软件工程基础及实践

**Python学习和开发环境的建立**

03014421 黄阳鹏

[年]

目录

[1. Python的简介及开发环境建立 3](#_Toc477721275)

[1.1 Python的简介 3](#_Toc477721276)

[1.2 python基本开发环境 4](#_Toc477721277)

[1.2.1 Python软件的安装 4](#_Toc477721278)

[1.2.2 使用IDLE编程 4](#_Toc477721279)

[2.python扩展软件包的安装 5](#_Toc477721280)

[2.1科学计算扩展包安装 5](#_Toc477721281)

[2.2过程中出现的问题： 6](#_Toc477721282)

[3.交互计算 Jupyter 7](#_Toc477721283)

[3.1交互计算示例 8](#_Toc477721284)

[4.选择安装编程软件 9](#_Toc477721285)

[4.1 Pycharm 9](#_Toc477721286)

[4.2 Visual Studio Code 10](#_Toc477721287)

[4.2.1 基本环境的搭建 10](#_Toc477721288)

[4.2.2 功能特点 11](#_Toc477721289)

[4.2.3Git 11](#_Toc477721290)

[4.3 Eclipse + Pydev 11](#_Toc477721291)

[4.3.1安装Java SDK 11](#_Toc477721292)

[4.3.2 安装Eclipse 13](#_Toc477721293)

[4.3.3 配置Eclipse工作空间 14](#_Toc477721294)

[4.3.4Python开发插件PyDev 14](#_Toc477721295)

[4.4配置PyDev以提高效率 16](#_Toc477721296)

[4.4 1修改编辑器配色提高可读行 16](#_Toc477721297)

[4.4.2 配置任务标签 16](#_Toc477721298)

[4.5 PEP8检查和修改 19](#_Toc477721299)

[4.6 使用Pylint 20](#_Toc477721300)

[4.7 Eclipse配置和使用 21](#_Toc477721301)

[4.7.1 文件字符编码方式 21](#_Toc477721302)

[4.7.2 插件补充和更新 21](#_Toc477721303)

[4.7.3 联机指南 22](#_Toc477721304)

[4.7.4 Markdown插件 22](#_Toc477721305)

[4.8 出现的错误 23](#_Toc477721306)

[5.Python 编程示例 24](#_Toc477721307)

[6.个人小结 24](#_Toc477721308)

[7.参考文献 25](#_Toc477721309)

Python开发环境的建立

# 1. Python的简介及开发环境建立

# 1.1 Python的简介

Python是一种少有的宣称既简单又强大的编程语言。它令人惊喜的地方在于它更注重问题的解决方法，而不是语言的语法和结构。Python的官方介绍如下：python是一种易学友好且强大的语言。它有高效的高级数据结构和有效且简单地面向对象的编程方法。Python优雅的语法和动态类型和它的语言解释性使得它成为能在很多平台完成编程和迅速的应用以应用于几乎所有领域的理想编程语言

**Python的特点如下：**

1. 简单。Python是一门简单精小的语言。阅览一个好的python程序就几乎像是在读非常严谨的英文。Python的伪代码的特性是一项强大的功能。你可以集中注意力解决问题而不是关注语言本身。
2. 易学。显而易见，Python的起步是极为简单地。Python有极为简单的语法，正如上文所提到的那样。
3. 免费而且开源。Python是FLOSS的一个典范。简单地说，你可以随意地散发这个软件的副本，阅读它的源代码，对它进行改变和在新的免费项目中使用它的片段。FLOSS基于一个共享知识的社区概念。Python虽已被创造但是仍然在持续不断地被一个力图使它变得更好的社区改进，就是python很棒的原因之一。
4. 高层次语言。当你用python写程序时，你觉不需要为比如为你的程序安排内存这样的低层次细节而烦恼。
5. 通用性。由于它的开源性，python可以在许多平台上使用。如果你足够小心地避免系统依赖性问题，所有的Python程序不需要任何更改就可以在任何平台上运行。
6. 已被编译。这一点需要一点解释。像用C或者C++那样的汇编语言写的程序需要用具有许多标志和选项的编译器转换成计算机的机器语言（二进制）。当我们运行程序的时候，连接加载软件从硬盘中复制程序送进内存然后开始运行程序。而对于Python而言，就不需要编译成二进制机器语言。你只要直接从源代码运行程序。本质上说，Python是把源代码转换成了一个叫bytecode的过渡类型，然后开始把这个过渡类型转换成机器语言然后运行程序（这样造成python在第一次运行时较慢但是后面就会很快）。实际上，这样就使得python的使用相较其他语言来说简单了去多，因为你不需要担心程序的编译，只要确保适当的库被连接和加载。
7. 面向对象。Python既支持面向过程的编程也支持面向对象的编程。在面向过程的语言中，程序围绕围绕作为程序中不重要但是可以重复使用的流程和功能建立的。在面向对象的语言中，程序是围绕着集聚数据和功能的对象建立的。Python能够强大而简单地进行面向对象编程，尤其与如C++或者Java这样的大型编程语言相比。
8. 可延展。如果你需要关键的程序运行得非常快或者不想要算法公开，你可以用C或者C++写那部分程序写好，然后在你的python程序中用。
9. 可嵌入。你可以吧python嵌入到你的C/C++程序中为你的程序用户提供脚本功能
10. 广泛的库**。**Python的标准库真的很大，能够帮助你实现很多事情比如正则表达式、文档生成、单元测试、线程、数据库、Web浏览器、CGI、FTP、电子邮件、XML、HTML、XML RPC，WAV文件，密码，GUI（图形用户界面），以及其他系统相关的东西。记住，这一切都是可以实现的只要安装了python。除了标准软件包，在 [Python Package Index](http://pypi.python.org/pypi)（我们在这个手册中使用<http://www.lfd.uci.edu/~gohlke/pythonlibs/>上面的软件包）你还能找到很多高质量的其他软件包。

Python确实是一个令人兴奋的和强大的语言。它具有合适的性能和特点，在Python中的乐趣和轻松的写作程序组合。

## 1.2 python基本开发环境

### 1.2.1 Python软件的安装

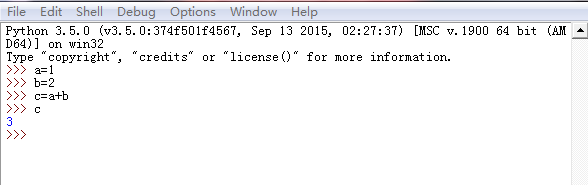
我们可以直接访问<https://www.python.org/downloads/> ，然后根据操作系统选择我们要下载的版本。可以看到有两个系列2.X系列和3.X系列，最新的是3.60版。2.X系列不会有新的版本发布了，而3.X系列还在不断发展和发布新版本，而且3.X系列并没有对2.X向下兼容（2.X有着许多C语言的遗传血统，存在许多比较奇葩的问题，Python的核心团队厌烦了解释各种限制，而是选择解决这些问题，3.X便应运而生，所以3.X的本意就是不继承2.X的一些特征。）就意味着最近发布的标准软件包只适用于3.X系列。两者的区别不在此多言，关键在于作为入门者的我们要怎么选择。虽然3.X系列的软件包也许比不上2.X的庞大而成熟，但是3.X系列也应该是够用的，而且2.X也不是如很多人说的那样完美；其次对于初学者而言，学习2.X是比较痛苦的，3.X相较而言更加python！最后，“活着”的语言才是有前景的。

所以我们选择3.X系列比较好，而最新出的3.60版，由于刚发布，软件包比较少，会影响我们的使用。所以选择3.5.X系列较好。

在安装过程中要注意：1.勾选Add Python 3.5 to PATH这一选项。2.更改安装地址，勾选Customize installation,然后点Next,设置安装地址为：C：\python35 。你可以选择下载Launcher，这无关紧要，Launcher是用于在下载的python个版本之间的转换。

