

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
2 Python 语言	3
2.1 Python3.5.3 安装过程	3
2.2 Python3.5.3 使用示例	5
3 安装扩展模块.....	6
3.1 安装科学计算包.....	7
3.2 安装规范格式扩展包.....	8
3.3 IF97 物性计算	8
4 交互计算环境 Jupyter Notebook.....	10
4.1 Jupyter Notebook 安装过程	10
4.2 Jupyter Notebook 使用示例	10
5 集成开发环境 Eclipse	12
5.1 Eclipse 安装过程	12
5.1.1 安装 Java SDK 并配置环境变量	12
5.1.2 安装 Eclipse 及配置工作空间.....	15
5.1.3 安装 PyDev 插件及配置.....	15
5.2 Eclipse 使用示例	18
6 问题及解决	20
7 个人小结.....	23
8 参考资料.....	24

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, 会有很多个下载的连接, 如图 1。以 64 位的 win10 系统为例, 应该下载 Windows x86-64 executable installer。

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		6192f0e45f02575590760e68c621a488	20656090	SIG
XZ compressed source tarball	Source release		57d1f8bfabf4f2500273fb0706e6f21	15213396	SIG
Mac OS X 32-bit i386/PPC installer	Mac OS X	for Mac OS X 10.5 and later	4994f588ebad17c4bf12148729b430d5	26385455	SIG
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6f9ee2ad1fceb1a7c66c9ec565e57102	24751146	SIG
Windows help file	Windows		91600322a55cff692dd7fbc2fb0d841	7794982	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64, not Itanium processors	1264131c4c2f3f935f34c455bceedee1	6913264	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64, not Itanium processors	333d536b5f76f95a6118fb2ecd623351	30261960	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64, not Itanium processors	b6be1ce6e69ac7dcd6b3316c91bebd95	974352	SIG

