

# python初级教程：入门详解

版本：v1.1

Crifan Li

## 摘要

本文是针对Python的初学者，从无到有的介绍Python语言如何入门，主要包括了：Python的简介，如何下载Python，如何安装Python，如何使用终端、Shell，IDE等各种开发环境进行Python开发，Python中的语法和基本知识、概念和逻辑，以及继续深入学习Python的方法，另外还整理一些值得参考的资料。



## 本文提供多种格式供：

在线阅读	<a href="#">HTML</a> <sup>1</sup>	<a href="#">HTMLs</a> <sup>2</sup>	<a href="#">PDF</a> <sup>3</sup>	<a href="#">CHM</a> <sup>4</sup>	<a href="#">TXT</a> <sup>5</sup>	<a href="#">RTF</a> <sup>6</sup>	<a href="#">WEBHELP</a> <sup>7</sup>
下载（7zip压缩包）	<a href="#">HTML</a> <sup>8</sup>	<a href="#">HTMLs</a> <sup>9</sup>	<a href="#">PDF</a> <sup>10</sup>	<a href="#">CHM</a> <sup>11</sup>	<a href="#">TXT</a> <sup>12</sup>	<a href="#">RTF</a> <sup>13</sup>	<a href="#">WEBHELP</a> <sup>14</sup>

HTML版本的在线地址为：

[http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/html/python\\_beginner\\_tutorial.html](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/html/python_beginner_tutorial.html)

有任何意见，建议，提交bug等，都欢迎去讨论组发帖讨论：

[http://www.crifan.com/bbs/categories/python\\_beginner\\_tutorial/](http://www.crifan.com/bbs/categories/python_beginner_tutorial/)

## 修订历史

修订 1.1	2013-10-09	crl
1. 把之前教程的地址移过来		
2. 合并帖子的内容：【整理】【多图详解】如何在Windows下开发Python：在cmd下运行Python脚本+如何使用Python Shell（command line模式和GUI模式）+如何使用Python IDE		

<sup>1</sup> [http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/html/python\\_beginner\\_tutorial.html](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/html/python_beginner_tutorial.html)

<sup>2</sup> [http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/htmls/index.html](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/htmls/index.html)

<sup>3</sup> [http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/pdf/python\\_beginner\\_tutorial.pdf](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/pdf/python_beginner_tutorial.pdf)

<sup>4</sup> [http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/chm/python\\_beginner\\_tutorial.chm](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/chm/python_beginner_tutorial.chm)

<sup>5</sup> [http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/txt/python\\_beginner\\_tutorial.txt](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/txt/python_beginner_tutorial.txt)

<sup>6</sup> [http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/rtf/python\\_beginner\\_tutorial.rtf](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/rtf/python_beginner_tutorial.rtf)

<sup>7</sup> [http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/webhelp/index.html](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/webhelp/index.html)

<sup>8</sup> [http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/html/python\\_beginner\\_tutorial.html.7z](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/html/python_beginner_tutorial.html.7z)

<sup>9</sup> [http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/htmls/index.html.7z](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/htmls/index.html.7z)

<sup>10</sup> [http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/pdf/python\\_beginner\\_tutorial.pdf.7z](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/pdf/python_beginner_tutorial.pdf.7z)

<sup>11</sup> [http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/chm/python\\_beginner\\_tutorial.chm.7z](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/chm/python_beginner_tutorial.chm.7z)

<sup>12</sup> [http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/txt/python\\_beginner\\_tutorial.txt.7z](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/txt/python_beginner_tutorial.txt.7z)

<sup>13</sup> [http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/rtf/python\\_beginner\\_tutorial.rtf.7z](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/rtf/python_beginner_tutorial.rtf.7z)

<sup>14</sup> [http://www.crifan.com/files/doc/docbook/python\\_beginner\\_tutorial/release/webhelp/python\\_beginner\\_tutorial.webhelp.7z](http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/webhelp/python_beginner_tutorial.webhelp.7z)

---

# python初级教程：入门详解:

Crifan Li

版本：v1.1

出版日期 2013-10-09

版权 © 2013 Crifan, <http://crifan.com>

本文章遵从：[署名-非商业性使用 2.5 中国大陆\(CC BY-NC 2.5\)](#)<sup>15</sup>

---

<sup>15</sup> [http://www.crifan.com/files/doc/docbook/soft\\_dev\\_basic/release/html/soft\\_dev\\_basic.html#cc\\_by\\_nc](http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#cc_by_nc)

---

---

# 目录

前言 .....	vii
1. 本文目的 .....	vii
2. 看此文之前你所要掌握的知识 .....	vii
1. Python简介 .....	1
1.1. 什么是Python .....	1
1.1.1. Python脚本的由来 .....	1
1.2. Python能干什么 .....	1
1.3. Python的特点 .....	1
1.3.1. Python的优点 .....	1
1.3.2. Python的缺点 .....	2
1.4. Python中的2.x版本和3.x版本 .....	2
2. 下载Python .....	3
3. 安装Python .....	4
4. Python开发环境 .....	5
4.1. 如何在Windows环境下开发Python .....	5
4.1.1. Python的最原始的开发方式是什么样的 .....	7
4.1.1.1. 找个文本编辑器，新建个.py文件，写上Python代码 .....	7
4.1.1.2. 打开Windows的cmd，并且切换到对应的python脚本所在目录 .....	12
4.1.1.2.1. 方法1：手动打开cmd，并cd到对应路径 .....	12
4.1.1.2.2. 方法2：通过Notepad++的Open current dir cmd .....	12
4.1.1.3. 在cmd中去运行你的Python脚本（.py文件） .....	13
4.1.2. 利用Python的shell进行交互式开发又是什么样的 .....	15
4.1.2.1. 命令行版本的Python Shell – Python (command line) .....	15
4.1.2.2. 带图形界面的Python Shell – IDLE (Python GUI) .....	17
4.1.2.3. 关于（command line或GUI版本的）Python Shell的用途 .....	23
4.1.3. 利用第三方Python的IDE进行Python开发又是怎么回事 .....	23
4.1.3.1. 为何会有Python的IDE .....	24
4.1.3.2. 目前常见的一些Python的IDE .....	24
4.1.3.3. Python的IDE和Python代码编辑器，Windows的cmd，等的关系 .....	25
4.1.3.4. 使用IDE时所遇到的一些常见的问题 .....	26
4.1.3.4.1. IDE只能够打开了文件，并不代表就已经在shell中运行了该文件 .....	26
4.1.3.4.2. 需要注意，确保有可以运行的Python起始部分的代码 .....	28
4.1.4. 总结：到底使用哪种环境去开发Python .....	29
4.1.4.1. 对初学者的建议：如何选用Python的开发环境 .....	29
4.1.5. 如何在Windows环境下使用Python脚本 .....	29
4.1.5.1. 如何在Windows下的cmd中运行BlogsToWordpress.py .....	32
4.2. 如何在Mac环境下运行Python脚本 .....	33
5. Python的语法和基础知识 .....	34
6. Python的学习方法 .....	35
7. Python的相关学习资料 .....	36
参考书目 .....	37

---

## 插图清单

4.1. 在Windows下的cmd下面运行Python脚本的样子 .....	30
4.2. 开始菜单中找到的Python (Command Line) .....	31
4.3. Python (Command Line)的界面 .....	31
4.4. 动画演示如何在Windows的cmd中运行Python脚本BlogsToWordpress.py .....	32
4.5. 在Mac下的Terminal中运行Python脚本：BlogsToWordpress .....	33

---

## 范例清单

4.1. 举例：用Python的IDLE去做URL解码 .....	23
-----------------------------------	----

---

## 公式清单

4.1. 什么是IDE .....	24
4.2. 什么是Python的IDE .....	24

---

# 前言

## 1. 本文目的

本文目的在于，让原先对于Python不熟悉的，甚至没什么概念的人。

看完本系列教程后，从Python的小白，变成，对于Python可以算是入门了。

## 2. 看此文之前你所要掌握的知识

最好有其他计算机语言的基础

比如C语言，Java语言等。

当然，如果没有，也是可以看此文的。

我会在必要的时候，进行相应的提示的。

---

# 第 1 章 Python简介

在开发Python之前，需要先搞懂Python是啥。

下面就是介绍一下，概念性的东西。



## 相关旧帖

[【整理】Python语言简介](#)<sup>1</sup>

## 1.1. 什么是Python



### 什么是解释性语言

简单说就是，无需编译源码为可执行文件，直接使用源码，就可以运行的语言。

此处即，对于Python的源代码，文本文件，直接就可以去执行了，不需要像C，C#等语言，还需要编译一下，才可以运行。

更详细的解释，参见：TODO：解释性语言的详细解释。

### 1.1.1. Python脚本的由来

正是因为Python是门解释型语言，所以才叫Python为脚本的。

## 1.2. Python能干什么

## 1.3. Python的特点

1. 不需要编译，即可运行
2. 有很多特性，使得语言本身很方便编程者使用

### 1.3.1. Python的优点

---

<sup>1</sup> [http://www.crifan.com/simple\\_intro\\_what\\_is\\_python\\_and\\_how\\_to\\_run\\_python\\_script](http://www.crifan.com/simple_intro_what_is_python_and_how_to_run_python_script)



### 1.3.2. Python的缺点

## 1.4. Python中的2.x版本和3.x版本



#### 相关旧帖

[【整理】总结Python2\(Python 2.x版本\)和Python3 \(Python 3.x版本\)之间的区别](#)<sup>2</sup>

[【整理】关于Python 3.x中,使用print函数时出现的语法错误 \(SyntaxError: invalid syntax\) 的问题的原因](#)<sup>3</sup>

在具体介绍, 如何下载Python, 安装Python, 进行Python开发之前, 一个不得不先要解释的话题就是, Python中的版本的问题。

主要包括2.x系列的和3.x系列的, 两者不兼容。

下面详细解释一下。

---

<sup>2</sup> [http://www.crifan.com/summary\\_the\\_difference\\_between\\_python2\\_and\\_python3](http://www.crifan.com/summary_the_difference_between_python2_and_python3)

<sup>3</sup> [http://www.crifan.com/summary\\_reason\\_of\\_python\\_3\\_x\\_print\\_syntaxerror\\_invalid\\_syntax](http://www.crifan.com/summary_reason_of_python_3_x_print_syntaxerror_invalid_syntax)

---

## 第 2 章 下载Python

开发Python之前，肯定是要先搞清楚，如何下载对应的Python。

本来下载一个东西是很简单的事情，但是由于Python有很多版本方面的事情，所以需要搞清楚很多基本逻辑和概念，才知道自己需要下载哪个版本的。



### 相关旧帖

[【教程】如何下载最新版的，各种版本的，包括Python 2.x和Python 3.x的Python](#) <sup>1</sup>

---

<sup>1</sup> [http://www.crifan.com/tutorial\\_how\\_to\\_download\\_latest\\_misc\\_2\\_x\\_3\\_x\\_version\\_python](http://www.crifan.com/tutorial_how_to_download_latest_misc_2_x_3_x_version_python)

---

## 第 3 章 安装Python

在下载完毕Python后，就是去安装Python了。



### 相关旧帖

[【教程】如何在Windows系统中安装Python](http://www.crifan.com/tutorial_how_to_install_python_on_windows) <sup>1</sup>

---

<sup>1</sup> [http://www.crifan.com/tutorial\\_how\\_to\\_install\\_python\\_on\\_windows](http://www.crifan.com/tutorial_how_to_install_python_on_windows)