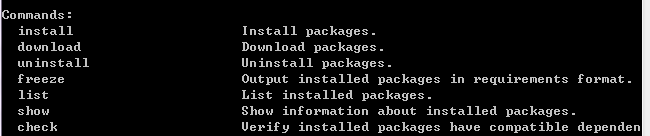
### 1.2.2 使用IDLE编程

IDLE是Python软件包自带的一个集成开发环境，初学者可以利用它方便地创建、运行、测试和调试Python程序。



# 2.python扩展软件包的安装

pip是用于下载安装python的软件包的工具。有如下几条命令以及其作用的解释：

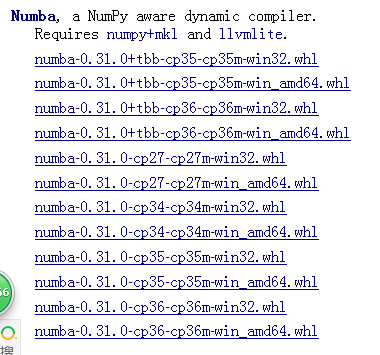


在使用pip安装软件包时要首先更新pip在联网的情况下，在DOS命令行下，输入>python -m pip install -U pip就可以实现pip的检查和更新。接下来就可以进行扩展软件包的安装了。

## 2.1科学计算扩展包安装

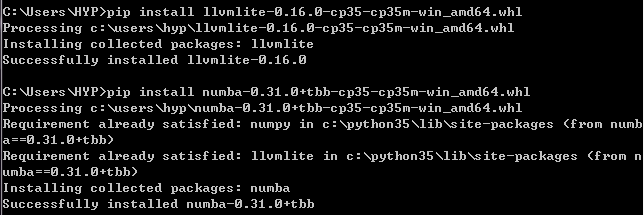
推荐从南加州大学欧文分校提供的Windows32/64位Python编译版扩展包。http://www.lfd.uci.edu/~gohlke/pythonlibs/ 里面有详实，丰富，及时的各种编译版扩展包。我们所需要的扩展包主要包括：1.**基本科学计算扩展包**：对于基本应用而言numpy、scipy、matplotlib就够了。2.**科学计算可选包：**Pandas: 基于 Numpy 构建，有更高级的数据结构和数据分析；Scikit-learn: 机器学习软件包；Numba: 计算加速包，安装Numba前，需要预先安装llvmlite。以上软件包都可以在以下流程下完成（其中先后顺序要注意）：

1.在<http://www.lfd.uci.edu/~gohlke/pythonlibs/> 中根据python版本和计算机操作系统版本下载相应的扩展包；



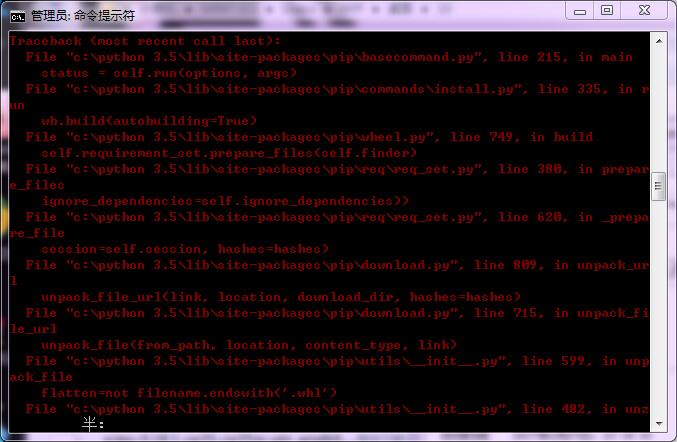
比如上图是Numba扩展包，图中cp3X是python版本名，32,64是操作系统的位数。

2.再在DOS命令行下，执行：pip install 扩展包文件名。



## 2.2过程中出现的问题：

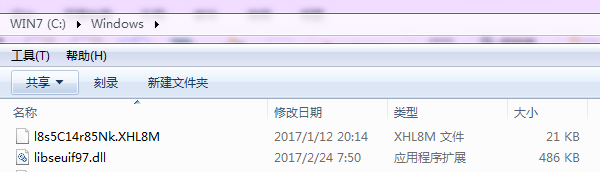
没有把下载的软件扩展包放在相应的文件夹，造成安装过程找不到文件。

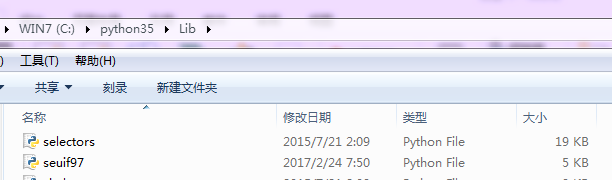


**2.3 SEUIF97水和水蒸汽物性计算**

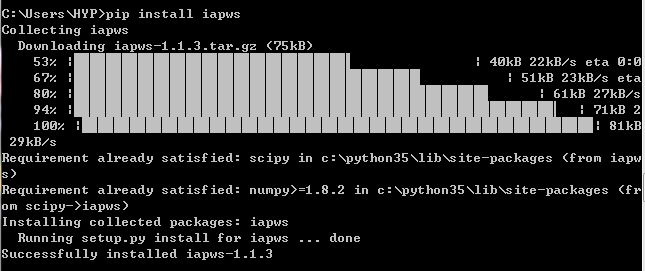
C语言IF97共享库及其Python封装包仓库:<https://github.com/PySEE/SEUIF97>

下载仓库的zip文件，解压，将和操作系统对应版本的：seuif97.dll和 seuif97.py： 1) seuif97.dll拷贝到 c:\windows\system 2) seuif97.py拷贝到 c:\python35\Lib





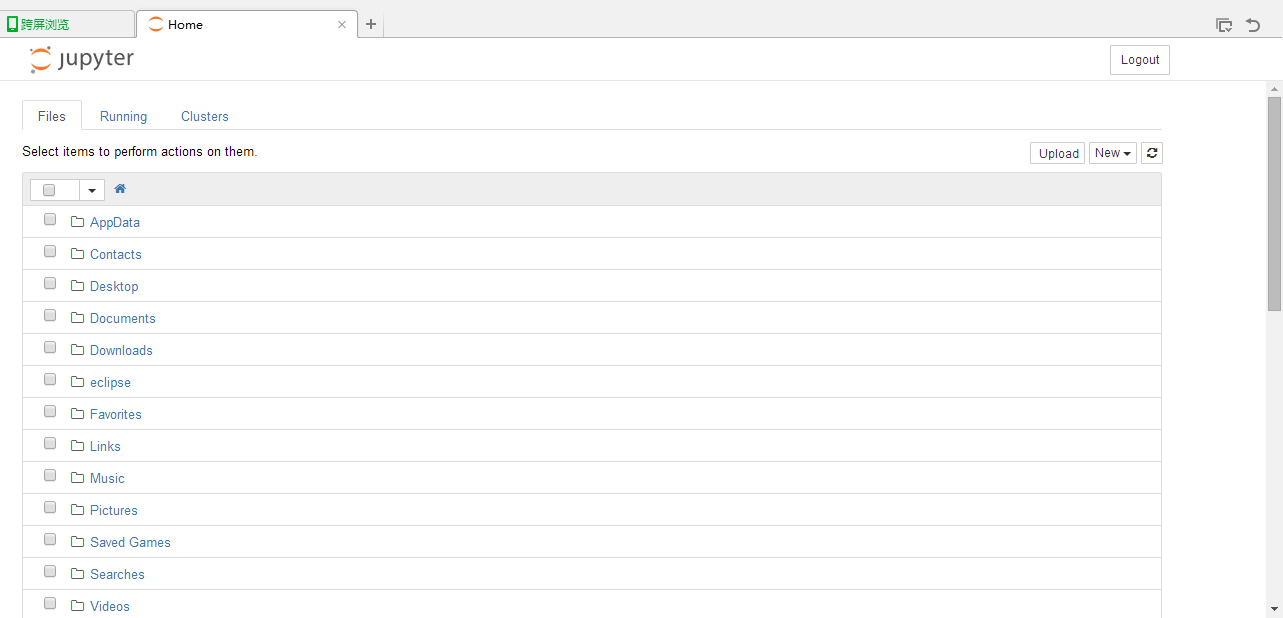
Python语言计算包，在线安装：>pip install iapws

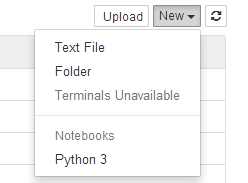
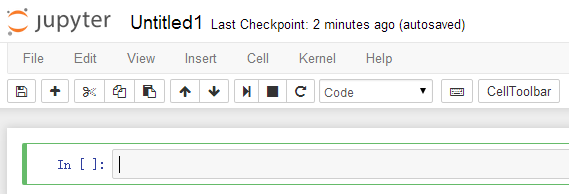


