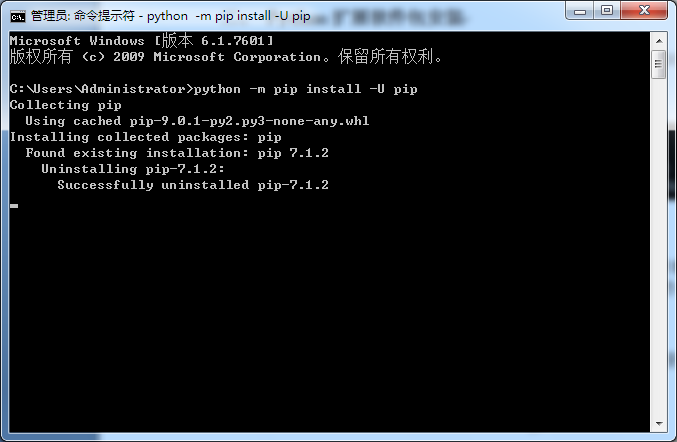


图10 pip版本更新

当当前版本或更新完毕后版本满足要求时，输入相同指令，会得到如下的提示窗口：



图11 pip版本检测

**2.2 编码规范**

好的软件源码需遵循一定的约定，Python语言将这些约定总结为PEP8，可以使用软件包，检查修改源码，使其满足PEP8的要求，提高源码质量。

在DOS命令行环境下，输入如下指令：

>pip installautopep8

保证网络环境良好，即可进行安装，安装时，界面如下：

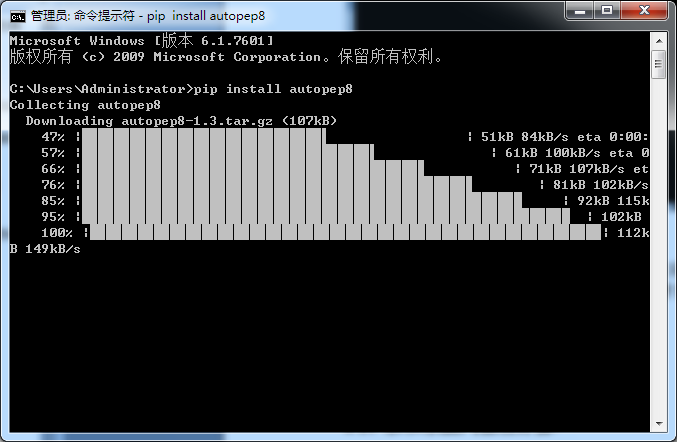


图12安装过程

经过一段时间后，即可安装成功，此时会给出相应的提示信息，以及安装的版本号信息，如本人电脑安装成功后的提示界面如下图所示：

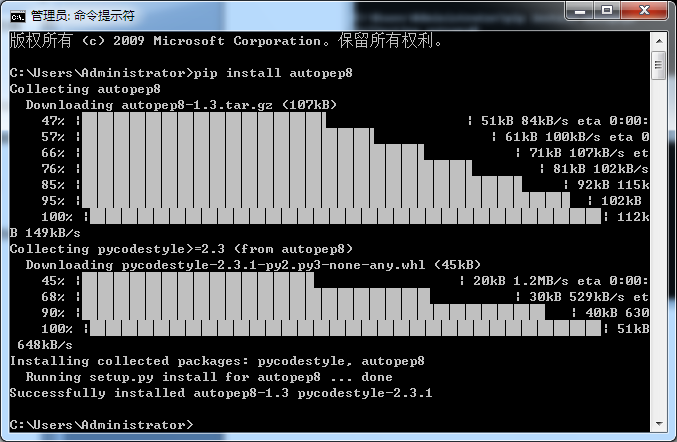


图13安装成功提示界面

此步完成后，在DOS命令行环境下，输入如下指令：

>pip installpylint

进行下一步安装

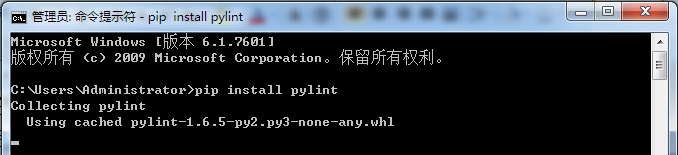


图14安装过程

安装完成后，会给出相应的提示信息如下：

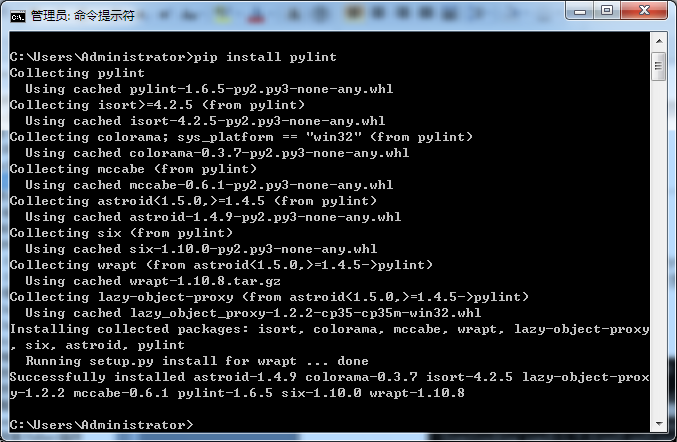


图15安装成功提示界面

**2.3 科学计算软件包**

Scipy官网：<http://www.scipy.org/>,提供有全面的科学计算软件包。基本应用安装numpy、scipy、matplotlib在3个软件包即可。

Scipy 软件包依赖一些C/Fortran软件库。在线安装时，会下载这些库的源码到本地编译，但Windows下通常缺少这些源码的本地编译环境，因此，Windows下安装Scipy 软件包，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：。

通过网址<http://www.lfd.uci.edu/~gohlke/pythonlibs/>进入加州大学欧文分校的编译版软件包下载网站，其网站如下图所示，从中可以找到所需要的安装的软件包numpy、scipy、matplotlib，点开，即可进入相应下载的地方

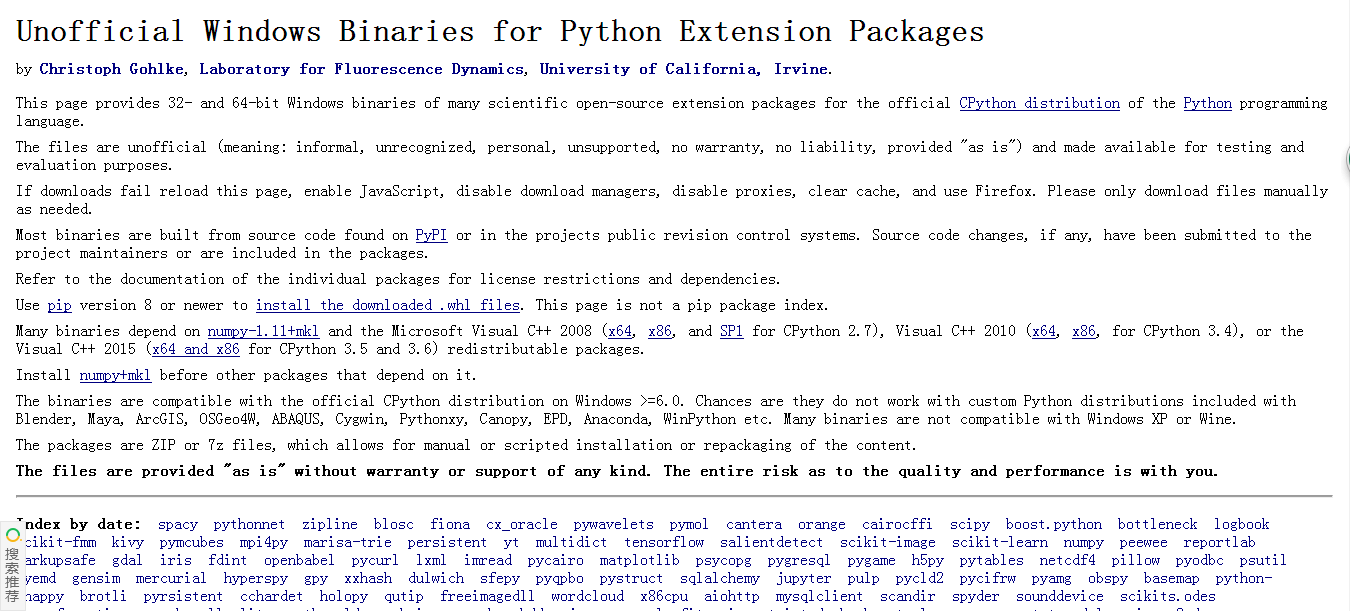


图16科学计算软件包下载网页截图

以numpy为例，点开，即进入如下的下载页面，可见提供了多样的下载版本。其中第一个数表示NumPy的版本，mkl是科学计算用的，cp35对于的是python3.5大的版本，最后是指系统。一定选择和电脑以及python版本适配的进行下载。

