2017

**软件工程基础及实践**

课程实践作业二

吉珣碧 03014402 能源与环境学院

**目录**

[1 Python及其开发环境简介 2](#_Toc476247879)

[2 Python语言 3](#_Toc476247880)

[2.1 Python3.5.3安装过程 3](#_Toc476247881)

[2.2 Python3.5.3使用示例 6](#_Toc476247882)

[3 安装扩展模块 7](#_Toc476247883)

[3.1 安装科学计算包 7](#_Toc476247884)

[3.2 安装规范格式扩展包 8](#_Toc476247885)

[3.3 IF97物性计算 9](#_Toc476247886)

[4 交互计算环境Jupyter Notebook 10](#_Toc476247887)

[4.1 Jupyter Notebook安装过程 10](#_Toc476247888)

[4.2 Jupyter Notebook使用示例 11](#_Toc476247889)

[5 集成开发环境Eclipse 12](#_Toc476247890)

[5.1 Eclipse安装过程 12](#_Toc476247891)

[5.1.1 安装Java SDK并配置环境变量 12](#_Toc476247892)

[5.1.2 安装Eclipse及配置工作空间 15](#_Toc476247893)

[5.1.3 安装PyDev插件及配置 16](#_Toc476247894)

[5.2 Eclipse使用示例 18](#_Toc476247895)

[6 问题及解决 21](#_Toc476247896)

[7 个人小结 23](#_Toc476247897)

[8 参考资料 24](#_Toc476247898)

**建立Python基本开发环境**

# 1 Python及其开发环境简介

全世界差不多有600多种编程语言，但流行的编程语言就20多种。其中Python就是一个十分流行的语言。相对于其他语言来说，Python以简介著称，也比较容易入门。许多大型网站就是用Python开发的，例如YouTube、Instagram，还有国内的豆瓣。很多大公司，包括Google、Yahoo等，甚至NASA（美国航空航天局）都大量地使用Python。

我们知道C语言是可以编写操作系统的语言，有些开发环境就是用它来编写，相对更贴近于硬件（也就是说比较低层，但是还是比汇编语言高级很多的），所以C语言适合开发那些追求运行速度、充分发挥硬件性能的程序。而Python是用来编写应用程序的高级编程语言。Python就为我们提供了非常完善的基础代码库，覆盖了网络、文件、GUI、数据库、文本等大量内容，被形象地称作“内置电池（batteries included）”。用Python开发，许多功能不必从零编写，可以直接使用现成的。除了内置的库外，Python还有大量的第三方库，也就是别人开发的，可以直接使用的东西。

Python的优点是简洁，适合开发网络应用、日常需要的小工具，也可以把其他语言开发的程序包装起来，应用广泛。但是所有语言都不是完美的，对于Python来说，第一个缺点就是运行速度慢，因为Python是解释型语言，代码在执行时需要翻译成CPU能理解的机器码，过程非常耗时。第二个缺点就是代码不能加密。如果要发布Python程序，实际上就是发布源代码，而C语言不用发布源代码，只需要把编译后的机器码（也就是常见的xxx.exe文件）发布出去。要从机器码反推出C代码是不可能的，所以，凡是编译型的语言，都没有这个问题，而解释型的语言，则必须把源码发布出去。

使用Python进行开发，首先要有开发环境，也就是有地方编写代码，并且让这个语言翻译成为机器可以识别的语言。首先需要安装的就是Python。有了这个之后，基本的环境就建立好了，IDLE是开发python程序的基本IDE (Integrated Development Environment)，已经是一个集成开发的环境，不过是一个很轻量级的工具，功能比较简单。为了更方便的使用，人们往往会添加许多扩展安装包，或者采用具有交互功能的开发环境和更加重量级的集成开发环境，比如Jupyter Notebook和Eclipse。

# 2 Python语言

## 2.1 Python3.5.3安装过程

首先从Python官网（<https://www.python.org/> ）到download区，找到Python3.5.3的版本，同一个版本由于系统和安装方式不同，会有很多个下载的链接，如图1。以64位的win10系统为例，应该下载Windows x86-64 executable installer。



图1 Python3.5.3下载区

双击运行安装软件，出现图2，选中Add Python 3.5 to PATH（若不选中，在命令提示符中就无法运行python，这是因为Windows会根据一个Path的环境变量设定的路径去查找python.exe，如果没找到，就会报错。如果在安装时漏掉了勾选Add Python 3.5 to PATH，那就要手动把python.exe所在的路径添加到Path中）[1]。

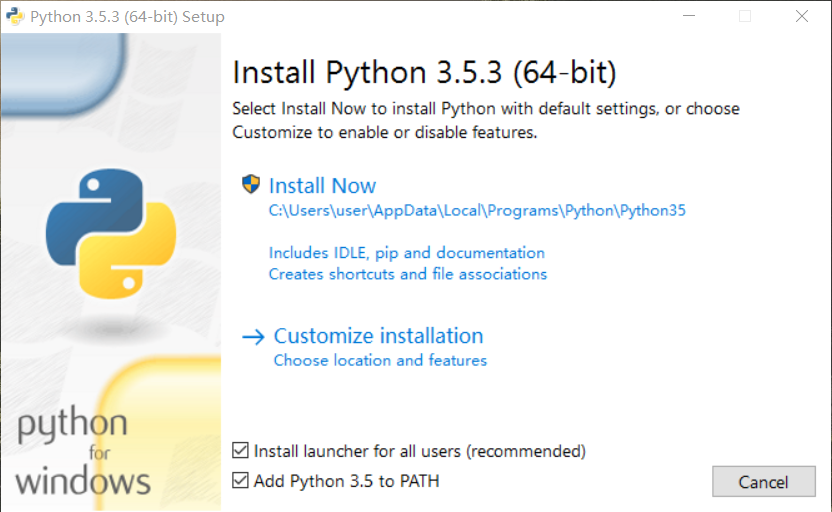


图2 运行Python3.5.3安装文件后出现的窗口

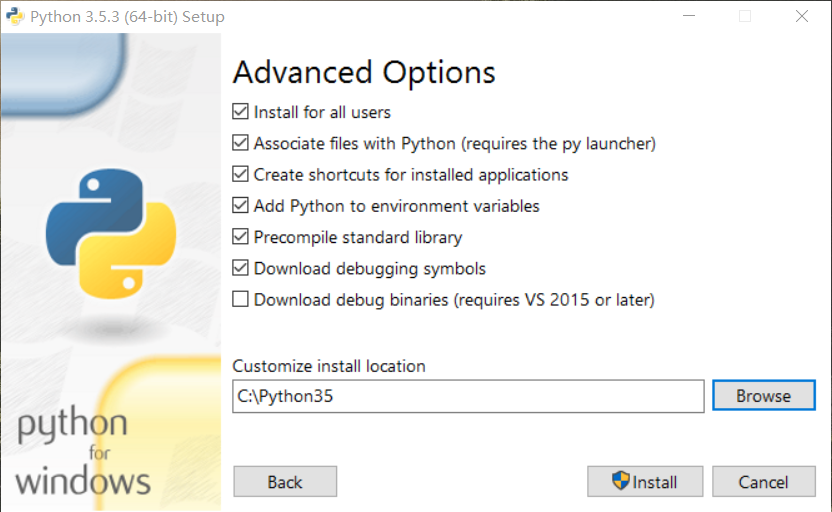


图3 Python3.5.3自定义安装窗口

点击Customize installation，选中上面6个选项，如果未安装vs2015则不要勾选最后一个选项，最后一个是指可以借助vs2015进行调试。自己定义安装文件夹，直接装在C盘而不是默认的文件夹下，便于以后查找。

开始安装，进度条可能停止不动，最终安装成功出现如下界面：

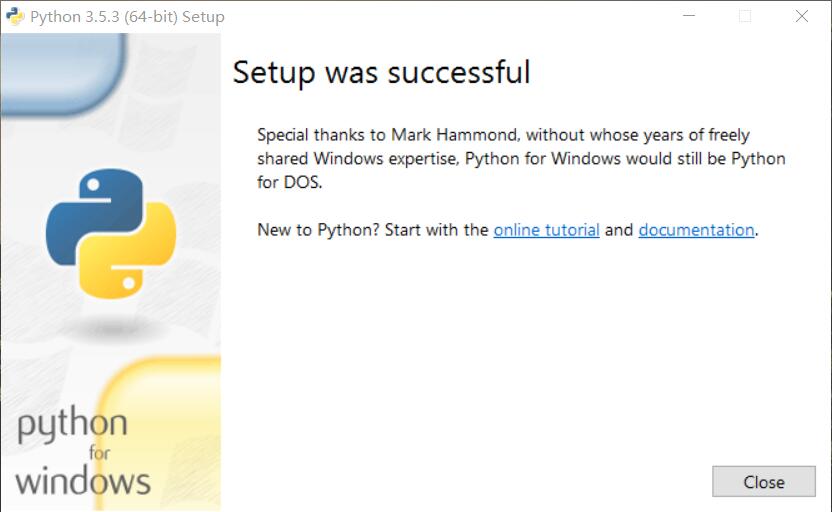


图4 Python3.5.3安装成功窗口

下面打开命令提示符，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，发现已经最新，可以使用pip进一步的安装。