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）。



图 2 运行 Python3.5.3 安装文件后出现的窗口

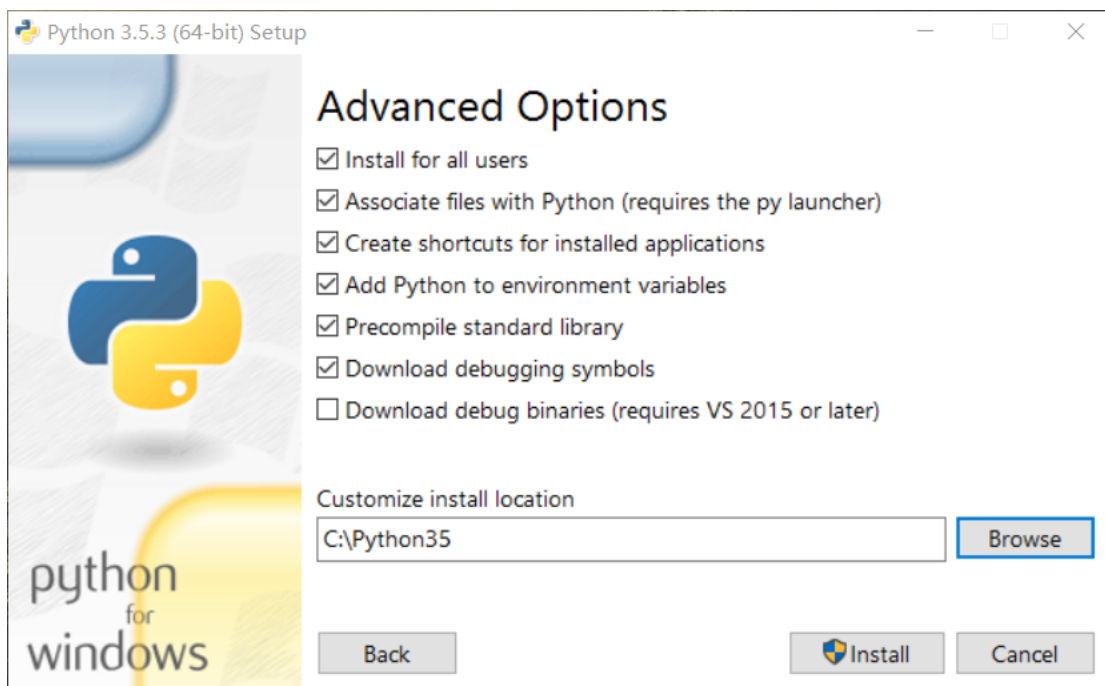


图 3 Python3.5.3 自定义安装窗口

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：

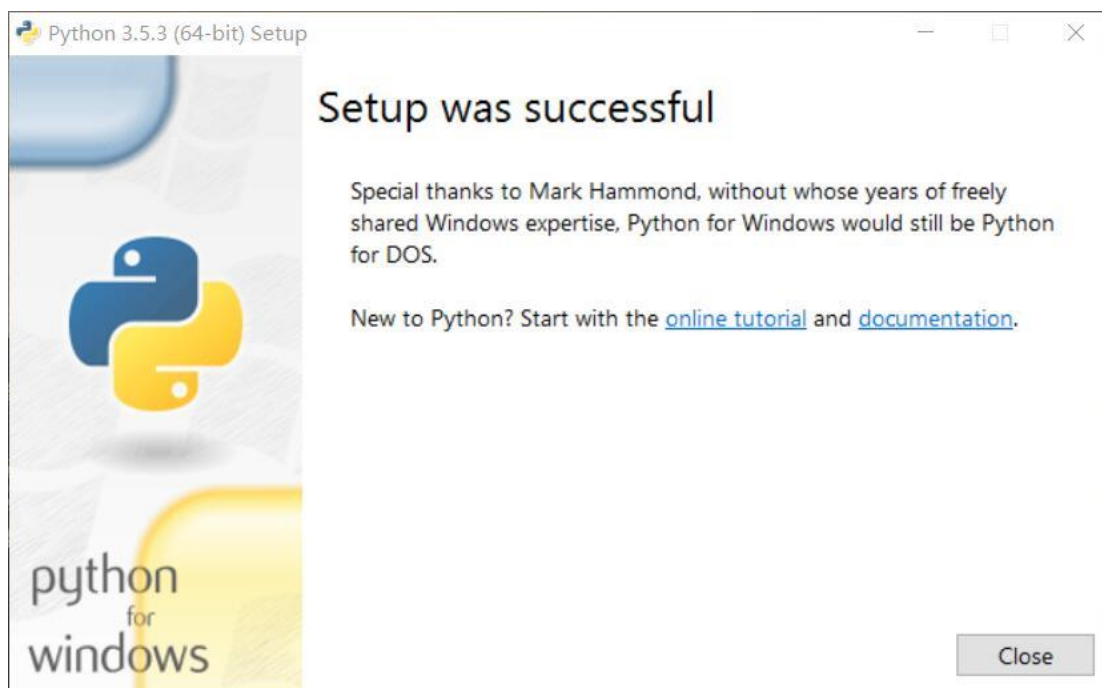


图 4 Python3.5.3 安装成功窗口

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。

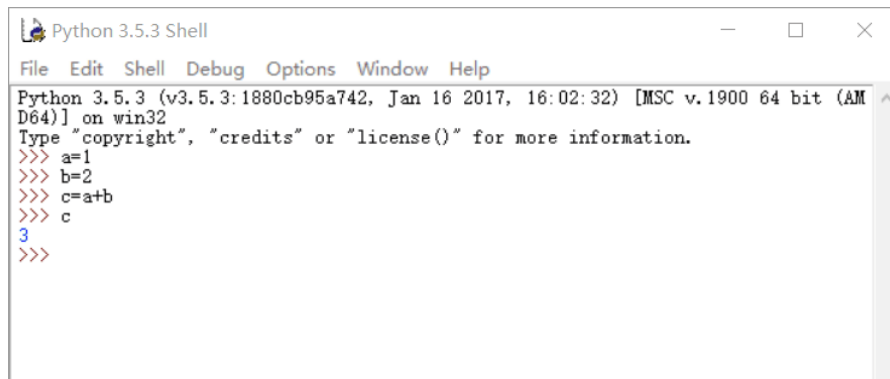


图 6 开发环境 IDLE

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。

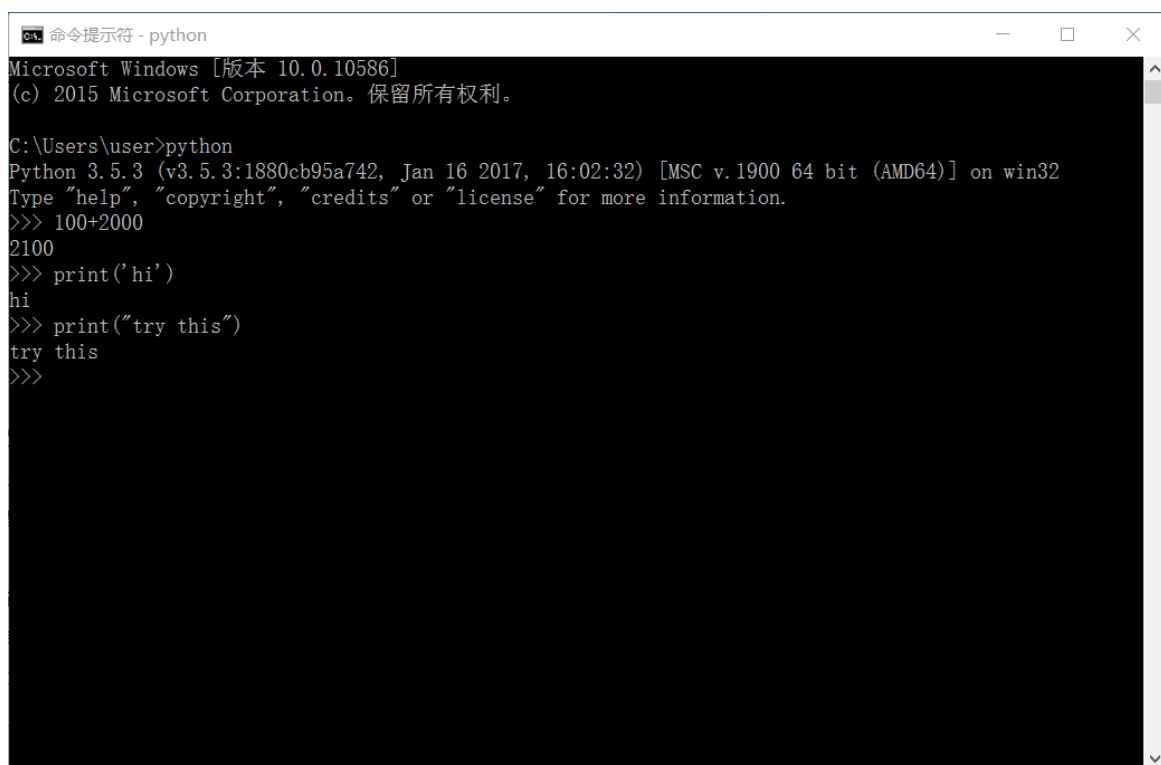


图 7 在命令提示符里打开 Python

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。

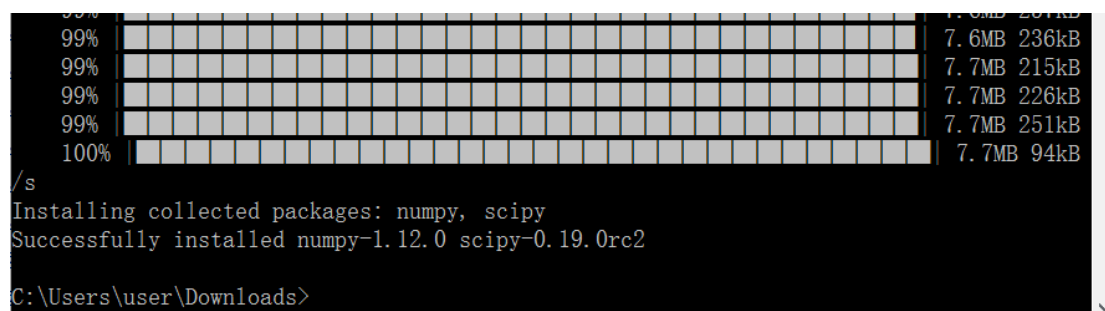
NumPy, a fundamental package needed for scientific computing with Python.
 Numpy+MKL is linked to the [Intel® Math Kernel Library](#) and includes required DLLs in the numpy.core directory.

[numpy-1.11.3+mkl-cp27-cp27m-win32.whl](#)
[numpy-1.11.3+mkl-cp27-cp27m-win_amd64.whl](#)
[numpy-1.11.3+mkl-cp34-cp34m-win32.whl](#)
[numpy-1.11.3+mkl-cp34-cp34m-win_amd64.whl](#)
[numpy-1.11.3+mkl-cp35-cp35m-win32.whl](#)
[numpy-1.11.3+mkl-cp35-cp35m-win_amd64.whl](#)
[numpy-1.11.3+mkl-cp36-cp36m-win32.whl](#)
[numpy-1.11.3+mkl-cp36-cp36m-win_amd64.whl](#)
[numpy-1.12.0+mkl-cp27-cp27m-win32.whl](#)
[numpy-1.12.0+mkl-cp27-cp27m-win_amd64.whl](#)
[numpy-1.12.0+mkl-cp34-cp34m-win32.whl](#)
[numpy-1.12.0+mkl-cp34-cp34m-win_amd64.whl](#)
[numpy-1.12.0+mkl-cp35-cp35m-win32.whl](#)
[numpy-1.12.0+mkl-cp35-cp35m-win_amd64.whl](#)

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。



```

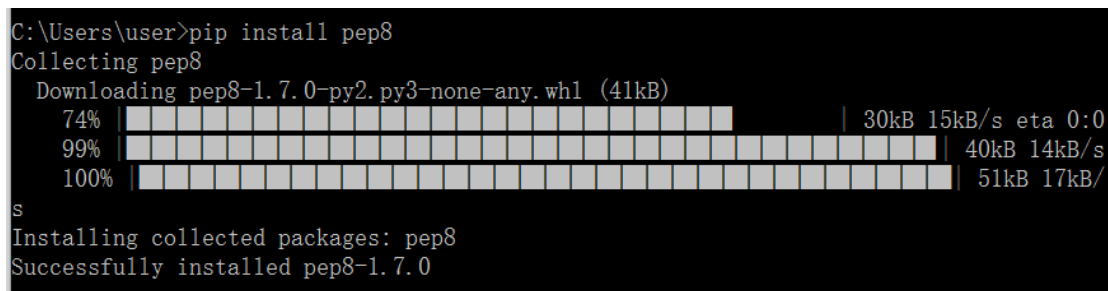
99% | 7.6MB 236kB
99% | 7.7MB 215kB
99% | 7.7MB 226kB
99% | 7.7MB 251kB
100% | 7.7MB 94kB
/s
Installing collected packages: numpy, scipy
Successfully installed numpy-1.12.0 scipy-0.19.0rc2
C:\Users\user\Downloads>

```


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。



```
C:\Users\user>pip install pep8
Collecting pep8
  Downloading pep8-1.7.0-py2.py3-none-any.whl (41kB)
    74% |#####| 30kB 15kB/s eta 0:0
    99% |#####| 40kB 14kB/s
   100% |#####| 51kB 17kB/s
Installing collected packages: pep8
Successfully installed pep8-1.7.0
```