# 3.交互计算 Jupyter

首先安装jupyter的环境包，只要在联网情况下，在DOS命令行下执行：pip install jupyter如：C:\Users\HYP\AppData\Roaming\Tencent\Users\1047386577\QQ\WinTemp\RichOle\3)0ULEX63[6@[IXF1PYO`TQ.png，整个过程比较长，但是一般不会出现错误。以下为jupyter的使用指南：

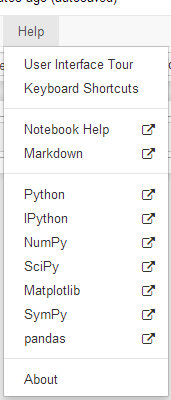
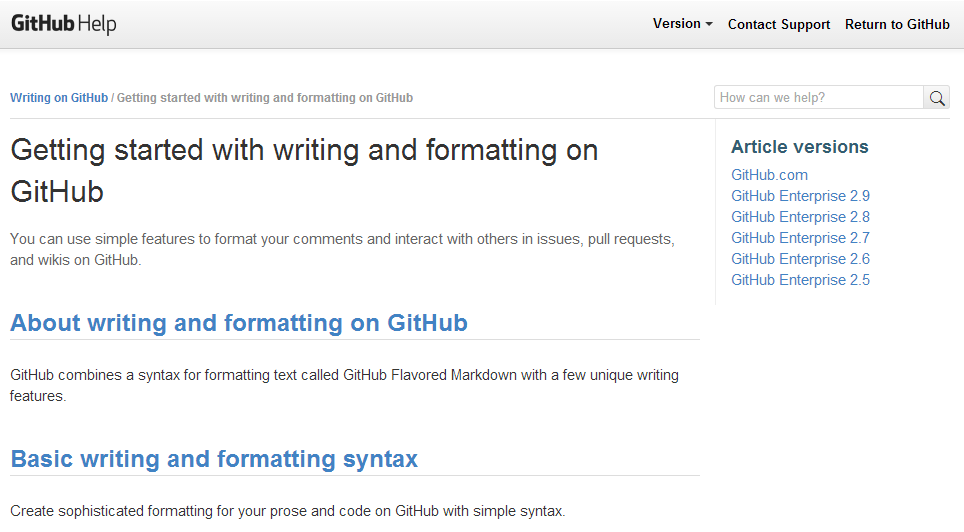
1.可以在Dos命令行中执行：jupyter notebook如：C:\Users\HYP\AppData\Roaming\Tencent\Users\1047386577\QQ\WinTemp\RichOle\9%{N0_XY7)63ZGAYS9XRRDN.png然后就会在网页中出现以下页面：



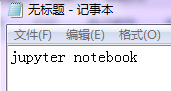
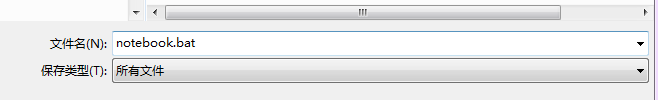
2.点NEW，新建一个文件，选择python类型。然后就会出现右侧界面：

基本的操作是一目了然的，程序的键入，运行等等

关于一些科学计算和Markown等问题可以在Help中寻找解决方法还有一些起步的教程与方法也可以在这找到答案~：

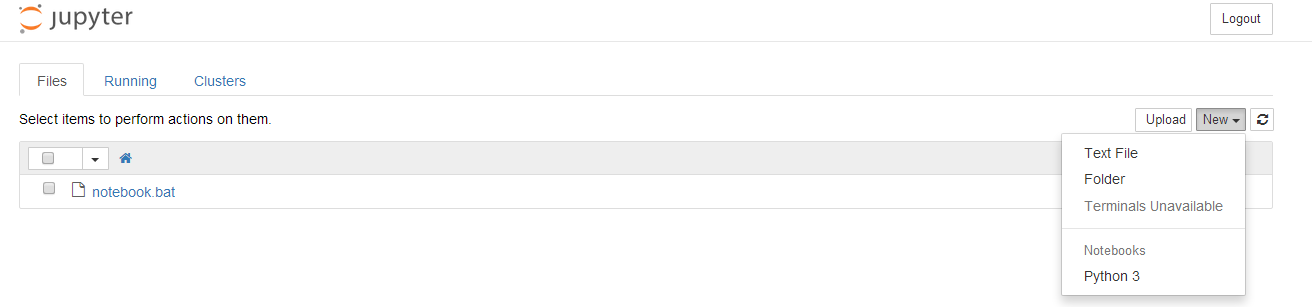
如：

## 3.1交互计算示例

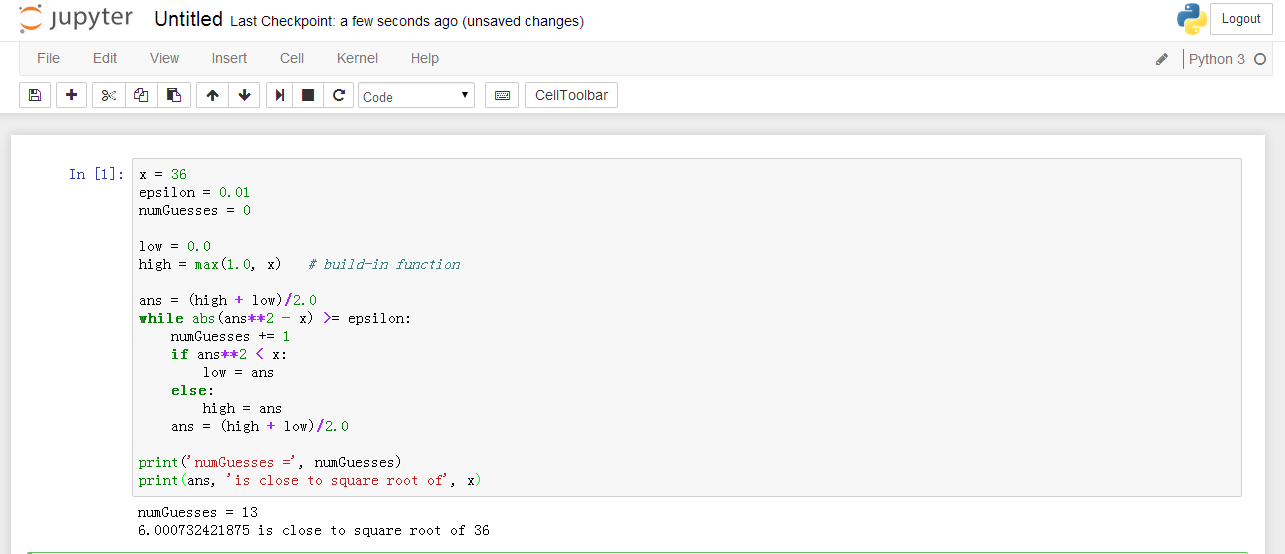
1. 在E盘中新建一个文件夹，取名为workplace for notebook（简写成workplace），然后打开写字板，在内打出 jupyter notebook，然后另存到workplace时将文件名的后缀改为.bat,就把文本文件转化成了批处理文件。