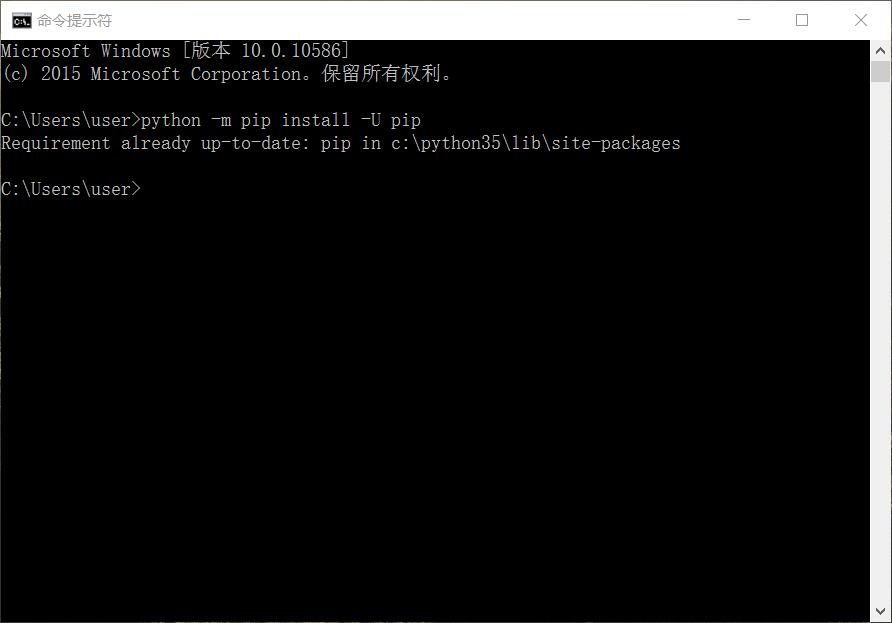


图5 命令提示符输入指令后显示pip已经最新

## 2.2 Python3.5.3使用示例

此时的Python开发环境IDLE已经建立，可以编写代码，示例如下图。

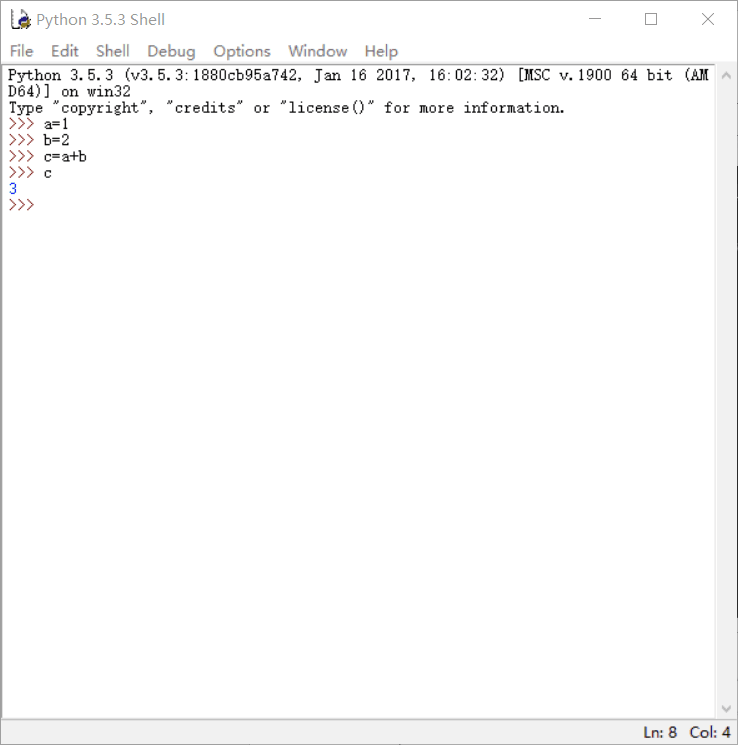


图6 开发环境IDLE

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，坏处是没法保存，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。[2]

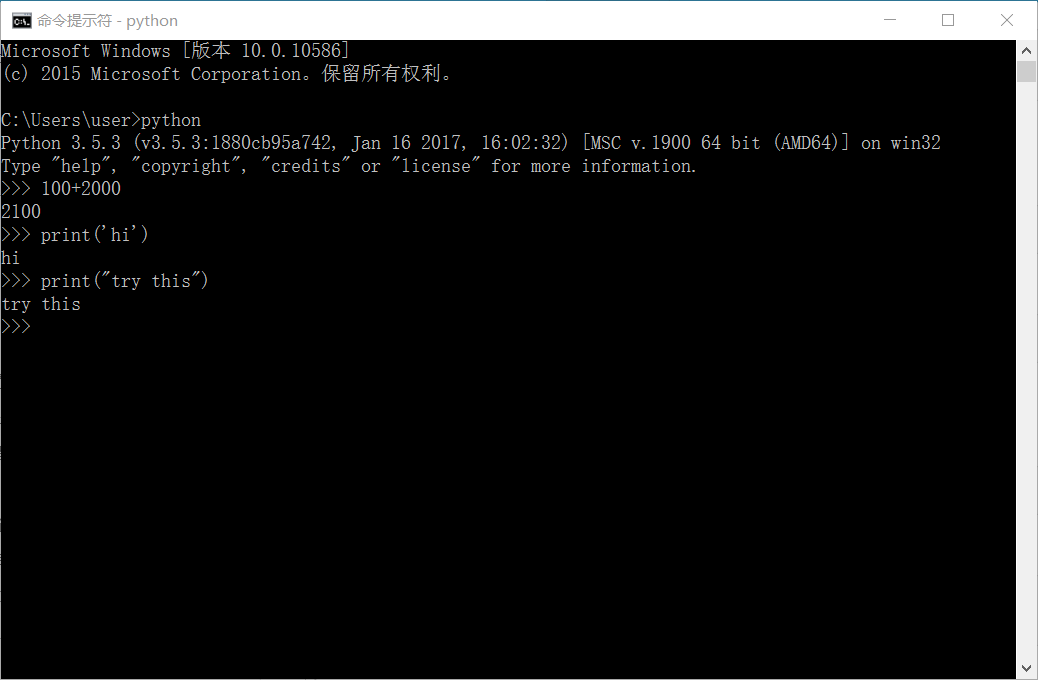


图7 在命令提示符里打开Python

# 3 安装扩展模块

## 3.1 安装科学计算包

基本的科学计算包有 SciPy、NumPy、Matplotlib。

在加州大学欧文分校的网站上（<http://www.lfd.uci.edu/~gohlke/pythonlibs/>）可以找到对应版本的安装包，下载后用命令提示符进行pip安装。以NumPy为例。



图8 NumPy扩展包下载选项

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。

打开命令提示符，输入 >pip install \*.whl。\*用扩展包名字替换，如：>pip install numpy-1.12.0+mkl-cp35-cp35m-win\_amd64.whl，要注意的是，需要cd到安装包所下载的目录下才能找到并且安装。有些安装包有依存关系，所以输入一个安装指令可能会装两个包，如图9。要注意，这里自动安装的numpy是不含mkl的，所以还是要另外安装一次包含mkl的numpy。