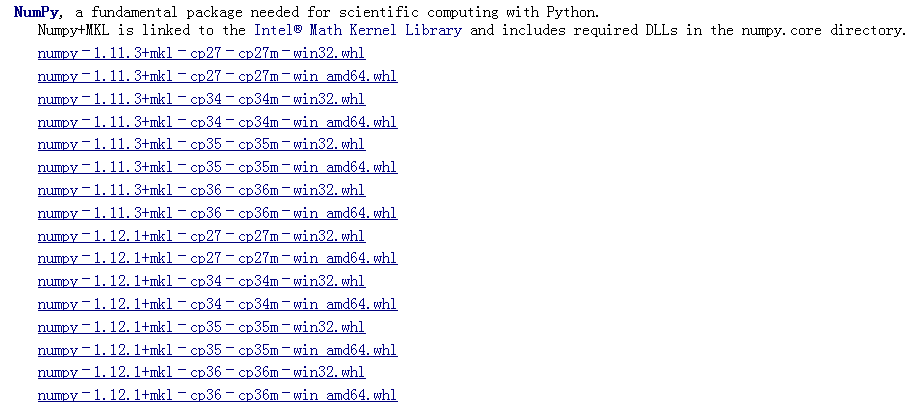


图17numpy版本截图

打开命令提示符，输入 >pip install\*.whl。\*用扩展包名字替换，如：>pip install numpy-1.12.1+mkl-cp35-cp35m-win32.whl，即可进行安装。

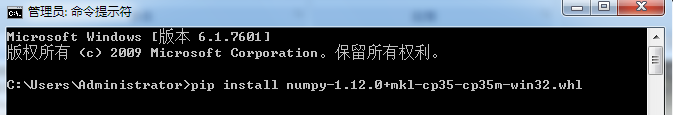


图18科学计算软件包numpy的安装



图19科学计算软件包numpy的安装

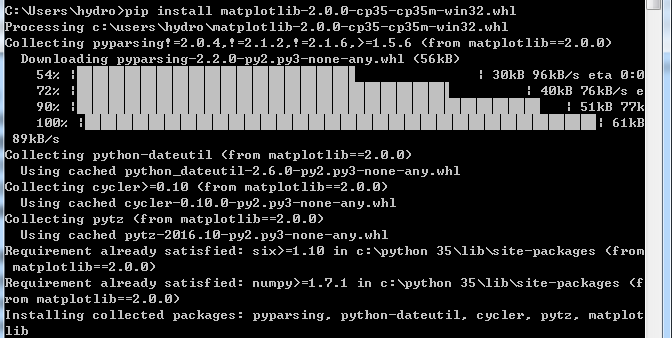


图20科学计算软件包matplotlib的安装

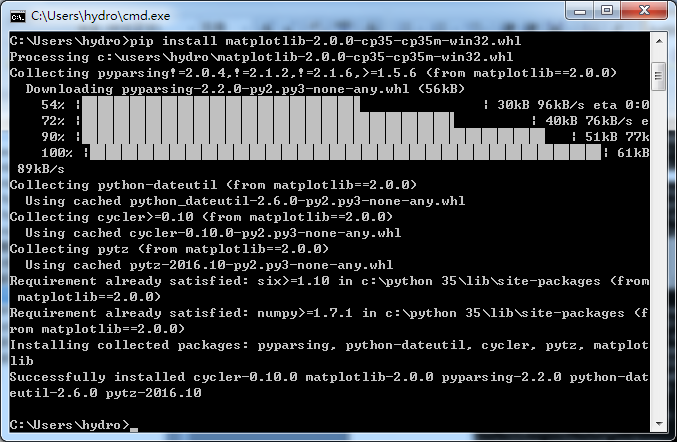


图21科学计算软件包matplotlib的安装

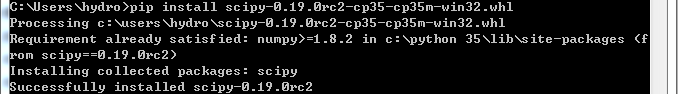


图22科学计算软件包scipy的安装

**2.4 IF97水和水蒸汽物性计算**

Python语言计算包，在线安装：>pip installiapws

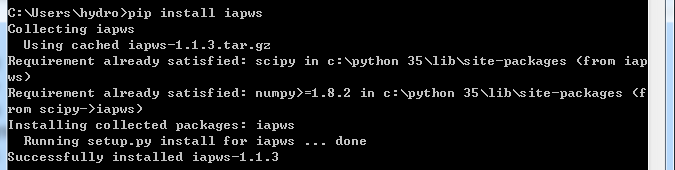


图23科学计算软件包iapws的安装

**第三章 交互计算环境和程序示例**

**3.1 安装Jupyter环境包**

Jupyter Notebook是一个交互式开发环境，可以在安装Python的基础上直接用命令提示符进行安装。只要联网，然后输入命令>pip install jupyter即可安装。

在线安装Jupyter ：

>pip installjupyter

Jupyter安装过程中，需要安装很多软件包，易出现超时现象，耐心等待。此时的DOS环境界面如下图所示：

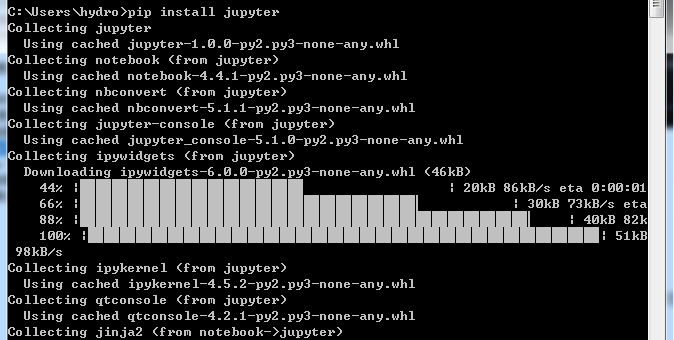


图24Jupyter的安装

经过一段时间后，即可安装成功，会给出相应的提示及安装信息如下图所示：

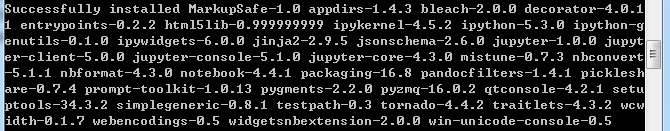


图25Jupyter的安装

**3.2 创建Jupyter Notebook文件**

可以自己建立一个文件夹，作为Jupyter Notebook的工作空间。在文件夹内新建一个记事本文件，输入jupyter notebook，然后另存为“.bat”的所有文件格式，双击即可运行，在这个工作目录下，可以新建自己的文件（虽然是用网页模式打开，但实际上还是一个本地的文件），在这个文件里可以看见一行行可以输入的空间，也称为cell。即使同一个文件里的每一个cell的格式也是可以不一样的，比如示例中既可以有markdown的代码，也可以有python的代码，并且都可以独自编译。