这样每次就不用在命令行输入jupyter notebook了。

1. 打开notebook.bat,就到了jupyter notebook的网页，点New，选择python3,就新建了一个python文件。



1. 开始编程



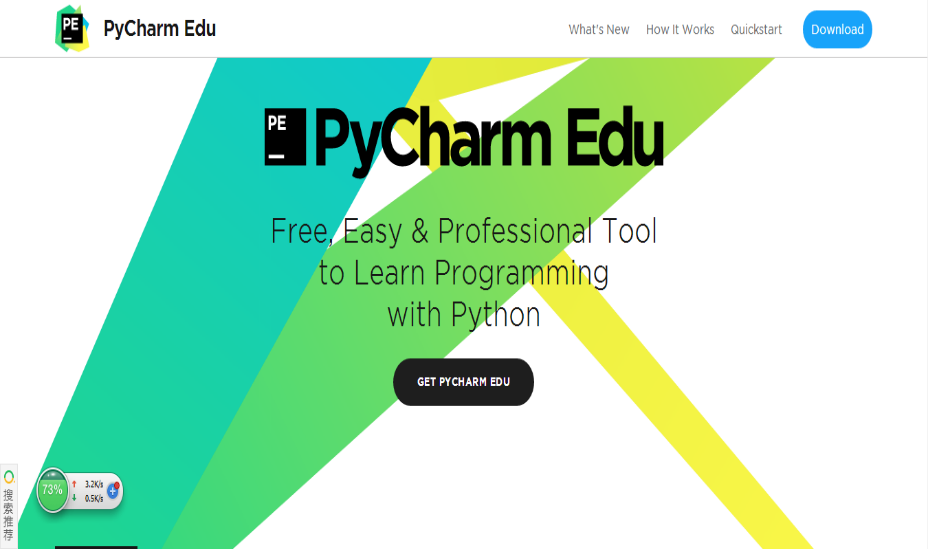
# 4.选择安装编程软件

我们不能每次都在IDLE中打出程序，再来运行。所以，我们必须要把程序存成文件而且可以任意次地运行。为了建立python源代码文件，我们需要一个你可以写，存程序的编程软件。一个好的编程软件可以使得你的编程生涯更为轻松。因此，编程软件的选择实际上是很重要而关键的。选择一个编程软件就和你选择买什么车一样。一个好的编程软件会帮助你很轻松地写程序，使你的旅途更加舒服而且能让你更快而安全地到达终点。

我们常用的python编辑器包括：Vim（for linux）, Visual Studio Code , Emacs ， Pycharm，Eclipse+Pydev 等等。这些软件几乎都有Highlight（语法高亮）功能，这是一个友好的编程软件的最基础的要求。下面介绍Visual Studio Code ， Pycharm ，Eclipse+Pydev三种软件的安装和基本功能。

## 4.1 Pycharm

Pycharm 是一个集成编程环境软件，提供智能编程，编程监视，传输过程错误高亮，快速修复，自动代码重构和丰富的导航功能。Pycharm可以在JetBrain的官网就可以获取<http://www.jetbrains.com/pycharm/> ，我们主要进行科学计算，因此下载科学计算版。 值得一提的是PyCharm有专门的Edu版本，可以很好地帮助教学者教学和初学者来学习python编程，可以在<http://www.jetbrains.com/pycharm-edu/> 中免费下载



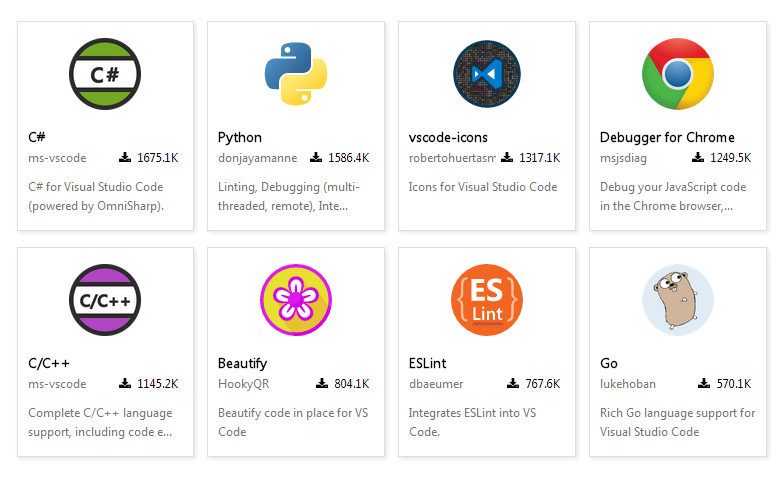
## 4.2 Visual Studio Code

VS Code 很小，只包含几乎所有开发工作流所共有的最基础的组件。只有最基本的功能：编辑器，文件管理，窗口管理和偏好设置。（VSCode可以在<https://code.visualstudio.com/>

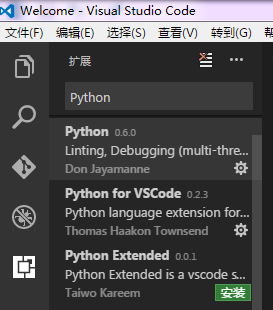
中下载，安装。）

### 4.2.1 基本环境的搭建

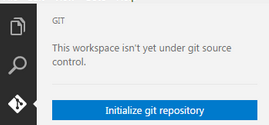
因此大部分的用户需要根据特定的需求来安装其他的组件。在VS Code官网中有相应的组件，如下图：



比如我们要用VS Code做python编程，也要下载安装相应的组件（插件）来搭建一个Python的编程环境。

如左图：我们可以点开边栏最后一个图标（扩展应用商店），然后输入所需要的插件类型，然后再选择插件进行安装。这样就基本完成了环境的搭建。

### 4.2.2 功能特点

Vc Studio Code 有语法高亮，智能代码补全、自定义热键、括号匹配、代码片段、代码对比 Diff、GIT 等功能，VC Studio Code还有一个功能就是 Zen Mode，也就是讲编辑框全屏，让程序员专注于程序本身，而暂时不考虑其他问题，有如禅宗里的打坐禅定。（Control+K+Z）

### 4.2.3Git

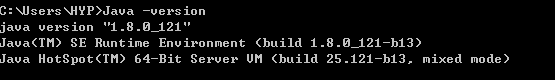
下面就Git 版本控制的方法介绍一下：1.先建立一个工作空间，用以存放程序2.然后在边框中点第二个，很显然我们要初始化Git库，选择新建一个文件夹，就是我们刚才新建的工作空间，然后再在文件夹下新建文件，这样就同时有了版本控制。

## 4.3 Eclipse + Pydev

### 4.3.1安装Java SDK

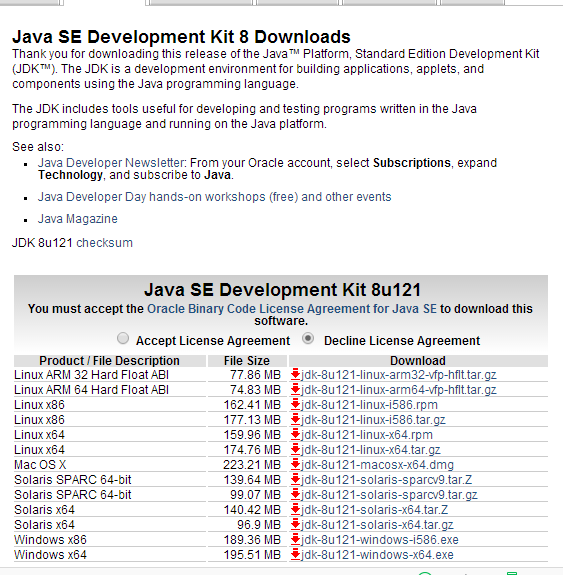
Eclipse IDE（集成编程环境）是用Java写的，所以Eclipse要运行的话需要有Java运行环境，那么至少要预装有JRE（能保证Eclipse至少能运行起来），但更推荐安装JDK。

首先，我们先检查是否已经预装有 Java JDK/JRE，在命令行下输入 Java-version。



然后，如果没有安装JDK或者JDK版本太低不能满足Eclipse的使用的话要到Oracle的官网下载：[**http://www.oracle.com/technetwork/java/javase/downloads/index.html**](http://www.oracle.com/technetwork/java/javase/downloads/index.html)