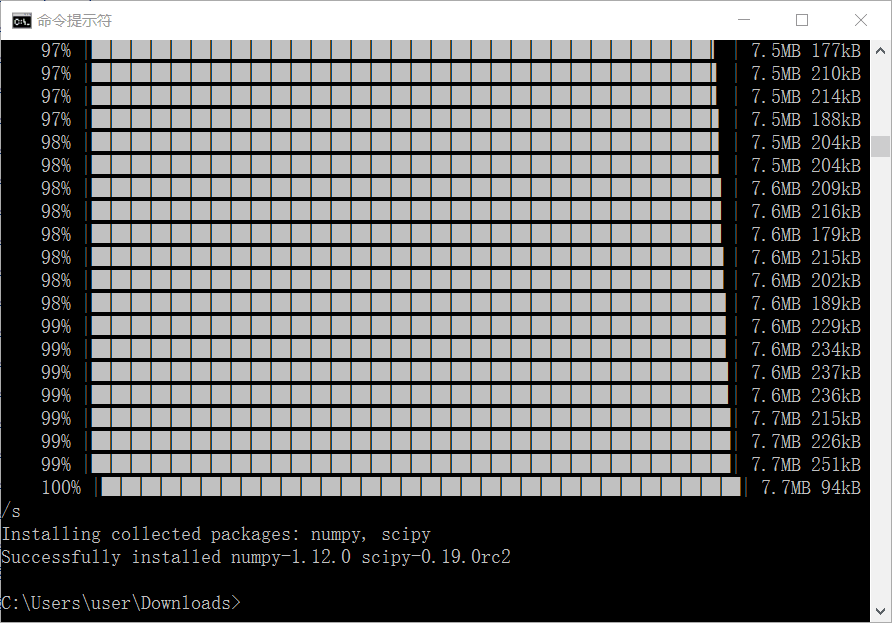


图9 Scipy科学扩展包安装成功

## 3.2 安装规范格式扩展包

安装扩展模块pep8、autopep8 和 pylint，一般不用另外下载，直接输入pip install pep8即可在线安装。这一步安装python之后就进行，这样后续环境的建立才比较顺利。

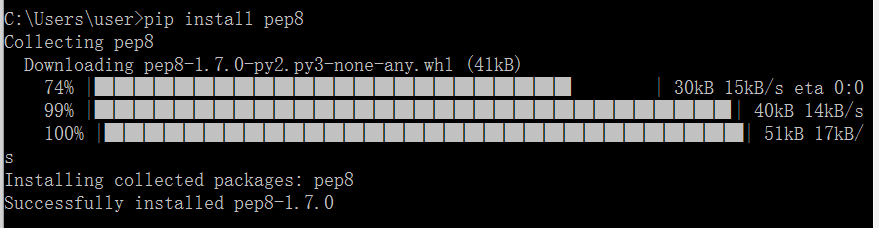


图9 格式规范扩展包安装示例

## 3.3 IF97物性计算

IF97物性计算可使用基于C语言实现共享库、纯Python语言包。C语言共享库计算速度远快于Python语言包。C语言IF97共享库及其Python封装包仓库：:

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

下载仓库的zip文件并解压，将和操作系统对应版本的seuif97.dll和 seuif97.py，seuif97.dll拷贝到 c:\windows\system，seuif97.py拷贝到 c:\python35\Lib。