# 第 4 章 Python开发环境



## 相关旧帖

- [【整理】Python语言简介](#)<sup>1</sup>
- [【整理】各种Python的IDE\(集成开发环境\)的总结和对比](#)<sup>2</sup>
- [【记录】使用Python的IDE : PyScripter](#)<sup>3</sup>
- [【已解决】PyScripter启动出错 : Python could not be properly initialized. We must quit.](#)<sup>4</sup>
- [【记录】使用Python的IDE : Ulipad](#)<sup>5</sup>
- [【已解决】安装Ulipad后，选择启动Ulipad，结果无法启动](#)<sup>6</sup>
- [【记录】使用Python的IDE : Eclipse+PyDev](#)<sup>7</sup>
- [【教程】在Eclipse中安装PyDev](#)<sup>8</sup>
- [【教程】在Eclipse中配置刚安装好的PyDev插件](#)<sup>9</sup>
- [【教程】在Eclipse中使用PyDev进行Python开发](#)<sup>10</sup>
- [【已解决】把Eclipse中的PyDev中的Python代码中的很难看的中文换个好看点的字体](#)<sup>11</sup>
- [【已解决】Eclipse+PyDev无法调试Python : Unexpected IO Exception in Pydev debugger](#)<sup>12</sup>
- [【记录】折腾IDE工具 : Aptana Studio 3](#)<sup>13</sup>
- [【已解决】Aptana Studio 3中通过Auto Config配置PyDev中Python出错 : java.io.IOException: Cannot run program "python": CreateProcess error=2, The system cannot find the file specified](#)<sup>14</sup>

## 4.1. 如何在Windows环境下开发Python

此部分内容的目的：

希望对于，如何在Windows下，写Python代码，进行Python开发，运行Python脚本的人，看了此部分内容后，懂得了：

- 什么是cmd下面去运行Python脚本
- 什么是Python的交互式的shell

<sup>1</sup> [http://www.crifan.com/simple\\_intro\\_what\\_is\\_python\\_and\\_how\\_to\\_run\\_python\\_script](http://www.crifan.com/simple_intro_what_is_python_and_how_to_run_python_script)

<sup>2</sup> [http://www.crifan.com/summary\\_common\\_python\\_ide\\_pyscripter\\_ulipad\\_eclipse\\_pydev\\_eric](http://www.crifan.com/summary_common_python_ide_pyscripter_ulipad_eclipse_pydev_eric)

<sup>3</sup> [http://www.crifan.com/try\\_with\\_python\\_ide\\_pyscripter](http://www.crifan.com/try_with_python_ide_pyscripter)

<sup>4</sup> [http://www.crifan.com/pyscripter\\_start\\_error\\_python\\_could\\_not\\_be\\_properly\\_initialized\\_we\\_must\\_quit](http://www.crifan.com/pyscripter_start_error_python_could_not_be_properly_initialized_we_must_quit)

<sup>5</sup> [http://www.crifan.com/try\\_with\\_python\\_ide\\_ulipad](http://www.crifan.com/try_with_python_ide_ulipad)

<sup>6</sup> [http://www.crifan.com/ulipad\\_after\\_install\\_finish\\_not\\_launch](http://www.crifan.com/ulipad_after_install_finish_not_launch)

<sup>7</sup> [http://www.crifan.com/try\\_with\\_python\\_ide\\_eclipse\\_pydev](http://www.crifan.com/try_with_python_ide_eclipse_pydev)

<sup>8</sup> [http://www.crifan.com/eclipse\\_install\\_plugin\\_pydev](http://www.crifan.com/eclipse_install_plugin_pydev)

<sup>9</sup> [http://www.crifan.com/eclipse\\_configure\\_newly\\_installed\\_plugin\\_pydev](http://www.crifan.com/eclipse_configure_newly_installed_plugin_pydev)

<sup>10</sup> [http://www.crifan.com/eclipse\\_use\\_pydev\\_develop\\_python](http://www.crifan.com/eclipse_use_pydev_develop_python)

<sup>11</sup> [http://www.crifan.com/eclipse\\_pydev\\_change\\_ugly\\_zhcn\\_char\\_to\\_another\\_font](http://www.crifan.com/eclipse_pydev_change_ugly_zhcn_char_to_another_font)

<sup>12</sup> [http://www.crifan.com/eclipse\\_pydev\\_python\\_unexpected\\_io\\_exception\\_in\\_pydev\\_debugger](http://www.crifan.com/eclipse_pydev_python_unexpected_io_exception_in_pydev_debugger)

<sup>13</sup> [http://www.crifan.com/try\\_dev\\_ide\\_aptana\\_studio\\_3](http://www.crifan.com/try_dev_ide_aptana_studio_3)

<sup>14</sup> [http://www.crifan.com/apatana\\_studio\\_3\\_auto\\_config\\_pydev\\_error\\_java\\_io\\_ioexception\\_cannot\\_run\\_program\\_python\\_createprocess\\_error\\_2](http://www.crifan.com/apatana_studio_3_auto_config_pydev_error_java_io_ioexception_cannot_run_program_python_createprocess_error_2)

- 什么是Python的IDE

看此部分内容之前，需要具有以下前提：

- Python语言的基本知识  
包括知道其代码就是普通文本等基础知识

不了解的，可以先去看：

[【整理】计算机语言基础知识介绍](#)<sup>15</sup>

以了解关于计算机语言的宏观介绍；

再去看：

[【整理】Python语言简介](#)<sup>16</sup>

就明白了。

- 已经在Windows中安装好了Python  
关于此部分内容，详见：

[第 3 章 安装Python](#)

另外提示一下，关于版本的选择：

[【整理】总结Python2\(Python 2.x版本\)和Python3 \(Python 3.x版本\)之间的区别](#)<sup>17</sup>

下面，通过最简单的Python代码,此处只是打印一些Python版本信息和系统信息：

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
"""
Function:
【整理】如何在Windows下开发Python（如何运行Python脚本）
http://www.crifan.com/how\_to\_do\_python\_development\_under\_windows\_environment

Author: Crifan Li
Version: 2012-12-06
"""

import platform;

pythonVersion = platform.python_version();
uname = platform.uname();