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。

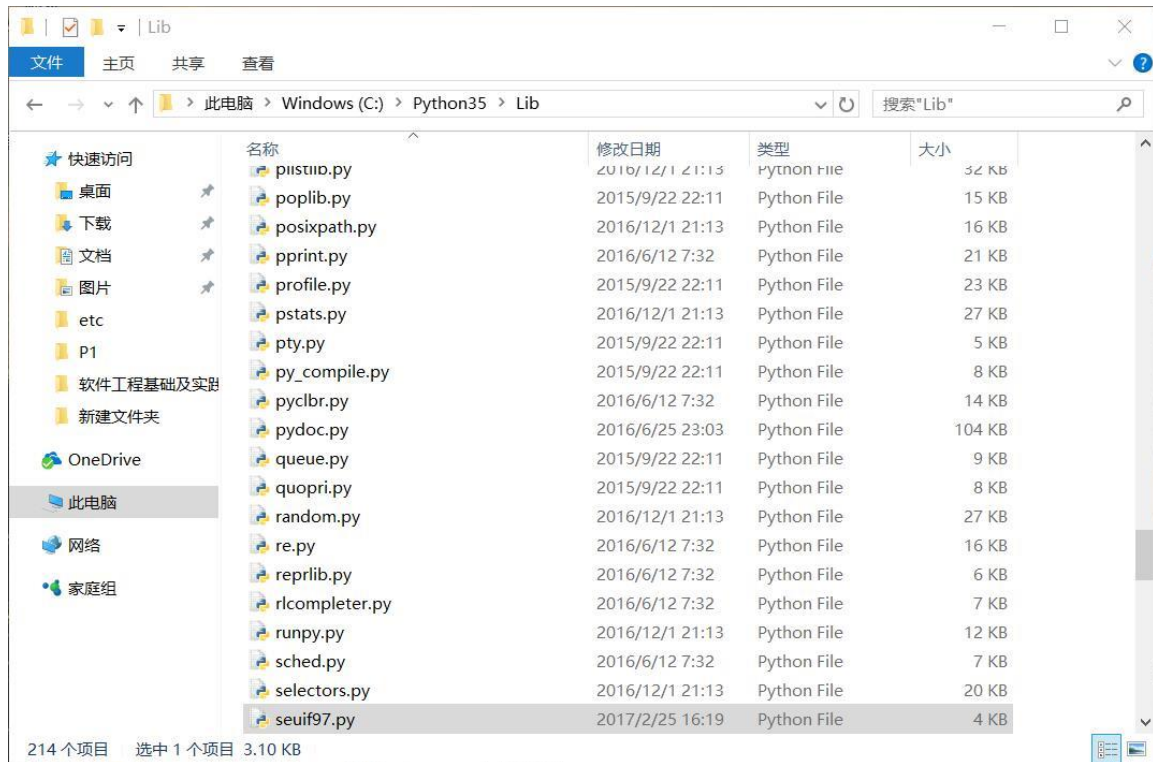


图 10 拷贝 seuif97.py 到 c:\python35\Lib

Python 语言计算包，在线安装：>pip install iapws

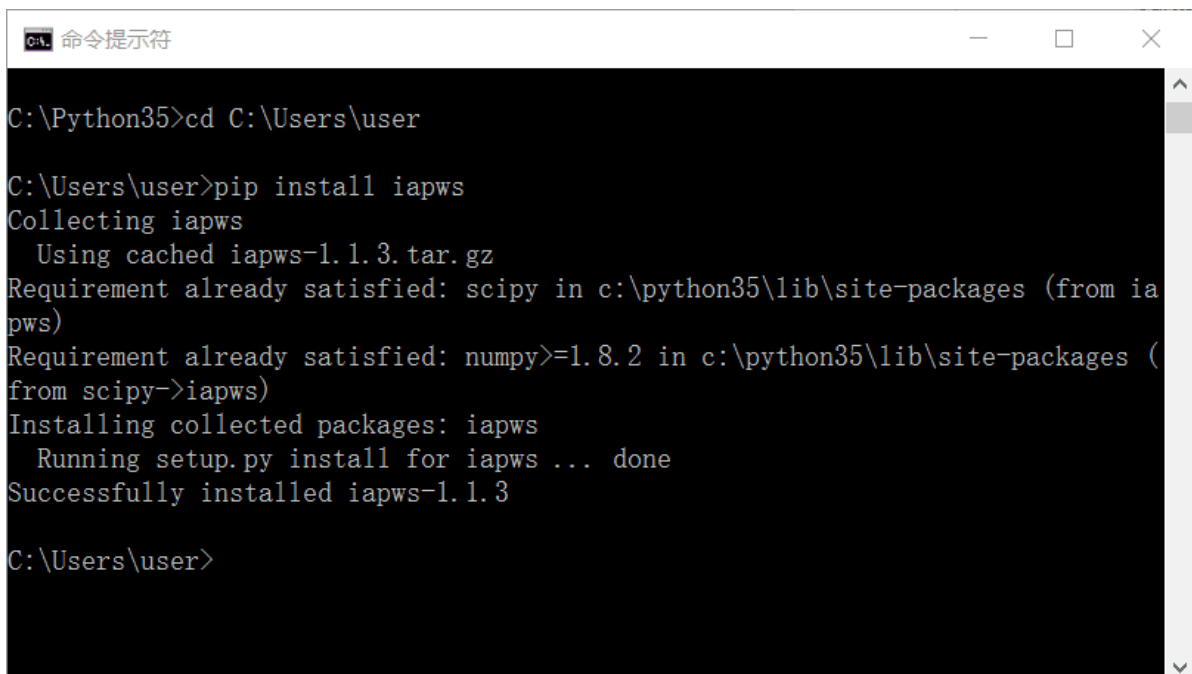


图 11 成功安装 iapws

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。

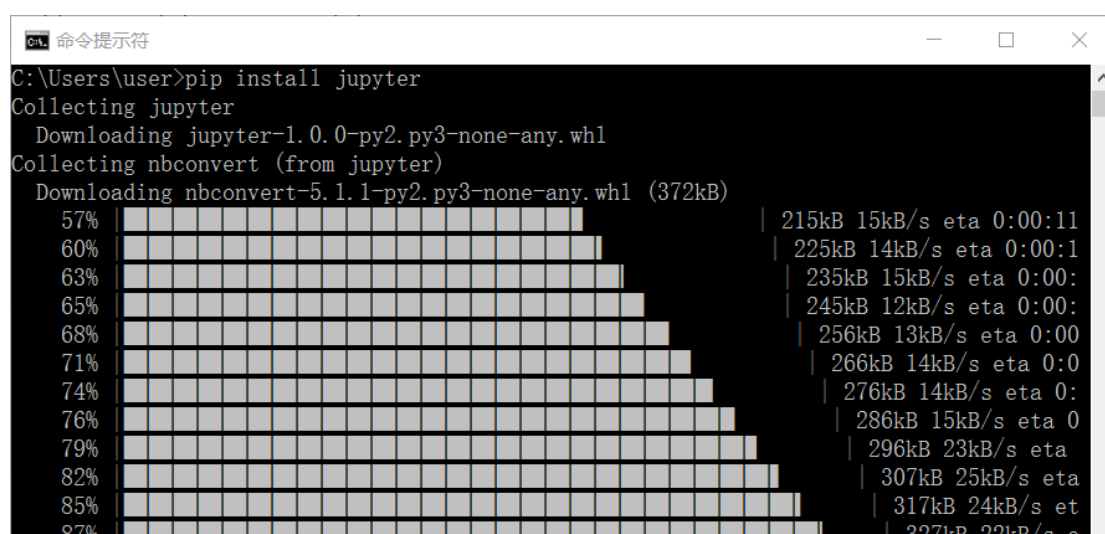


图 12 在线安装 Jupyter Notebook

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。