除了直接编写代码，也可以在这个文件中导入其他的代码。输入 %load Mycode.py 就可以直接把Mycode.py里的代码贴进一个cell，这一句命令也自动转为注释，再一次运行是就可以看到这一个代码文件运行的结果，十分方便。导入网上的文件也同理，具体操作见网站 (<http://blog.csdn.net/tina_ttl/article/details/51031113>)。

老师的课件也可以用Jupyter Notebook打开。首先找到notebook 下的StartNB.bat文件，双击运行，即可用Jupyter Notebook打开Home的目录。也可以通过命令提示符，cd到课件所在文件夹Home，在该目录下运行Jupyter Notebook。

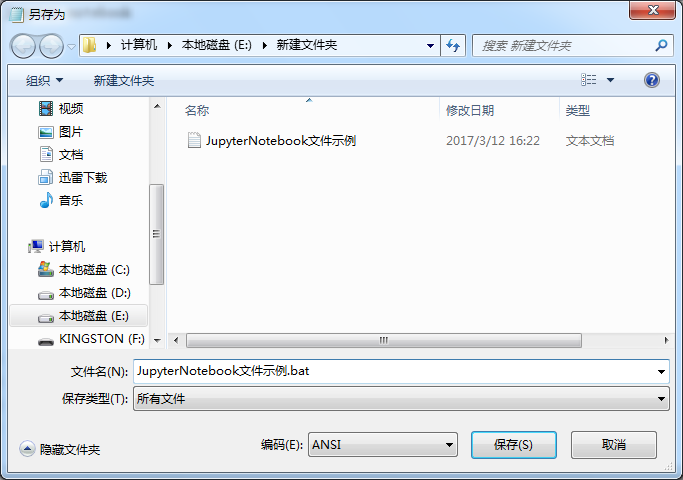


图26创建Jupyter Notebook文件



图27创建Jupyter Notebook文件

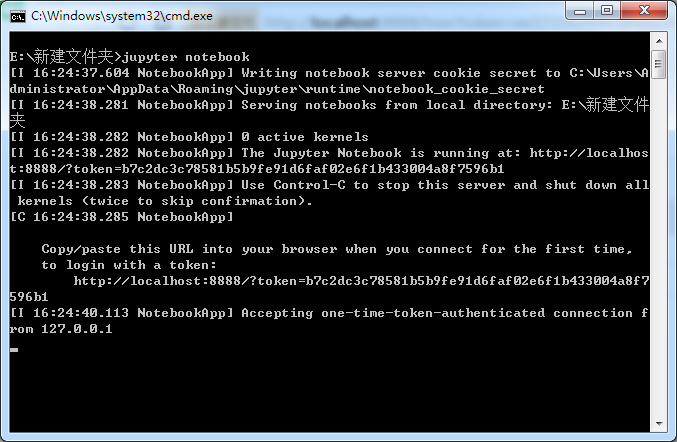


图28创建Jupyter Notebook文件

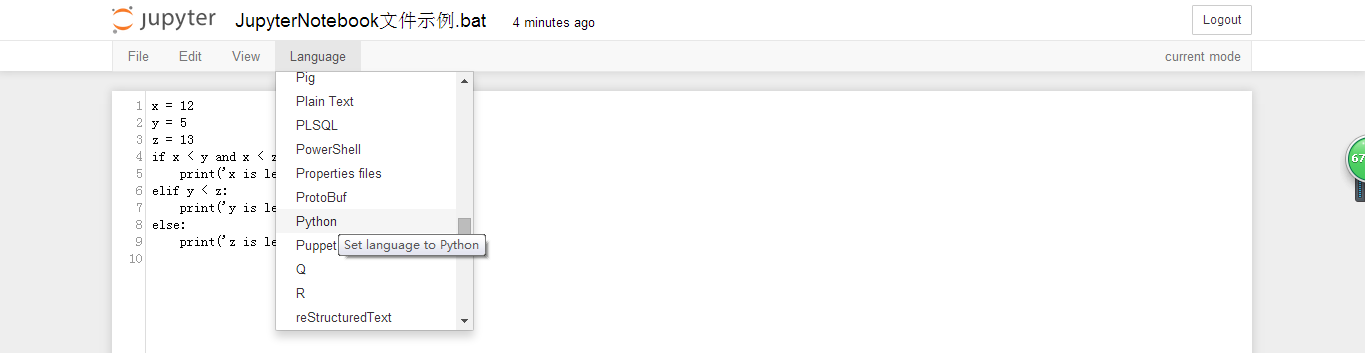


图29创建Jupyter Notebook文件

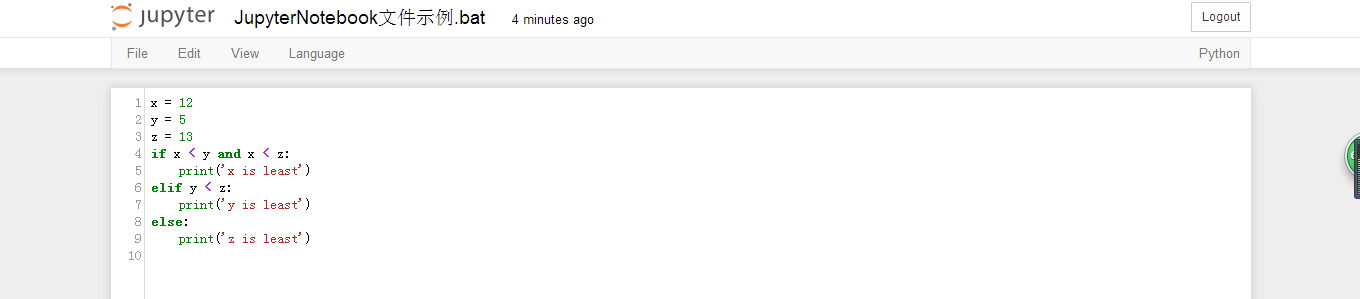


图30创建Jupyter Notebook文件

**第四章 集成开发环境和程序示例**

**4.1 安装Java SDK并配置环境变量**

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 ,会显示有关版本信息。如果没有安装Java包或者版本不能满足Eclipse IDE最低要求，需要从Oracle下载Java包。Java JDK的Oracle官方下载地址为：

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

安装完成后，需要配置java环境变量，在我的电脑上右击-属性-高级系统设置，在弹出的窗口中点击右下角的“环境变量”按钮，接下来可以在用户环境变量或者系统环境变量中配置具体的环境变量，新建名为“classpath”的变量名，值为“.;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\tools.jar 。同样，新建名为”JAVA\_HOME“的变量名，变量值为之前安装jdk的目录。在已有的系统变量“path“的变量值加上”;%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin“，自此配置完成。

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 ,会显示有关版本信息：



图31 Java版本检测

如果没有安装Java包 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>

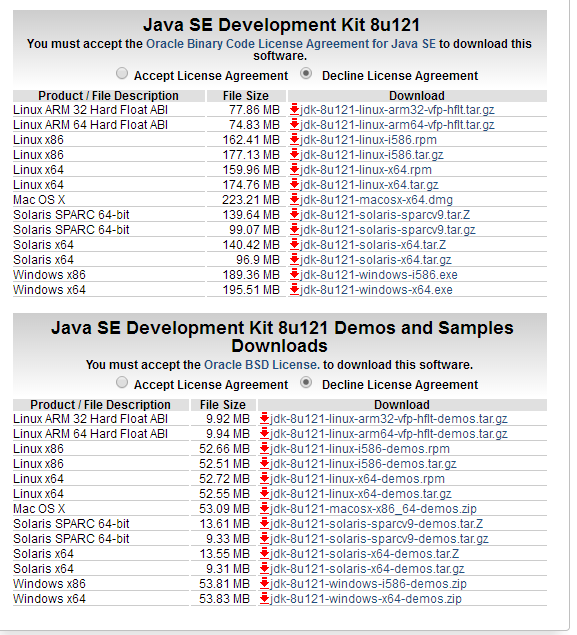


图32Java版本选择

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.

