# 

# 软件工程基础与实践

**课程实践作业二**

**03014329 孔祥琛**

**能源与环境学院**

[1 Python基本开发环境和程序示例 2](#_Toc476669662)

[1.1 Python语言简介 2](#_Toc476669663)

[1.2 Python安装与程序示例 2](#_Toc476669664)

[**1.2.1** **Python3.5.3安装过程** 2](#_Toc476669665)

[**1.2.2** **Python程序示例** 5](#_Toc476669666)

[2 扩展模块 5](#_Toc476669667)

[2.1 安装科学计算包 5](#_Toc476669668)

[2.2 安装规范格式扩展包 6](#_Toc476669669)

[2.3 IF98物性计算包 7](#_Toc476669670)

[3 交互计算环境Jupyter Notebook 8](#_Toc476669671)

[3.1 Jupyter Notebook安装过程 8](#_Toc476669672)

[3.2 Jupyter Notebook使用示例 8](#_Toc476669673)

[4 集成开发环境 Eclipse 10](#_Toc476669674)

[4.1安装和配置Eclipse 10](#_Toc476669675)

[**4.1.1安装Java SDK** 10](#_Toc476669676)

[**4.1.2 安装Eclipse** 11](#_Toc476669677)

[**4.1.3 配置Eclipse工作空间** 11](#_Toc476669678)

[4.2 Python开发插件PyDev 12](#_Toc476669679)

[**4.2.1 安装PyDev** 12](#_Toc476669680)

[**4.2.2 配置Python解释器** 12](#_Toc476669681)

[5 基于PyDev的语言规范静态检查 12](#_Toc476669682)

[5.1 PEP8检查和修改 13](#_Toc476669683)

[5.2 使用PyLint 14](#_Toc476669684)

[6 Eclipse的使用 14](#_Toc476669685)

[6.1 编程的准备工作 15](#_Toc476669686)

[6.2 Eclipse编程示例 16](#_Toc476669687)

[7 建立开发环境等过程中遇到的问题和解决办法 19](#_Toc476669688)

[8 总结 19](#_Toc476669689)

[9 参考资料 19](#_Toc476669690)

# 1 Python基本开发环境和程序示例

* 1. **Python语言简介**

Python（英国发音：/ˈpaɪθən/ 美国发音：/ˈpaɪθɑːn/）, 是一种面向对象的解释型计算机程序设计语言，由荷兰人Guido van Rossum于1989年发明，第一个公开发行版发行于1991年。

Python是纯粹的自由软件， 源代码和解释器CPython遵循 GPL(GNU General Public License)协议。

Python语法简洁清晰，特色之一是强制用空白符(white space)作为语句缩进。

Python具有丰富和强大的库。它常被昵称为胶水语言，能够把用其他语言制作的各种模块（尤其是C/C++）很轻松地联结在一起。常见的一种应用情形是，使用Python快速生成程序的原型（有时甚至是程序的最终界面），然后对其中有特别要求的部分，用更合适的语言改写，比如3D游戏中的图形渲染模块，性能要求特别高，就可以用C/C++重写，而后封装为Python可以调用的扩展类库。需要注意的是在您使用扩展类库时可能需要考虑平台问题，某些可能不提供跨平台的实现。

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。

* 1. **Python安装与程序示例**
     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