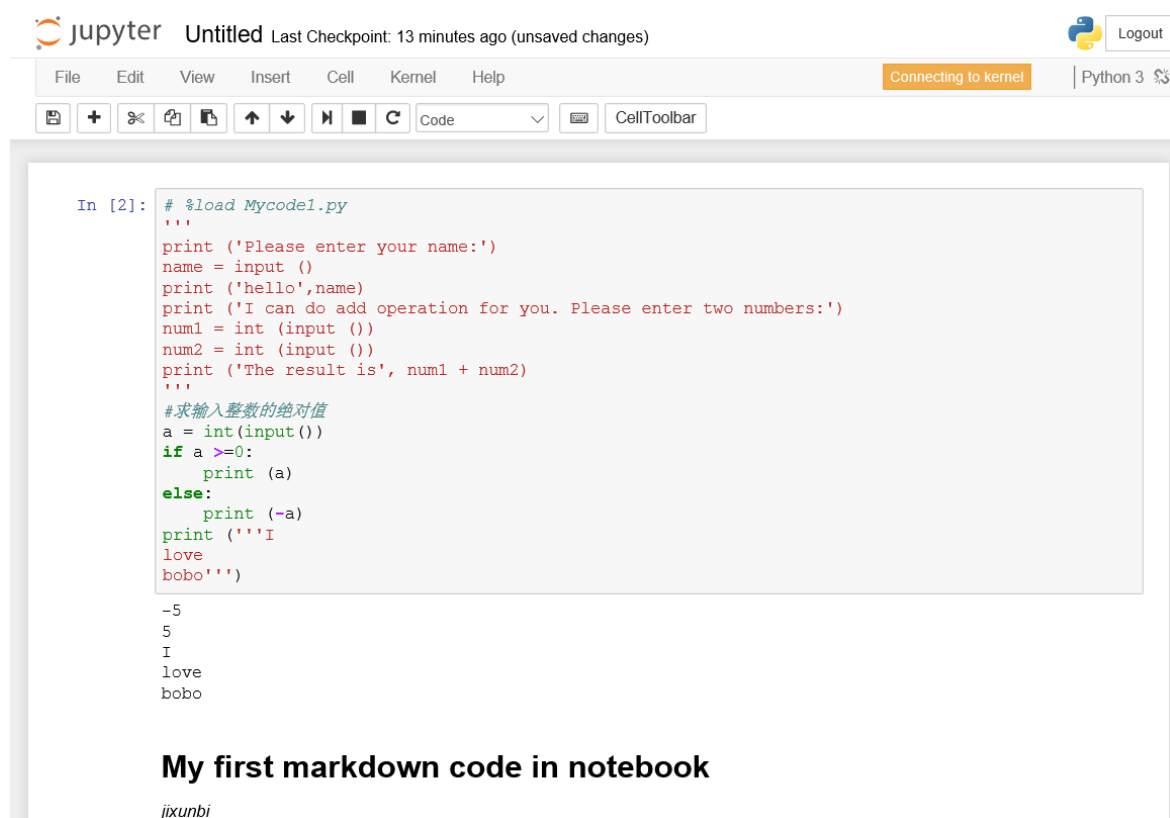


图 13 Jupyter Notebook 使用示例截图

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。

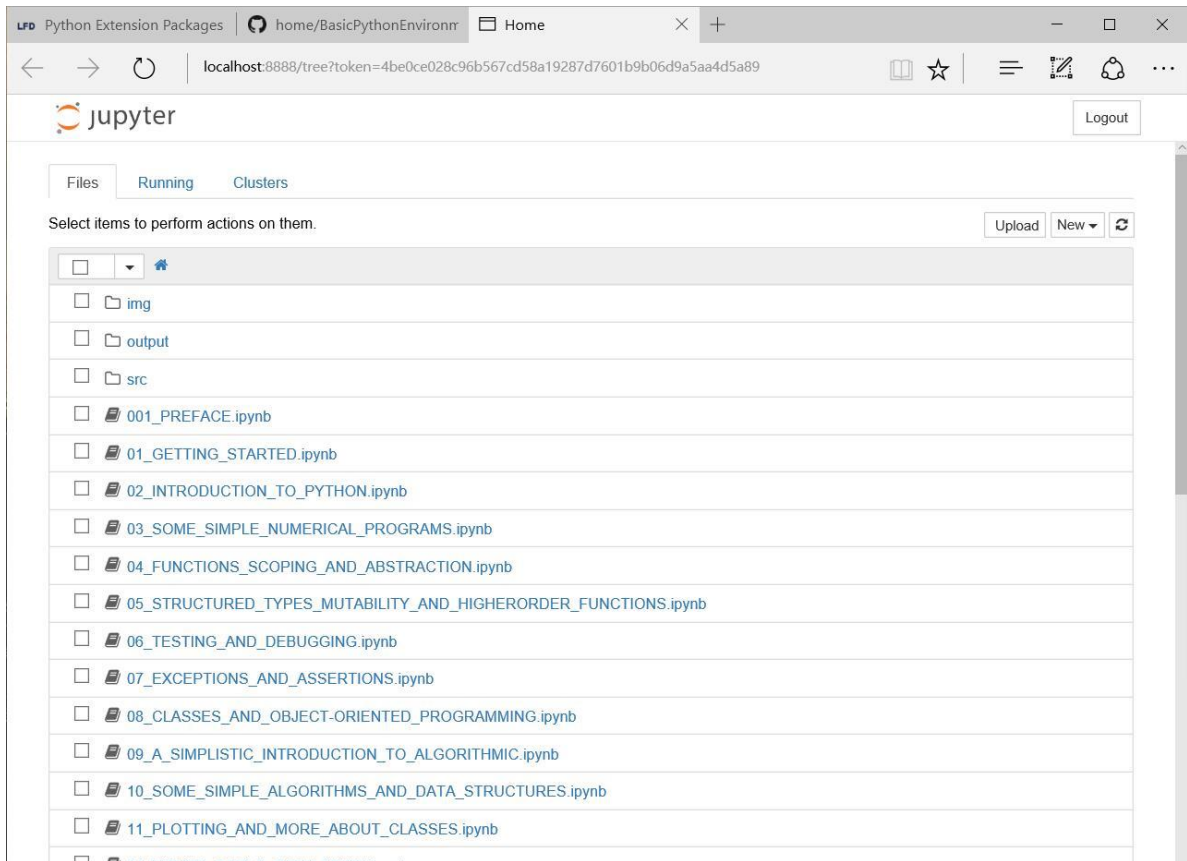


图 14 Jupyter Notebook 运行课件目录的界面截图

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。

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”。

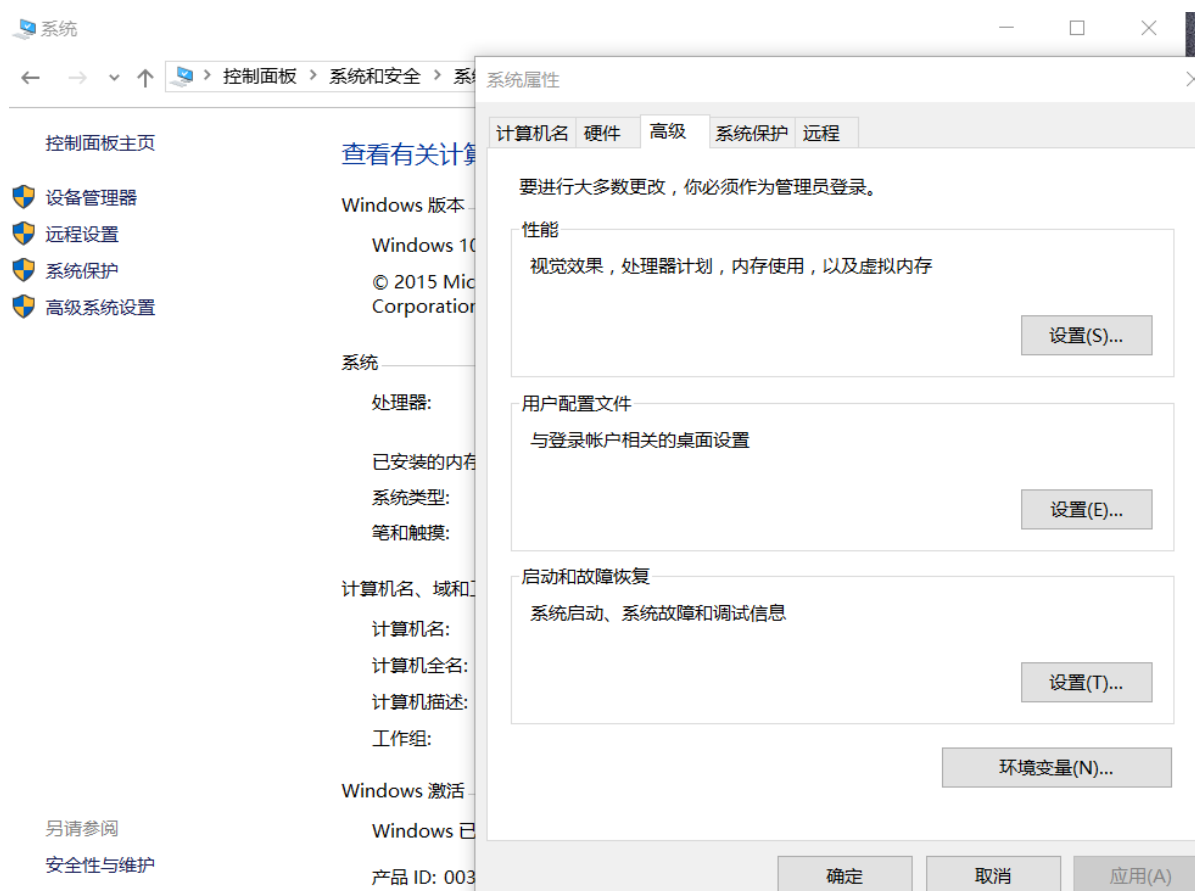


图 15 环境变量所在界面

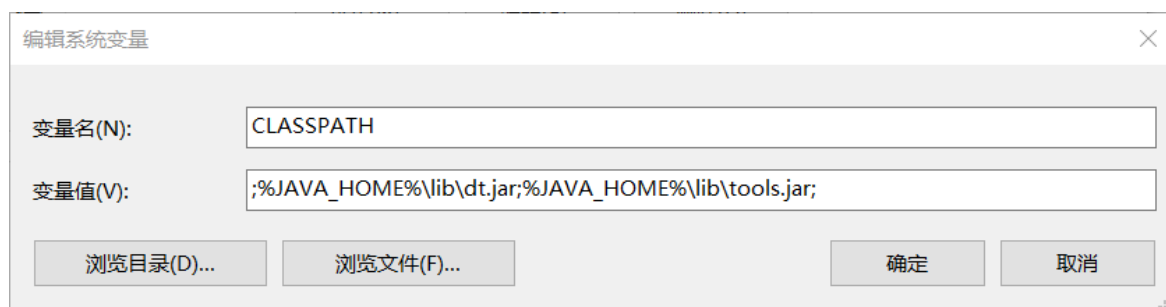


图 16 设置变量名和变量值