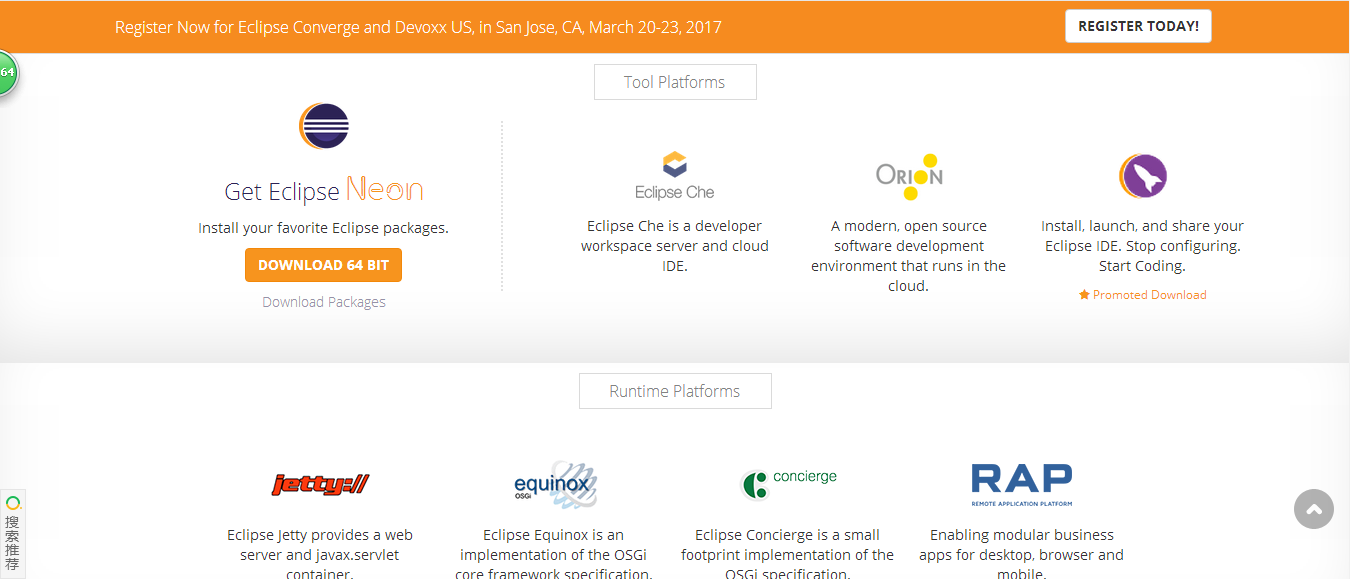


根据系统版本选择要下载的JDK。Win32选择i586,Win64选择X64。在Java安装完成后，注意对环境变量的配置。安装完成后，需要配置java环境变量，在我的电脑上右击-属性-高级系统设置，在弹出的窗口中点击右下角的“环境变量”按钮，接下来可以在用户环境变量或者系统环境变量中配置具体的环境变量，新建名为“classpath”的变量名，值为“.;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\tools.jar 。同样，新建名为”JAVA\_HOME“的变量名，变量值为之前安装jdk的目录。在已有的系统变量“path“的变量值加上 ”;%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin“，自此配置完成。下面进行Eclipse的直接安装。

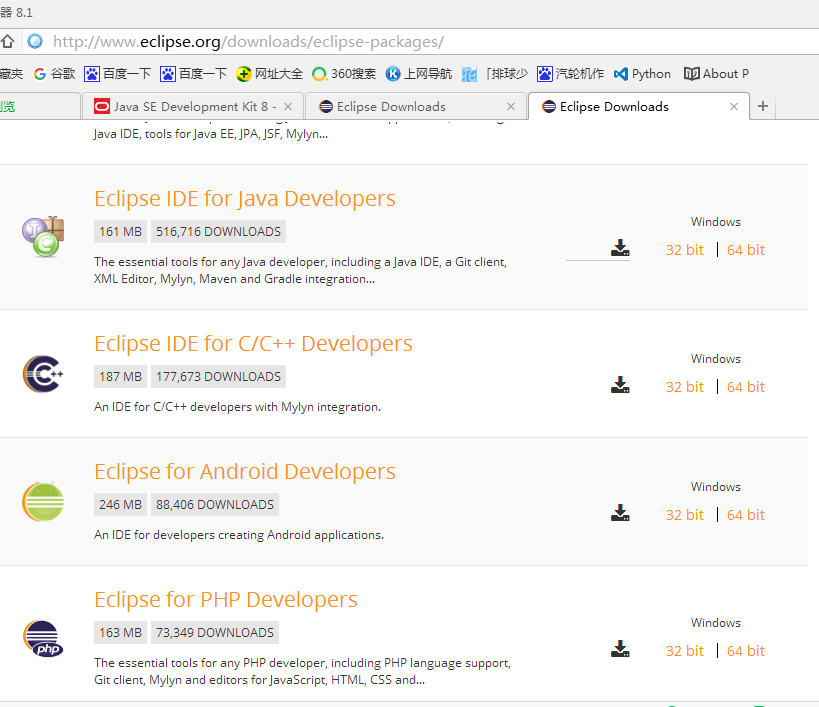
### 4.3.2 安装Eclipse

Eclipse IDE是一种插件型集成开发环境，用户需要根据自己的需求来安装插件和设置环境变量。Eclipse基金会在<http://www.eclipse.org/users/> ，提供了大量资源。安装和使用中的问题，都可以在其中找到相应的帮助文档。

在以下网址： http://www.eclipse.org/downloads/下载好Eclipse Neon



然后在安装过程中选择forC/C++ Developer,因为Python可能会要用到C/C++写部分程序，只不过这里只有64位的；其实可以直接在<http://www.eclipse.org/downloads/eclipse-packages/> 根据系统位数下载Eclipse IDE for C/C++ Developer



在安装过程中勾选将Eclipse 锁定在任务栏，方便使用。

### 4.3.3 配置Eclipse工作空间

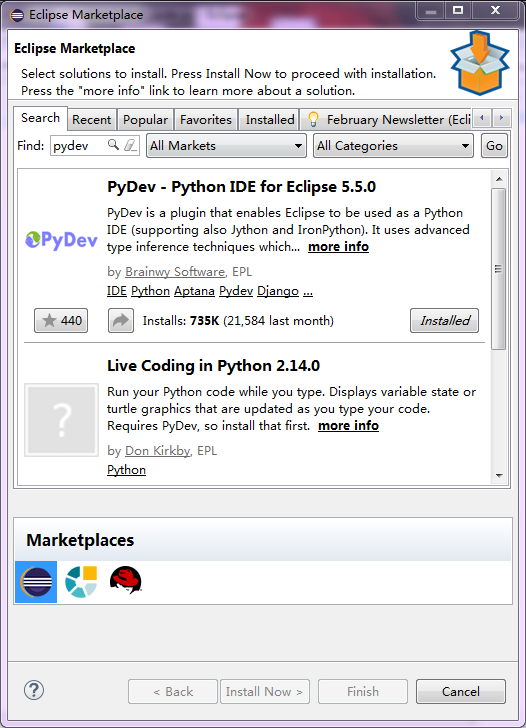
在首次打开Eclipse的时候，会提示Eclipse默认的工作空间，不建议直接使用这个工作空间（在C盘，占用系统内存），而是在D盘中新建一个文件夹（取名pyworkspace），然后再browse，选择这个新文件夹作为工作空间。第二次打开时，仍选择这个文件夹作为工作空间，并勾选Use this as the default and do not ask again这样之后的工作空间就默认为pyworkspace 这个文件夹。如果要建立多个工作空间目录，可以在file下switch workspace ->other新建目前工作空间。