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）。

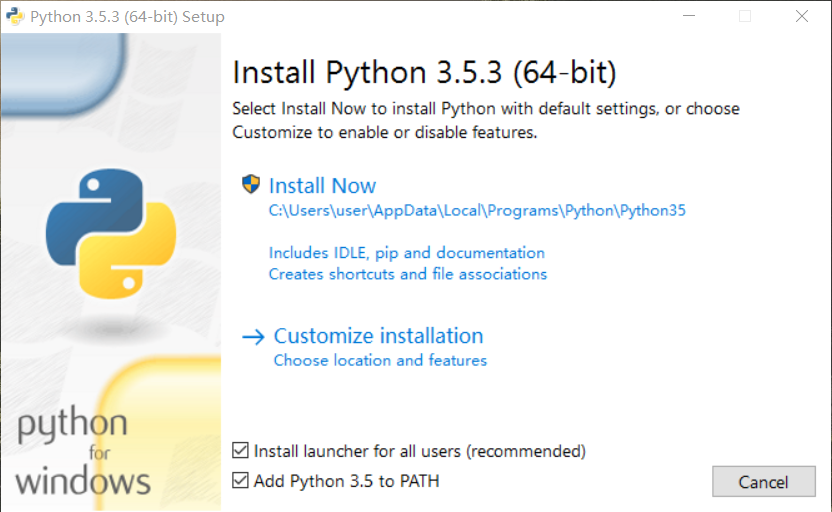


图2

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。如图3。

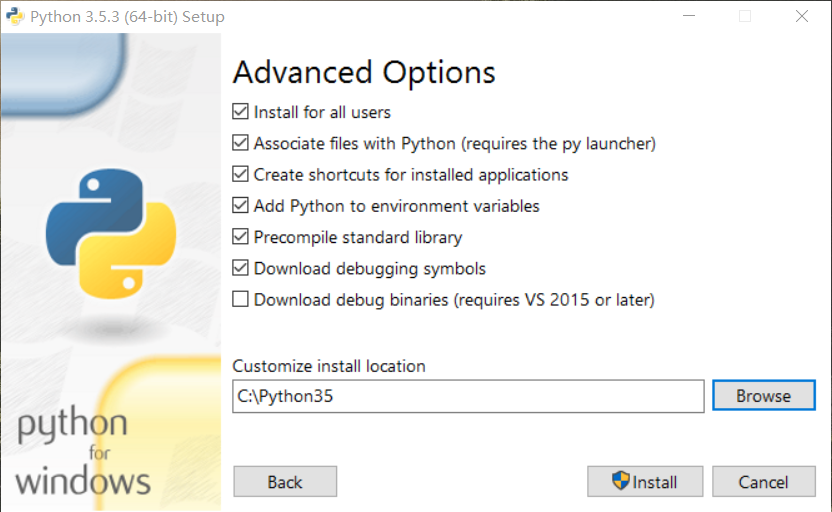


图3

经过一段时间等待，安装完成，如图4。

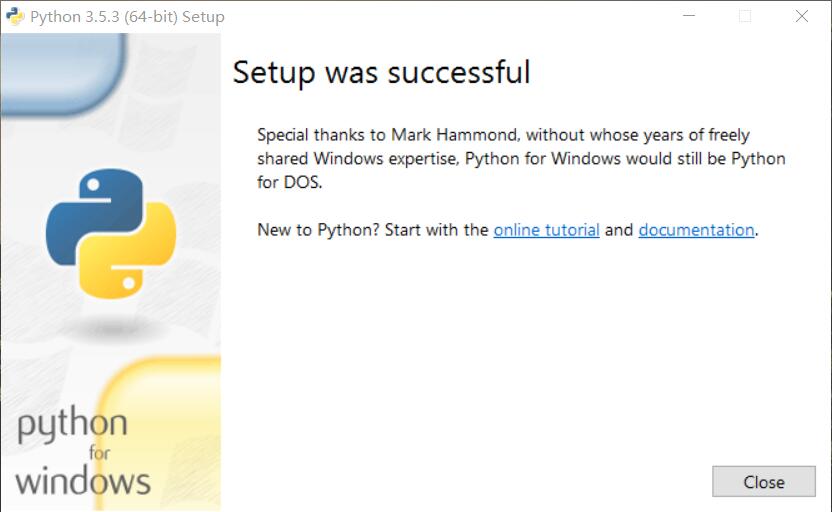


图4

接下来打开DOS，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，如图5。

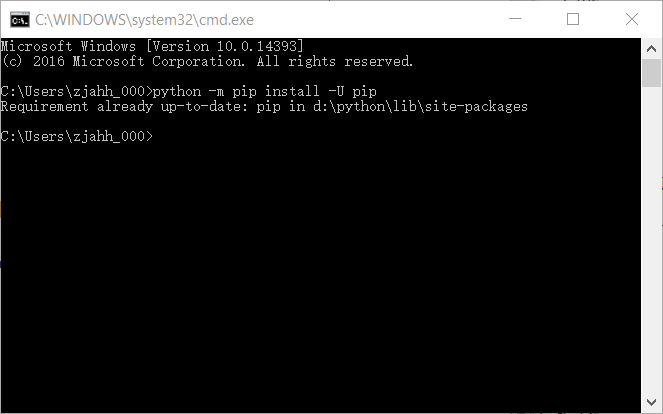


图5

* + 1. **Python程序示例**

可以通过IDLE编写代码，如图6。

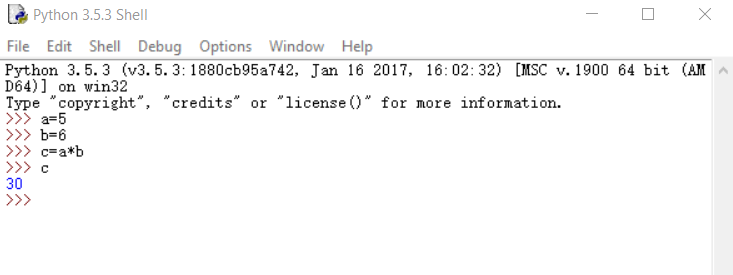


图6

同时，在DOS上也可以直接运行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，如图7。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。

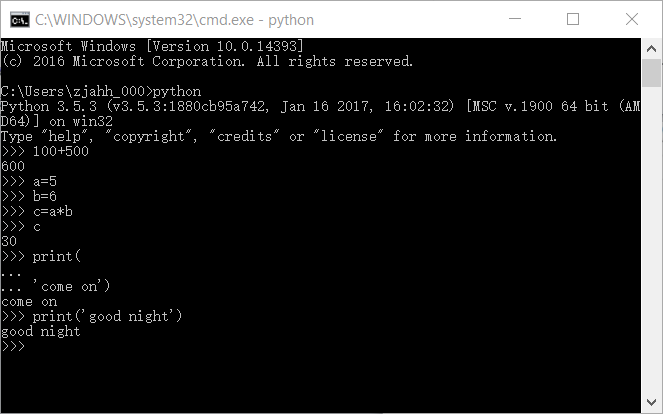


图7

1. **扩展模块**
   1. **安装科学计算包**

在这我们只安装基本的科学计算包 SciPy、NumPy和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SciPy为例。