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。

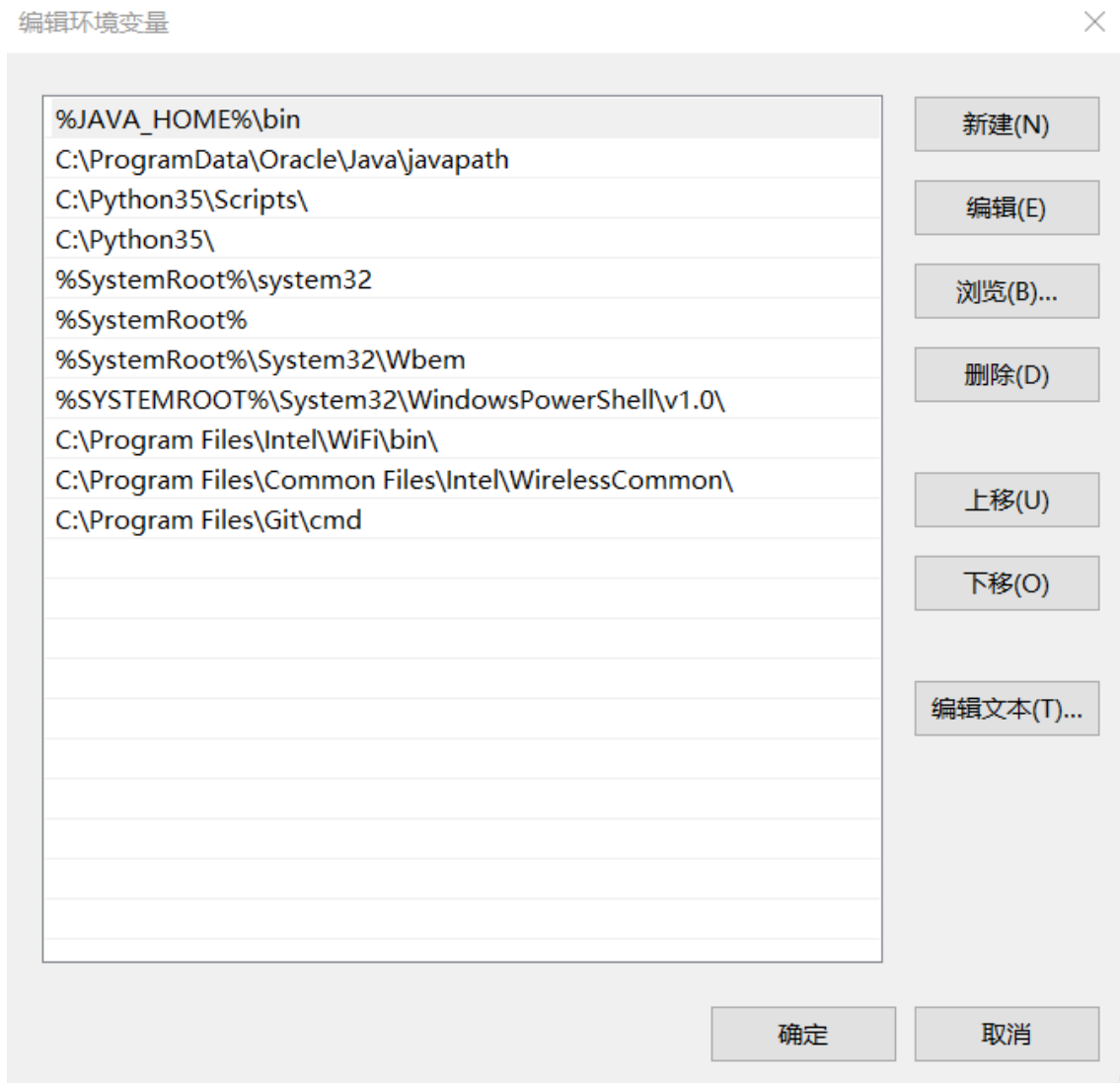


图 17 编辑已有系统变量并新建路径

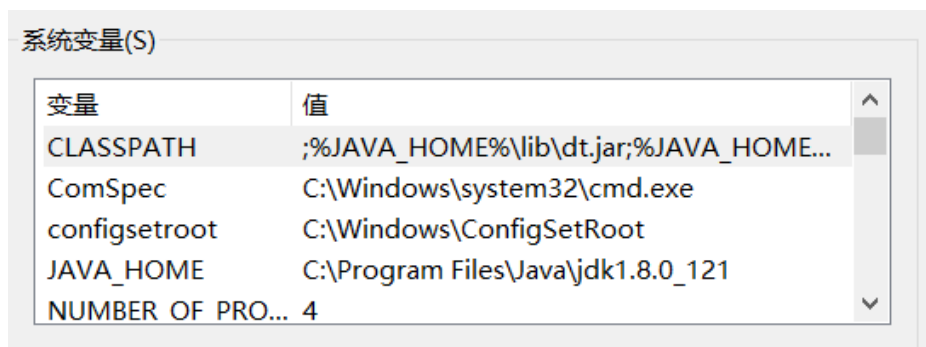


图 18 添加后的系统变量

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。

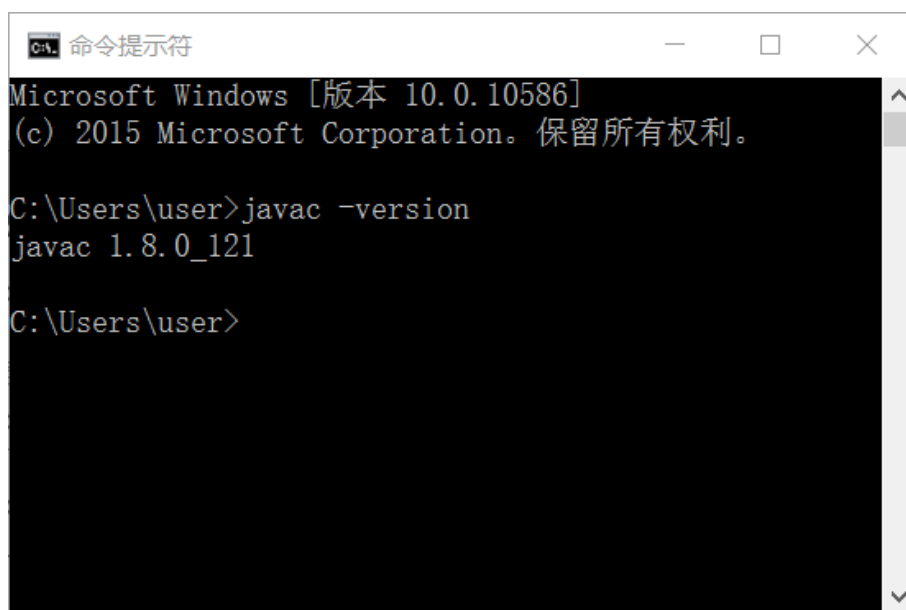


图 19 环境变量配置成功

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->Interpereters->Python Interpreter 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。

