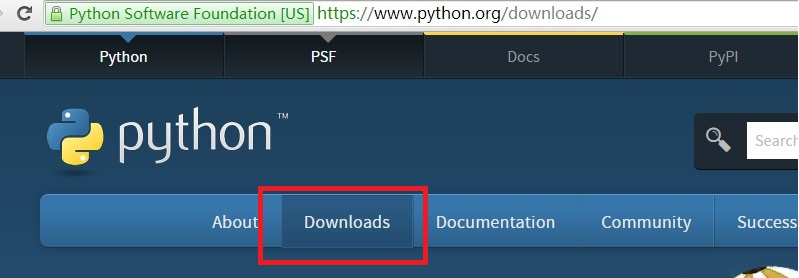
**Python开发环境建立指导手册**

1. **Python基本开发环境建立及适用领域**

Python是一种面向对象、解释型计算机程序设计语言，语法简洁清晰，特色之一是强制用空白符作为语句缩进。Python具有丰富和强大的库，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。

对于开发环境的基本建立，可以从官网：<http://www.python.org>



下载和操作系统（32/64位）对应版本的Python3软件。由于个人电脑配置的原因，所以选择64位操作系统对应的Python3。不选择Python2主要是因为主流Linux操作系统Ubuntu16.04(4月正式版），自身将不默认安装Python2

所以，现在必须放弃Python2因此选择了Python3。

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”



然后，命令行下更新pip到新版本：

>pip install --upgrade pip

1. **Python扩展模块的建立**

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 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 交互计算 Jupiter**

1） 安装Jupiter ：

>pip install jupyter 在线安装

2） 安装Python语言内核

>pip install ipython ，支持Python语言

3） 安装依赖包

>pip install pyreadline

>pip install sympy

4）支持显示数学符号、公式，安装MathJax：

命令行下，

>IPython

打开一个IPython的shell，然后，在其中键入如下代码：

from IPython.external.mathjax import install\_mathjax

install\_mathjax()

5） 运行notebook:

在iPython notebook文件所在目录下，打开命令行窗口：

>jupyter notebook

**2.2.3 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. **基于Eclipse的Python集成开发环境**

选择Eclipse作为Python的集成开发环境主要是因为老师的推荐，因为其实对这几个开发环境并不是有很深的了解，所以就选择用老师所推荐的东西。

Eclipse是著名的跨平台的自由集成开发环境（IDE）。最初主要用来 Java语言开发，通过安装不同的插件Eclipse可以支持不同的计算机语言，比如C++和Python等开发工具。Eclipse的本身只是一个框架平台，但是众多插件的支持使得Eclipse拥有其他功能相对固定的IDE软件很难具有的灵活性。许多软件开发商以Eclipse为框架开发自己的IDE。

**3.1 Java的安装**

**3.1.1安装JavaSDK**

因为没有安装java，所以需要先从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。

**3.1.2 Eclipse IDE**

　　 Eclipse IDE是插件型开发环境，有很多版本可以下载。因为本课程主要使用Python语言，会涉及C/C++开发，所以选择下载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32/63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。再将运行eclipse.exe，固定到任务栏，以方便日常的使用。（在eclipse.exe文件名上，点鼠标右键即可）



C:\Users\Administrator\Documents\Tencent Files\523125383\FileRecv\MobileFile\Image\[HZTP5X@~Y83VB~31]QGTKO.png

**3.2 安装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）配置解释器为Python3.5；

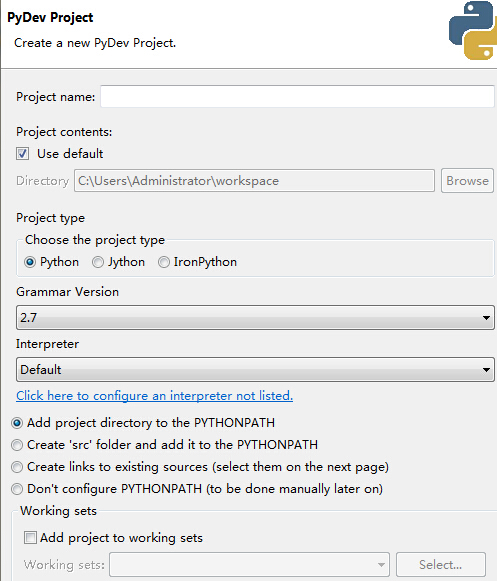
通过Windows->Eclipse Marketplaces进入市场，输入Pydev，找到Pydev安装/更新项目，在线安装即可：