print "Just for demo how to do python development under windows:";
print "Current python version info is %s"%(pythonVersion);
print "uname=",uname;
```

<sup>15</sup> [http://www.crifan.com/computer\\_language\\_basic\\_knowledge\\_introduction](http://www.crifan.com/computer_language_basic_knowledge_introduction)

<sup>16</sup> [http://www.crifan.com/simple\\_intro\\_what\\_is\\_python\\_and\\_how\\_to\\_run\\_python\\_script](http://www.crifan.com/simple_intro_what_is_python_and_how_to_run_python_script)

<sup>17</sup> [http://www.crifan.com/summary\\_the\\_difference\\_between\\_python2\\_and\\_python3](http://www.crifan.com/summary_the_difference_between_python2_and_python3)

来说明，如何在Windows下，进行Python开发。

### 4.1.1. Python的最原始的开发方式是什么样的

相对来说，最原始的，最基本的，开发Python的方式，只是：

#### 4.1.1.1. 找个文本编辑器，新建个.py文件，写上Python代码

Python代码，本身就只是文本；

所以，从原理上来说，只需要找个合适的文本编辑器，即可；

换句话说，如果只是简单的写写几行Python代码，你用Windows的记事本（Notepad），都是可以的；

只不过，由于Notepad功能太弱，所以在此个人推荐Notepad++。



#### 为何推荐用Notepad++去写Python代码

此处，推荐用Notepad++去写Python代码，做Python开发，的原因

除了本身Notepad++作为文本编辑器，好用之外，

还有一个，相对来说，比较重要的问题：字符编码

很多时候，由于Python开发者对于Python文件的编码，不太了解

导致在开发期间，出现很多，相对比较常见的字符编码的问题

所以，为了，更加深入的了解和学习，真正的掌握Python

有必要，从一开始学习Python的时候

就选用合适的工具，这样，避免后期，一些该知道的细节，尤其是文件的编码，不知道而犯各种常见的错误

而关于Notepad++本身，不会使用，不熟悉，则可参考：

[【crifan推荐】轻量级文本编辑器，Notepad最佳替代品：Notepad++](#)<sup>18</sup>

关于Notepad++的几乎一切的内容，都可以在上述的教程中找到。

此处呢，对应的就是：

用我所推荐的Notepad++，新建一个文件，然后包括代码进入，存为对应的一个.py的文件。

而关于如何操作，此处也一点点截图，透彻的说明一下：

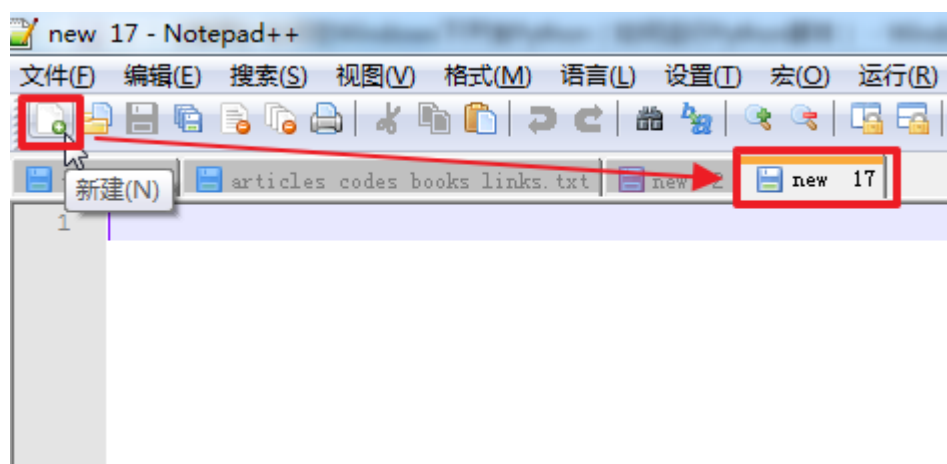
此处再次提示一下，一下Notepad++的所有功能，特性，用法，上面那个帖子中，都有专门的介绍。

想要更加详细了解的相关的Notepad++的某个特定功能的用法的，自己去看即可。

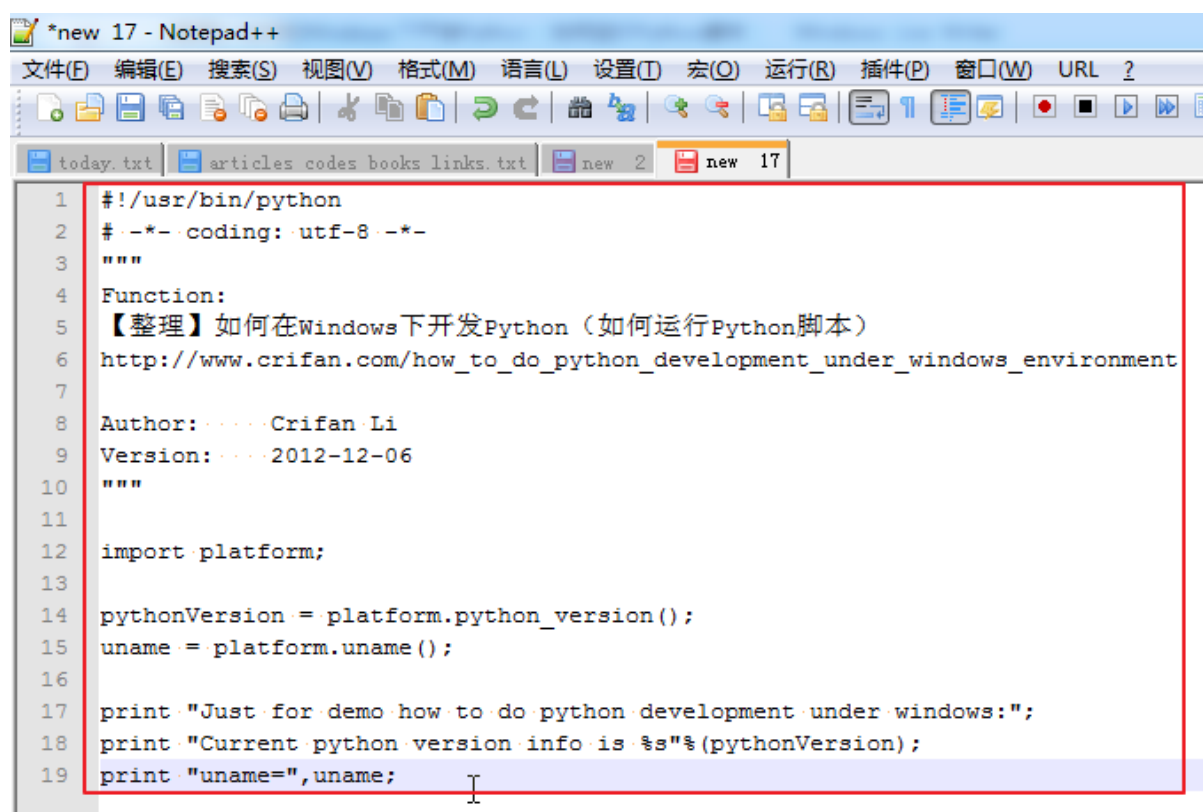
<sup>18</sup> [http://www.crifan.com/files/doc/docbook/rec\\_soft\\_npp/release/html/rec\\_soft\\_npp.html](http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/html/rec_soft_npp.html)

下面的截图，就不再事无巨细的解释Notepad++的功能特点了。只是截图解释如何操作而已。

打开Notepad++后，点击新建，新建出一个新的文件：



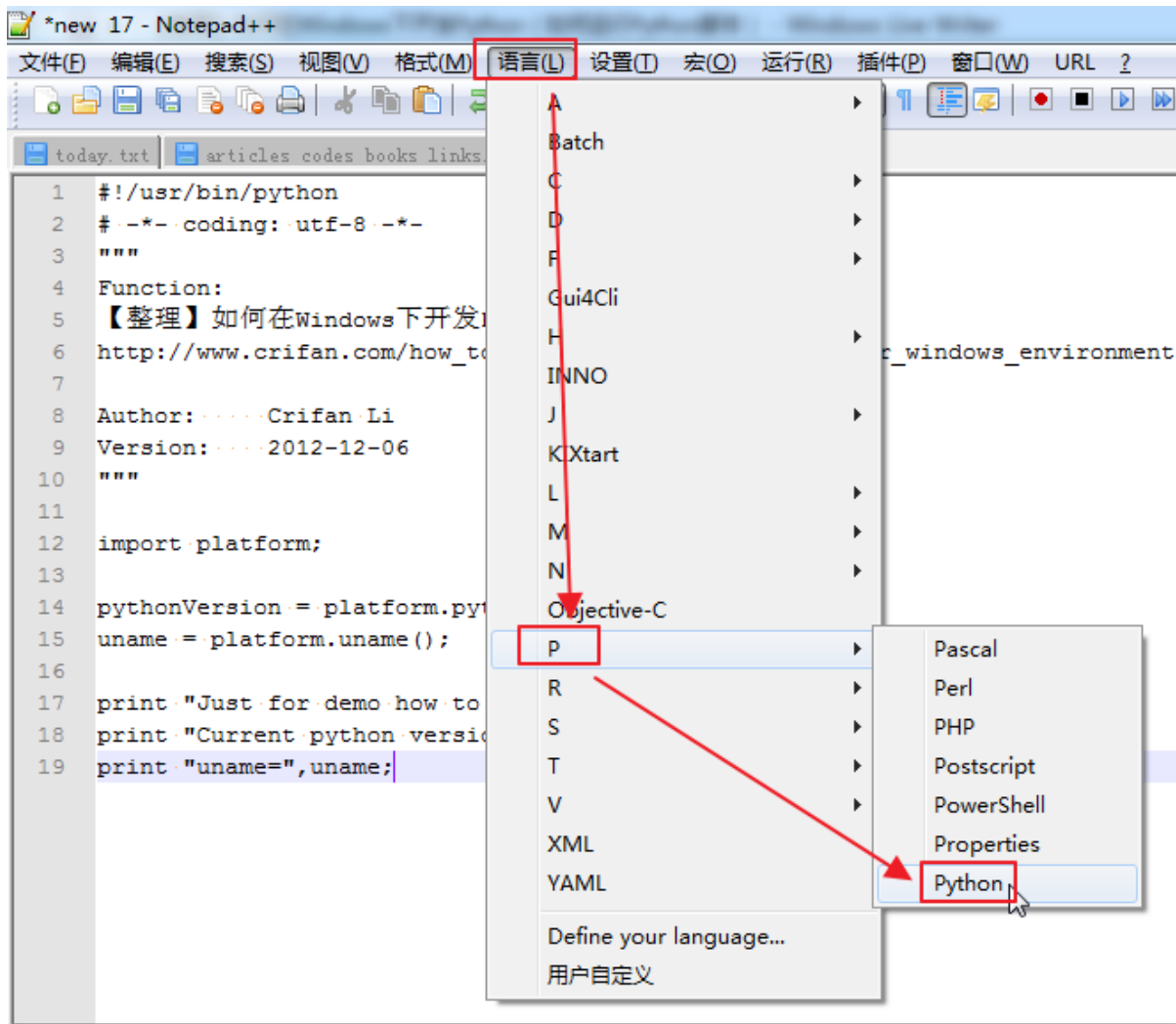
然后把上面给你们的代码，拷贝进去，就变成了：



此处由于是新建的空白文件，所以Notepad++，不知道你是Python代码，没法帮你自动实现语法高亮，

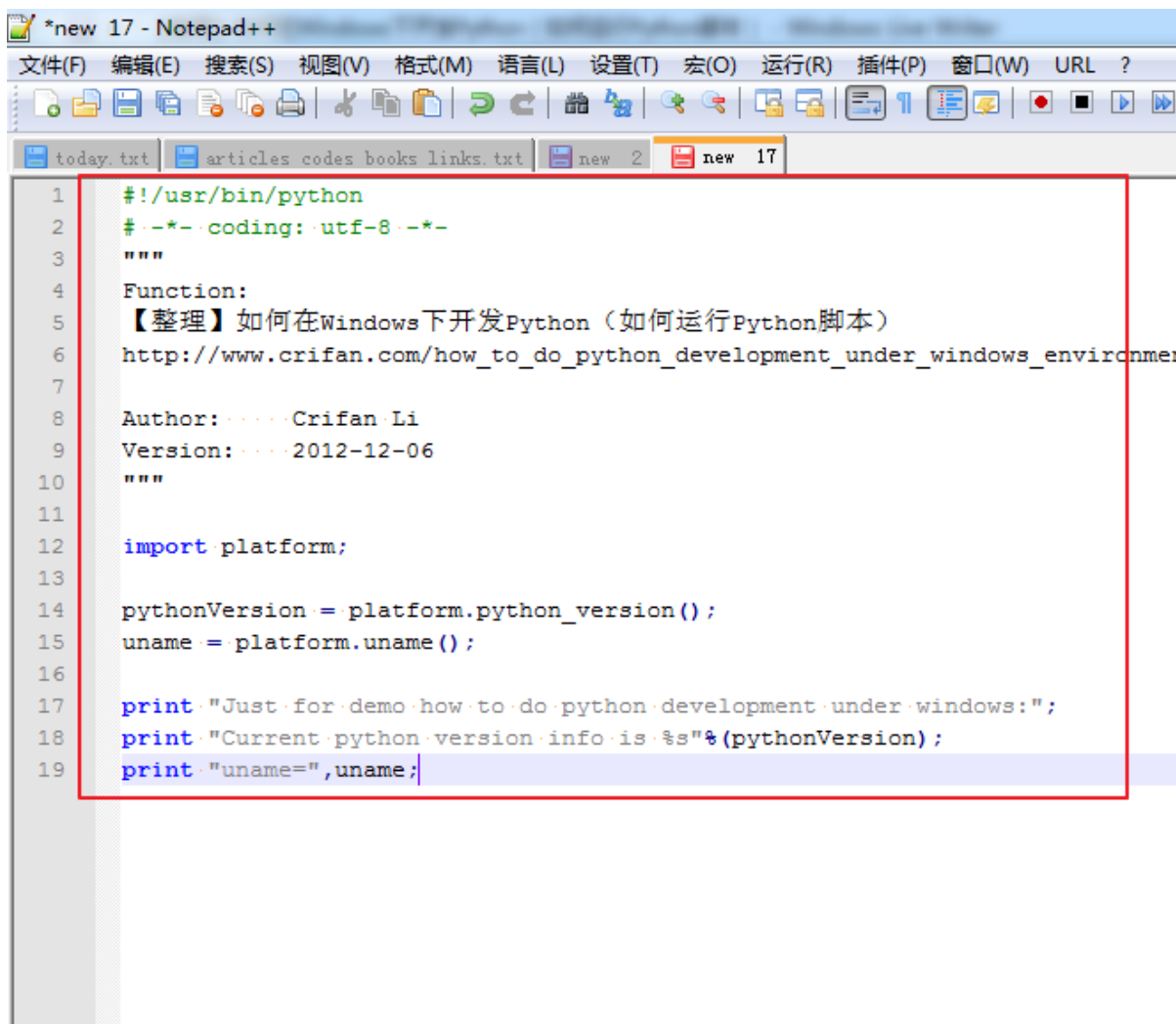
需要手动去设置一下：

语言⇒P⇒Python



就可以看到Python代码的语法高亮的效果了：