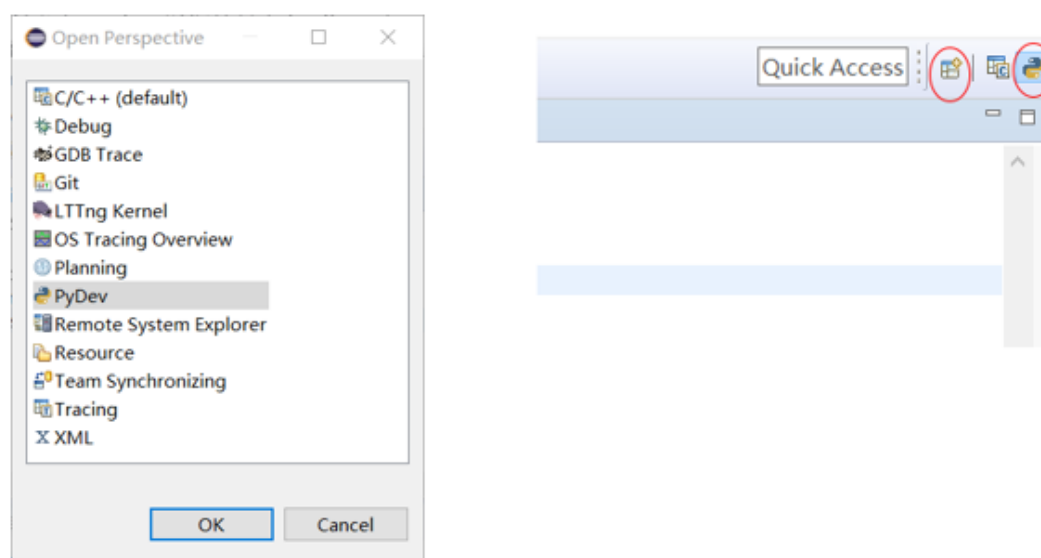


图 20 添加 Python 场景到工作页面

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。

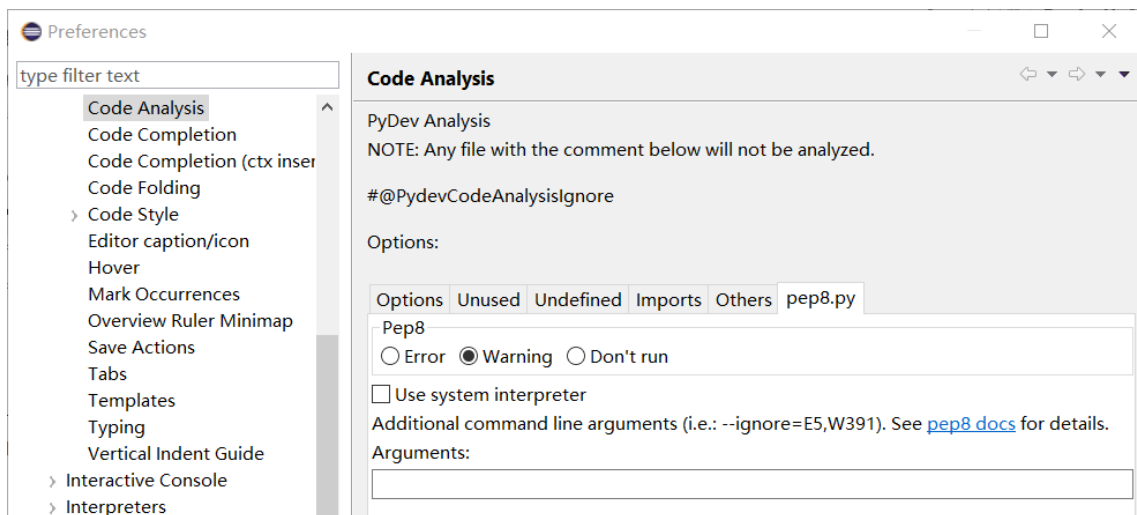


图 21 开启 pep8 检查

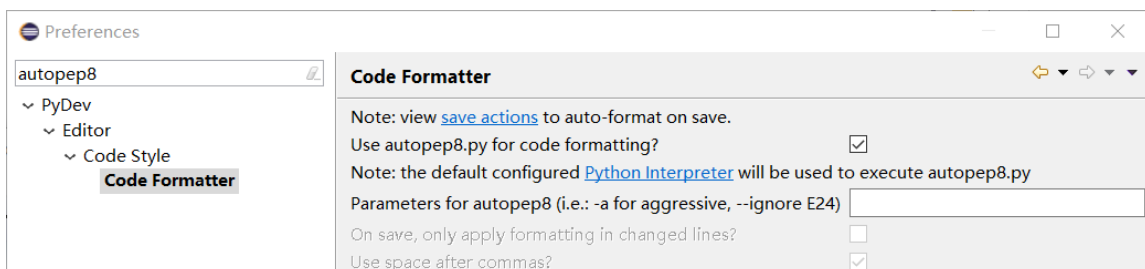


图 22 开启 autopep8

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。

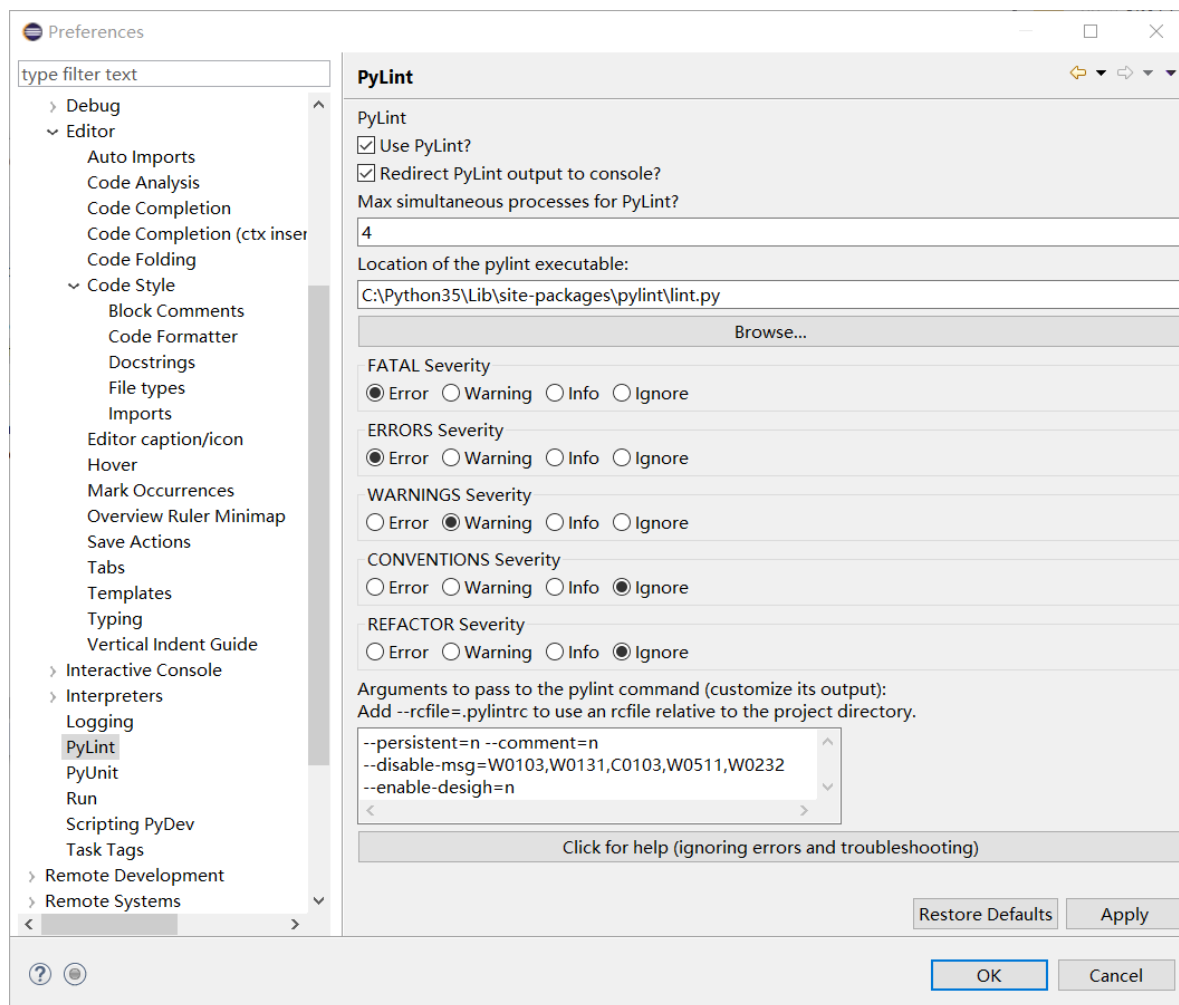


图 23 开启 Pylint

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）。

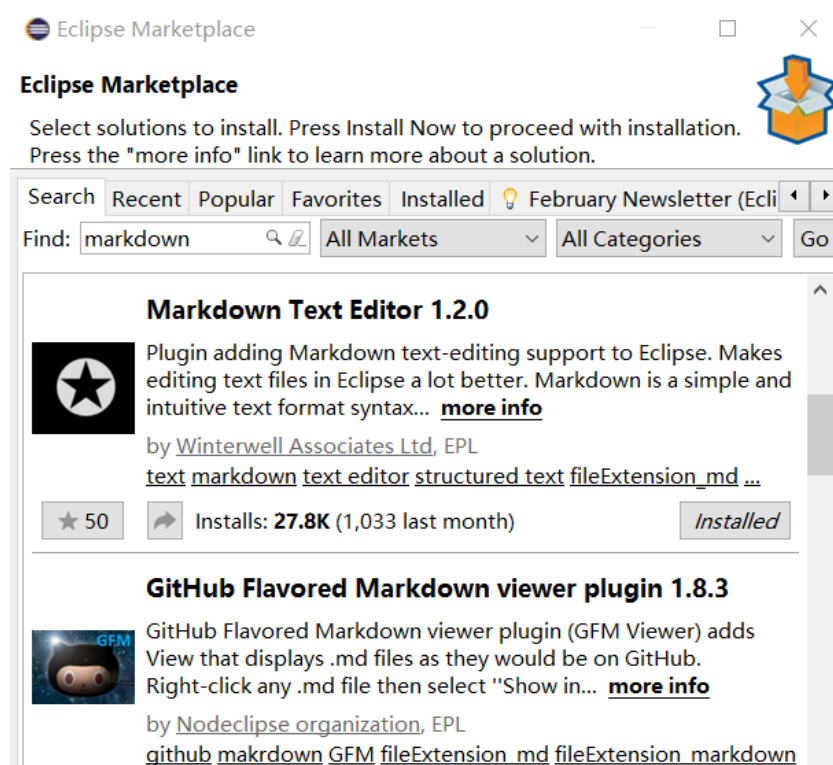


图 24 Eclipse Marketplace（插件安装后）

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。

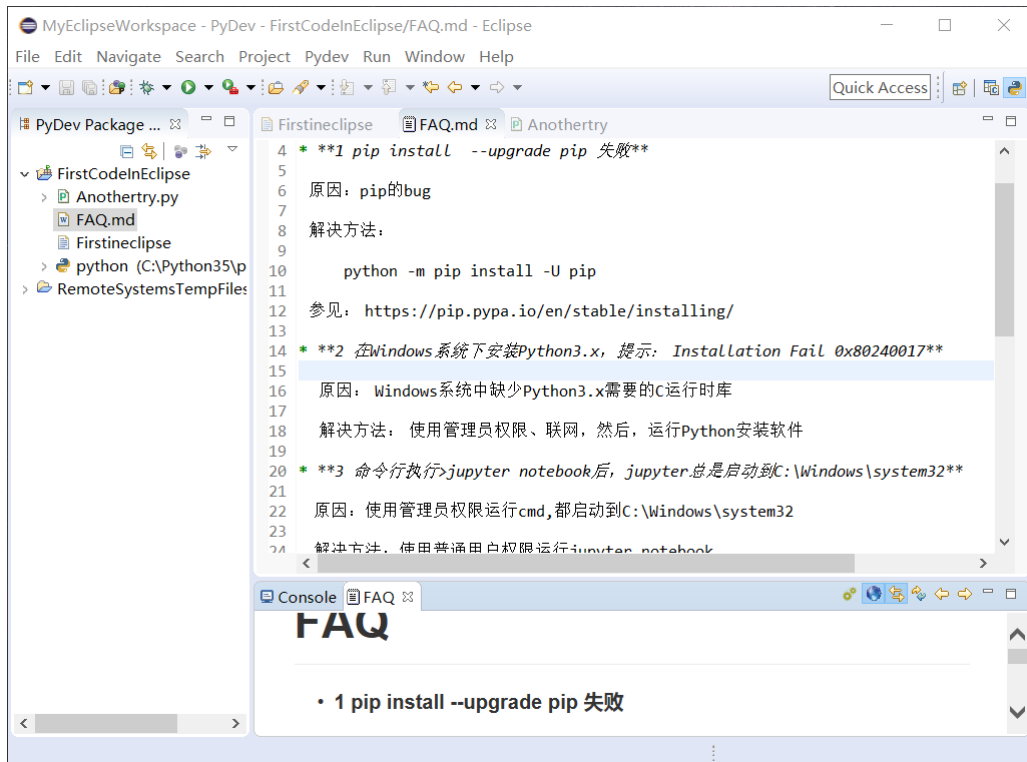


图 25 打开 md 文件编辑和显示视图

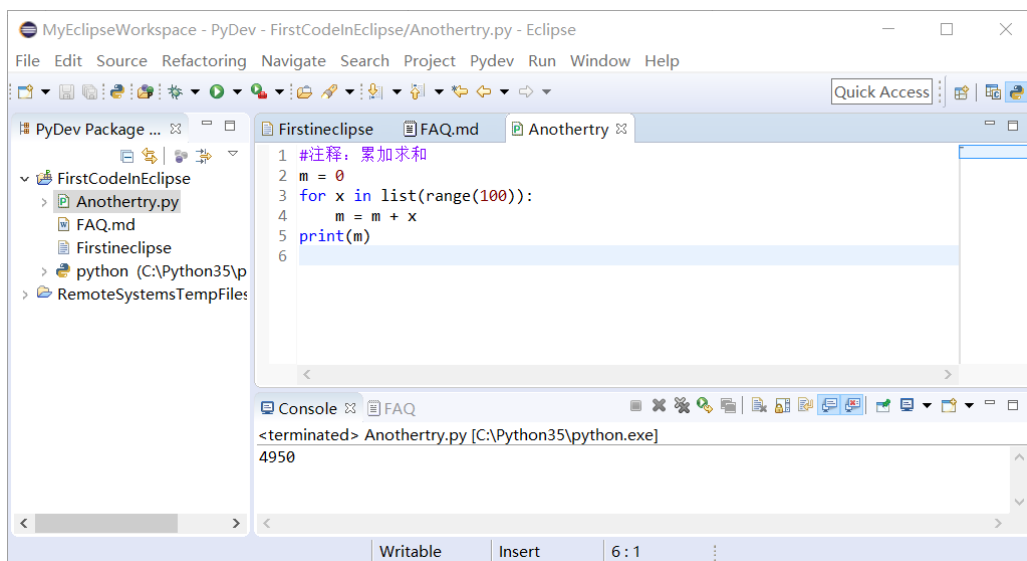


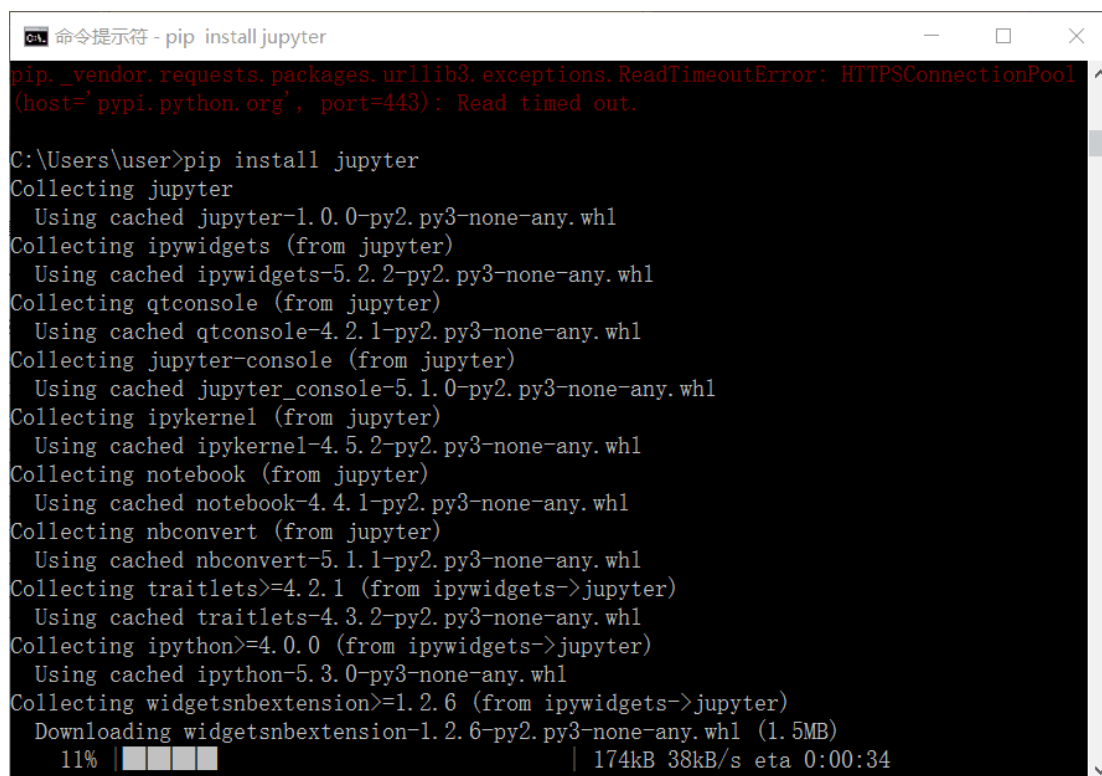
图 26 新建项目文件编写代码及运行结果示例

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

- a) **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。



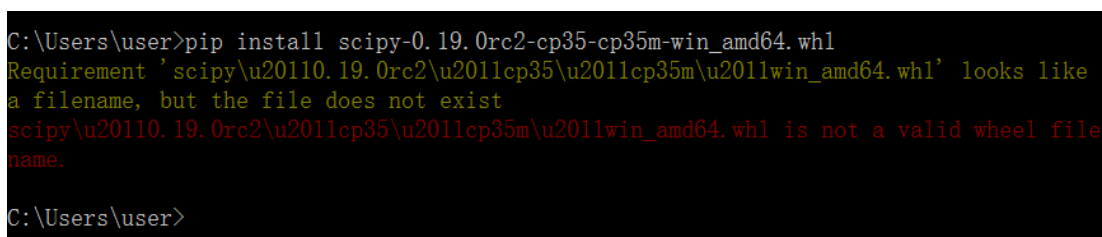
```
命令提示符 - pip install jupyter
pip._vendor.requests.packages.urllib3.exceptions.ReadTimeoutError: HTTPSConnectionPool
(host='pypi.python.org', port=443): Read timed out.

C:\Users\user>pip install jupyter
Collecting jupyter
  Using cached jupyter-1.0.0-py2.py3-none-any.whl
Collecting ipywidgets (from jupyter)
  Using cached ipywidgets-5.2.2-py2.py3-none-any.whl
Collecting qtconsole (from jupyter)
  Using cached qtconsole-4.2.1-py2.py3-none-any.whl
Collecting jupyter-console (from jupyter)
  Using cached jupyter_console-5.1.0-py2.py3-none-any.whl
Collecting ipykernel (from jupyter)
  Using cached ipykernel-4.5.2-py2.py3-none-any.whl