### 4.3.4Python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。

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。

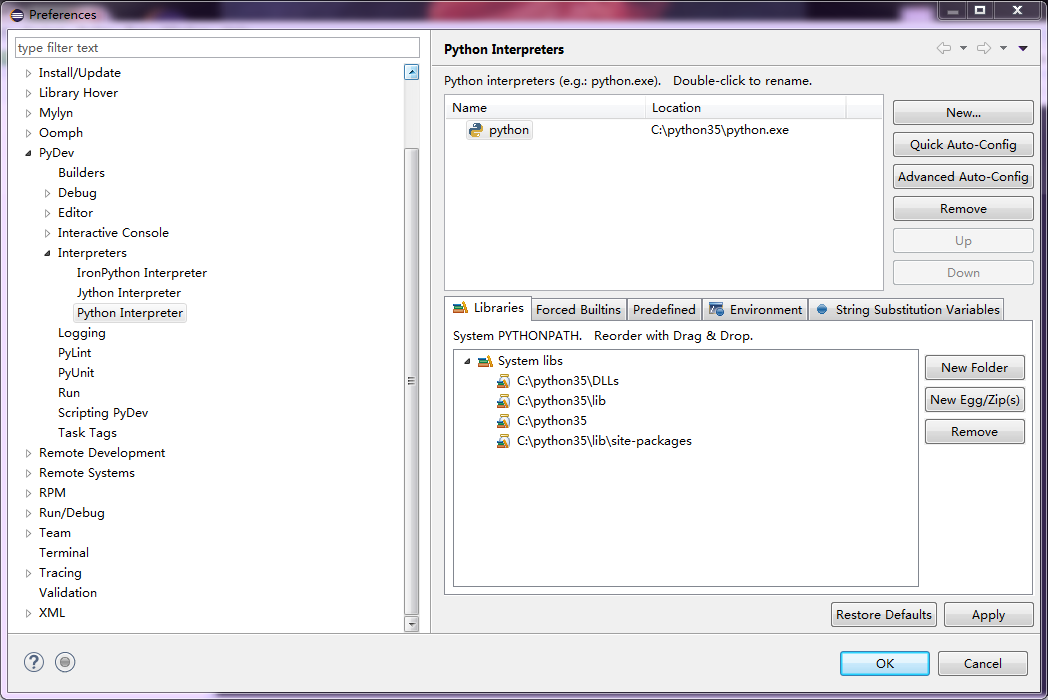


一般情况下可以很好地在线下载安装，如有问题则本地安装。

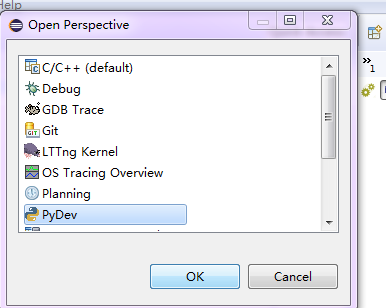
**4.3.5 配置 Python解释器**

安装好后，重新打开Eclipse在：

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：

<-open perspective

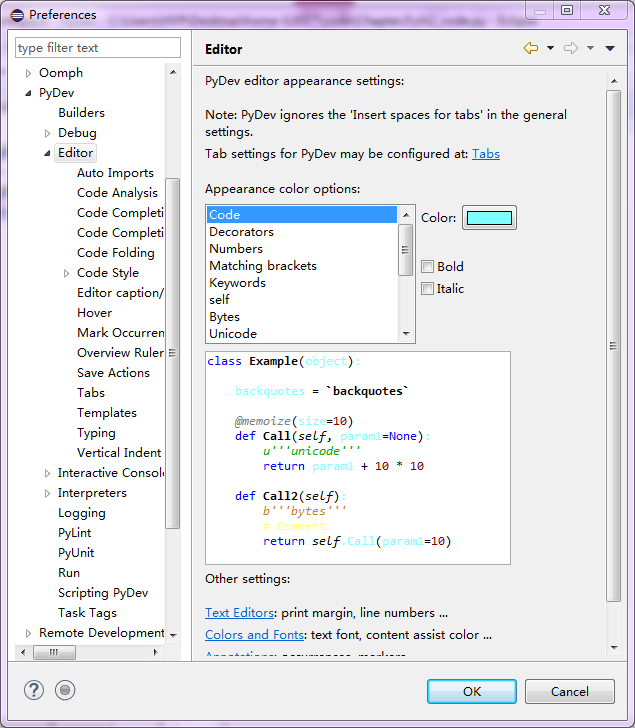
点open perspective后再 选择Pydev

就可以进行Python开发了。

## 4.4配置PyDev以提高效率

### 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：

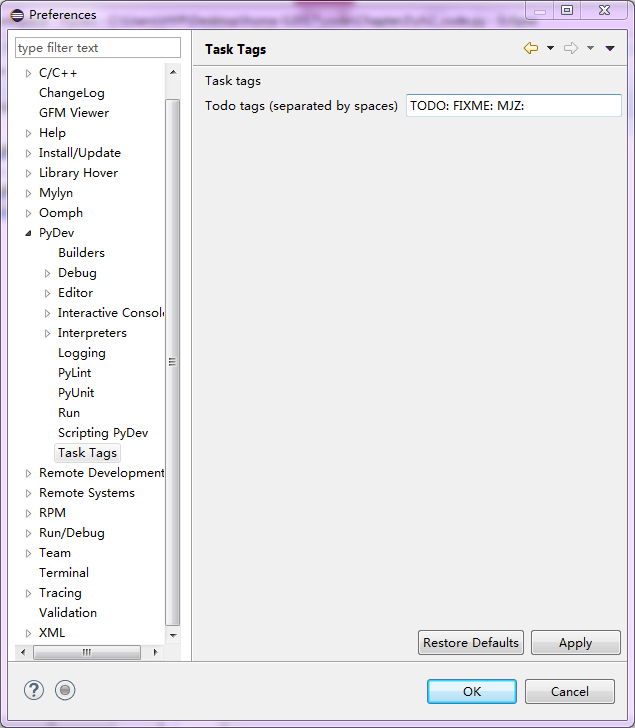
****

### 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。

1. **配置任务标签：**

通过PyDev->Task Tags中配置任务标签:



可以根据不同的任务类型，自己增加自定义标签

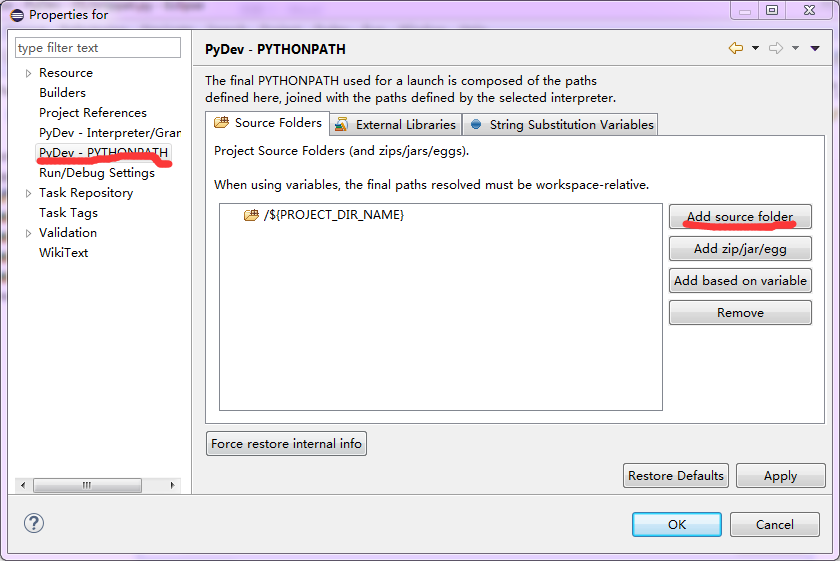
1. **添加任务标签注释**

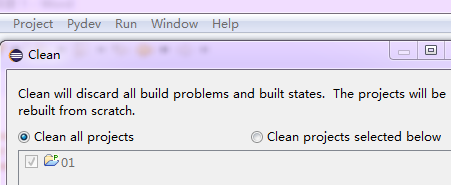
在程序源码中加入使用任务标签的注释（是注释，但是使用了“任务标签”关键词开头，PyDev用任务标签作为关键字，检索生成任务列表的注释）：

C:\Users\HYP\AppData\Roaming\Tencent\Users\1047386577\QQ\WinTemp\RichOle\WEHXFKE%QMRLD[X93J1ZIIB.png

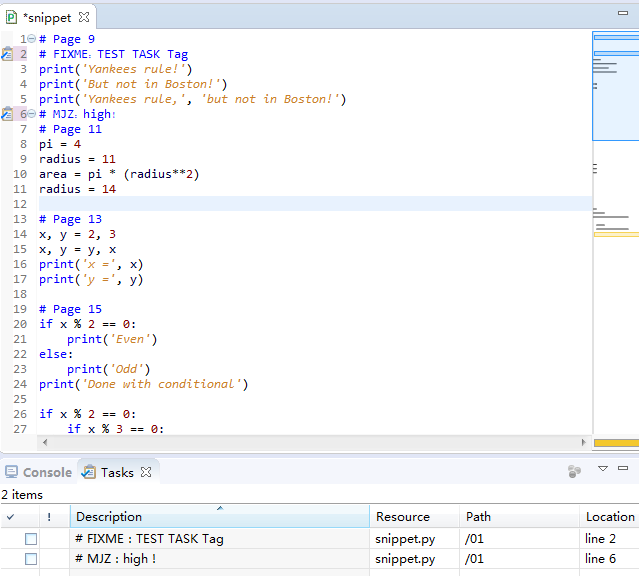
1. **自动识别任务**

需要配置当前工程的PyDev-PYTHONPATH的source folder。在源码工程的Properties->PyDev-PYTHONPATH配置项中，点“Add source folder”将源码目录加入。