```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  """
4  Function:
5  【整理】如何在Windows下开发Python（如何运行Python脚本）
6  http://www.crifan.com/how_to_do_python_development_under_windows_environment
7
8  Author: . . . . Crifan Li
9  Version: . . . . 2012-12-06
10 """
11
12 import platform;
13
14 pythonVersion = platform.python_version();
15 uname = platform.uname();
16
17 print "Just for demo how to do python development under windows:";
18 print "Current python version info is %s"%(pythonVersion);
19 print "uname=",uname;
```

然后去保存到某个位置。

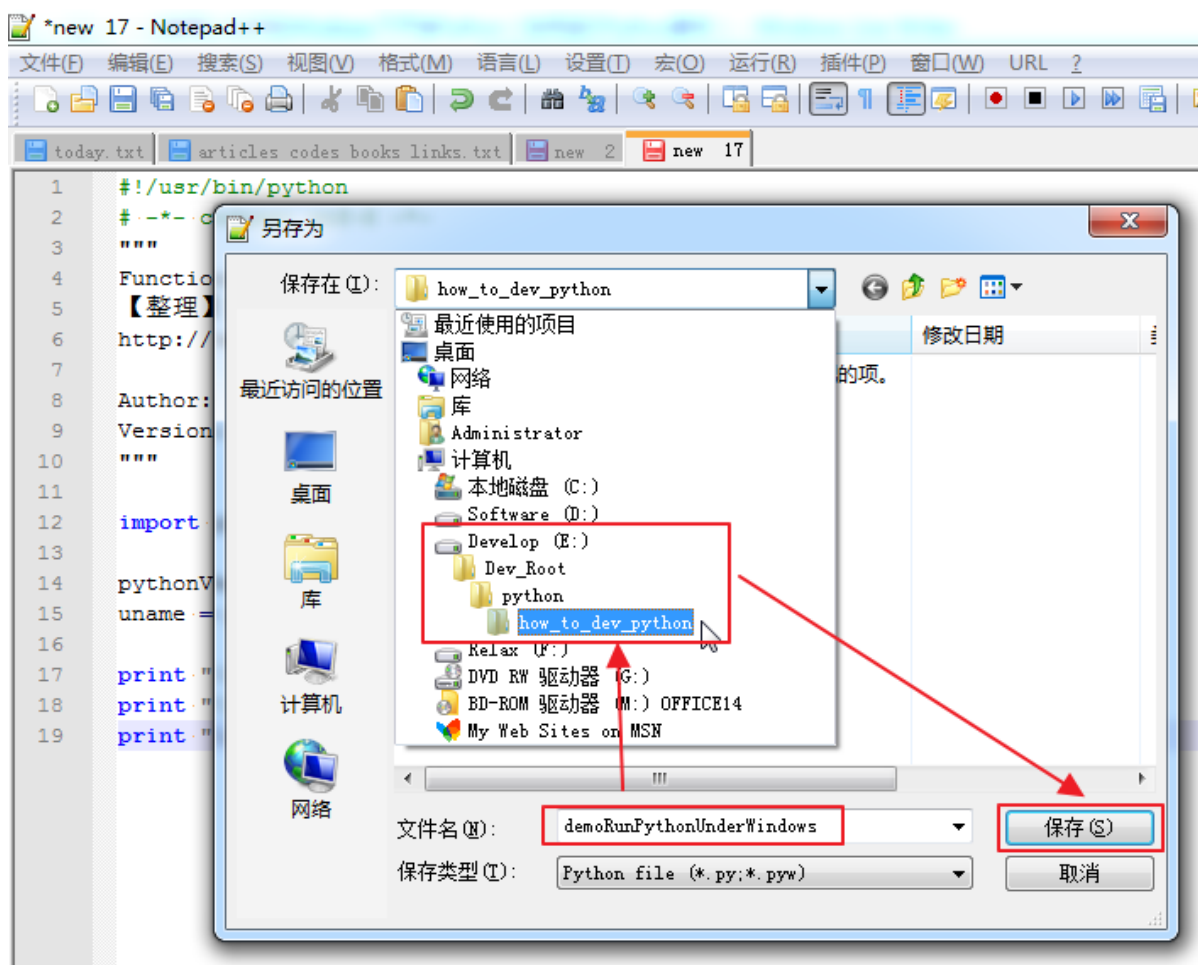
文件⇒另存为

在弹出的对话框中，输入要保存的文件名：demoRunPythonUnderWindows

提示：其中可以看到[Notepad++](http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/html/rec_soft_npp.html)<sup>19</sup>自动帮你写好了.py后缀，那是因为你之前设置了Python语法高亮。

然后再选择对应的路径去保存：

<sup>19</sup> [http://www.crifan.com/files/doc/docbook/rec\\_soft\\_npp/release/html/rec\\_soft\\_npp.html](http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/html/rec_soft_npp.html)



## 路径中不要包含（中文，空格等）特殊字符

对于保存文件来说，需要注意的是：

Python，以及其他语言，开发期间，最好都不要让路径中带有特殊字符

此处所谓的特殊字符，指的是非（ASCII）英文字符，主要包括：

- 中文字符
- 空格
- 其他特殊字符

而只包含普通的英文字符，即字母数字下划线。

比如，我此处的路径，用的是：

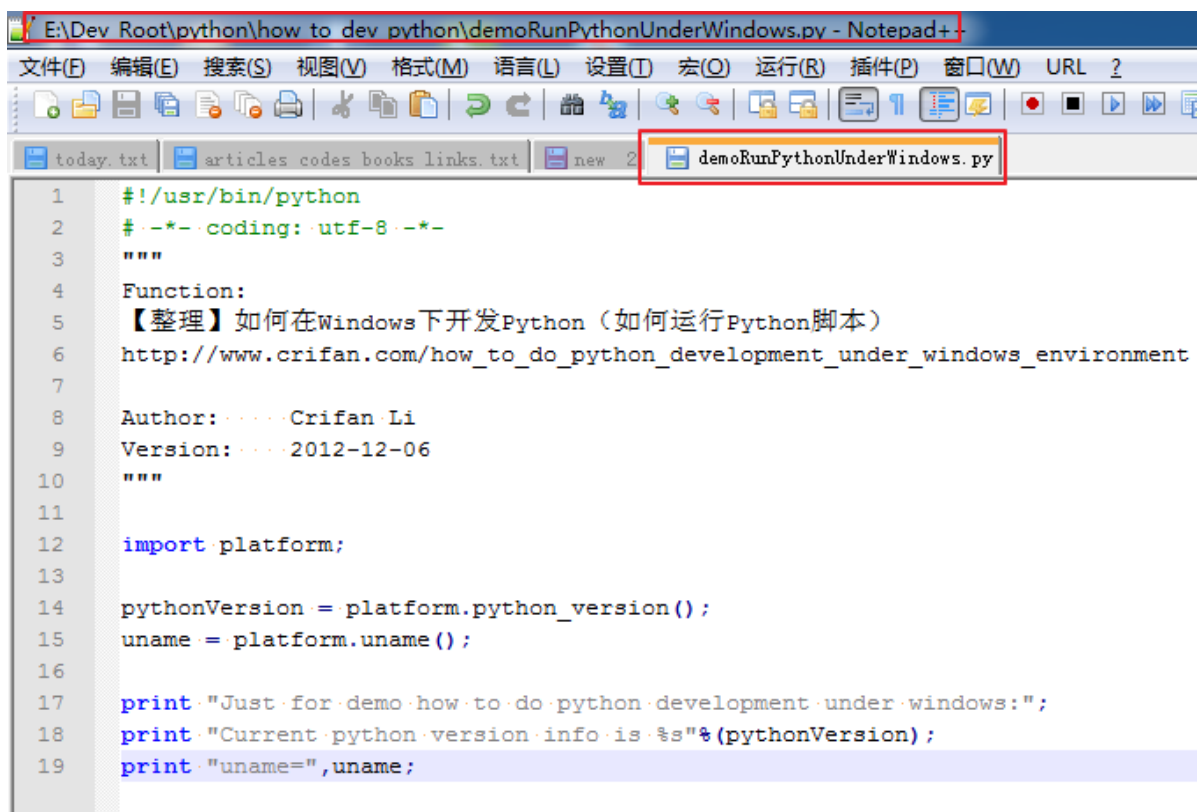
E:\Dev\_Root\python\how\_to\_dev\_python\

否则，很容易在后期开发期间，出现一些异常现象

比如找不到文件，文件夹等问题

以及常见的中文路径无法识别等问题。

保存好后，就是这样的了：