**4.2 Eclipse安装过程**

Eclipse IDE有很多版本。本课程使用Python语言，可能会涉及C/C++开发，所以下载Eclipse CDT(Eclipse IDE for C/C++)版。Eclipse CDT官方下载地址：

http://www.eclipse.org/downloads/eclipse-packages/

下载之后直接解压到指定目录，运行eclipse.exe即可。首次打开Eclipse软件时,软件会提示设置默认的工作空间。在非系统盘中，如D盘中建立一个目录，然后打开Eclipse，在其提示设置默认的工作空间时，“Browser”到该目录，勾选“Use this as the default and do not ask again",就将该目录配置为Eclipse默认当前工作空间目录了。

可以根据工作需要建立多个工作空间目录，通过“File->Switch Workspace->other”改变当前工作空间目录。

首次打开Eclipse软件时,软件会提示设置默认的工作空间（就是存放所有开发项目信息的目录）。

注意：不要使用Eclipse软件默认的工作空间目录。

建议配置自己的工作空间：首先，在非系统盘中，如D盘中建立一个目录，如:D:\myworkspace，然后，打开Eclipse，在其提示设置默认的工作空间时，“Browser”到D:\myworkspace，勾选“Use this as the default and do not ask again",就将该目录配置为Eclipse默认当前工作空间目录了，你开发的程序代码等都在这个目录下。

使用Eclipse IDE作为Python开发环境，需要：1）安装PyDev插件；2）配置使用的Python解释器版本。

**4.3 Python开发插件PyDev**

**4.3.1安装PyDev插件**

通过Help->Eclipse Marketplaces进入市场，输入pydev，找到Pydev项目，点“install”在线安装。

在线安装过程简单，但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.