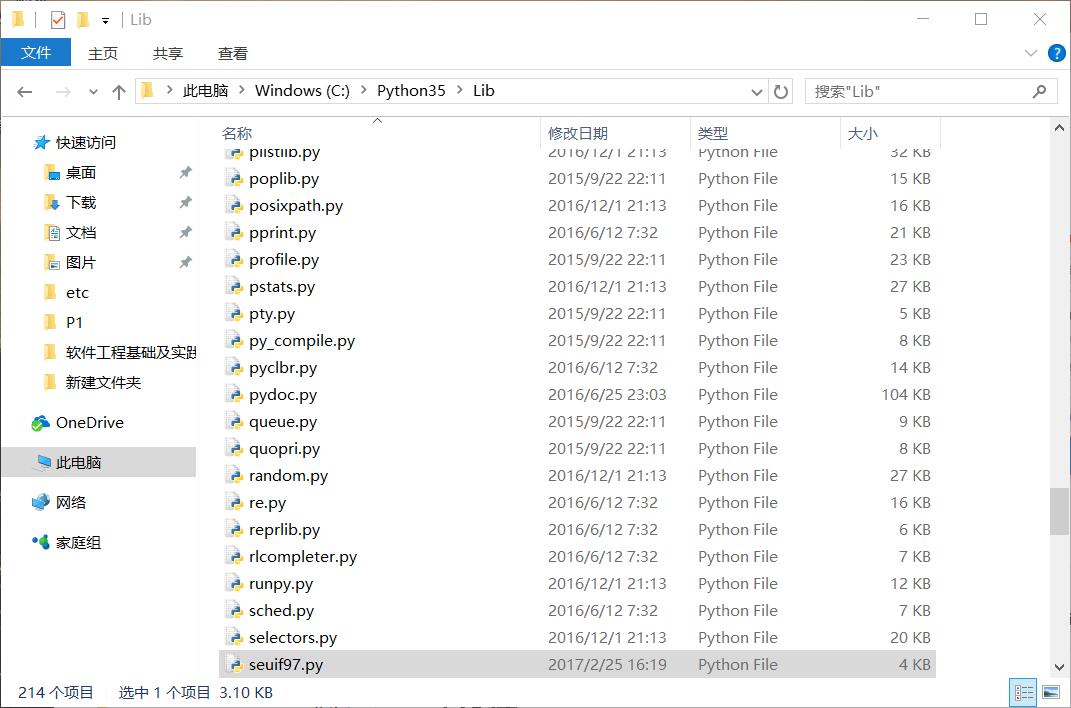


图10 拷贝seuif97.py到 c:\python35\Lib

Python语言计算包，在线安装：>pip install iapws

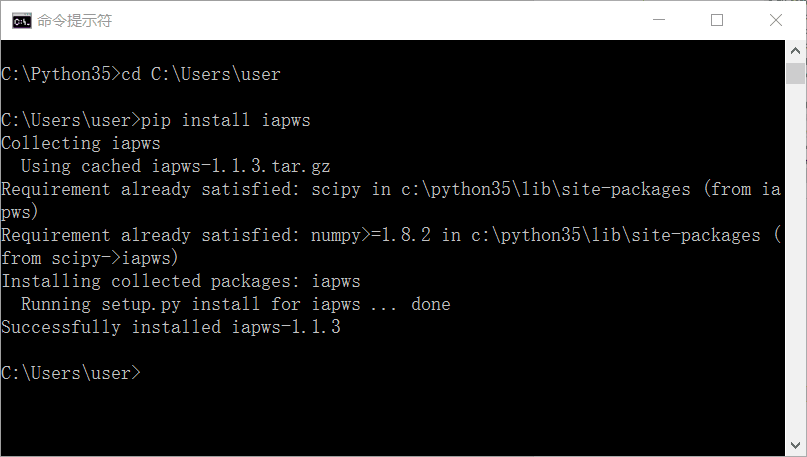


图11 成功安装iapws

# 4 交互计算环境Jupyter Notebook

## 4.1 Jupyter Notebook安装过程

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。

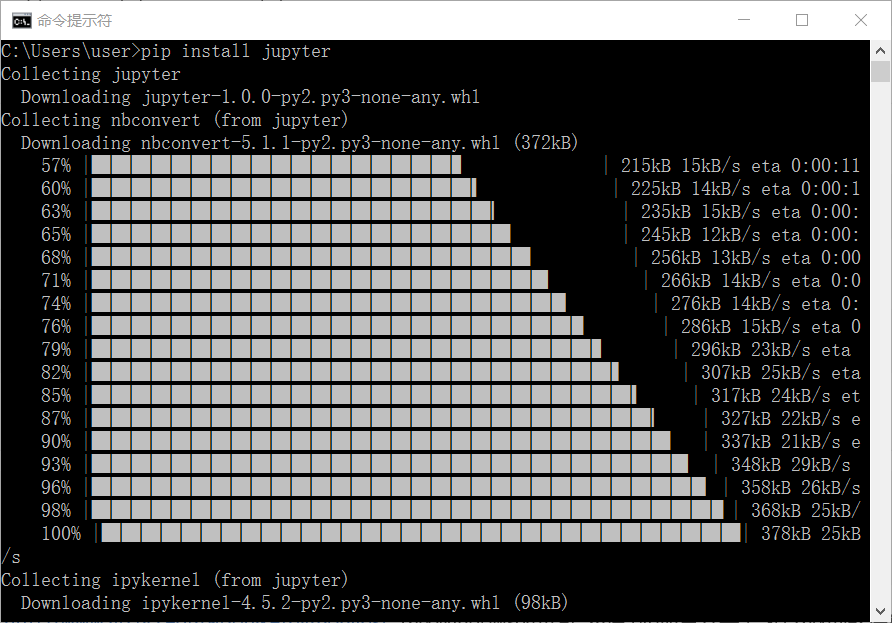
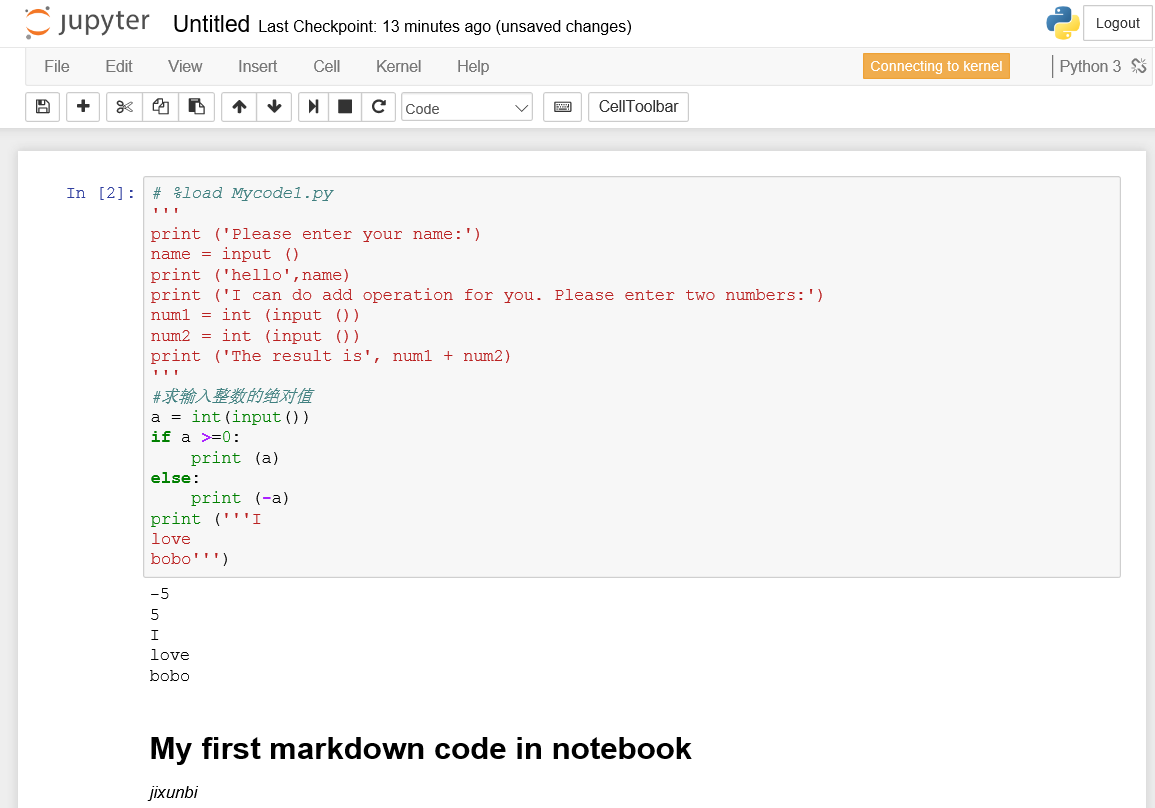


图12 在线安装Jupyter Notebook

在制定目录下输入>jupyter notebook 就可以运行并用notebook模式打开这一目录。

## 4.2 Jupyter Notebook使用示例

我们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，我们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[3]。