```
1  #!/usr/bin/python
2  #-*-coding:utf-8-*-
3  """
4  Function:
5  【整理】如何在Windows下开发Python（如何运行Python脚本）
6  http://www.crifan.com/how_to_do_python_development_under_windows_environment
7
8  Author: .....Crifan Li
9  Version: .....2012-12-06
10 """
11
12 import platform;
13
14 pythonVersion = platform.python_version();
15 uname = platform.uname();
16
17 print "Just for demo how to do python development under windows:";
18 print "Current python version info is %s"%(pythonVersion);
19 print "uname=",uname;
```

#### 4.1.1.2. 打开Windows的cmd，并且切换到对应的python脚本所在目录

且换到对应的，Python文件所在的，文件夹，有两种办法：

##### 4.1.1.2.1. 方法1：手动打开cmd，并cd到对应路径

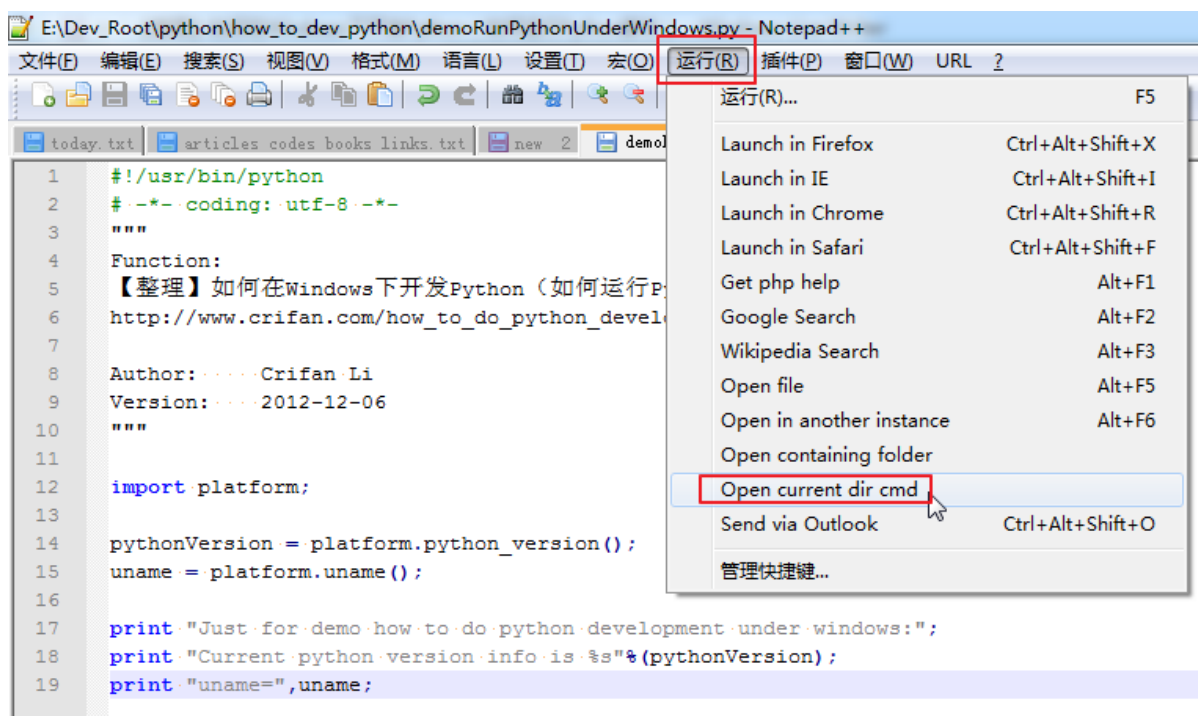
而对于，什么是windows的cmd，如何打开cmd，如何切换到对应的路径等内容，不熟悉的话，可以去参考：

[Windows的命令行工具: cmd](http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#win_cmd) <sup>20</sup>

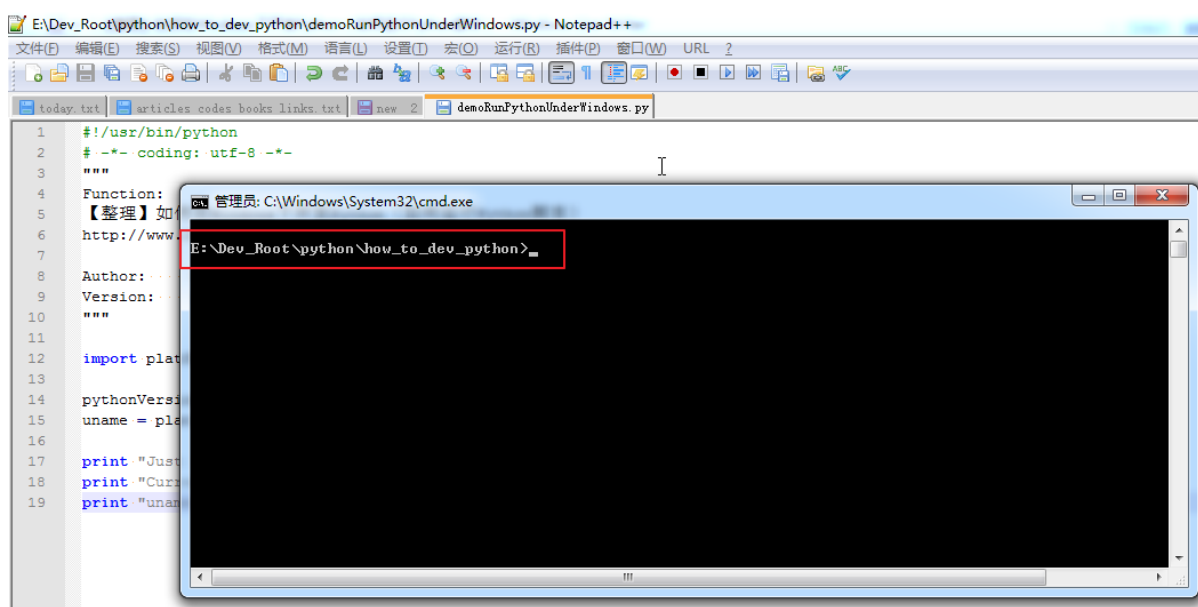
##### 4.1.1.2.2. 方法2：通过Notepad++的Open current dir cmd

此处，使用一个更方便的办法，利用Notepad++中功能，直接打开cmd，并切换到对应路径：

<sup>20</sup> [http://www.crifan.com/files/doc/docbook/soft\\_dev\\_basic/release/html/soft\\_dev\\_basic.html#win\\_cmd](http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#win_cmd)



然后就直接实现，打开了cmd，并且切换到对应的当前文件所在路径了：



#### 4.1.1.3. 在cmd中去运行你的Python脚本（.py文件）

然后就是，在cmd中，输入你的Python脚本，即.py文件的完整的文件名。

此处是：

demoRunPythonUnderWindows.py



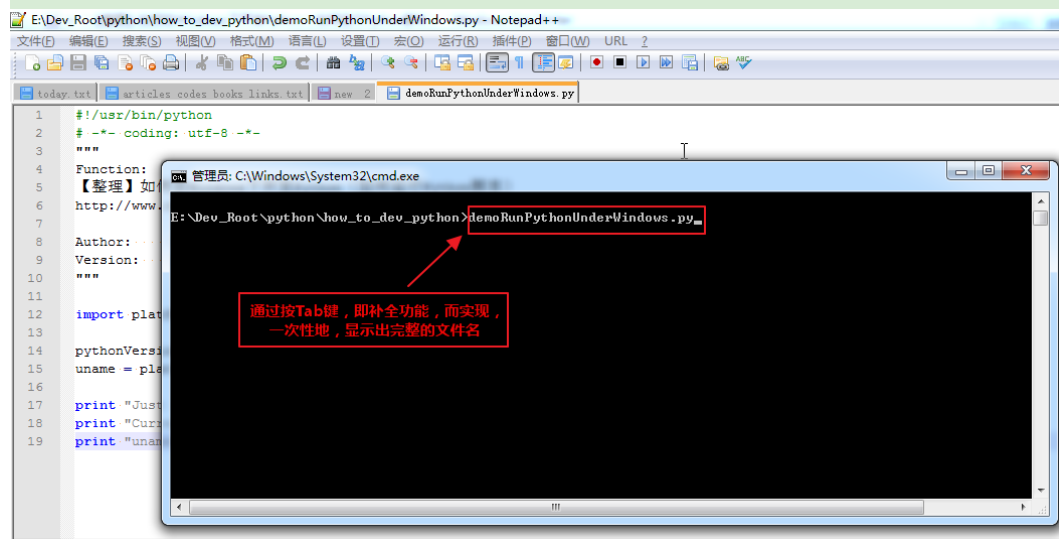
#### 利用TAB键实现自动输入

正常的是通过手动输入对应的Python的完整的文件名，即，对于此处的demoRunPythonUnderWindows.py，一个个字母的输入

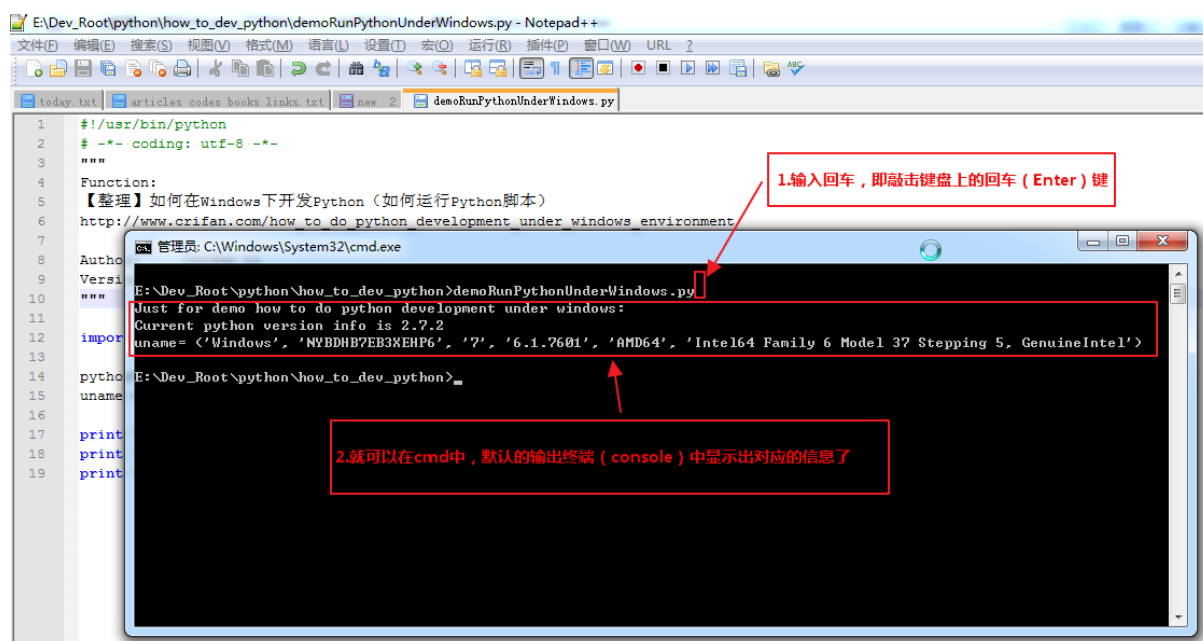
不过，此处有个小技巧，可以利用Windows（和Linux中通用的）**Tab**键，然后会自动显示出你当前目录的文件；

如果当前文件夹有多个文件，多次按**Tab**键，会在多个文件之间切换。

此处就一个文件，所以通过按**Tab**键，就可以一下子就显示出对应的整个文件名了：



然后输入回车，即可运行对应的Python脚本，接着在cmd中也就可以看到输出的结果了：



如此，就是一个，完整的，在windows的cmd中，运行Python脚本的流程了。

而接下来，作为正常的开发Python的流程就只是：

- 你继续去修改你的Python代码，添加新的代码，然后保存python文件，
- 再回到cmd中，重新再次运行Python，以验证程序运行是否正常，是否获得了你所期望的结果。
- 如此反复，一点点，由少到多，一点点写出足够复杂的Python代码，实现你的复杂的功能。

## 4.1.2. 利用Python的shell进行交互式开发又是怎样的

Python有个shell，提供一个Python运行环境。方便你交互式开发。

即写一行代码，就可以立刻被运行，然后方便查看到结果。

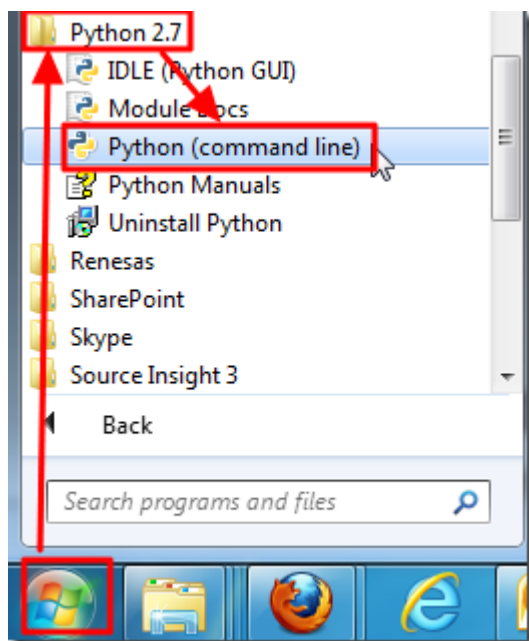
而Python的Shell，在Windows环境下，又分两种：

- Python (command line)
- IDLE (Python GUI)

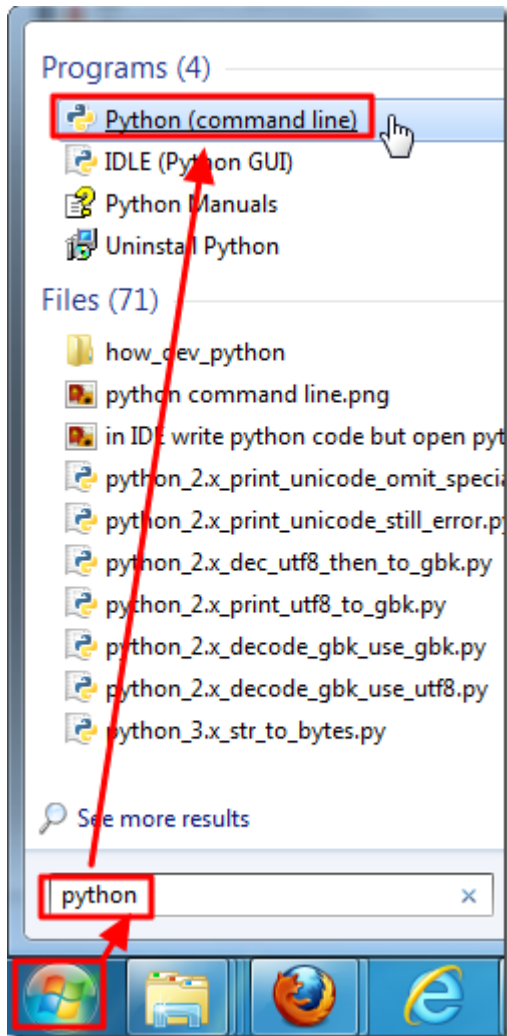
下面分别介绍一下。

### 4.1.2.1. 命令行版本的Python Shell – Python (command line)

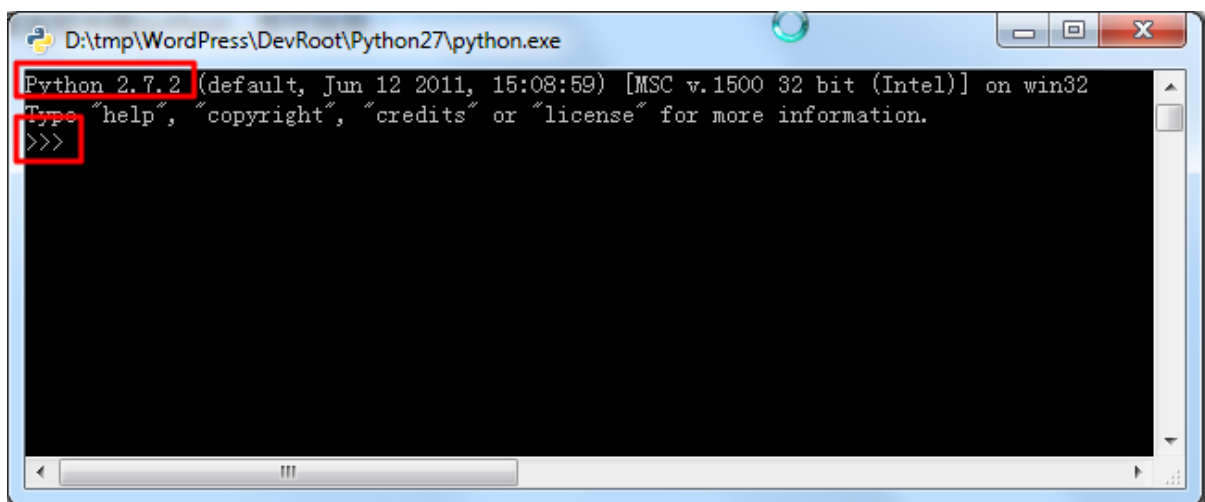
Windows下，安装好Python后，你可以在开始菜单中，找到对应的command line版本的Python Shell的：



其实，Win7中，有个更方便的方式，直接在搜索框中搜python，即可找到：



打开后，就是这个样子的：



其中可以显示出来对应的python版本信息和系统信息。

然后就是三个大于号>>>接下来，你就可以像在之前所说的，

普通文本中输入python代码一样，在此一行行输入代码，然后就可以显示对应的信息了：