正常情况，Eclipse会自动解压，安装好插件。如解压中异常，可手工解压插件包，将解压后的所有文件，拷贝到Eclipse的插件目录plugins下。

个人在按装过程中采用了较为简单方便的在线安装方式。



图33安装PyDev插件

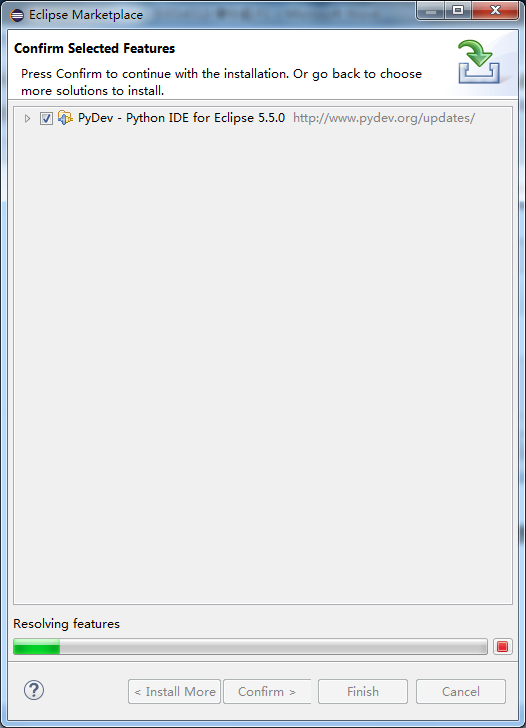


图34安装PyDev插件

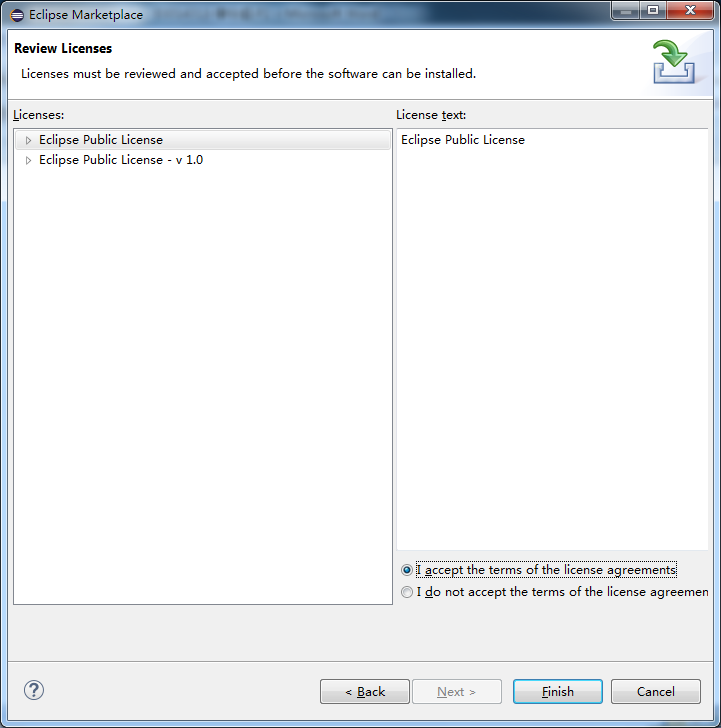


图35安装PyDev插件

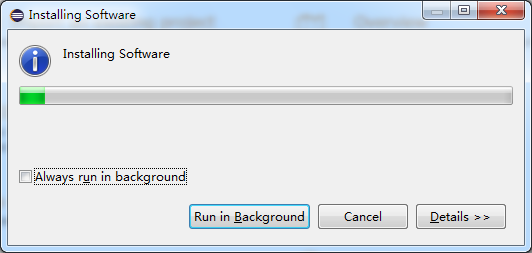


图36安装PyDev插件

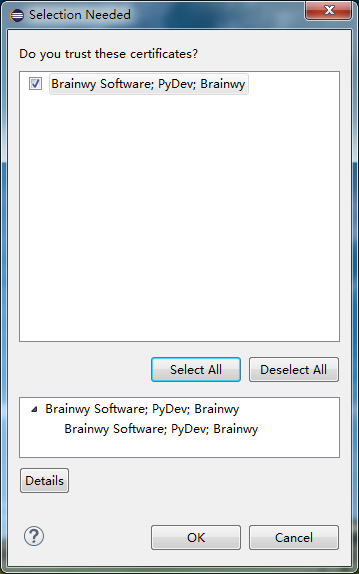


图37安装PyDev插件

**4.3.2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:

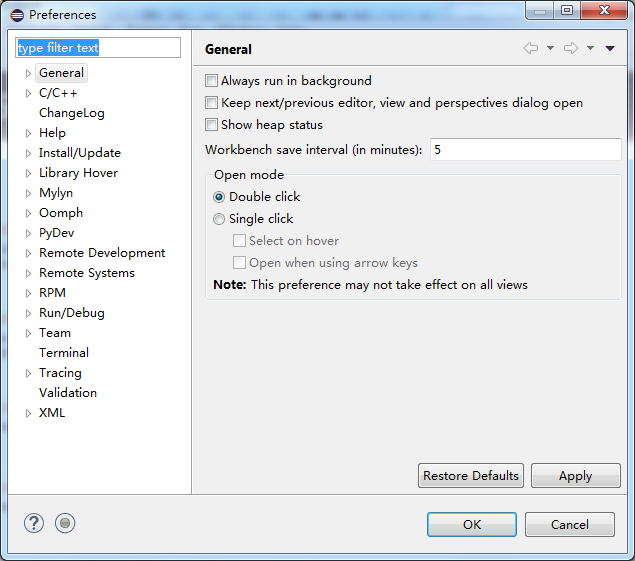


图38配置 Python解释器

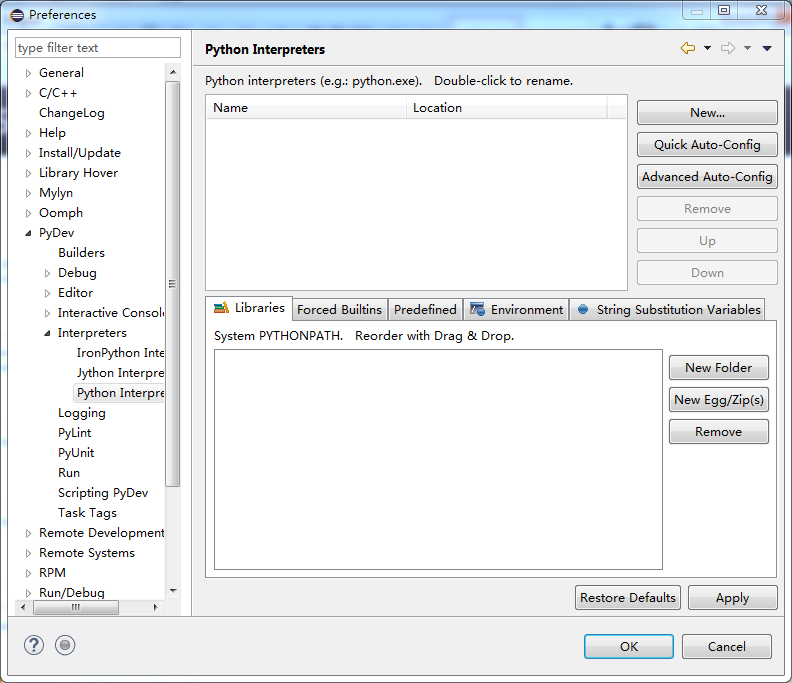


图39配置 Python解释器

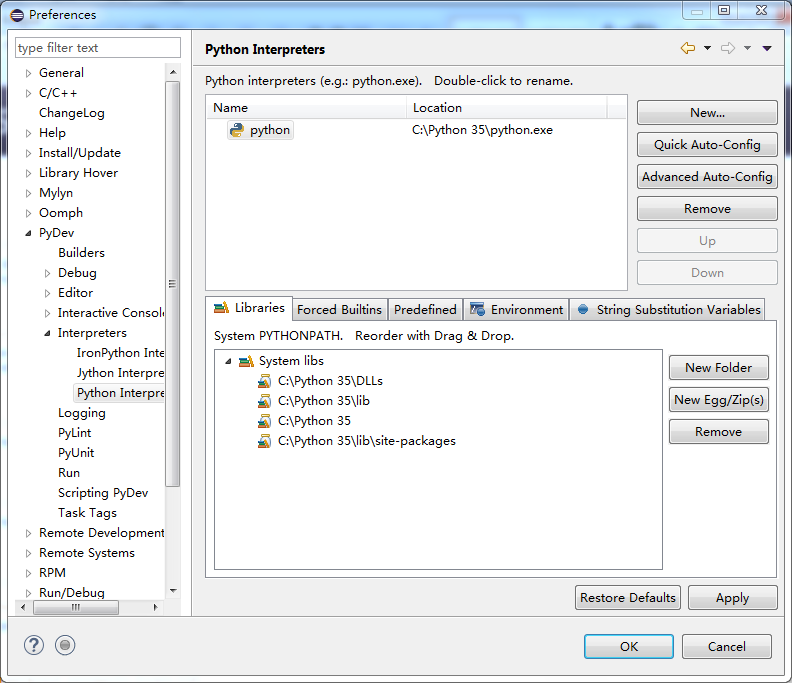


图40配置 Python解释器