图13 Jupyter Notebook使用示例截图

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>)。

我们课件也可以用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。

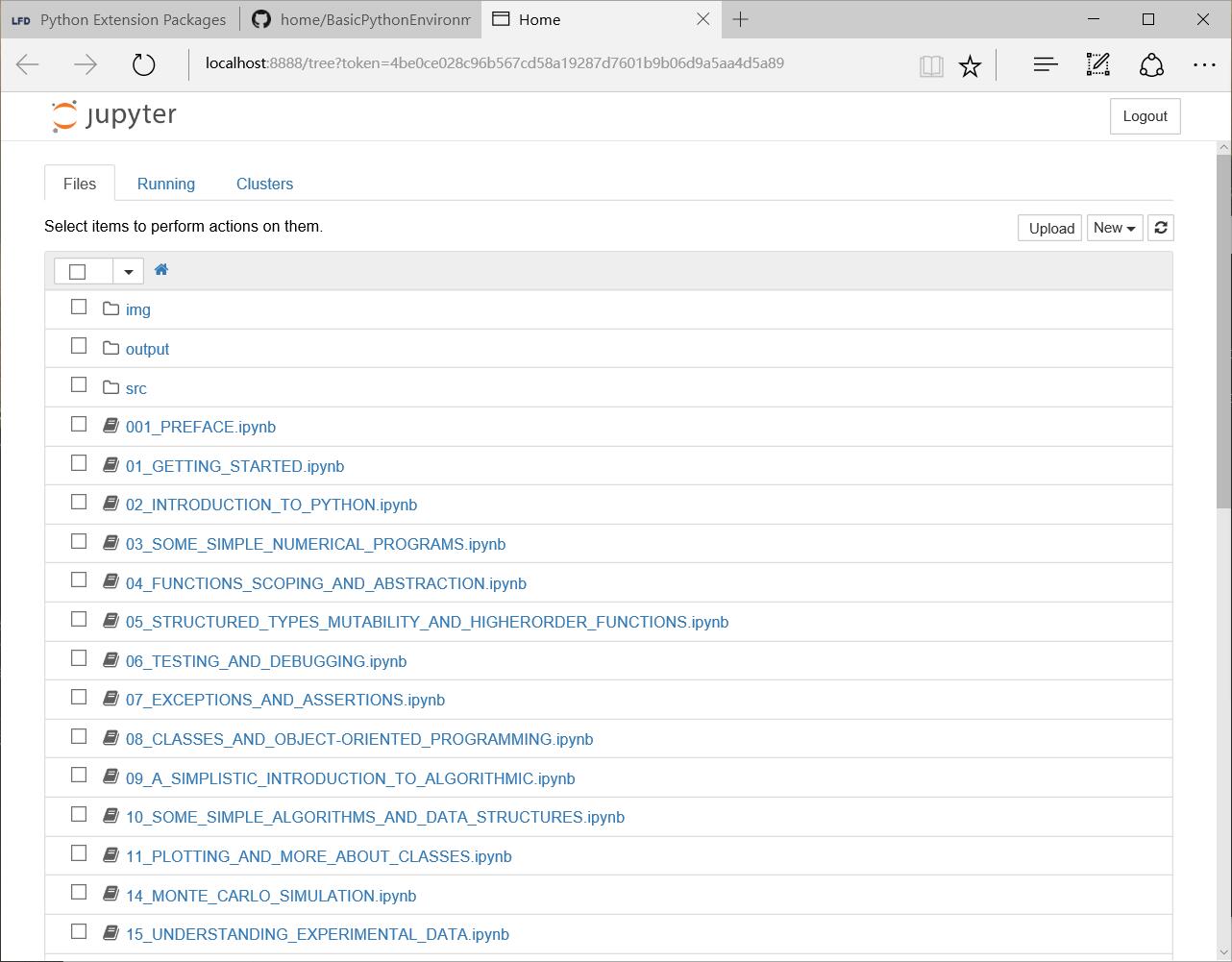


图14 Jupyter Notebook运行课件目录的界面截图

# 5 集成开发环境Eclipse

## 5.1 Eclipse安装过程

### 5.1.1 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[4]。

Java JDK的Oracle官方下载地址为：

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

运行下载Java安装包，即可安装好Java环境，安装时需要配置环境变量。配置环境变量的方法可以参考以下网址：

<http://jingyan.baidu.com/article/9f63fb91d87fb0c8400f0e93.html>

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 。

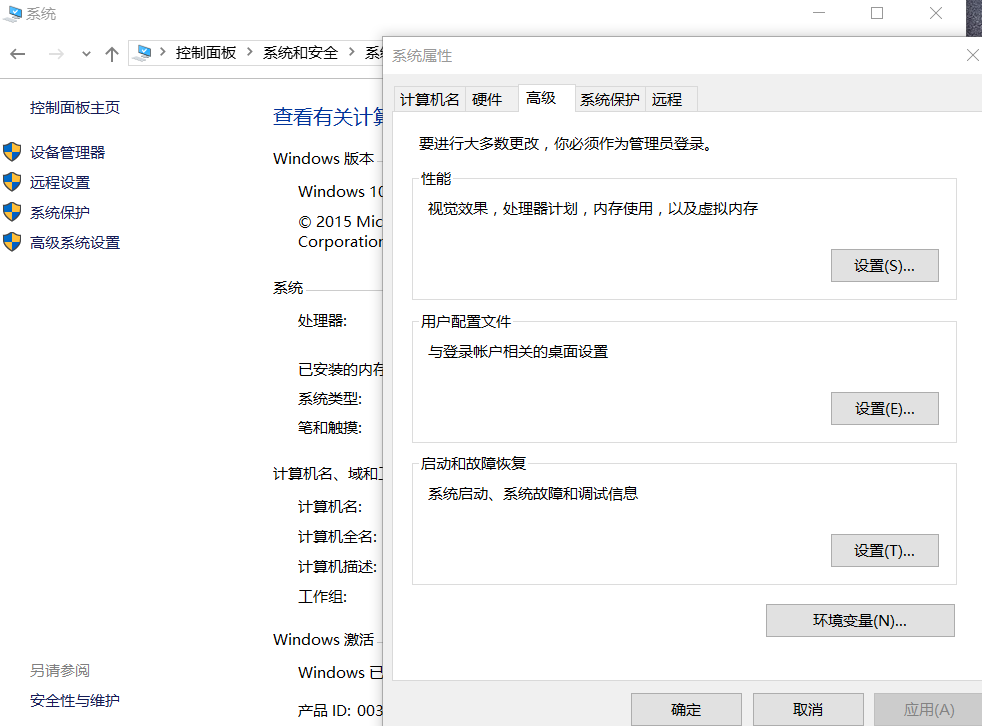


图15 环境变量所在界面

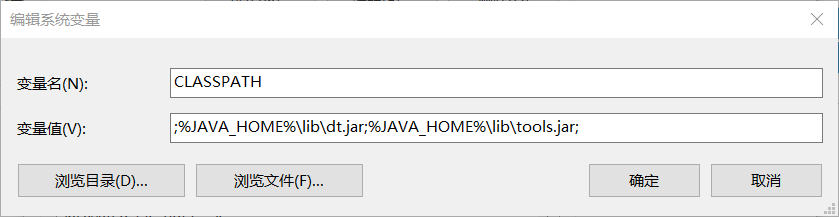


图16 设置变量名和变量值

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。

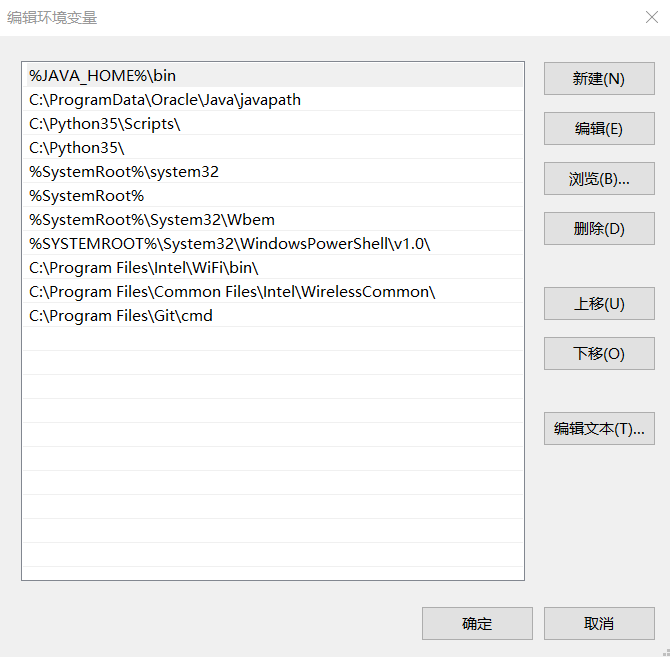


图17 编辑已有系统变量并新建路径

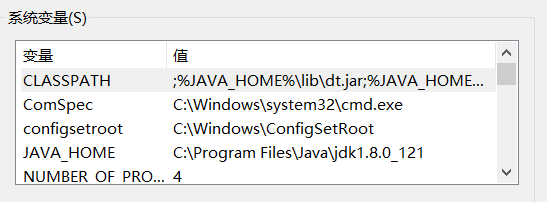


图18 添加后的系统变量

打开命令提示符，输入>javac -version检验是否配置成功。显示版本号说明配置成功。

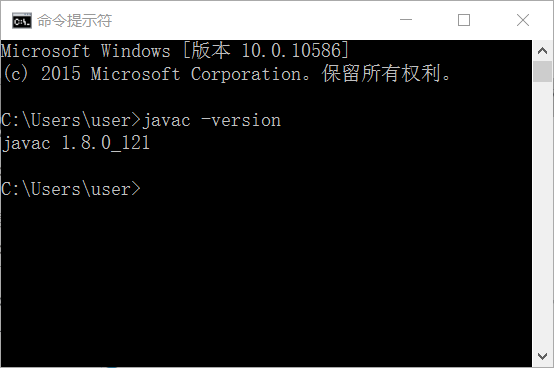


图19 环境变量配置成功

### 5.1.2 安装Eclipse及配置工作空间

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。

### 5.1.3 安装PyDev插件及配置

通过Help->Eclipse Marketplaces进入市场，输入pydev，找到Pydev项目，点“install”在线安装，安装好重新启动Eclipse。下一步是配置Python解释器，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，添加Python场景到工作页面。

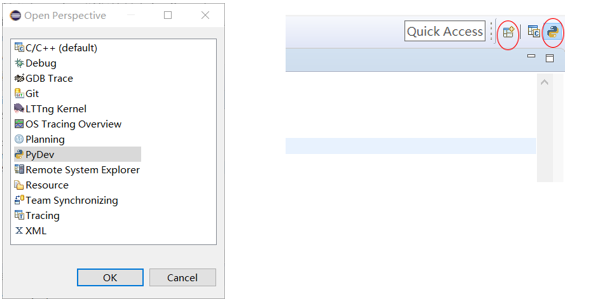


图20 添加Python场景到工作页面

然后就可以进行开发Python了。为了方便使用，还可以对Pydev进行配置，比如说修改编辑器颜色提高可读性、添加任务标签注释等等。这些都可以根据个人的喜好和需要进行设置，具体操作见参考资料【Python学习和开发环境的建立(教学版).docx】，本文不再赘述。

在配置中，比较重要的一点是基于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。而语法检查比较重要，所以在此说明。在程序开发的过程中，大妈的编写也需要注意规范，但是过多的注意也会影响进程，所以开发过程中默认检查关闭，在一个阶段结束时开启规范性检查。设置的方法如下：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。启动autopep8自动修改：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。