```

D:\tmp\WordPress\DevRoot\Python27\python.exe
Python 2.7.2 (default, Jun 12 2011, 15:08:59) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import platform;
>>> pythonVersion = platform.python_version();
>>> uname = platform.uname();
>>> print "Just for demo how to do python development under windows:";
Just for demo how to do python development under windows:
>>>

```



## 为何叫做交互式shell (interactive shell)

其中可以看到，当你输入对应的，第一行print时，此处命令行 (command line) 版本的Python Shell中，

就可以，动态的，交互式地，显示出对应的信息了。

正由于，此处可以，直接地，动态的，交互性地，显示出对应的信息，

所以，才被叫做Python 的交互式的Shell，简称Python Shell。

对应的，把前面的代码都输入完毕，结果显示为：

```

D:\tmp\WordPress\DevRoot\Python27\python.exe
Python 2.7.2 (default, Jun 12 2011, 15:08:59) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import platform;
>>> pythonVersion = platform.python_version();
>>> uname = platform.uname();
>>> print "Just for demo how to do python development under windows:";
Just for demo how to do python development under windows:
>>> print "Current python version info is %s"%(pythonVersion);
Current python version info is 2.7.2
>>> print "uname=",uname;
uname= ('Windows', 'PC-CLI-1', '7', '6.1.7601', 'AMD64', 'Intel64 Family 6 Model 42 Stepping 7, GenuineIntel')
>>>

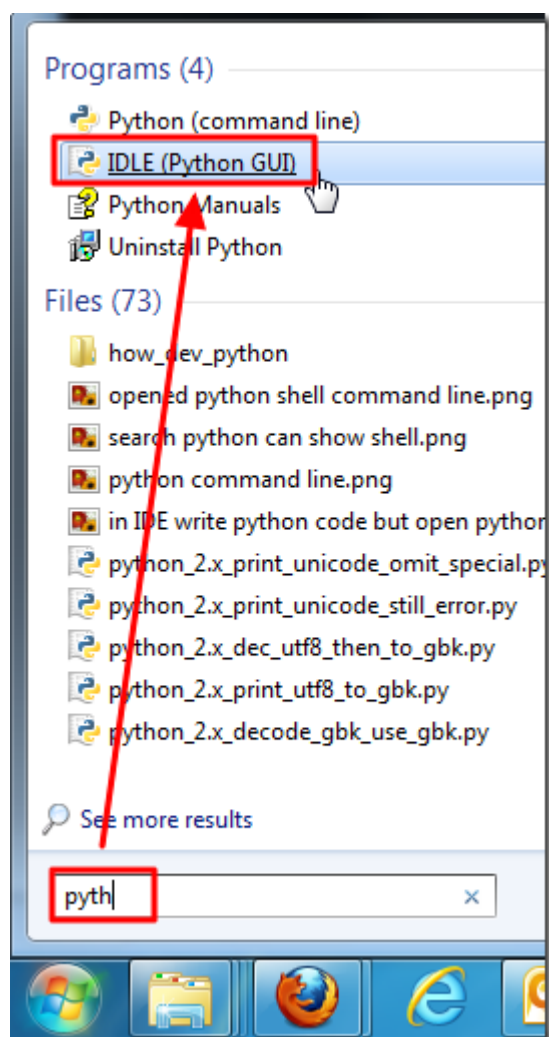
```

### 4.1.2.2. 带图形界面的Python Shell – IDLE (Python GUI)

下面将要介绍的，同样是Python的Shell，但是是相对于command line版本而言，是GUI，带图形界面的版本的。

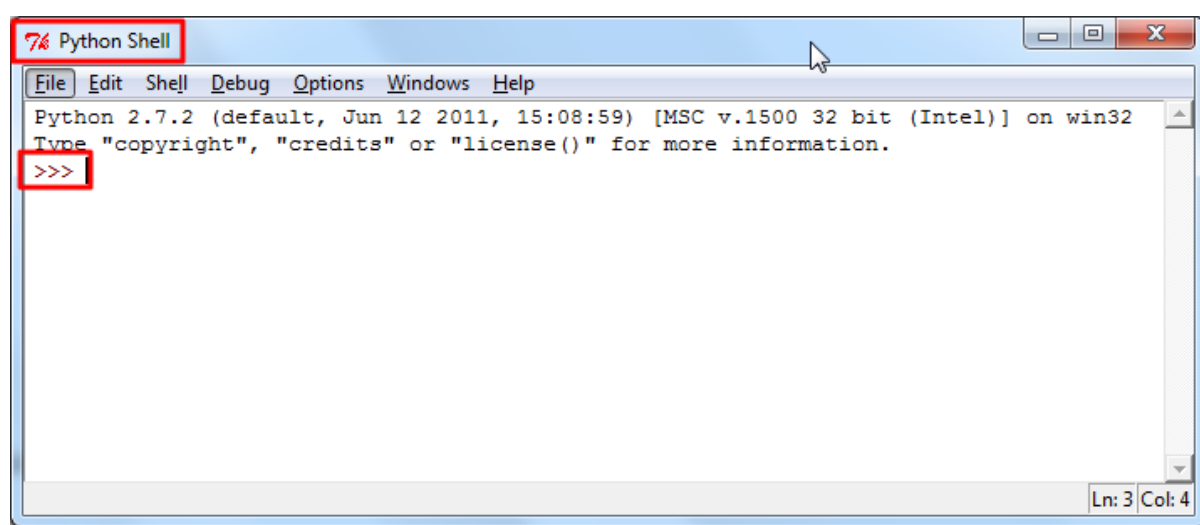
对应的打开方式，和上面类似，可以直接搜python而找到：





对应的，可以看到，其名字写的是IDLE

打开后就是这样的：



**IDLE左上角的红色TK图标表示IDLE是用Tkinter图形库开发的**

左上角的红色图标，好像是TK，是图形界面库的一种。

这个就是，很多Python教程中，

所常用来作为Python的开发环境，教别人写Python代码的那个IDLE。

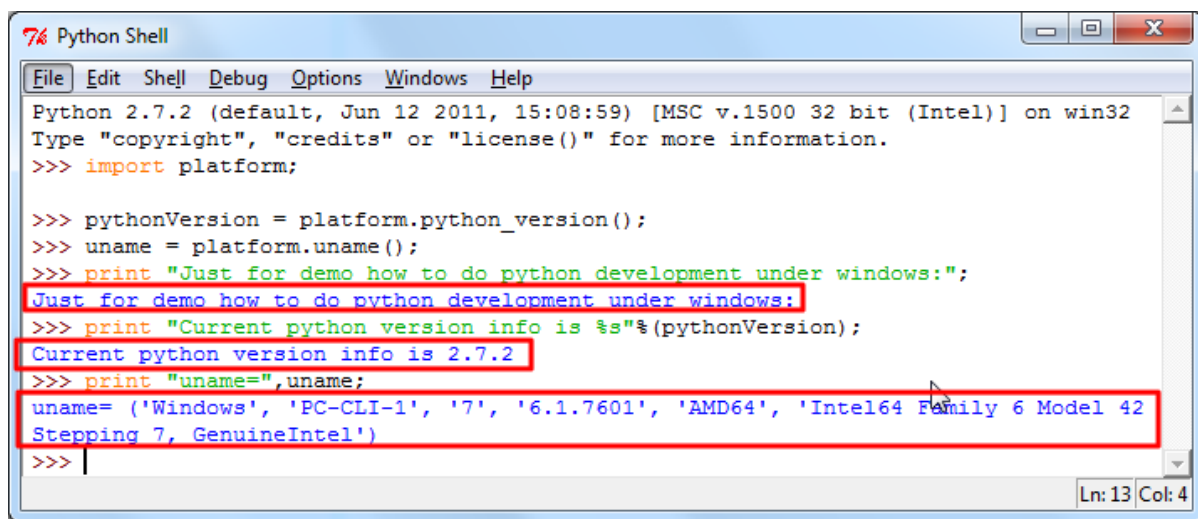
关于Python中的Tkinter图形库，详见：

[【整理】Python中的图形库](#)<sup>21</sup>

中的：

[【记录】折腾Python中的Tkinter](#)<sup>22</sup>

对应的，输入上述的代码，结果也是类似的：



```
Python 2.7.2 (default, Jun 12 2011, 15:08:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import platform;