Collecting notebook (from jupyter)
  Using cached notebook-4.4.1-py2.py3-none-any.whl
Collecting nbconvert (from jupyter)
  Using cached nbconvert-5.1.1-py2.py3-none-any.whl
Collecting traitlets>=4.2.1 (from ipywidgets->jupyter)
  Using cached traitlets-4.3.2-py2.py3-none-any.whl
Collecting ipython>=4.0.0 (from ipywidgets->jupyter)
  Using cached ipython-5.3.0-py3-none-any.whl
Collecting widgetsnbextension>=1.2.6 (from ipywidgets->jupyter)
  Downloading widgetsnbextension-1.2.6-py2.py3-none-any.whl (1.5MB)
    11% |■■■■■■■■■■| 174kB 38kB/s eta 0:00:34
```

图 27 失败后第二次安装成功

- b) **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。



```
C:\Users\user>pip install scipy-0.19.0rc2-cp35-cp35m-win_amd64.whl
Requirement 'scipy\u20110.19.0rc2\u2011cp35\u2011cp35m\u2011win_amd64.whl' looks like
a filename, but the file does not exist
scipy\u20110.19.0rc2\u2011cp35\u2011cp35m\u2011win_amd64.whl is not a valid wheel file
name.

C:\Users\user>
```

图 28 安装科学扩展包错误提示

- c) **问题 3:** 安装 Python 语言计算包 iapws 提示如图 29 的错误。

```

C:\Users\user>pip install iapws
Collecting iapws
  Downloading iapws-1.1.3.tar.gz (75kB)
    100% |#####| 81kB 123kB/s
  Complete output from command python setup.py egg_info:
  Traceback (most recent call last):
    File "<string>", line 1, in <module>
    File "C:\Users\user\AppData\Local\Temp\pip-build-xag9lkbd\iapws\setup.py", line 3, in <module>
      from iapws import __version__
    File "C:\Users\user\AppData\Local\Temp\pip-build-xag9lkbd\iapws\iapws\__init__.py", line 5, in <module>