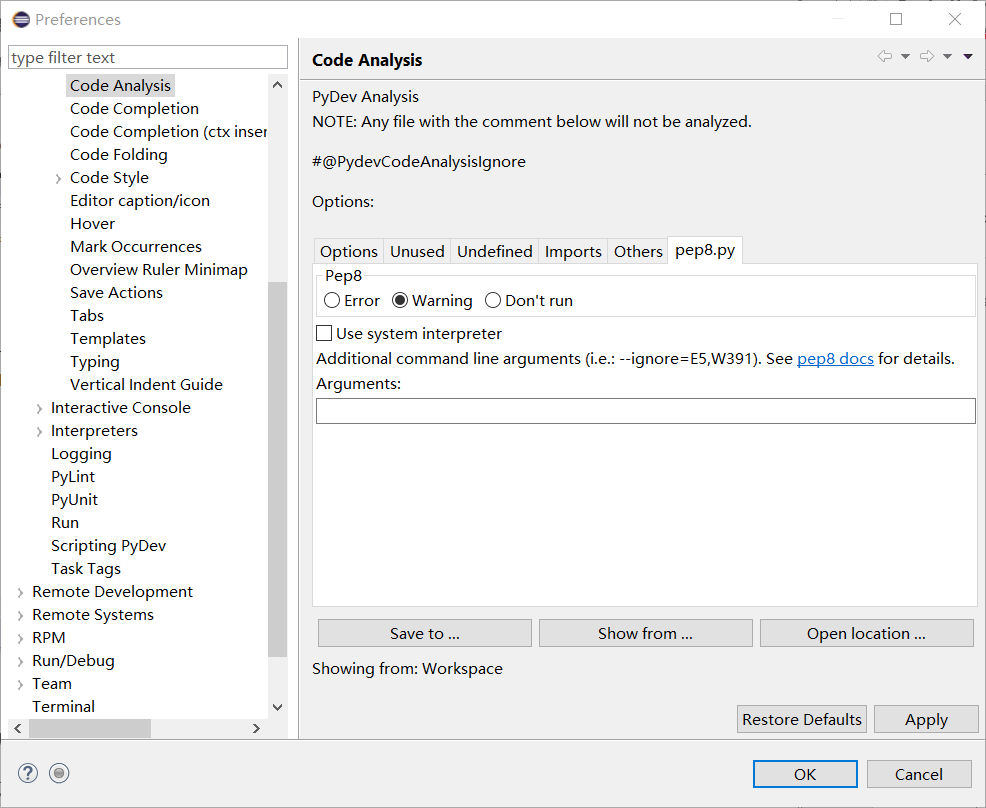


图21 开启pep8检查

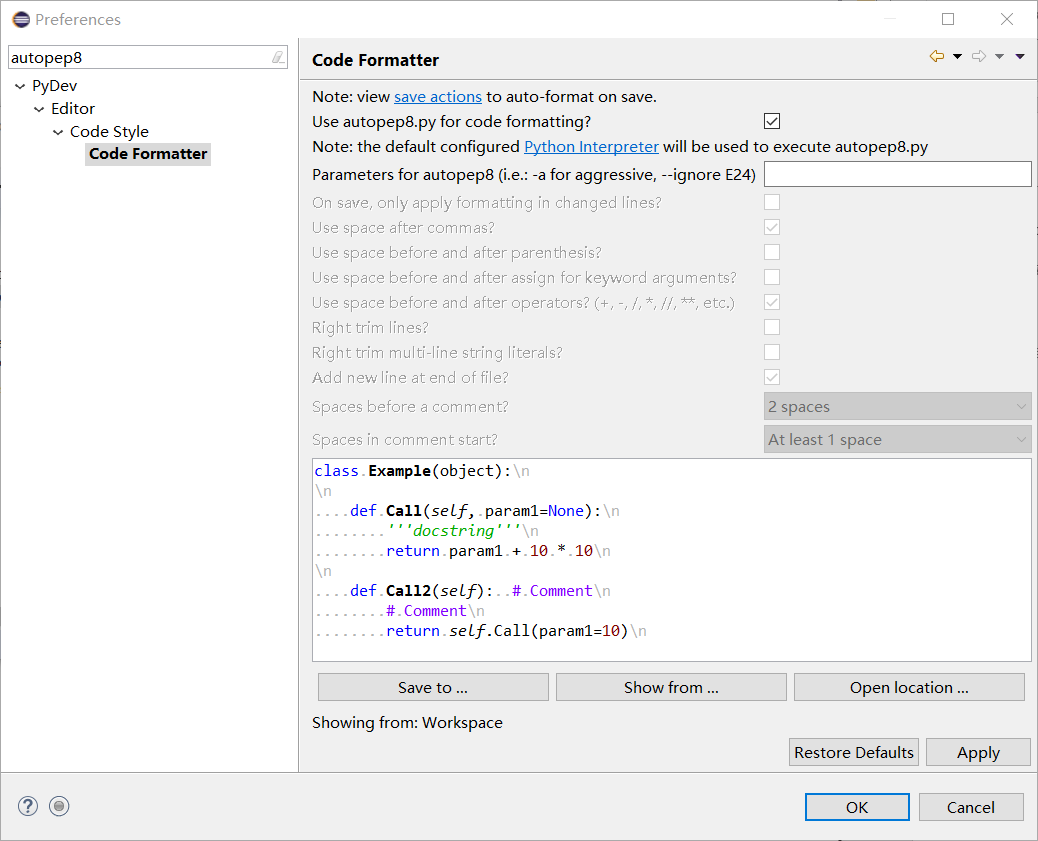


图22 开启autopep8

Pylint的开启：PyDev 默认不开启Pylint。通过Window -> preferences -> Pydev -> Pylint,选中"Use pylint?",“Browse…”找到安装好的lint.py的地址。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手动触发pylint。

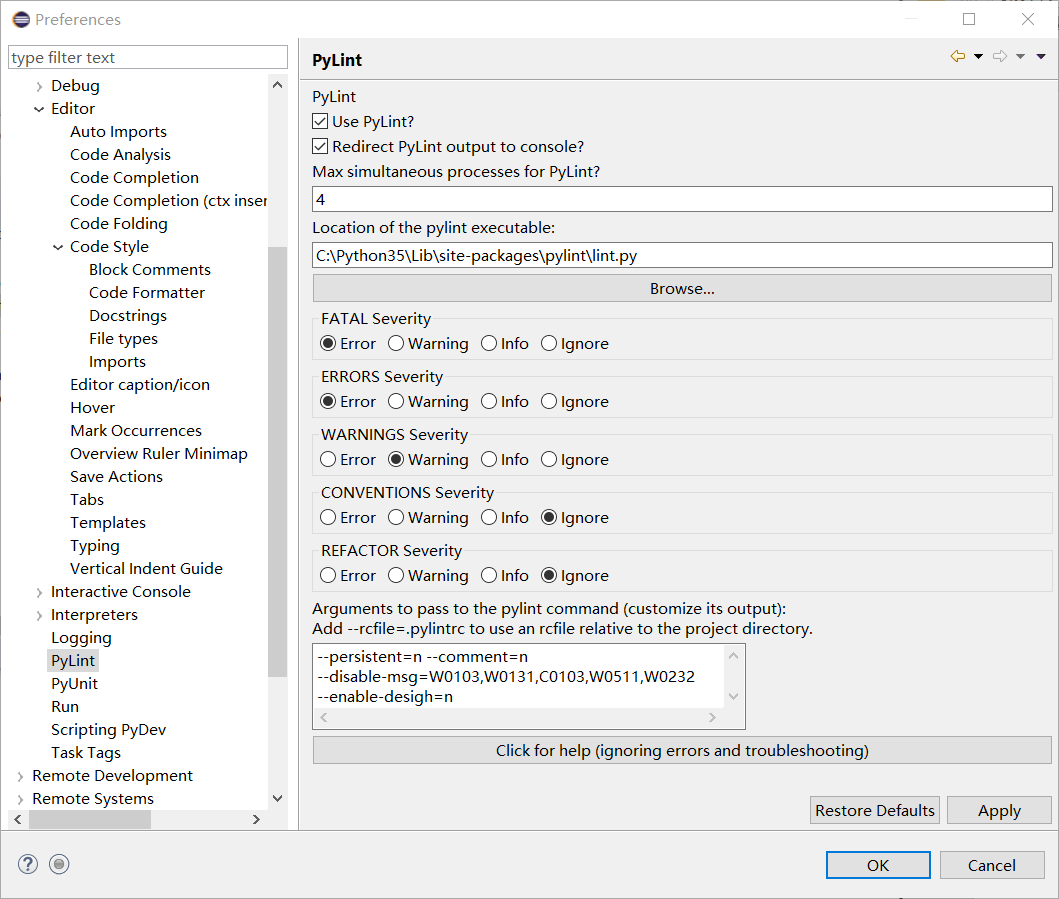


图23 开启Pylint

## 5.2 Eclipse使用示例

要想运用好Eclipse，光有以上配置也是不够的。在使用的时候还有一些注意点，比如说编码方式。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。

点“Window”->“Preference”，在弹出窗口中，点击“general”-“workspace”，修改“Text file encoding”为UTF-8。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“Text file encoding”为UTF-8(如果工作空间配置为UTF-8会继承过来)。

还有一个比较实用的就是Markdown插件。Eclipse默认安装对Markdown支持较弱，需要安装Markdown插件。Help->Eclipse Marketplace使用Markdown关键字可以找到2个插件：编辑和显示。Eclipse Marketplace 市场安装，安装后选择Markdown Editor作为md文件的编辑器（右键md文件，open with下也有这个选项）。

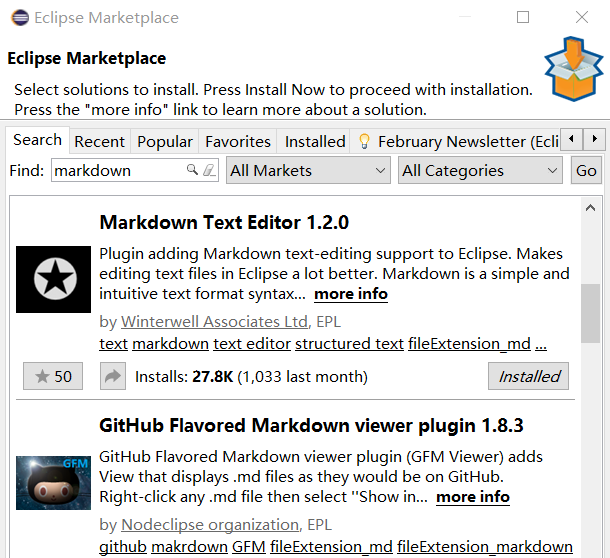


图24 Eclipse Marketplace（插件安装后）

Windows->Preferences->General->Editors->Text Editors->Markdown可以配置有关参数。Viewers插件的安装同理，安装后，Window -> Show View -> Other... and select entry GFM Support -> GFM Viewer。开启GFM Viewer窗口，就可以显示生成的文件内容式样了。