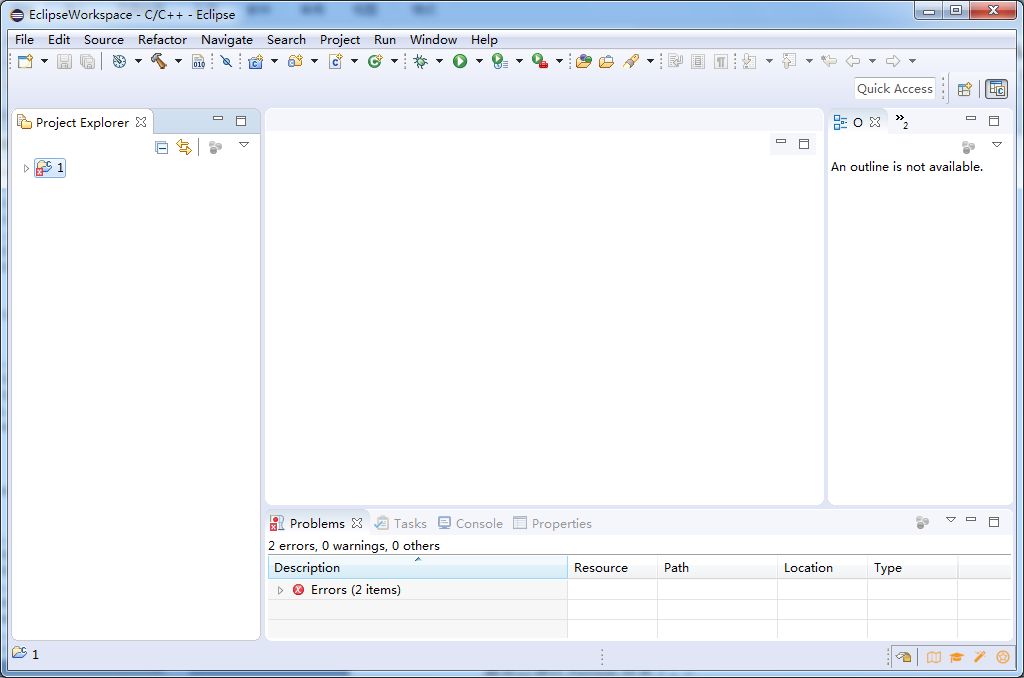


图41配置 Python解释器

配置好后，添加Python场景到工作页面：

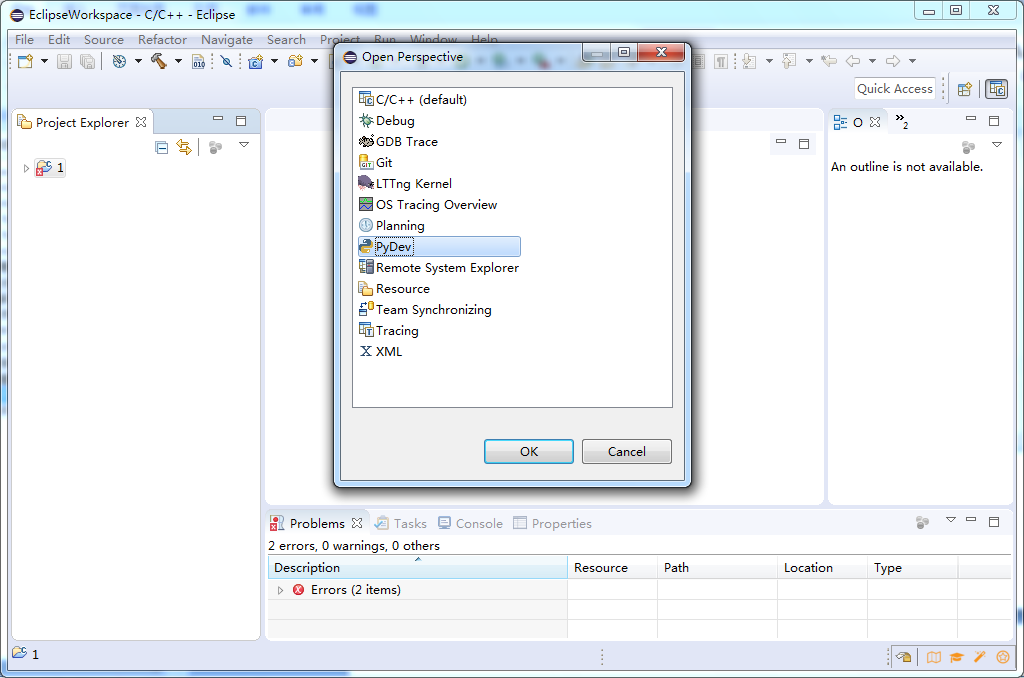


图42配置 Python解释器

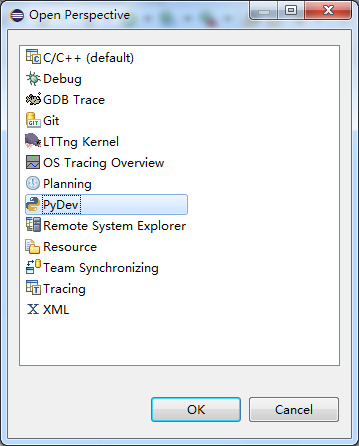


图43配置 Python解释器

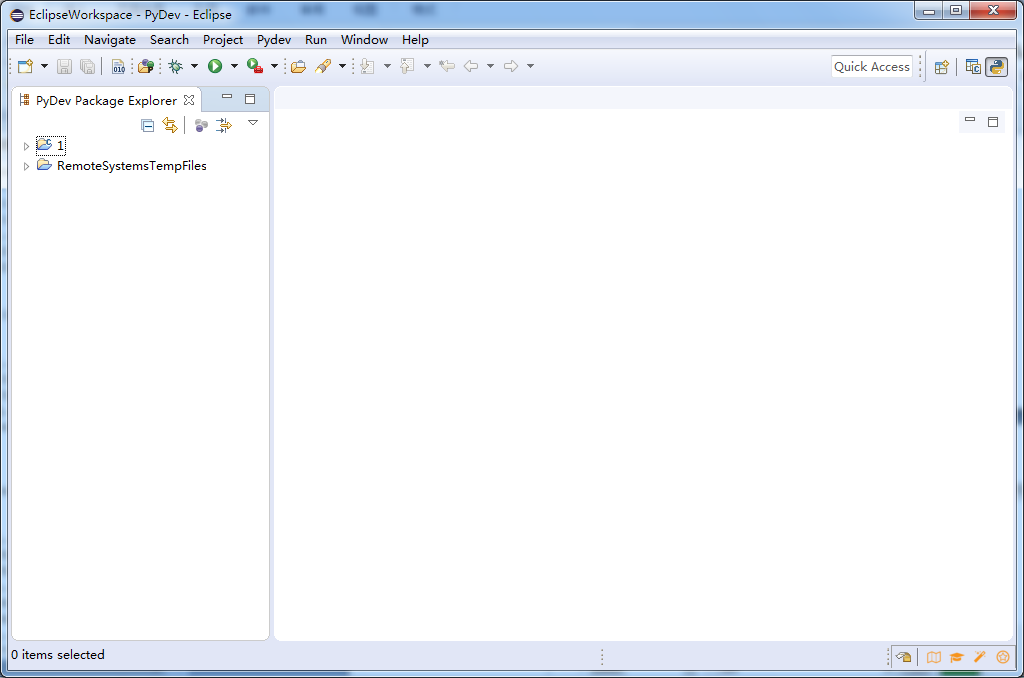


图44配置 Python解释器

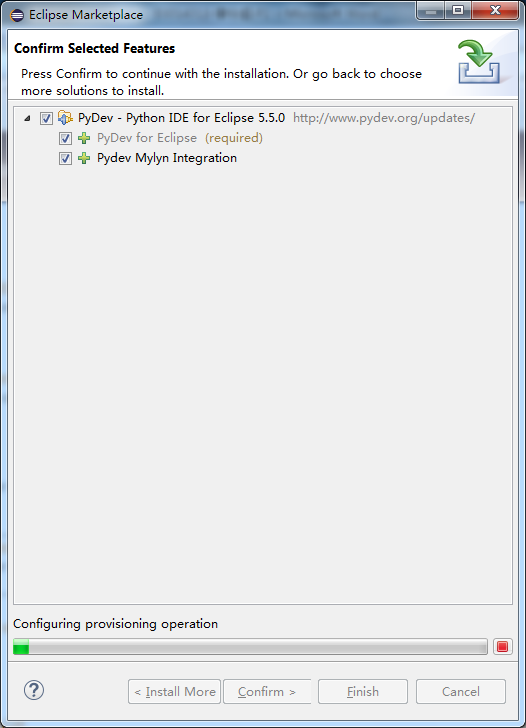


图45配置 Python解释器

**4.4 配置 PyDev提高效率**

**4.4.1修改编辑器配色提高可读行**

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

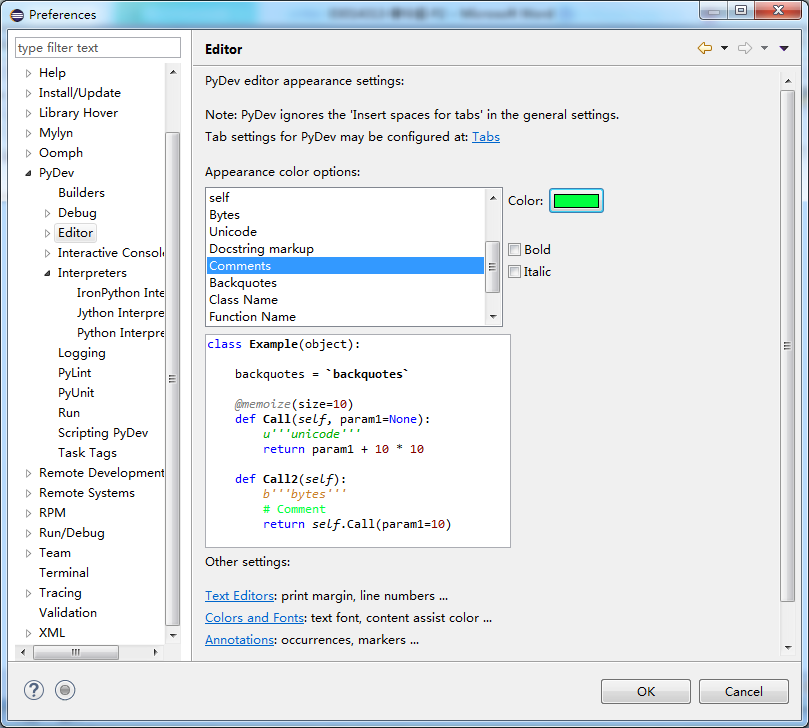


图46修改编辑器配色

**4.4.2配置任务标签：**

程序开发中，可在代码中标识当前任务状态，计划开发工作。使用“任务标签”在代码中标识任务，然后，让开发环境将其识别出来，加入工作空间的任务列表中。

通过PyDev->Task Tags中配置任务标签:

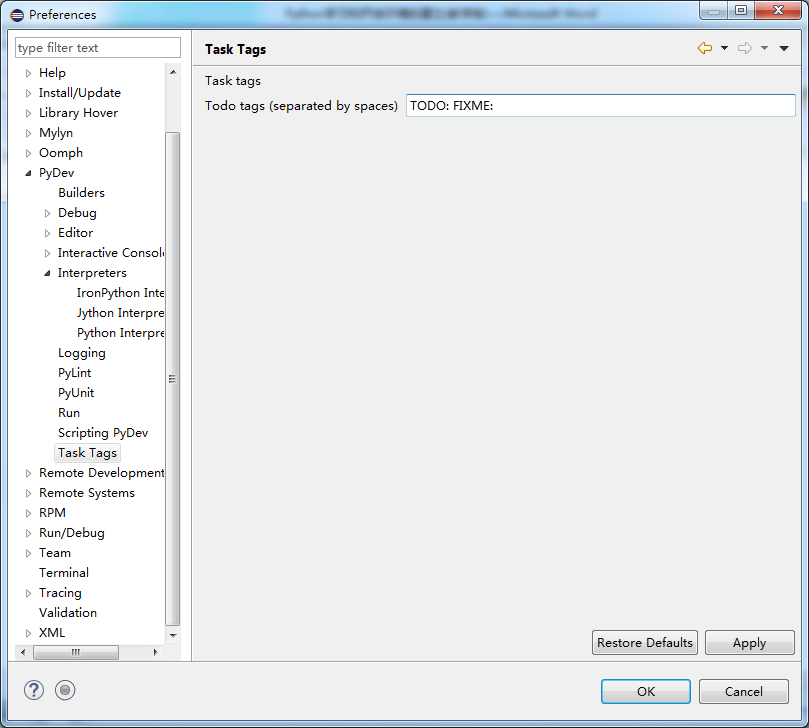


图47配置任务标签

可以根据不同的任务类型，自己增加自定义标签

**4.5 基于PyDev的语言规范静态检查**

PyDev中集成了PEP8、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，要有代码规范意识，但过分注意规范会影响开发进程。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，开发进程中默认关闭，在程序开发一个阶段结束时，开启规范性检查检查更好。

**4.6 Eclipse配置**

中文操作系统下，Eclipse默认工作空间编码方式为GBK, 这样的编码方式下，含中文字符的文件，在其他文本编辑器打开可能会乱码，因此，建议配置编码为UTF-8。

编码方式可以设定到：工作空间、工程和文件类型的不同等级的文件范围上。建议配置编码方式UTF-8到工作空间，工程继承工作空间配置的方式。

**4.6.1工作空间编码方式**

点“Window”->“Preference”，在弹出窗口中，点击“general”-“workspace”，修改“Textfileencoding”为UTF-8：

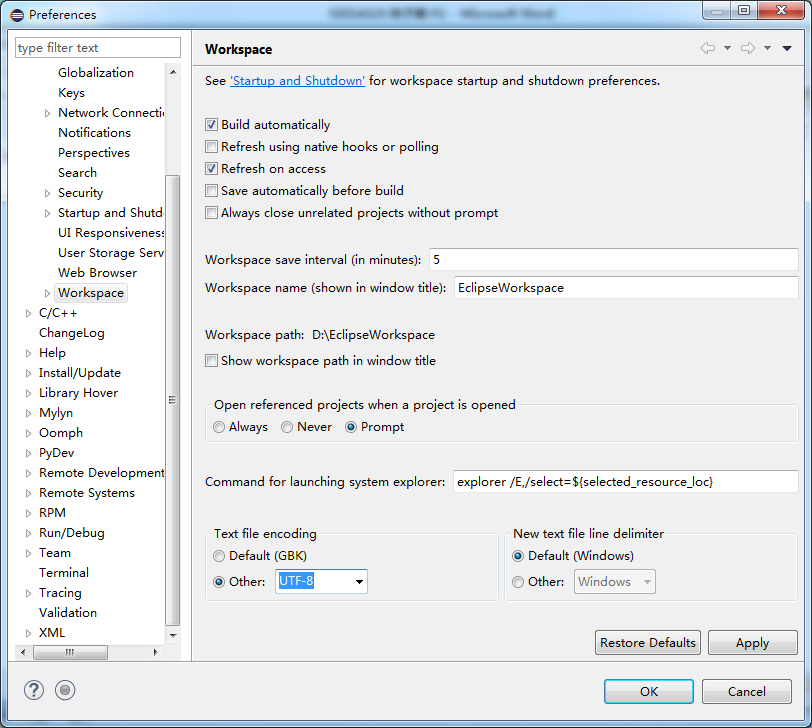


图48工作空间编码方式

**4.6.2 工程编码方式**

将鼠标移动到项目名上，点击右键，选择“properties”， 在弹出的对话框中 ，选中“resources”， 修改“Textfileencoding”为UTF-8(如果工作空间配置为UTF-8会继承过来)：