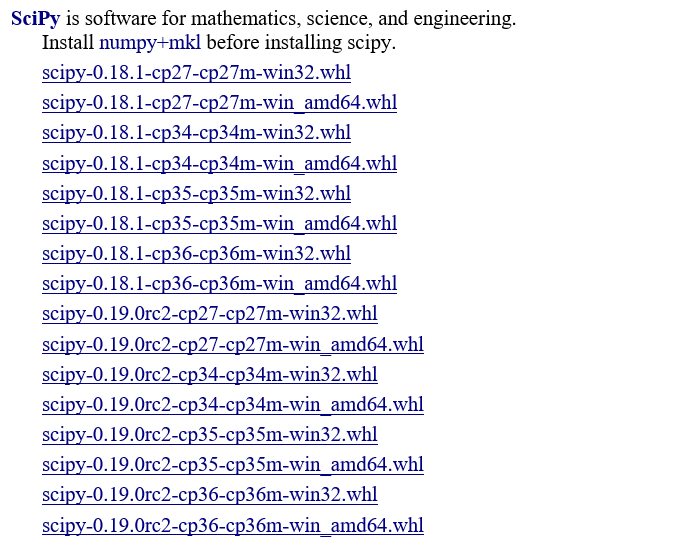
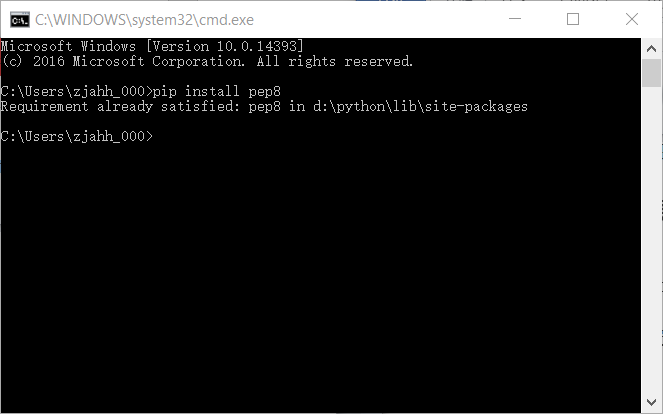


图8

打开DOS，输入 >pip install \*.whl。\*用扩展包名字替换，如：>pip install scipy-0.18.1-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。下载完SciPy、NumPy和Matplotlib对应的whl后，按如上过程依次安装即可。

* 1. **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。如图9，已经事先安装完成了。

图9

* 1. **IF98物性计算包**

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libseuif97.dll和 seuif97.py，seuif97.dll拷贝到 c:\windows\system，seuif97.py拷贝到 c:\python35\Lib。

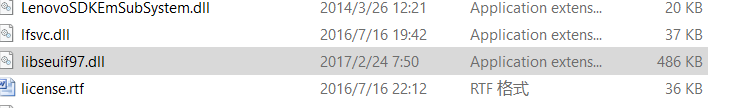


图10

![C:\Users\zjahh_000\AppData\Roaming\Tencent\Users\1324752516\QQ\WinTemp\RichOle\86X60@}CVV1@](FNW2MH1[4.png](data:image/png;base64,)

图11

Python语言计算包，在线安装：>pip install iapws

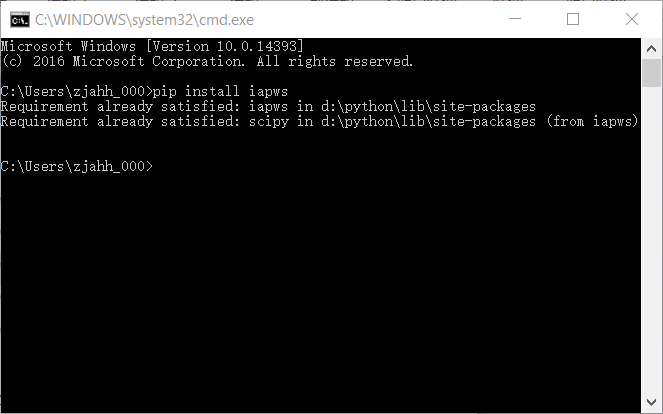


图12

1. **交互计算环境Jupyter Notebook**
   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。

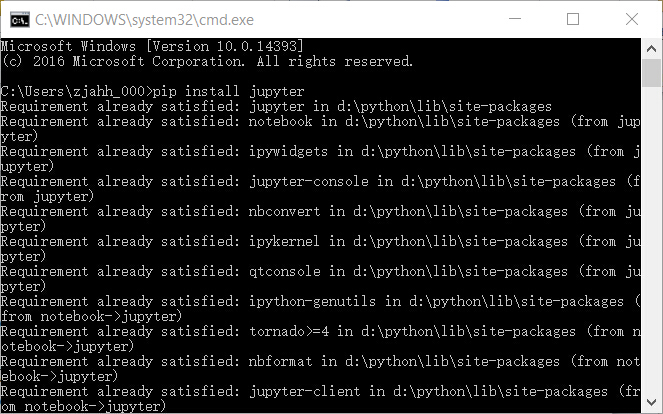


图13

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。

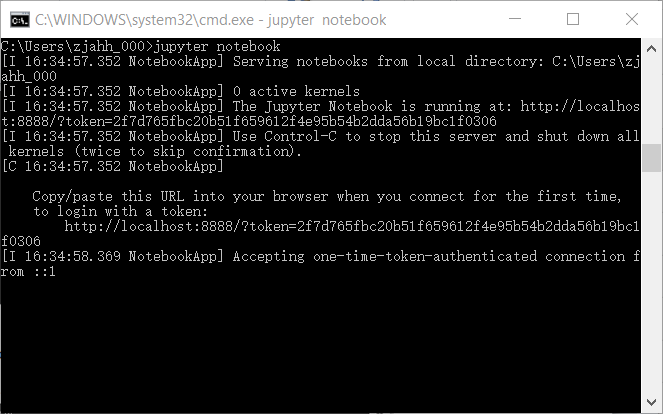


图14

* 1. **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。

C:\Users\zjahh_000\AppData\Roaming\Tencent\Users\1324752516\QQ\WinTemp\RichOle\1~`M4FQ~I]TZ5IHADS6ZP~7.png