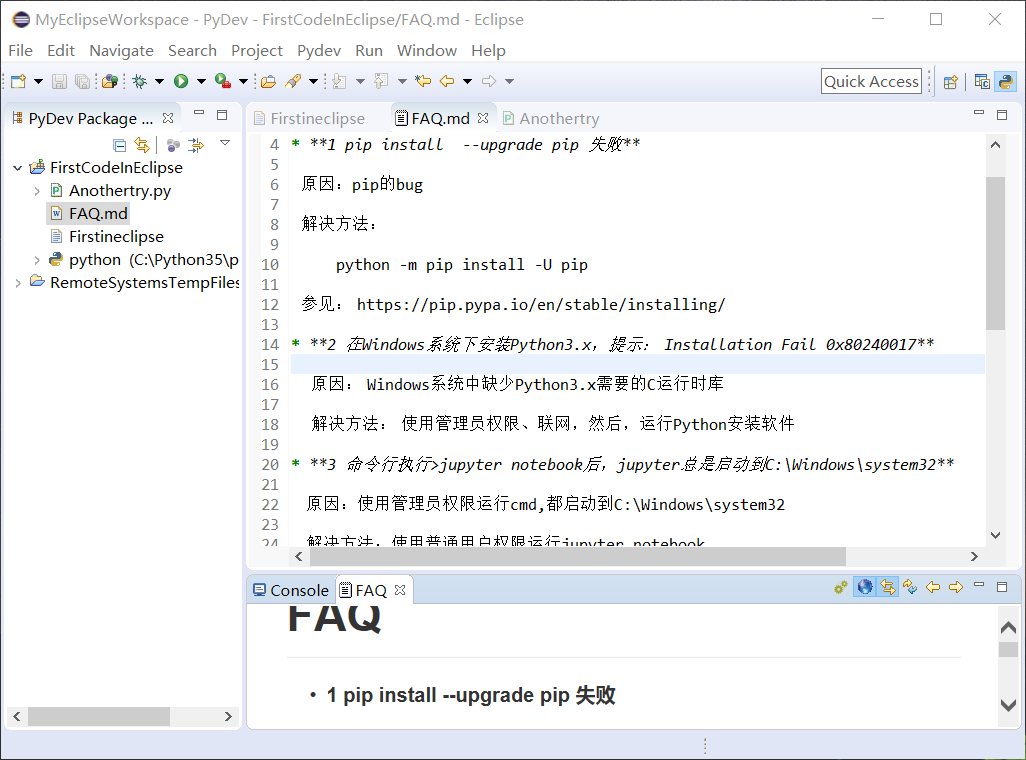


图25 打开md文件编辑和显示视图

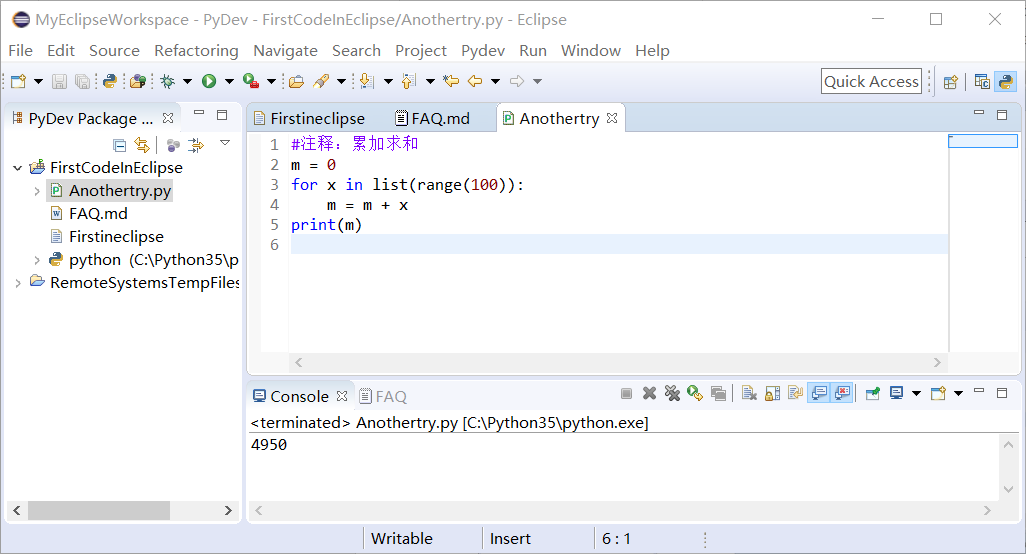


图26 新建项目文件编写代码及运行结果示例

在编写简单代码的过程中，可以发现之前的配置能极大的方便我们编程。比如说注释的颜色更为清晰，保存后可以自行检查语法错误，输入几个字母就可以提示函数的输入等等。

# 6 问题及解决

1. **问题1：**在线安装Jupyter的过程中出现红字，未能成功安装。

**解决办法：**可能因网络状况导致，因为Jupyter的安装需要的包比较多，很容易因为网络不好导致下载安装失败。检查联网，重新输入指令，即可解决。如果还不行，就在网上找到对应安装包添加到目录下再进行安装。

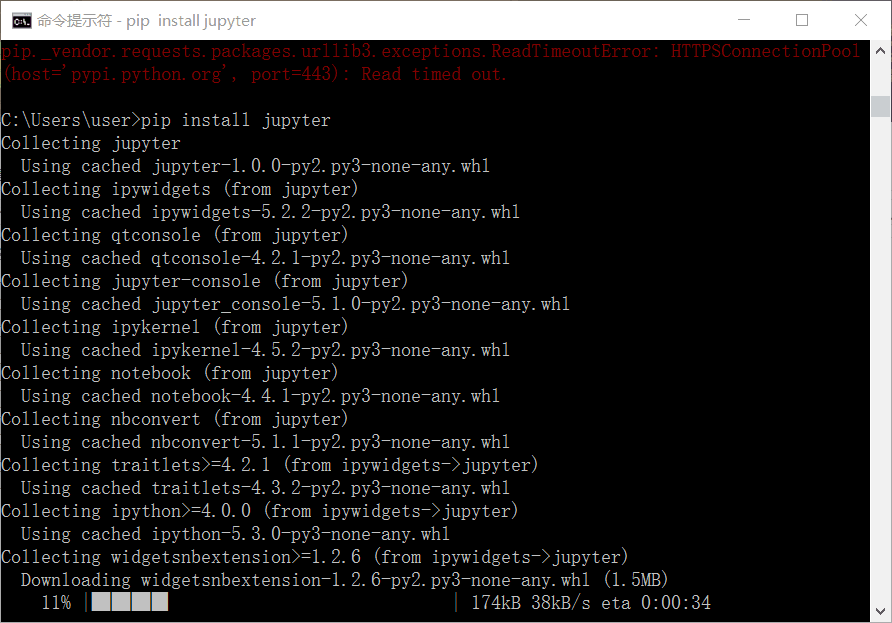


图27 失败后第二次安装成功

1. **问题2：**安装科学扩展包出现问题。

**解决办法：**如图28中的问题，找不到安装包。所以要cd一下路径，而且文件名不能输入错误，尤其是符号“-”的影响。

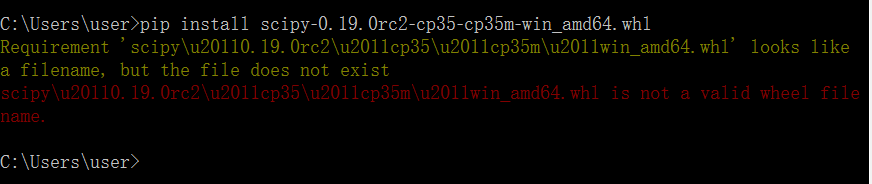


图28 安装科学扩展包错误提示

1. **问题3：**安装Python语言计算包iapws提示如图29的错误。

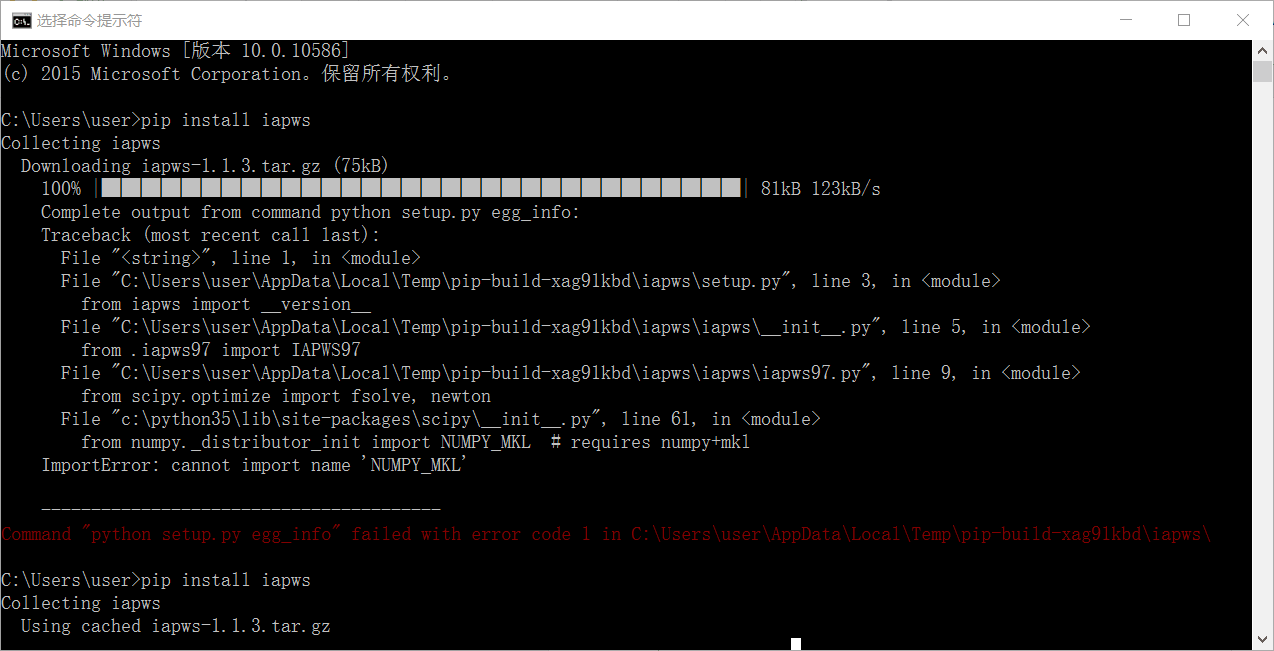


图29 安装语言计算包iapws错误提示

**解决办法：**查看错误提示，显示未安装numpy\_mkl，然后发现开始显示装的numpy的版本是不含mkl的，因此还要重新安装一次这个科学计算包。安装结束之后再进行iapws的安装，错误解决。

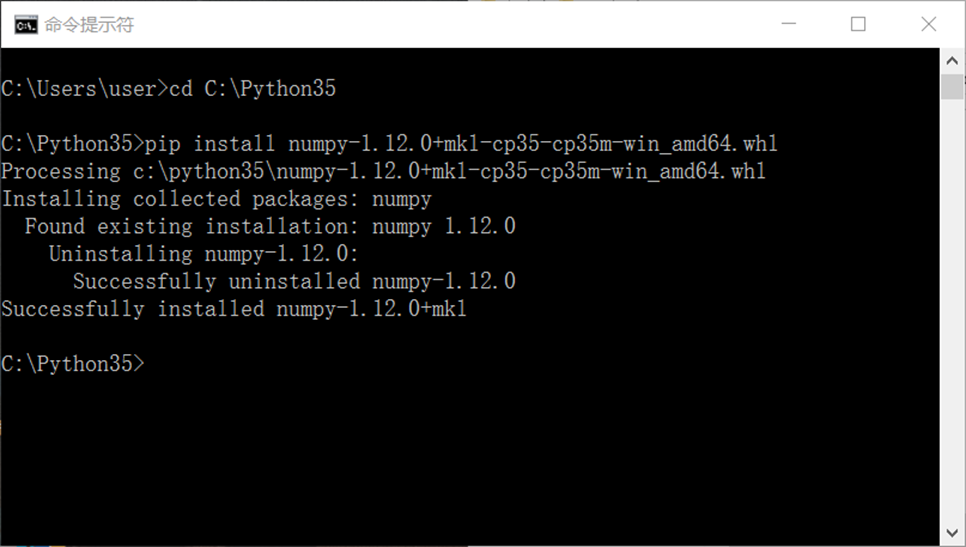


图30 重新安装NumPy

1. **问题4：**用Eclipse新建工程，发现编写代码并没有联想的功能，也没有区分度。