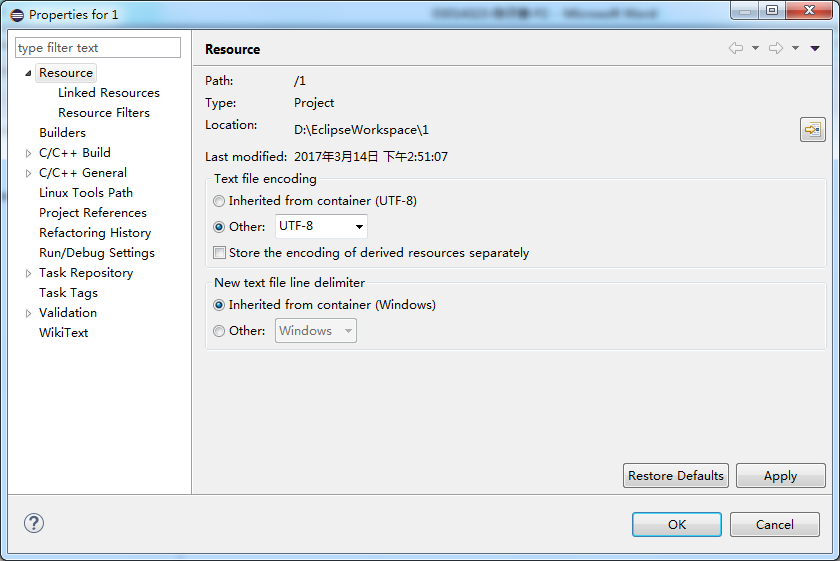


图49工程编码方式

**4.7 Eclipse使用示例**

首先创建PyDev项目，顺序创建PyDev Packages->PyDev Module.创建完成后，即可在.py文件内进行编程。

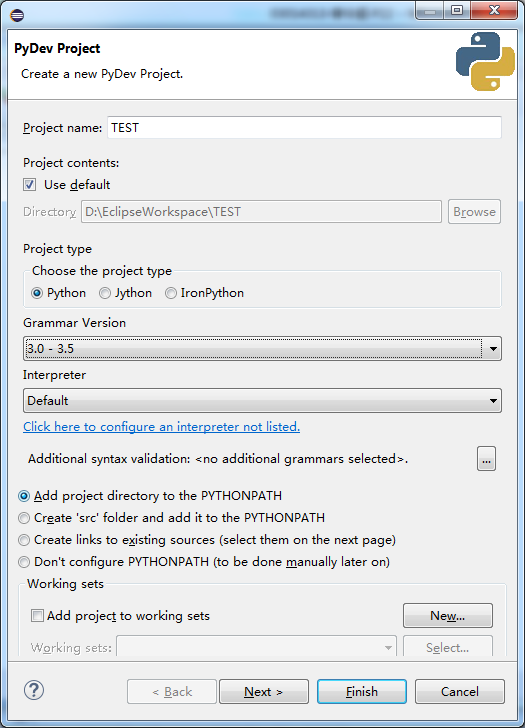


图 50 Eclipse使用示例：步骤一

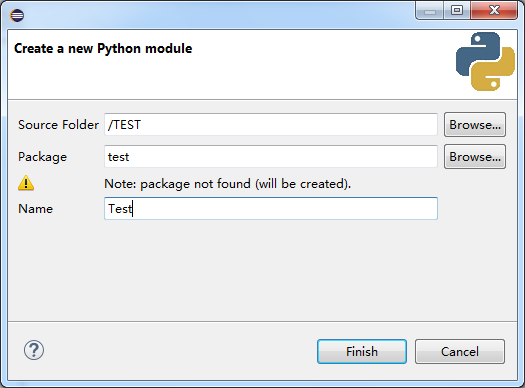


图 51 Eclipse使用示例：步骤二

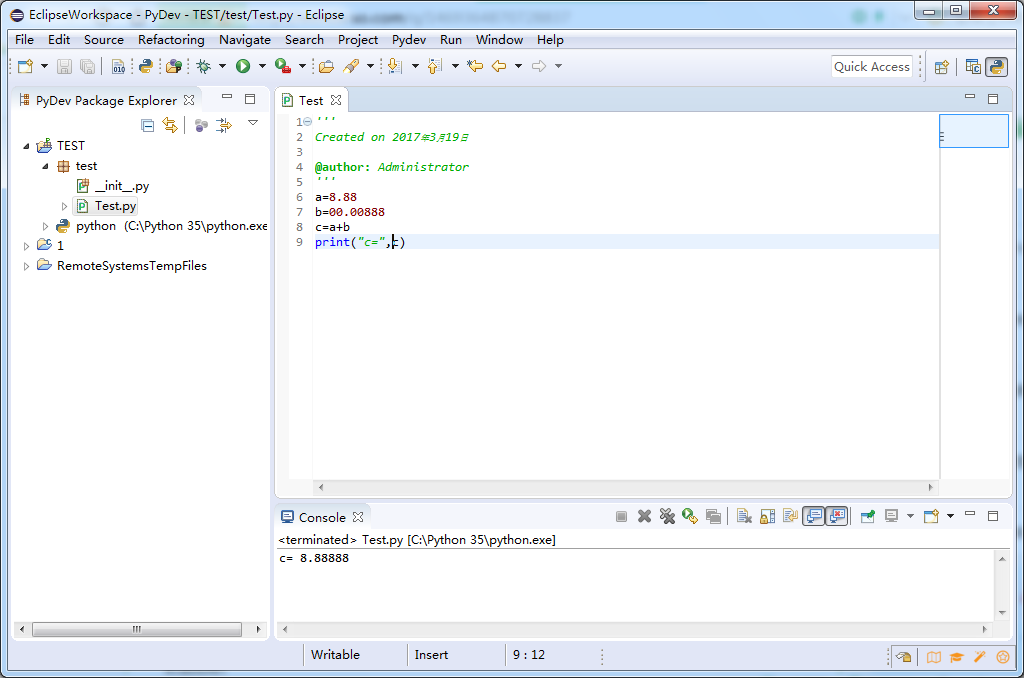


图 52 Eclipse使用示例：程序及其运行结果

图示程序为一个简单的求和程序

**第五章 建立开发环境等过程中出现的问题和解决方法**

**5.1 安装文件下载后点击无反应**

**问题描述**：以管理员身份运行已经下载好的安装文件没有反应。

**原因分析**：由于前一段时间电脑重新装了系统，没有搞清楚电脑的操作系统环境从64位变为了32位，没有下载正确的安装文件。

**解决方案**：从网站上重新下载与计算机操作系统环境相符的安装文件，再进行安装，即可解决问题。

**5.2 科学计算包安装过程出现问题**

**问题描述**：在科学计算安装包安装过程中出现中断，DOS命令行里显示红色提示字符。

**原因分析**：应该是由于科学计算包安装过程中网络出现问题造成的。

**解决方案**：在DOS命令行里重新输入指令再次执行即可，如果在此过程中出现相同问题，可在保证网络环境良好的条件下，再次输入指令执行，直至最终安装成功即可。

**5.3使用IDLE编程计算时出现问题**

**问题描述**：在IDLE编程计算6.66+8.88时，计算结果显示的并不是15.54（理论结果）。

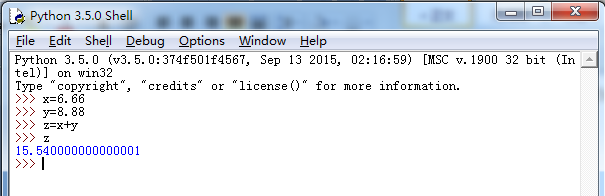


图 53 浮点数计算问题

**原因分析**：结果中显示的不是15.54，而是十分接近15.54.这是由于计算机内部通常采用二进制的原因。在大多数现代编程语言中，除了整数，其他数都表示为浮点数。可以用一对整数来表示数字：有效数字和指数。有效数字的位数决定了可以表示的数字的精度。由于有效位数总是有限的，所以用有限位的二进制数表示十进制小数，总会存在一定的差别，不过当有效位数较多时，由此带来的差别对计算结果几乎不会带来影响，不影响正常的使用。

**解决方案**：此种情况对于大多数情况不会带来较大计算误差。但对于判断两个数是否相等的情况，特别是浮点数，应该注意尽量避免直接用“==”判断，而通过判断两个数之差小于一个足够小的数字来判断。

**第六章 小结**

此次课程实践主要完成Python开发环境的建立，在完成过程中，持续时间较长，出现了一些问题，一方面是由于前一段时间电脑更换系统，对电脑的基本环境不是特别了解，另一方面对安装过程不是特别熟悉，不过在老师上课的指导，自己的不断琢磨以及身边同学的帮助下，最终还是完成了此次课程实践。

在软件安装过程中，持续时间很长，中间断断续续，出现了很多问题，有一部分的原因是由于在每一步的进行过程中，自己总是试图先能够利用已有环境进行一些相对复杂程序的编写，搞清楚这一步到底是为了以后的哪些工作做准备、做铺垫。例如，在Python基本开发环境建立后，就想通过编写一些相对复杂，而又不涉及扩展软件包的程序而达到对其基本编程语言熟练掌握的目的；在科学软件包安装之后，就想通过将其较好地应用到相对复杂一些的程序中去，达到掌握以及熟练应用的目的；在集成开发环境建立过程中，总想能够搞清楚安装过程中的每一步对将来的程序开发到底会带来哪些不同的效果……最终，很多问题还是没能够得到很好地解决。很多步骤只能根据老师提供的讲义，同学们的指导，示例来按部就班的完成。总之，自己还是不太适应这种在正式学习一门计算机语言之前就进行大量的软件安装及设置的方法，还是希望将开始能够在最基本的开发环境中熟练掌握其基本编程语言后，再进行软件的进一步安装及拓展，循序渐进。也许这是自己将来要注重培养的能力之一。

相对于第一次课程时间，由于电脑原因，只能在截止日期之前，临时抱佛脚，草草完成上交相比，这一次花费了一定的时间在上面，中间虽然出现了一定的问题，但最终还是得到了一定的解决。

在这之前，自己对Python并不是特别了解，应该说是一无所知。但通过老师的介绍，相关书籍的阅读以及网上的资源，自己对Python这门语言也有了一定程度的了解。由于Python语言的简洁性、易读性以及可扩展性，在国外用Python做科学计算的研究机构日益增多，其在科研工作中的重要性不言而喻。对于本专业的学生，能够熟练掌握和应用这门语言是很必要的。

**参考文献**

1. Python学习和开发环境的建立(教学版).docx

<https://github.com/PySEE/RecommendedPracticestree/S2016/P1>

1. Eclipse 4.3 (Kepler) for C/C++ Programming

<http://www3.ntu.edu.sg/home/ehchua/programming/howto/EclipseCpp_HowTo.html>

1. 为什么使用jupyter？，知乎

https://www.zhihu.com/question/37490497

1. Python 基础教程

http://www.runoob.com/python/python-tutorial.html

1. 廖雪峰的官方网站-Python教程

<http://www.liaoxuefeng.com/wiki/0014316089557264a6b348958f449949df42a6d3a2e542c000>