图15

我们可以写一个简单的程序，之后按SHIFT+ENTER即可运行，运行结果会显示在下方，如图16。

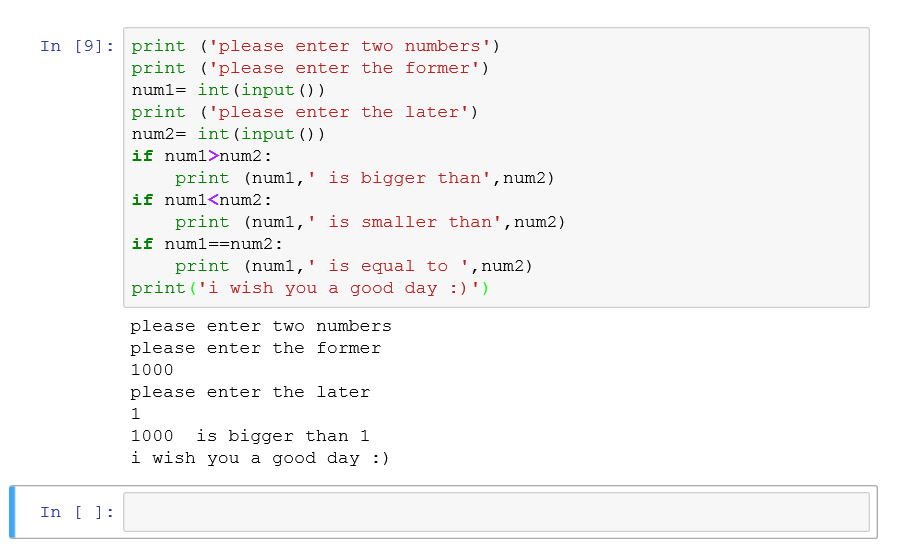


图16

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。

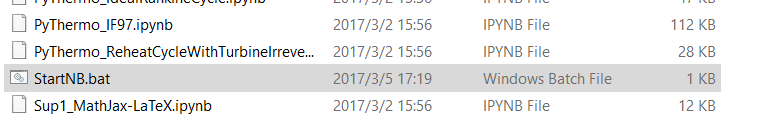


图17

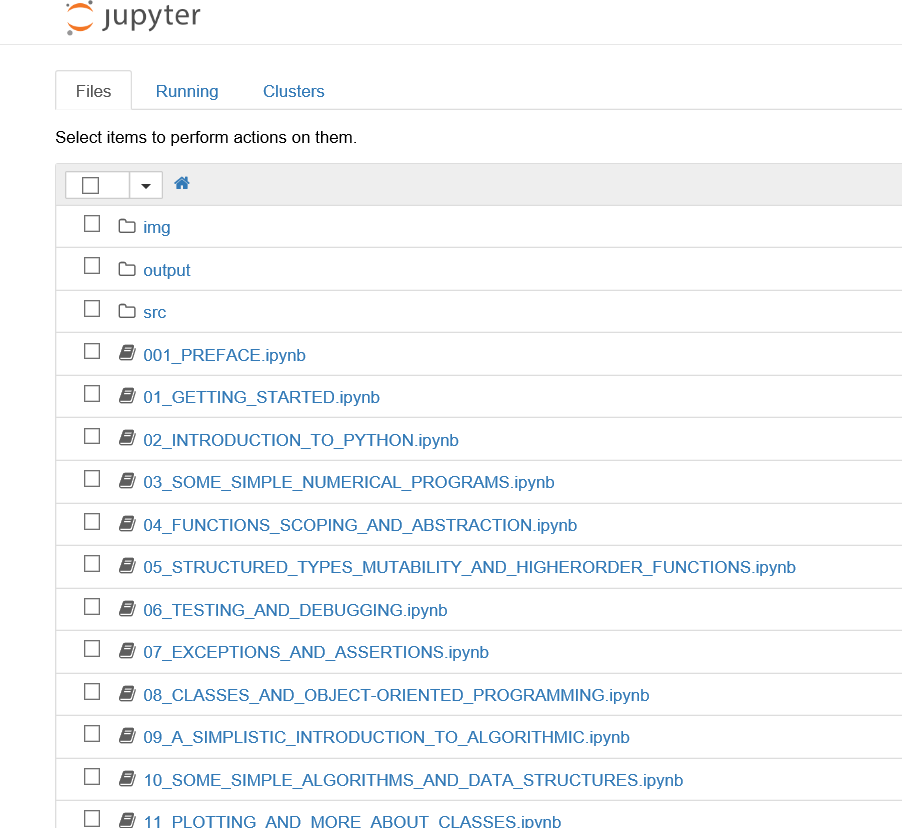


图18

1. **集成开发环境 Eclipse**

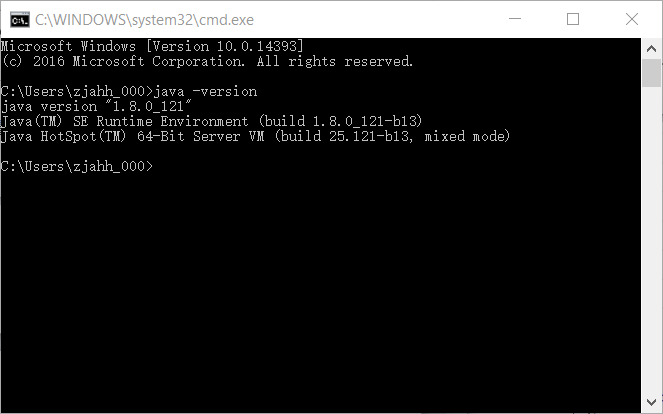
**4.1安装和配置Eclipse**

**4.1.1安装Java SDK**

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。



**4.1.2 安装Eclipse**

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。



图20

**4.1.3 配置Eclipse工作空间**

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。

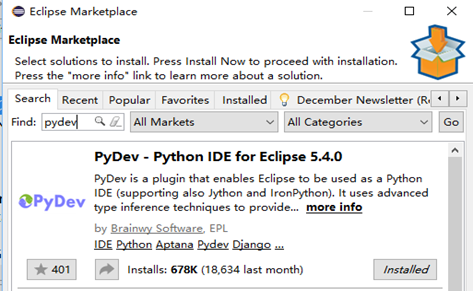
建议配置自己的工作空间：首先，在非系统盘中，如D盘中建立一个目录，如:D:\myworkspace，然后，打开Eclipse，在其提示设置默认的工作空间时，“Browser”到D:\myworkspace，勾选“Use this as the default and do not ask again",就将该目录配置为Eclipse默认当前工作空间目录了，你开发的程序代码等都在这个目录下。