****

保存新修改或运行程序一次（Ctrl+b）或选择 Project → Clean： 

就可将使用任务标签注释加入任务列表,相应的任务注释行左边会有标识#加任务标签关键字所在行为任务标识行。如果当前任务窗口包含“Tasks”，刚加的任务，就会立刻显示在“Tasks”窗口中。如“Tasks”窗口没有打开，可Window->Show View->Tasks显示任务窗口。



1. **手动添加任务**

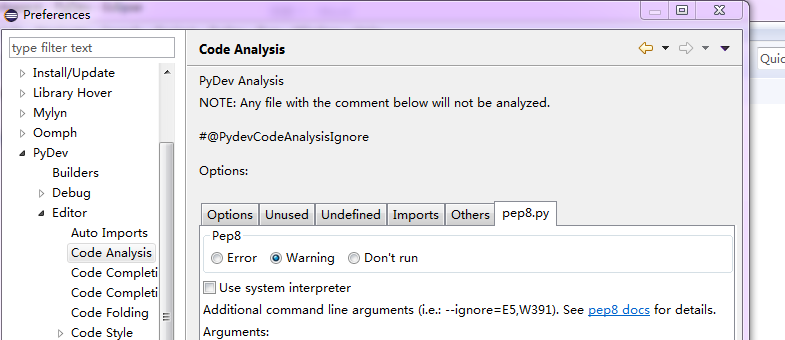
在PyDev 环境下，也可手动添加任务。将鼠标移动到需要添加为任务行的最左侧，点右键，选择“Add Task”,即将该行加入任务列表。

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。

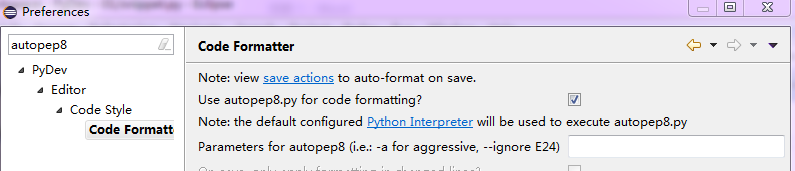
## 4.5 PEP8检查和修改

1.启动 pep8 检查：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. .

2.启动autopep8自动修改：点Windows -> Preferences ->输入 'autopep8' 作为搜索串.选择（Check）: Use autopep8.py for code formatting?

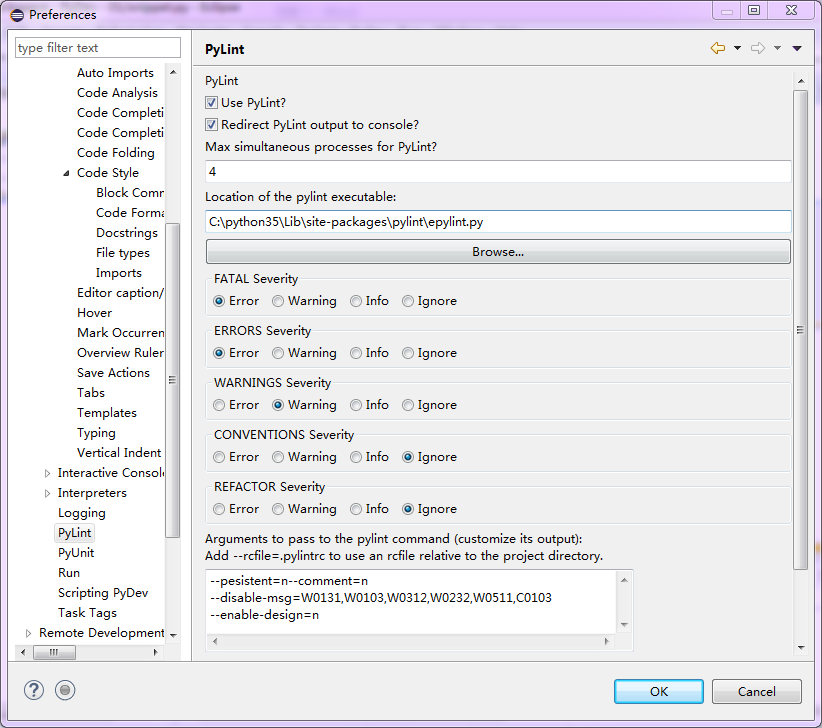


在Python源码窗口，按 CTRL-SHIFT-F 就可以自动修改代码

## 4.6 使用Pylint

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,例如"C:\Python36\Lib\site-packages\pylint\lint.py" 

配置参数，限制Pylint的输出

![C:\Users\HYP\AppData\Roaming\Tencent\Users\1047386577\QQ\WinTemp\RichOle\SGU97@T~G3G8%AZO3FG](SX.png](data:image/png;base64,)

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build触发pylint。

C:\Users\HYP\AppData\Roaming\Tencent\Users\1047386577\QQ\WinTemp\RichOle\3]Y01)_B$$Z_OCI9I`UEP26.png

## 4.7 Eclipse配置和使用

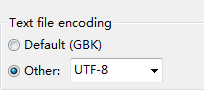
### 4.7.1 文件字符编码方式

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。

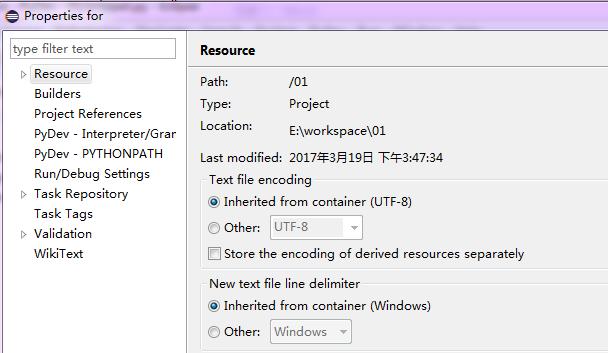
1.**工作空间编码方式**

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

****

**2.工程编码方式**

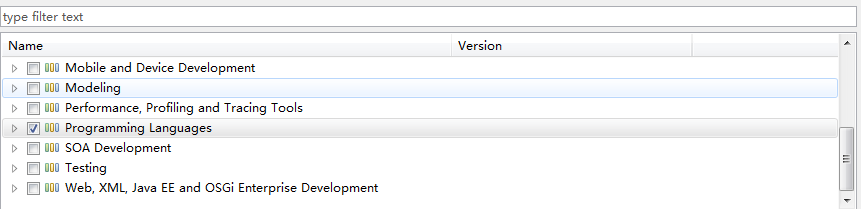
将鼠标移动到项目名上，点击右键，选择“properties”， 在弹出的对话框中 ，选中“resources”， 修改“Text file encoding”为UTF-8(如果工作空间配置为UTF-8会继承过来)：

****

### 4.7.2 插件补充和更新

以Eclipse当前版本插件为例说明插件的补充和更新：电脑联网，然后**，Help->Install New Software，**进入安装软件页面，选择插件的更新网址

