**Python学习和开发环境的建立**

Python解释器、标准库及扩展包构成Python的基本开发环境，满足学习和软件开发的基本要求。Python解释器及其标准库可从Python官网下载；其他扩展包可用pip命令等方法安装。

大型软件开发宜采用集成开发工具，常用的有: Eclipse，VS2015等，其中，Eclipse加PyDev插件的方式， 为较多Python程序开发人员使用。此外，还有使用编辑器(如Atom)构建开发环境的方法。

本文档给出的是Windows操作系统下开发环境的建立过程。

**注意：务必从官网下载软件，以免下载了被“污染”的软件**。

1. **Python基本开发环境建立**

从官网：

<http://www.python.org>

下载和操作系统（32/64位）对应版本的Python3软件。下面以Python3.5版本安装为例做简单说明

**Windows：** 建议使用定制方式安装:

1. 加Python安装目录到Windows系统的环境变量PATH中;
2. 自定义安装目录，目录名带版本号。现在安装Python3.5版，目录为：C:\Python35（注意：Python的安装目录可以自定义名称，但是，因为Python可以在一台机器上安装多个版本，然后，选择使用某个版本，使用目录名称中带版本号为好）

然后，命令行下更新pip到新版本：

>python -m pip install -U pip

1. **Python扩展包安装**

pip命令是软件包的基本安装方式，不过，有些软件包使用pip安装时，需要本地编译，编译条件如果不满足安装就会失败。

这时可使用： 加州大学欧文分校提供的Windows32/64位Python扩展包编译版，

<http://www.lfd.uci.edu/~gohlke/pythonlibs/>

加州大学欧文分校的编译版包，内容丰富、更新及时、安装方便。

**2.1 编码规范**

**>**pip install autopep8

>pip install pylint

**2.2 科学计算软件包**

**2.2.1 计算软件包SciPy**

Scipy官网：<http://www.scipy.org/>,提供有全面的科学计算软件包信息。基本应用安装以下3个即可：

numpy

scipy

matplotlib

**Window**环境下Scipy 软件包的安装，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：

>pip install \*.whl

**2.2.2 IF97 for Python**

Windows 32/64位版: 从

<https://github.com/Py03013052/SEUIF97>

下载：SEUIF97.dll和 seuif97.py，然后：

1) SEUIF97.dll拷贝到 c:\windows\system

2) seuif97.py拷贝到 c:\python35\Lib

1. **交互计算 Jupyter**

**3.1 建立Jupyter环境**

在线安装Jupyter ：

>pip install jupyter

启动jupyter notebook服务:

命令行执行：

>jupyter notebook

**3.2 创建jupyter notebook文件**

1. **基于Eclipse的Python开发环境**

**4.1 安装Eclipse IDE**

**4.1.1安装JavaSDK**

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**。**



如果没有安装java 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**](http://www.oracle.com/technetwork/java/javase/downloads/index.html)



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.

运行下载Oracle的Java安装包（安装时注意**配置环境变量**），即可安装好Java环境。

**4.1.2 Eclipse IDE**

　　Eclipse IDE是插件型开发环境，有很多版本可以下载。本课程主要使用Python语言，会涉及C/C++开发，所以，建议下载Eclipse CDT(Eclipse IDE　for　C/C++)版.

Eclipse CDT官方下载地址：

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



根据操作系统32/64位，下载相应的版本，然后将下载的Eclipse CDT解压到指定目录下，运行解压目录下的：eclipse.exe即可。

如果使用Windows7 以上版本操作系统，建议将运行eclipse.exe，固定到任务栏。（在eclipse.exe文件名上，点鼠标右键即可）





以后，从任务栏启动Eclipse方便很多（同样也可以将IDLE固定到任务栏）。

**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）配置使用那个版本的Python解释器。

**4.2.1 安装PyDev插件**

通过Help->Eclipse Marketplaces进入市场，输入Pydev，找到Pydev安装/更新项目，在线安装即可：



在线安装过程简单，但是，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.