>>> pythonVersion = platform.python_version();
>>> uname = platform.uname();
>>> print "Just for demo how to do python development under windows:"
Just for demo how to do python development under windows:
>>> print "Current python version info is %s"%(pythonVersion);
Current python version info is 2.7.2
>>> print "uname=",uname;
uname= ('Windows', 'PC-CLI-1', '7', '6.1.7601', 'AMD64', 'Intel64 Family 6 Model 42
Stepping 7, GenuineIntel')
>>>
```



## IDLE中一次性粘贴多行代码再运行则会出错

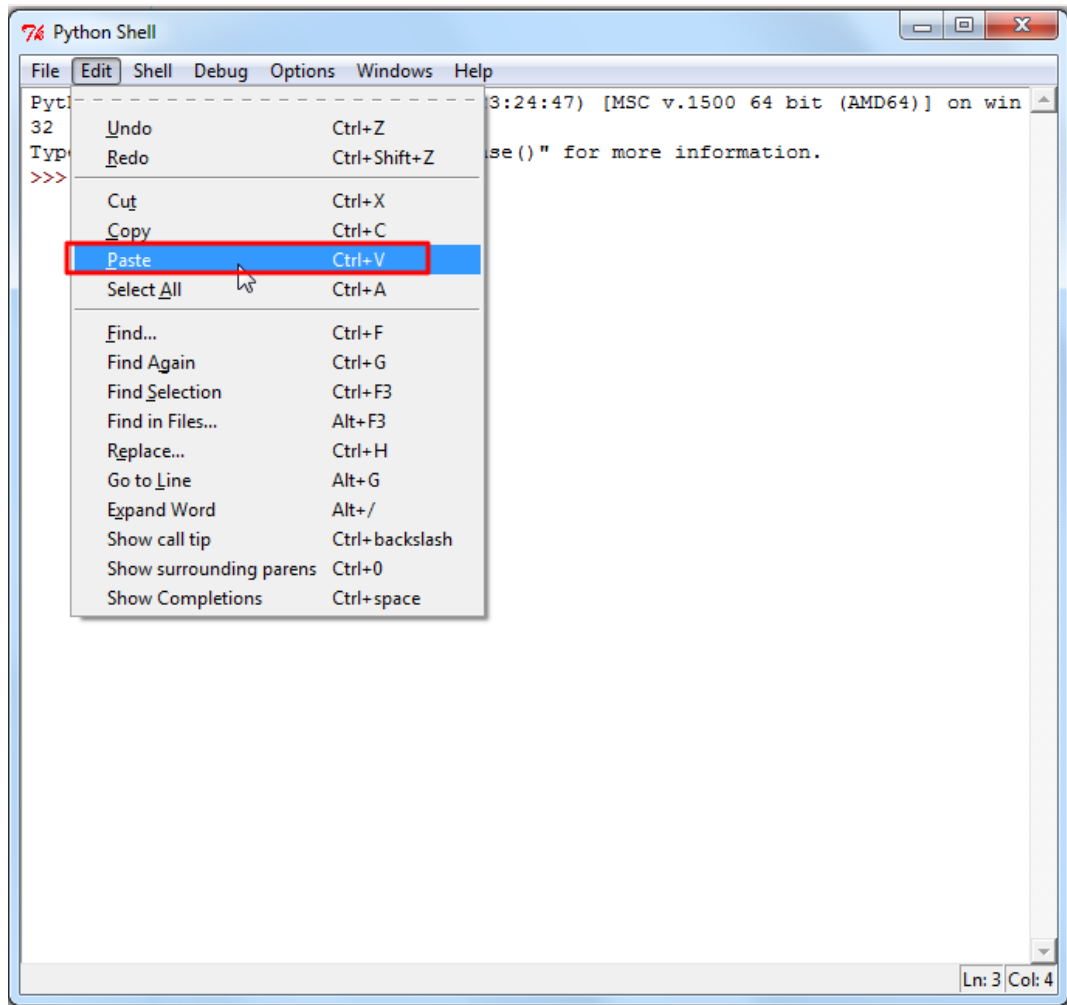
如果你对于IDLE的运行方式：交互式的，一行一行的输入，才能正常运行

不熟悉的话，作为新手，有些人会犯这样的错误：

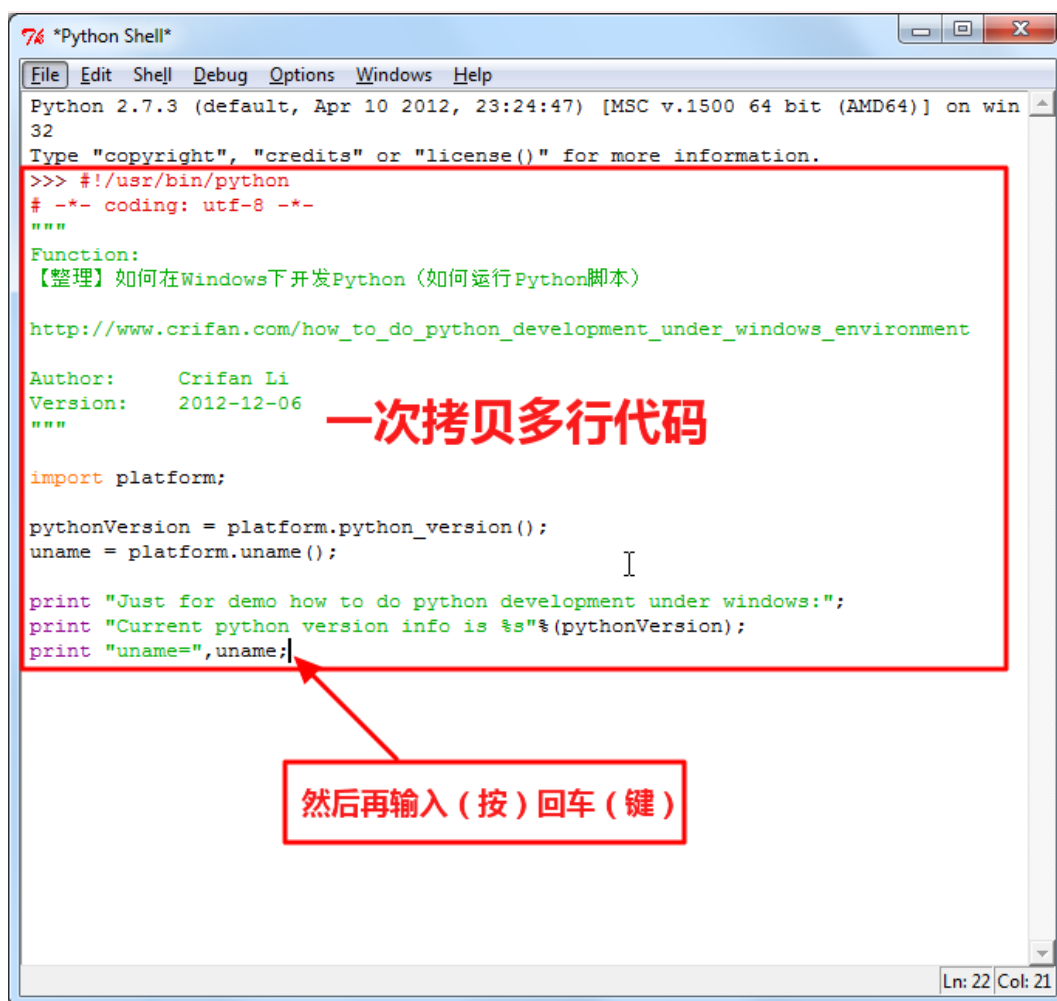
在别处拷贝Python示例代码，然后一次性的粘贴多行代码到IDLE中

<sup>21</sup> [http://www.crifan.com/summary\\_python\\_graphics\\_gui\\_libs\\_packages/](http://www.crifan.com/summary_python_graphics_gui_libs_packages/)

<sup>22</sup> [http://www.crifan.com/try\\_python\\_tkinter\\_module/](http://www.crifan.com/try_python_tkinter_module/)



然后输入回车去运行：



7% \*Python Shell\*

File Edit Shell Debug Options Windows Help

Python 2.7.3 (default, Apr 10 2012, 23:24:47) [MSC v.1500 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

```
>>> #!/usr/bin/python
# -*- coding: utf-8 -*-
"""
Function:
【整理】如何在Windows下开发Python (如何运行Python脚本)

http://www.crifan.com/how_to_do_python_development_under_windows_environment

Author:      Crifan Li
Version:     2012-12-06
"""

import platform;

pythonVersion = platform.python_version();
uname = platform.uname();

print "Just for demo how to do python development under windows:";
print "Current python version info is %s"%(pythonVersion);
print "uname=",uname;
```

一次拷贝多行代码

然后再输入 (按) 回车 (键)

Ln: 22 Col: 21

结果，很明显，会出错：

此处是不能正常输入，之前所期望的python版本信息，

而是出现其他异常情况：只是显示此处的注释代码

(因为此处最开始部分的代码，就只是注释代码)

(如果你是其他的多行代码，则对应的就是出现其他对应的错误了)

```

Python Shell
File Edit Shell Debug Options Windows Help
Python 2.7.3 (default, Apr 10 2012, 23:24:47) [MSC v.1500 64 bit (AMD64)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> #!/usr/bin/python
# -*- coding: utf-8 -*-
"""
Function:
【整理】如何在Windows下开发Python (如何运行Python脚本)

http://www.crifan.com/how_to_do_python_development_under_windows_environment

Author:      Crifan Li
Version:     2012-12-06
"""

import platform;

pythonVersion = platform.python_version();
uname = platform.uname();

print "Just for demo how to do python development under windows:";
print "Current python version info is %s"%(pythonVersion);
print "uname=",uname:
'\nFunction:\n\\xa1\xbe\xd5\xfb\xc0\xed\xa1\xbf\xc8\xe7\xba\xce\xd4\xdaWindows\xcf\x2\xbf\xaa\xb7\xa2Python\xa3\xa8\xc8\xe7\xba\xce\xd4\xcb\xd0\xd0Python\xbd\x5\xbf\xbe\xa3\xa9\n \nhttp://www.crifan.com/how_to_do_python_development_under_w
indows_environment\n \nAuthor:      Crifan Li\nVersion:     2012-12-06\n'
>>>

```

**IDLE中无法执行你所，一次性拷贝的多行代码，所以此处只是  
 显示出一些错误（不正常的信息，非正常代码执行的输出信息）**

而，如果想要正确的运行代码，

则应该和之前一样，正常的，一行一行的输入代码，并回车，去执行

才能正常的，显示出你所希望看到的信息

```

Python 2.7.3 (default, Apr 10 2012, 23:24:47) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> #!/usr/bin/python
# -*- coding: utf-8 -*-
"""
Function:
【整理】如何在Windows下开发Python (如何运行Python脚本)

http://www.crifan.com/how_to_do_python_development_under_windows_environment

Author:    Crifan Li
Version:   2012-12-06
"""

import platform;

pythonVersion = platform.python_version();
uname = platform.uname();