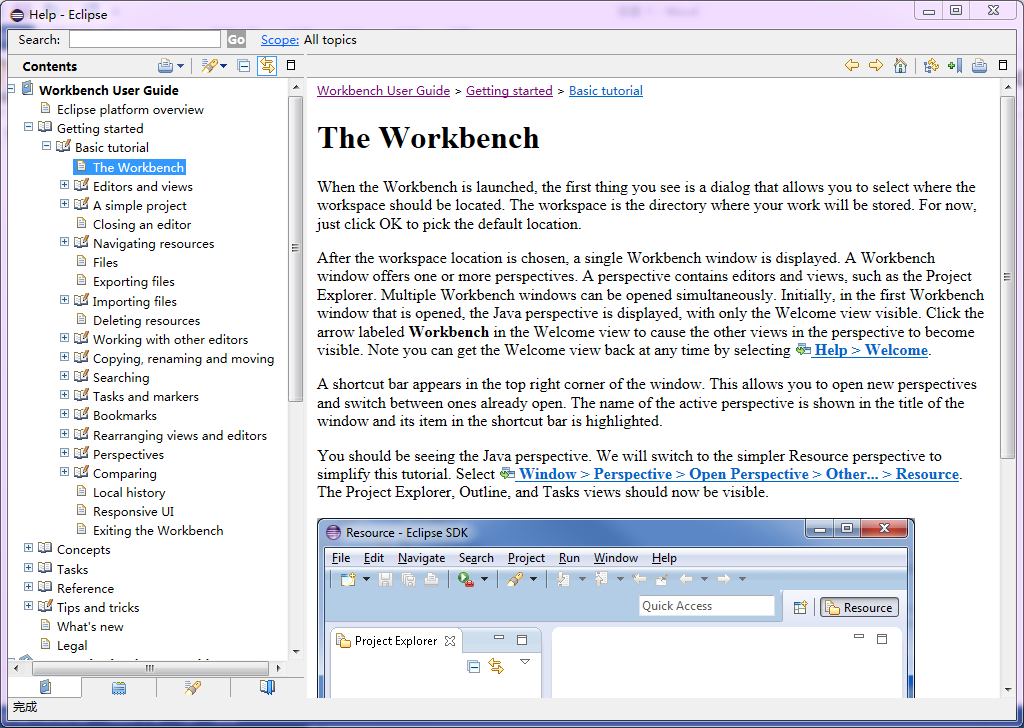
**![C:\Users\HYP\AppData\Roaming\Tencent\Users\1047386577\QQ\WinTemp\RichOle\%X{[)RUORJAIPHY]([O$JN3.png](data:image/png;base64,)**

稍后可看到插件列表，勾选需要安装的插件，按照要求继续下面步骤即可安装好指定的插件：

### 4.7.3 联机指南

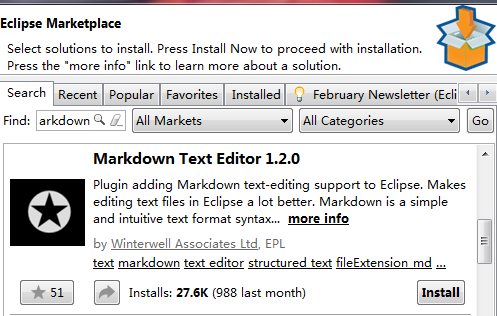
Eclipse的功能很多，软件使用中有疑问可以查询Eclipse内置的help：

**Help** -> **Help Contents**



### 4.7.4 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：

****

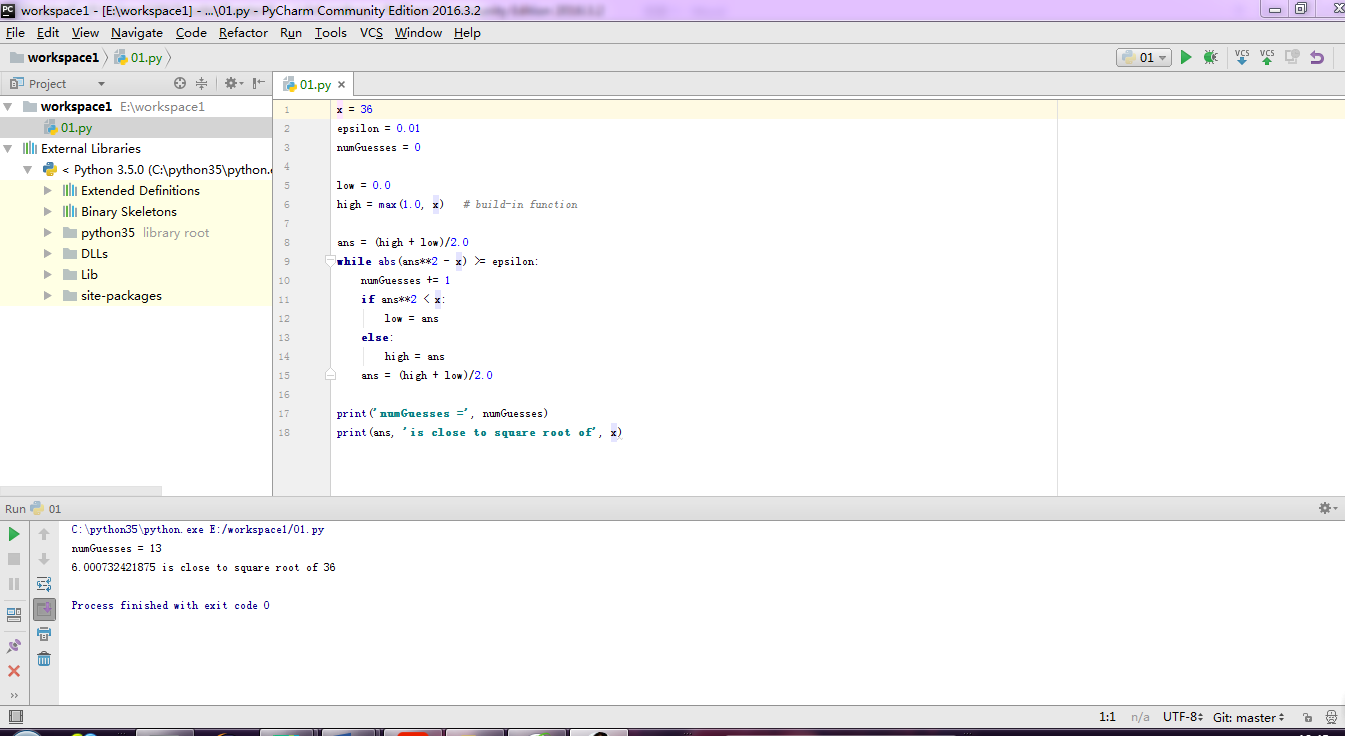
Eclipse Marketplace 市场安装，安装后选择其作为md文件的编辑器，即可.Markdown Editor插件Windows->Preferences->General->Editors->Text Editors->Markdown配置有关参数。

## 4.8 出现的错误



指令打错了，一度很无措，请教同学后方改正。

# 5.Python 编程示例



这是用Pycharm编的一段程序，用While循环求36的平方根。

在我目前使用的感觉觉得Pycharm还是相当不错的，比较简单（换言之就是比较傻瓜），可能对于入门者而言很好入门，但是学不到更多的有关软件的知识，实际上是不利于初学者学习掌握知识的，是不能带来裨益的。

# 6.个人小结

首先，这个过程让我感觉到了自己的不足，刚开始做的时候，我很茫然，我只会按照老师给的步骤做，但是却不是很了解为什么这么做，知其然而不知其所以然也。这种极为不适的危机感把我从舒适区里拉了出来，使我好好地审视了我的学习。

其次，我觉得我学到了很多知识。大一到大三，我对编程是一窍不通的，说句实话，对于一门编程语言的philosophy or architecture 是不大理解的，对于一些软件的构造也是所知甚少，只会在自己会的一亩三分地内编编“应试”的程，从来都没有摸清楚过编程这头大象的轮廓，由于所知甚少，所以还有着对于编程的恐惧。在这个过程中我一直是充满疑惑的，于是，我查看了许多的文献，对于一些语言创造者的想法，idea有了一定的了解，以及了解一些很基础的计算机知识，解答了我许多疑惑。

最后，这种体验改变了我对于编程的态度。我以前很恐惧编程，因为恐惧和惰性，我并没有笨鸟先飞，而是成了井底之蛙。老师把Github介绍给我们，让我们打开了视野，然后告诉我们官网的资料有着最好的学习资源，而且灌输给我们简单即美的观念，使我大有醍醐灌顶之感，顿时又燃起了学习编程的热情与兴趣。

# 7.参考文献

1. Python学习和开发环境的建立(教学版).docx

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

1. RecommendedPractices

<https://github.com/PySEE/RecommendedPractices/tree/S2016/P1>

1. Python With Visual Studio Code

<https://code.visualstudio.com/docs/languages/python>

1. A Byte of Python

<https://python.swaroopch.com/about_python.html>

1. Eclipse User Guide

<http://help.eclipse.org/neon/index.jsp?nav=%2F0>

1. <https://code.visualstudio.com/docs/>
2. 廖雪峰的官方网站-Python教程

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