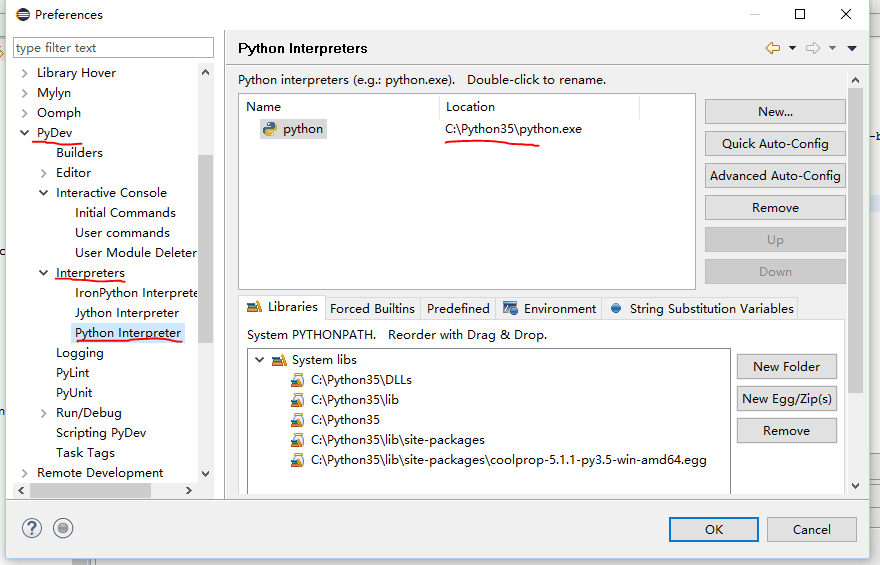


正常情况，Eclipse会自动解压，安装好插件。如果，解压中异常，可以，手工解压插件包，将解压后的所有文件，拷贝到Eclipse目录下。

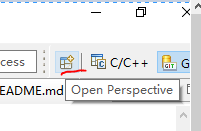
**4.2.2 配置 Python解释器**

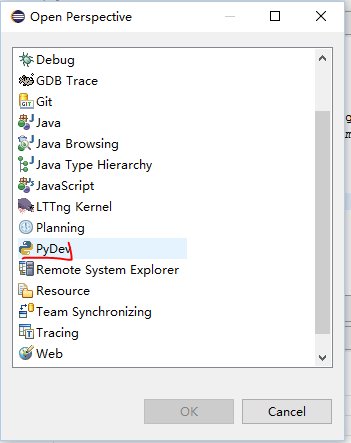
安装好后，重新启动。通过：

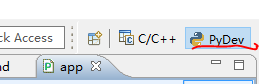
Windows->Preference->Pydev->Interperters->Python Interperter点其中的：Advanced Auto-config配置开发使用的 Python解释器版本:



配置好后切换到Python场景，







就可以进行Python开发了。

1. **配置 PyDev提高效率**

**5.1 修改编辑器配色提高可读行**

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





**5.2 配置任务标签**

在程序开发过程中可以在代码中标识，当前任务状态，计划开发工作。在代码中标识任务，可以使用任务标签，然后，让开发环境将其识别。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:



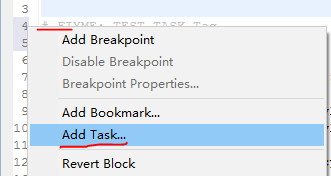
可以根据不同的任务类型，自己增加自定义标签

**2）添加任务标签注释**

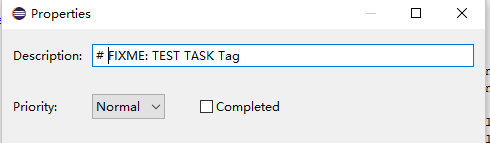
在程序源码中加入使用任务标签的注释（**是注释，但是使用了任务标签开头，是可以被开发环境用任务标签作为关键字，检索生成任务列表的注释**）：



并在该行左侧右键，选择Add Task



软件将自动，使用注释内容作为标签描述（可以修改，如删除# ）



确认后，一个任务就加好了：



**3）识别任务**

保存新修改、运行程序一次或者选择 Project → Clean，

就可将使用任务标签注释加入任务列表。如果当前任务窗口可视，刚加的任务，就会立刻显示在任务窗口中。



如任务窗口没有打开，，可以Window->Show View->Tasks开启任务窗口。

1. **基于PyDev的语言规范静态检查**

PyDev中集成了PEP8,AutoPEP8 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，要有规范意识，但不可能有很高的规范性，过分注意规范会影响开发进程。这时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，默认关闭是合适的。在程序开发一个阶段结果出来时，进行规范性检查更好。

**6.1 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： .



1）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

**6.2 使用Pylint**

PyDev 默认不开启Pylint。通过

     Window -> preferences -> Pydev -> Pylint,选中"Use pylint?",

找到安装好的lint.py的地址,例如"C:\Python34\Lib\site-packages\pylint\lint.py"



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build触发pylint。



1. **Eclipse配置和使用**

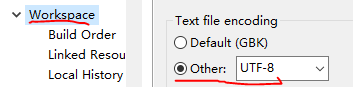
**7.1 设置Eclipse 中文件编码方式**

中文操作系统下，Eclipse默认工作空间编码方式为GBK, 这样的编码方式产生的含有中文的文件，在其他文本编辑器打开会乱码，因此需要修改为UTF-8。编码方式可以设定到：工作空间、工程和文件类型上。

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编码方式。

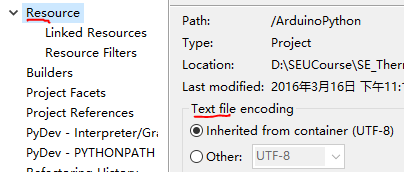
**7.1.1 工作空间编码方式**

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



**6.1.2 工程编码方式**

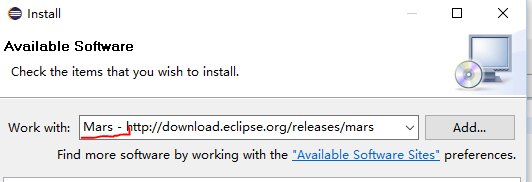
将鼠标移动到项目名上，点击右键，选择“properties”， 在弹出的对话框中 ，选中“resources”， 修改“Text file encoding”为UTF-8(如果工作空间配置为UTF-8会继承过来)：