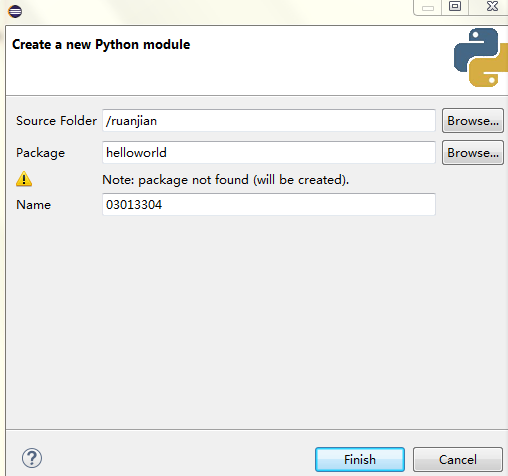
1. **程序的运行与实现**

对于第一次编写程序，绝大多数人应该都会从最基本的“hello world”开始，我也不例外。下面我就简单介绍一下这个最简单的程序的运行：

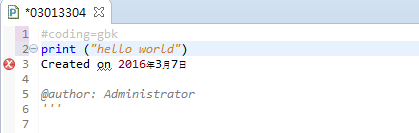
1. 打开Eclipse，通过File->New->Pydev project到以下界面新建一个工程：



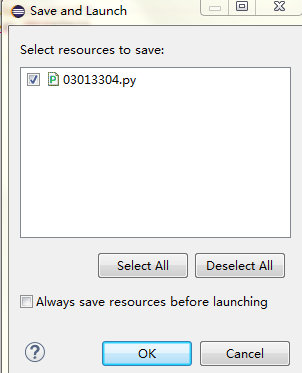
2．通过File->New->Pydev Module到以下界面新建一个Module，我们就是在这里面编写程序：



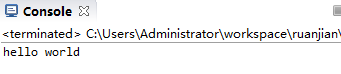
3．在编辑框内编写程序：



4.通过Run运行程序：



运行结果如下：



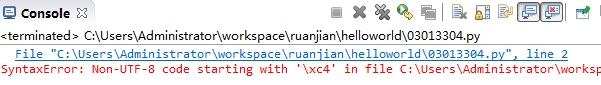
1. **个人小结**

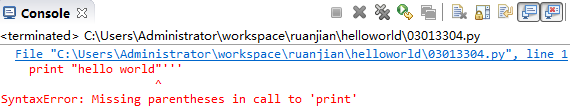
因为是初次接触Python，手头也并没有相关的教材，所以从Python的安装到最简单的“hello world”运行这整个过程中，也出了一些小问题，归结起来主要有以下三个：

1.在第一次安装的过程中，忘记加上环境变量了，导致在pip的安装过程中无法进行。未解决这个问题只能将原先装好的Python卸载干净之后重新安装；

2.在安装Java的过程中，由于没有仔细对照里面提供的版本，导致下载了错误的安装包，没能带起来；

3.在程序试验运行过程中，因为不熟悉操作界面，在开始的时候并没能找到Pydev的位置，没能建立起Module,经过多方询问之后才得以真正编写程序。而开始编写程序时，因为没在里面写过程序，所以对于代码的输入出了两个小问题，一是在语句开头漏加了“# -\*- coding: utf-8 -\*-”，二是(*"hello world"*) 漏加了括号，运行时出现以下提示：





总的来说，对Python的学习总算是正式开了个头。在整个过程中，感觉认真和细心真的是很重要的，一个细节的疏忽很容易导致自己多做许多的无用功，在接下来的学习中也更要重视这一点。在这里，也建议老师在上课过程中，如果时间允许能够尽量带着我们把整个过程走一遍，因为现在绝大多数同学对Python到底是什么可能还没有一个完整的概念，还处于一个懵懵懂懂的状态，所以也希望老师能尽量把东西能够讲得通俗一点。但愿接下来的学习过程能够更加顺利！