**4.2 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）配置使用的Python解释器版本。

**4.2.1 安装PyDev**

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

图21

**4.2.2 配置Python解释器**

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，添加Python场景到工作页面。

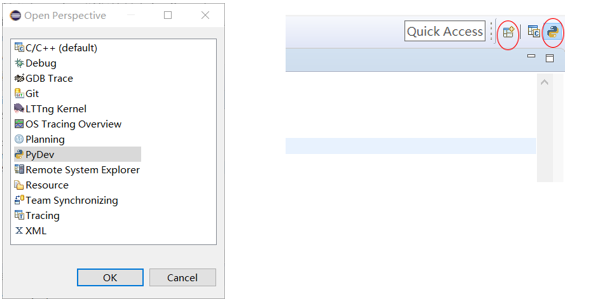


图22

接下来就可以进行Python开发了。当然，为了使开发过程效率更高，可以修改编译器颜色来提高可读性、添加任务标签注释等。

1. **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。

* 1. **PEP8检查和修改**

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。

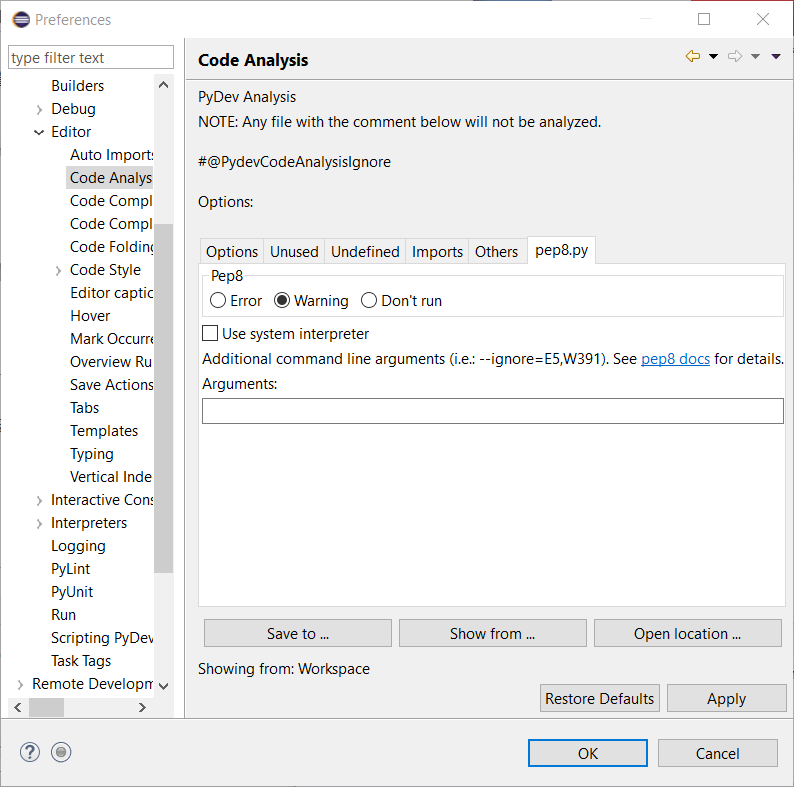


图23

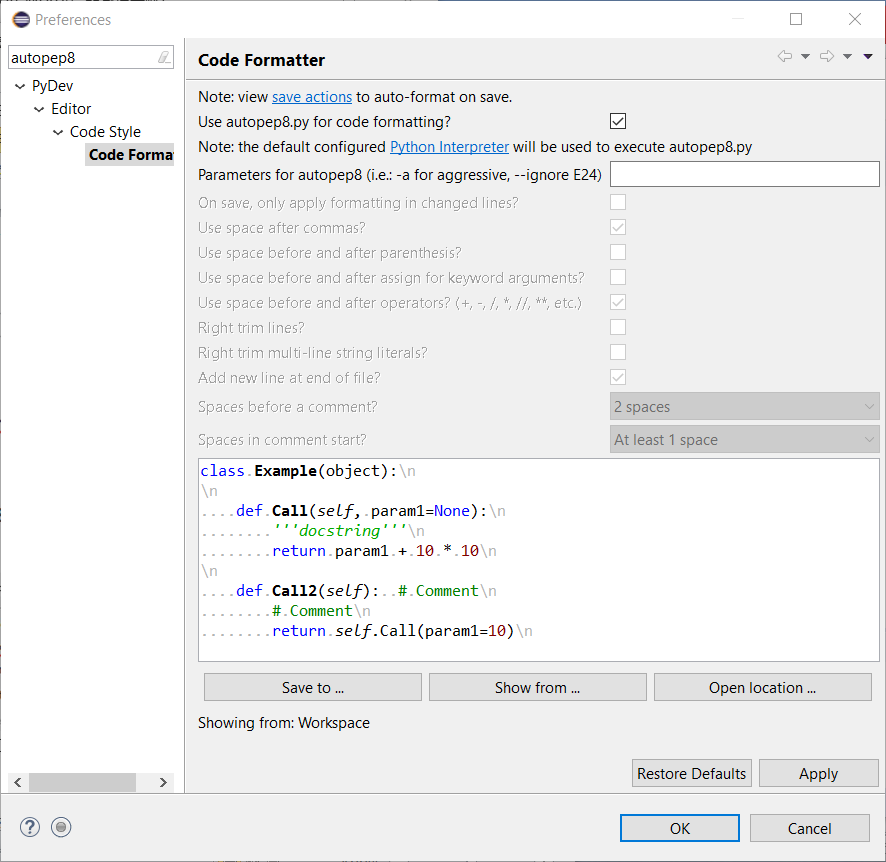


图24

* 1. **使用PyLint**

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。

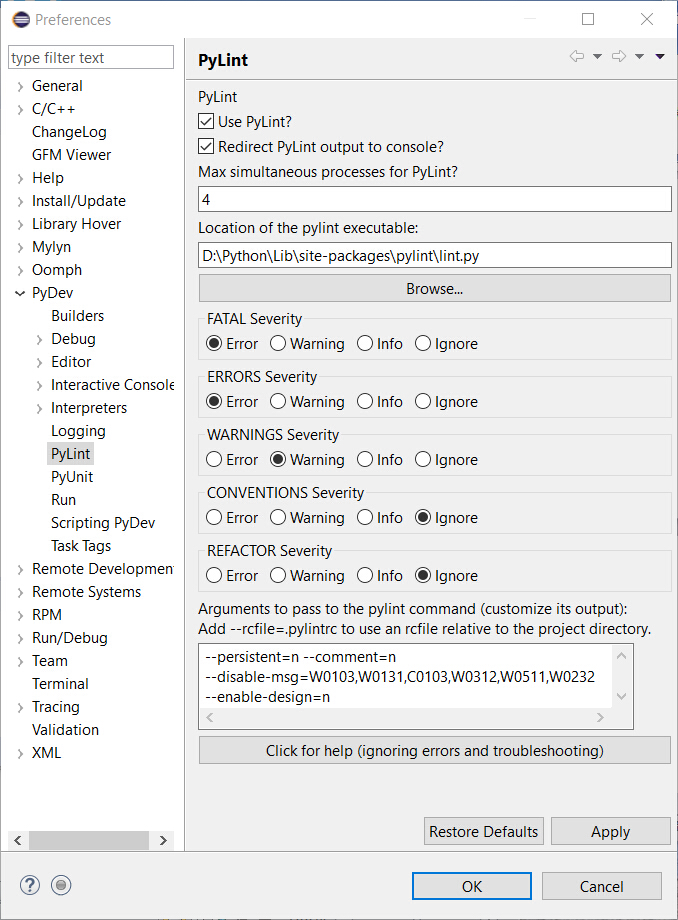


图25

1. **Eclipse的使用**

要更高效的使用Eclipse，除了以上的配置外，还有许多可以改进的地方，比如设置工作空间的编码方式，可以有效避免含中文字符的文件在其他文本编辑器打开时出现乱码，建议配置编码为UTF-8。编码方式最好配置到工作空间和工程集成工作空间。具体步骤为：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)。