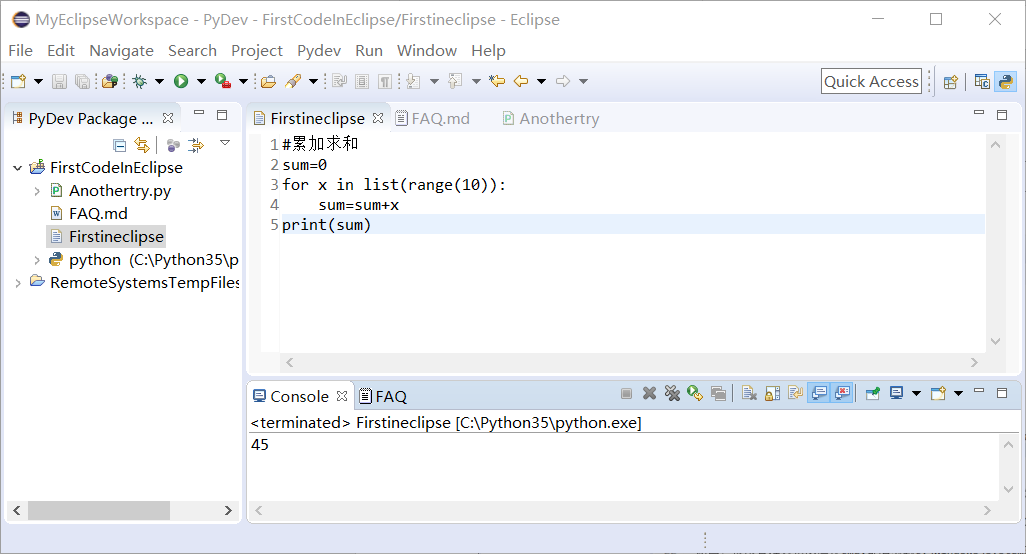


图31 不加后缀的文件显示

**解决办法：**新建的文件名是要自己加后缀”.py”的。不加后缀的文件可以写代码也可以编译，但是自动检查语法和提示等功能就不能使用。

# 7 个人小结

开始安装python并且建立比较完善的开发环境，断断续续花费了我好几天的时间。在这个过程中，可以说我的收获颇丰。从第一节课的不知所云，到现在我已经基本了解了各个软件的作用，以及简单的使用方法。

了解了python这个语言，也就能理解版本控制的概念和github的优势了，作为一个适合编写开源的解释型编程语言，可重复和可再生的特点也是它流行的原因。在建立开发环境的过程中，我熟悉的不只是新的写程序的语法，还提高了应用电脑的能力和自学的能力。安装的时候，可能因为联网问题或者没有注意的小细节，而出现问题，但是基本都是可以在网络上找到解决办法的。比如在线安装时候出现红字，提示存在问题，可以再输入一次命令重新安装，不一定要急着去网上找安装包，有时候只是网络连接的偶然原因导致失败。

电脑是一个很强大的工具，可是不同人用起来，同一台电脑开发的功能，可以产生的效用可能天壤之别。同样，开发环境也同样，我的理解就是，一个好的开发环境不仅仅是编写程序的必须，同样是一个提高效率的帮手，如果功能了解得多并且会使用的话，才能算是真正利用好了这个环境。

希望我在后面课程的学习中，对这个语言及其应用能有更深入的理解。

# 8 参考资料

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. 郑伟芳. PyDev for Eclipse简介.

<http://www.ibm.com/developerworks/cn/opensource/os-cn-ecl-pydev/>

1. 廖雪峰的官方网站-Python教程

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