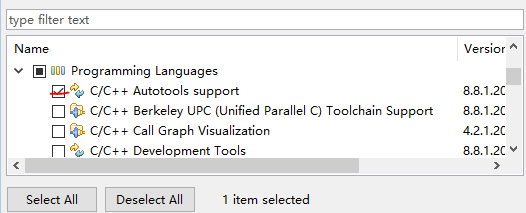
**7.2 插件补充和更新**

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：



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：



**7.3 联机指南**

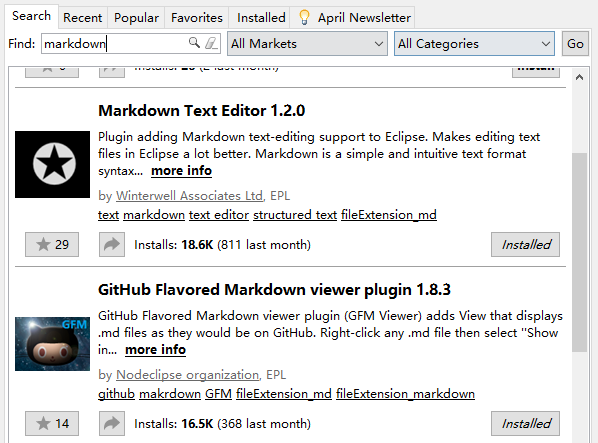
Eclipse的功能很多，软件使用中有疑问可以查询Eclipse内置的help：

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



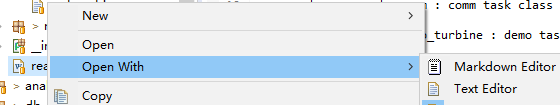
1. **Markdown插件**

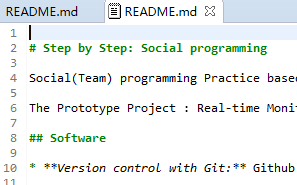
Eclipse默认安装对Markdown支持较弱，需要安装Markdown插件。Help->Eclipse Marketplace使用Markdown关键字可以找到2个插件：编辑和显示：



**8.1 Markdown Editor插件**

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插件可Windows->Preferences->General->Editors->Text Editors->Markdown配置有关参数。

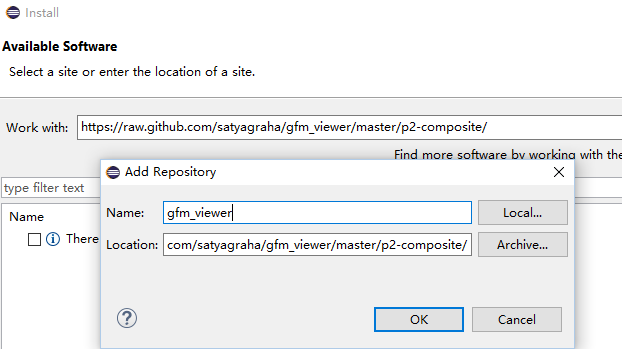
**8.2 GitHub Flavored Markdown Viewer插件**

GitHub Flavored Markdown Viewer插件源码仓库：

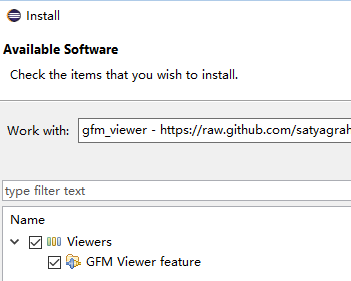
<https://github.com/satyagraha/gfm_viewer>。

其中,有插件安装地址，也可：Help -> Install New Software，拷贝插件地址到Work with:

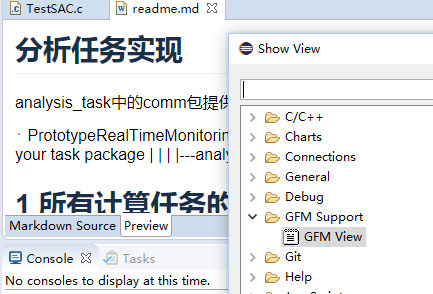
https://raw.github.com/satyagraha/gfm\_viewer/master/p2-composite/



中选择，安装Viewers插件：



安装后，Window -> Show View -> Other... and select entry GFM Support -> GFM Viewer



开启GFM Viewer窗口，点击Markdown文件就可以看到和GitHub上一样的显示效果了



**小结**

Python 学习和开发环境多种多样，建议同学们首先掌握好IDLE。通常，进一步用好Eclipse加PyDev就能满足开发需要了。其他，以后逐步学习掌握。

**参考文献**

1. Brainwy Software Ltd. PyDev Manual[EB/OL]. <http://www.pydev.org/manual_101_root.html>
2. 郑伟芳. PyDev for Eclipse简介[EB/OL]. <http://www.ibm.com/developerworks/cn/opensource/os-cn-ecl-pydev/> 2008.11．