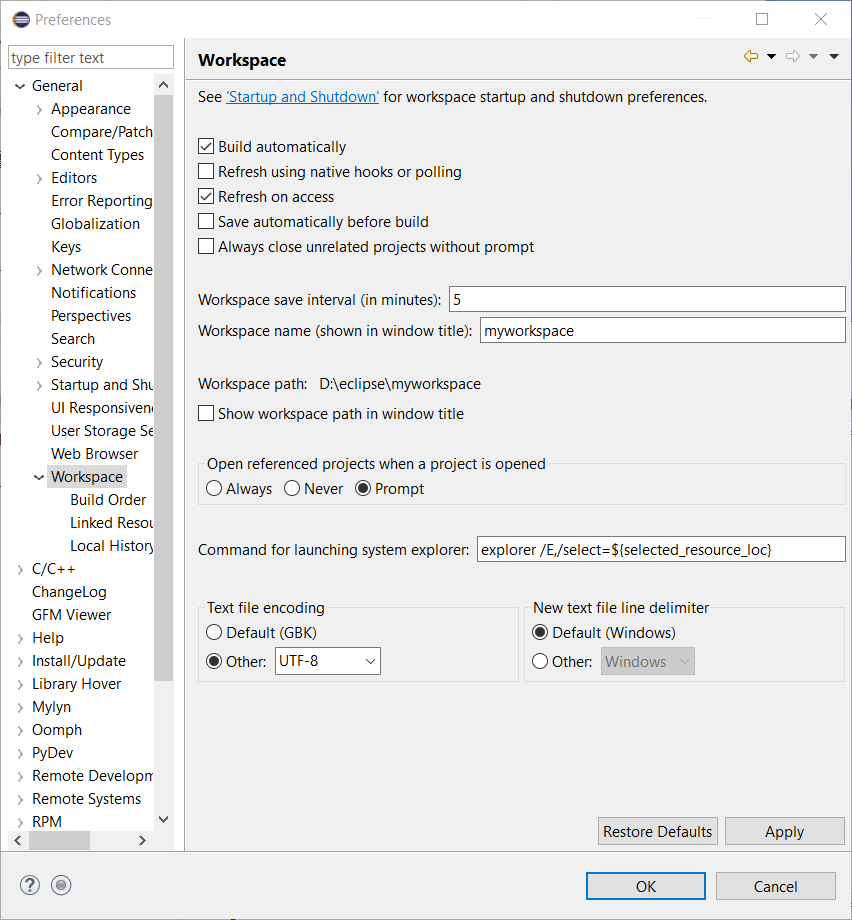


图26

**6.1 编程的准备工作**

为了能够在eclipse上使用熟悉的C/C++语言进行编译，需要安装一个编译器，这里我们选择MinGW，安装过程可以在

<https://github.com/PySEE/home/blob/S2017/guide/ProgrammingWithGCC.md>

查看。



图27

安装完成后，将MinGW的路径添加到环境变量，就可以正常使用了。

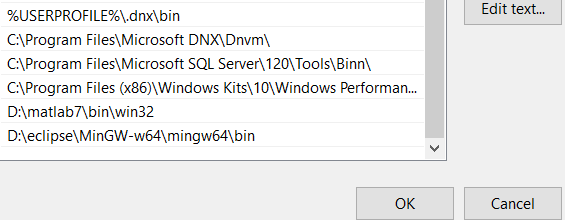
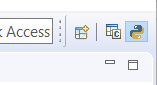
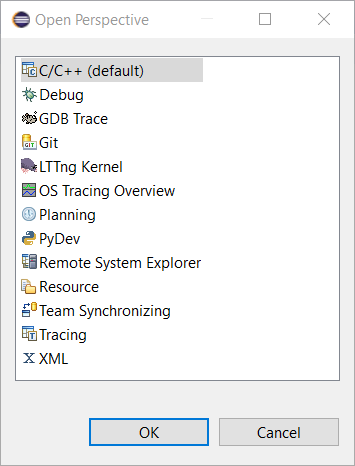