      from .iapws97 import IAPWS97
    File "C:\Users\user\AppData\Local\Temp\pip-build-xag9lkbd\iapws\iapws\iapws97.py", line 9, in <module>
      from scipy.optimize import fsolve, newton
    File "c:\python35\lib\site-packages\scipy\__init__.py", line 61, in <module>
      from numpy._distributor_init import NUMPY_MKL # requires numpy+mkl
  ImportError: cannot import name 'NUMPY_MKL'

  -----
  Command "python setup.py egg_info" failed with error code 1 in C:\Users\user\AppData\Local\Temp\pip-build-xag9lkbd\iapws\
C:\Users\user>pip install iapws
Collecting iapws
  Using cached iapws-1.1.3.tar.gz

```

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。

```

C:\Users\user>cd C:\Python35

C:\Python35>pip install numpy-1.12.0+mkl-cp35-cp35m-win_amd64.whl
Processing c:\python35\numpy-1.12.0+mkl-cp35-cp35m-win_amd64.whl
Installing collected packages: numpy
  Found existing installation: numpy 1.12.0
    Uninstalling numpy-1.12.0:
      Successfully uninstalled numpy-1.12.0
Successfully installed numpy-1.12.0+mkl

C:\Python35>

```

图 30 重新安装 NumPy

- d) **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。

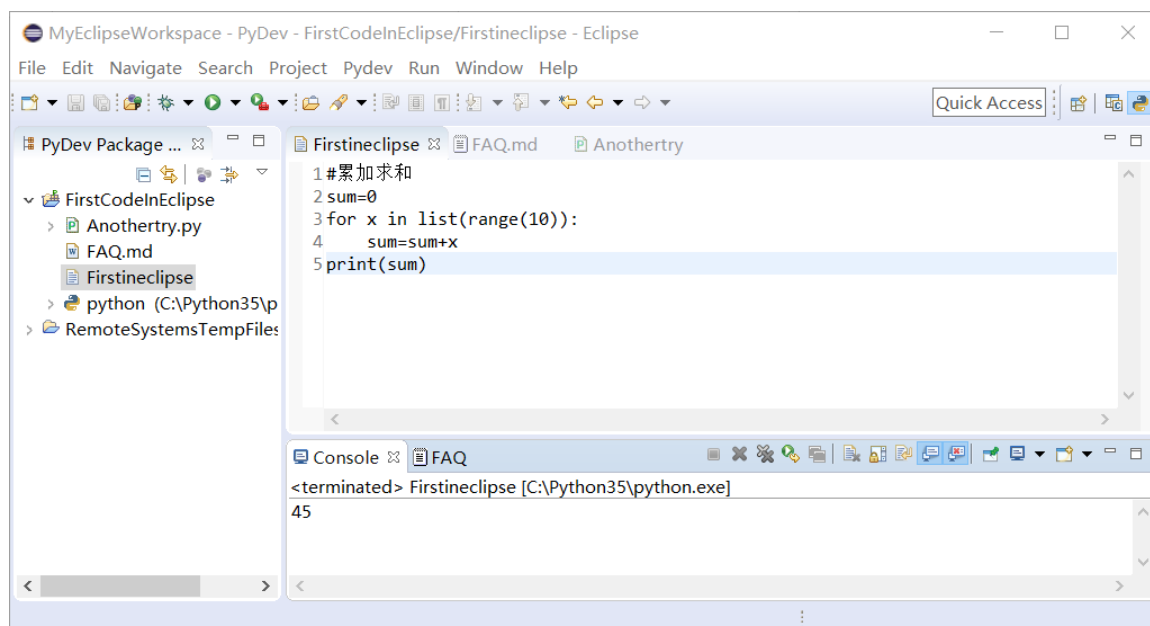


图 31 不加后缀的文件显示

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/RecommendedPracticetree/S2016/P1>

[2] Eclipse 4.3 (Kepler) for C/C++ Programming

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

[3] 郑伟芳. PyDev for Eclipse 简介.

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

[4] 廖雪峰的官方网站-Python 教程

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