图28

**6.2 Eclipse编程示例**

全部安装完成后，进去Eclipse，将编译环境切换到C++，点击图中左侧的按钮

图29 图30

为了方便，就直接新建一个hello word项目，以供测试，可以看到，多出了一个安装过的MinGW

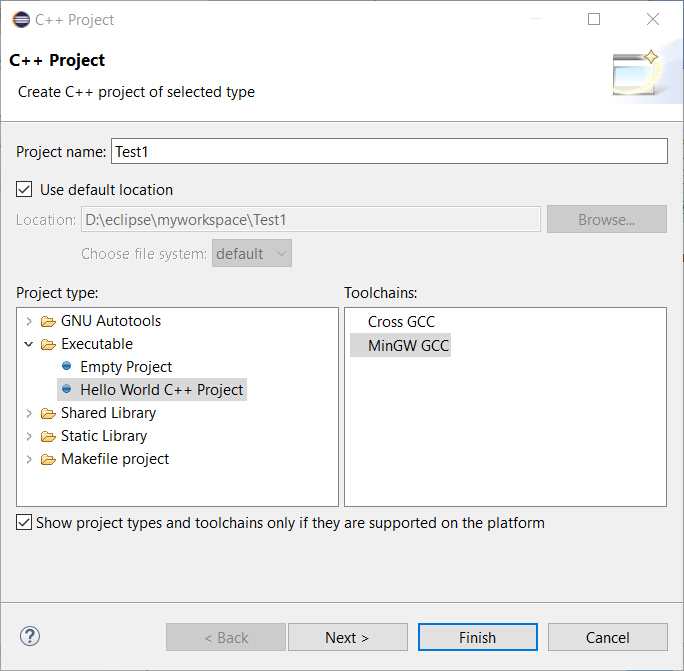


图31

编译运行后，就得到了我们熟悉的画面！

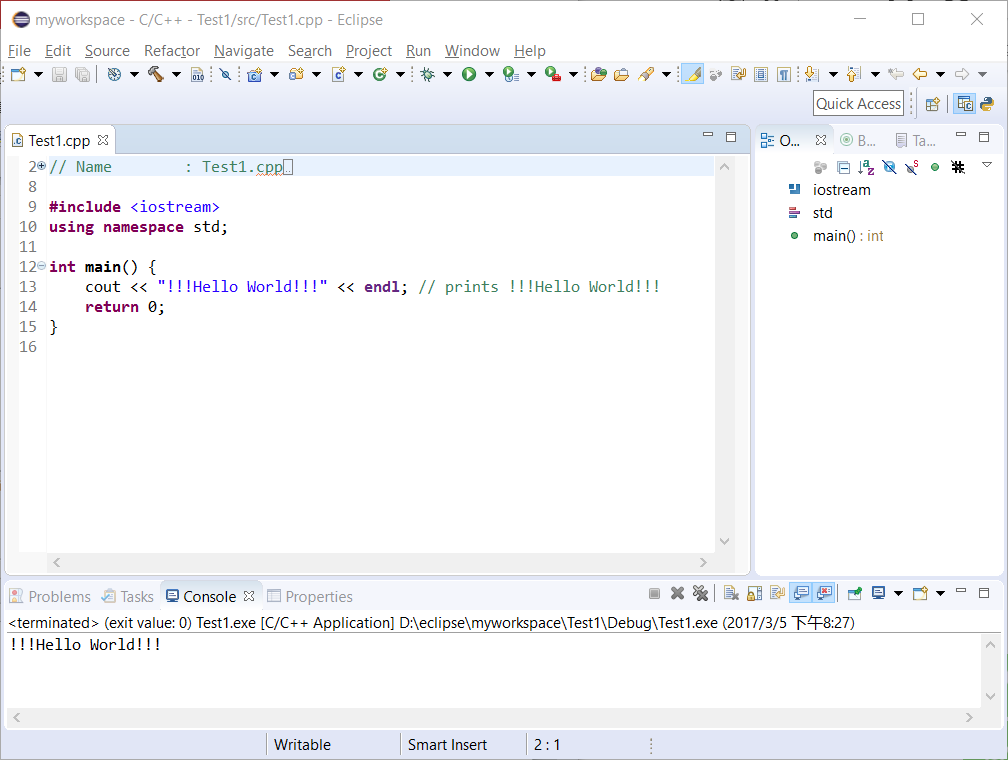


图32

如果想要自己写程序，则创建一个empty的项目，再向项目中添加源文件，注意，在添加源文件时，注意在文件名称后加文件类型.cpp。

编写完成后，可能会出现例如 “cout” could not be resolved 的错误，这时可以根据"Window -> Preferences -> C/C++ -> Build -> Environment", 增加两个新的环境变量： "LANG" and "LC\_ALL" 并把它们的值都设成是： "en\_US"。 如果还有错误提示，保存下工程，错误的提示就会消失了。

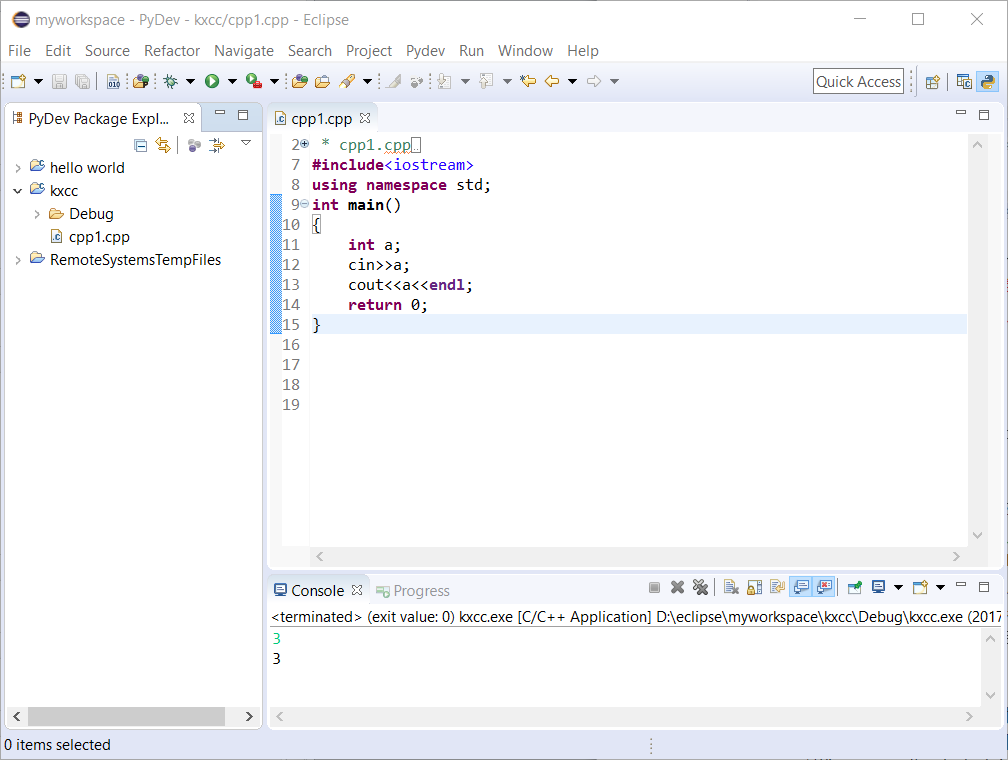


图32

# 7 建立开发环境等过程中遇到的问题和解决办法

1. 下载安装Jupyter Notebook时，出现超时现象怎么办？

解决方法：找一个网速好一点的地方再试一遍就好了。

1. 我只会C++语言，要如何在Eclipse上编程呢？

解决方法：Eclipse只是一个集成平台，如果要使用某种语言，如Java、C++等进行编译，需要创造相应的环境，对于C++语言，需要在Market中安装CKT，并且下载安装名为MinGW（也可以用别的编译器），之后便可以使用熟悉的C++语言在Eclipse上写程序了。

1. **总结**

说实话，刚开始接触这些软件的时候，我完全找不着北，只是按照老师给的步骤，一点点下载安装，却不知道具体如何使用，因为我只会C++语言，然而Eclipse是以Java为基础的平台，所以不知所措。之后，花了很大功夫在网上学习，终于知道了如何用C++在Eclipse上写程序。这个过程很麻烦，但却让我认识到，我之前所接触到的关于软件工程的知识，实在太狭隘了，原来软件知识是如此丰富，人们为了更方便的设计软件，竟能创造出如此多的工具。真希望能够通过这门课程，学到更多关于软件的知识，刷新我对软件领域的认知。

1. **参考资料**
2. 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>

[5] C/C++ Programming with GCC (for Windows)

<https://github.com/PySEE/home/blob/S2017/guide/ProgrammingWithGCC.md>