print "Just for demo how to do python development under windows:";
print "Current python version info is %s"%(pythonVersion);
print "uname=",uname;
'\nFunction:\n\xa1\xbe\xd5\xfb\xc0\xda\xa1\xbf\xc8\xe7\xba\xce\xda\xdaWindows\xcf\xc2\xbf\xaa\xb7\xa2Python\xa3\x
\xa8\xc8\xe7\xba\xce\xda\xcb\xd0Python\xbd\xc5\xbl\xbe\xa3\xa9\n \nhttp://www.crifan.com/how_to_do_python_dev
elopment_under_windows_environment\n \nAuthor:    Crifan Li\nVersion:   2012-12-06\n'
>>> import platform;
>>> pythonVersion = platform.python_version();
>>> uname = platform.uname();
>>> print "Just for demo how to do python development under windows:";
Just for demo how to do python development under windows:
>>> print "Current python version info is %s"%(pythonVersion);
Current python version info is 2.7.3
>>> print "uname=",uname;
uname= ('Windows', 'PC-CLI-1', '7', '6.1.7601', 'AMD64', 'Intel64 Family 6 Model 42 Stepping 7, GenuineIntel')
>>>

```

正常的，一行一行的，  
输入代码，才能正确执行  
所输入的代码，显示出  
对应的输入结果

#### 4.1.2.3. 关于 ( command line或GUI版本的 ) Python Shell的用途

而作为一般的Python开发的话，则很少有用这个Python Shell的，不论是command line的还是GUI版本的IDLE。

其比较适合用来测试，演示一些简单的代码的执行的效果。

好处是很方便，可以立刻看到代码执行的结果。

所以，结论就是：

对于python的shell，不论是command line版还是GUI版，都比较适合偶尔要测试少量的Python代码的情况下使用，而不适合长期的开发Python。

#### 例 4.1. 举例：用Python的IDLE去做URL解码

比如我之前就是借用Python的一些库函数，实现一些对于url解码的功能：

[【已解决】在用google搜索出来的链接无法打开的情况下，如何找到该链接的真实地址](#) <sup>23</sup>

### 4.1.3. 利用第三方Python的IDE进行Python开发又是怎么回事



#### 什么是IDE

关于IDE的基本概念，不了解的先去看：

[【整理】什么是IDE](#) <sup>24</sup>

<sup>23</sup> [http://www.crifan.com/find\\_real\\_link\\_from\\_google\\_link\\_when\\_failed\\_open\\_via\\_google/](http://www.crifan.com/find_real_link_from_google_link_when_failed_open_via_google/)

<sup>24</sup> [http://www.crifan.com/what\\_is\\_ide](http://www.crifan.com/what_is_ide)

即：

### 公式 4.1. 什么是IDE

IDE

= 集成开发环境

= 把开发相关的各种环境（和工具）都集成到一起

而Python的IDE，就是：

### 公式 4.2. 什么是Python的IDE

Python IDE

= Python的集成开发环境

= 把和Python开发相关的各种工具

- Python代码编辑器：替代你前面用的Notepad++等文本编辑器
- Python的运行环境：模拟或替代：Python的（命令行或GUI版本的，交互式）shell

集成在一起

另外再加上各种文件，代码，项目的组织，管理等方面的各种功能

以此去方便你运行对应的Python代码，

方便你进行Python项目的开发

### 4.1.3.1. 为何会有Python的IDE

而之所以会有Python的IDE的诞生，也很容易理解。

就是因为，如果开发Python过程中，写Python代码，调试Python代码，查找相关的函数的解释等等操作，

如果都是基于前面介绍的，用Notepad++等编辑器去编辑Python代码，写完代码了，再切换到windows的cmd中去运行，

往往觉得很麻烦。

尤其是大型项目的话，可能就更加显得不那么高效；

以及对应的需要一些额外的功能，比如调试复杂的Python代码，需要一点点跟踪调试，找到错误的根本原因等等。

上述的开发模式，就更显得力不从心。

所以，才会有：

Python发展到现在，已经有了很多第三方的，别人开发的，可以用于或者专门用于Python开发的一些集成开发环境，即Python的IDE。

### 4.1.3.2. 目前常见的一些Python的IDE

参考别人的一些讨论和总结：

[【python】【求助】关于python编辑器的选择](#) <sup>25</sup>

[python编辑器对比和推荐](#) <sup>26</sup>

罗列几个，相对用的比较广泛的（排名不分先后）：

- Ulipad
- PyScripter
- Wing IDE
- Eclipse + pydev插件

关于其中的一些IDE的效果，可以参考：

[【整理】各种Python的IDE\(集成开发环境\)的总结和对比](#) <sup>27</sup>

### 4.1.3.3. Python的IDE和Python代码编辑器，Windows的cmd，等的关系

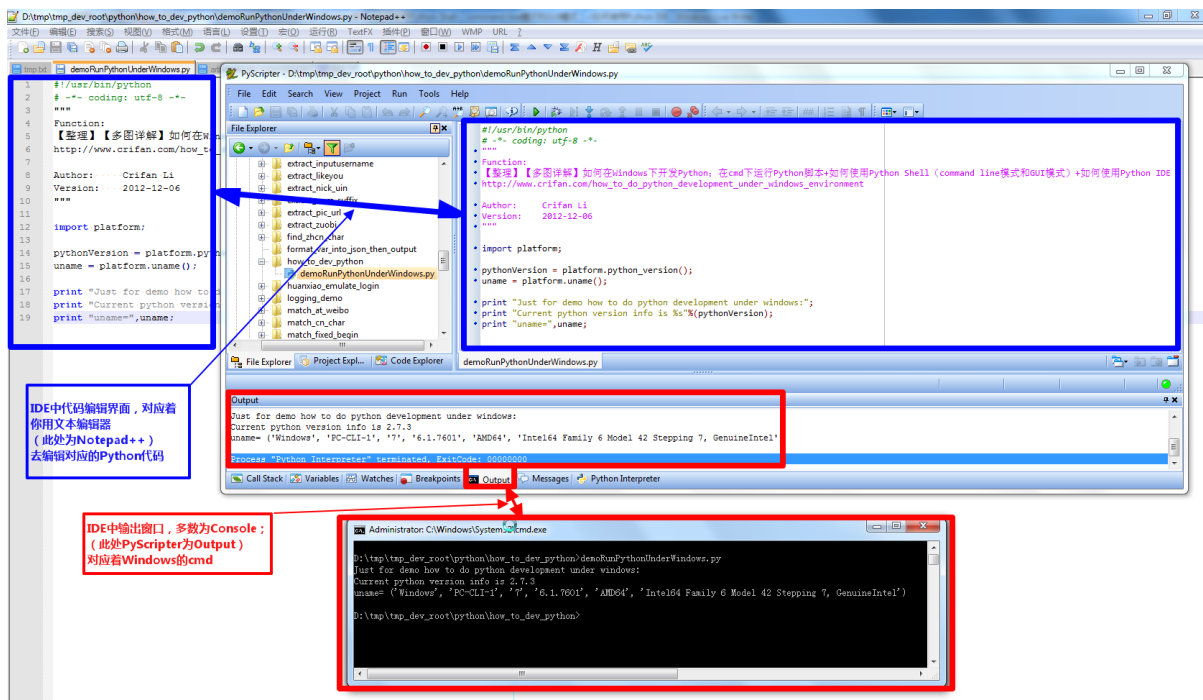
下面，随便找了一个Python的IDE，

比如

[【记录】使用Python的IDE：PyScripter](#) <sup>28</sup>

通过截图来对比性的解释，

这样你就更容易理解Python的原始开发环境和Python的IDE之间的关系了：



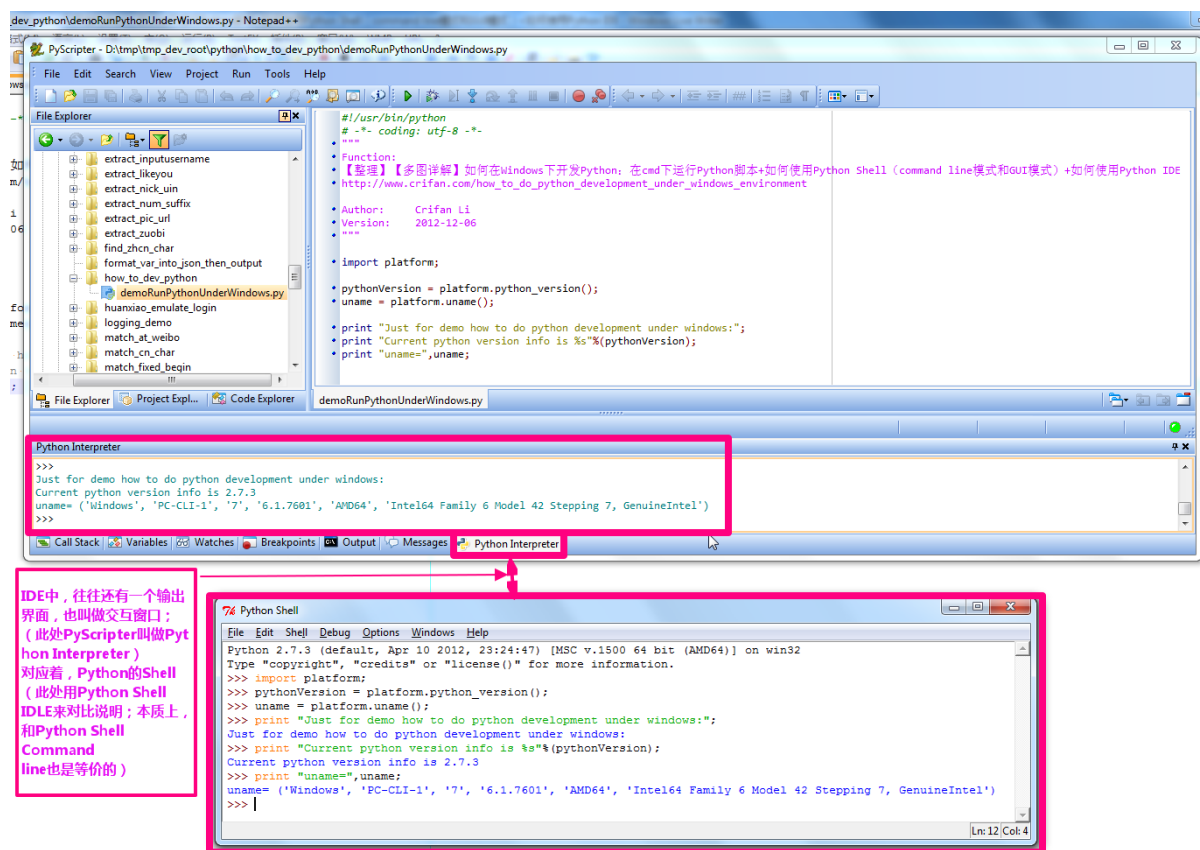
<sup>25</sup> <http://bbs.csdn.net/topics/390022660>

<sup>26</sup> <http://blog.csdn.net/cserchen/article/details/7036435>

<sup>27</sup> [http://www.crifan.com/summary\\_common\\_python\\_ide\\_pyscripter\\_ulipad\\_eclipse\\_pydev\\_eric](http://www.crifan.com/summary_common_python_ide_pyscripter_ulipad_eclipse_pydev_eric)

<sup>28</sup> [http://www.crifan.com/try\\_with\\_python\\_ide\\_pyscripter/](http://www.crifan.com/try_with_python_ide_pyscripter/)





这下，至少你应该对于：

- Windows的cmd
- Python的Shell
  - command line版本
  - GUI版本：IDLE
- Python的IDE

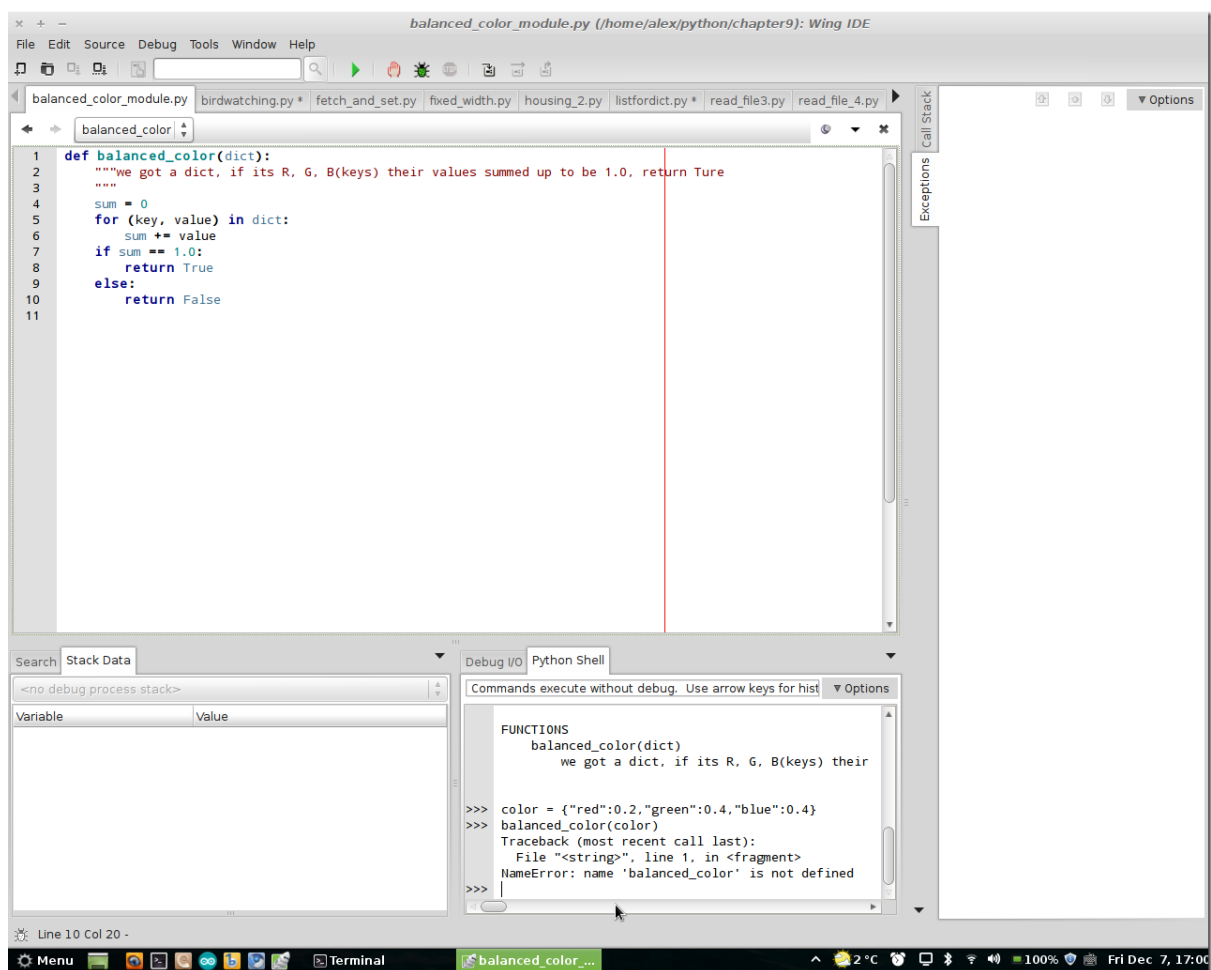
三者之间的关系，有个大概的认识了。

#### 4.1.3.4. 使用IDE时所遇到的一些常见的问题

##### 4.1.3.4.1. IDE只能够打开了文件，并不代表就已经在shell中运行了该文件

这里有一个错误的例子，比如[这里](http://zhidao.baidu.com/question/505232524.html)<sup>29</sup>所遇到的：

<sup>29</sup> <http://zhidao.baidu.com/question/505232524.html>



如图，其在当前的IDE中，打开了一个.py文件：balanced\_color\_module.py

按照正常的逻辑，应该到菜单中去运行代码，然后再去对应的console中（此处估计是Debug I/O），去查看运行的结果。

但是，很明显，对于此处的那个py文件（balanced\_color\_module.py），即使运行该文件，也不会有什么print输出信息

另外，其此处没有打开对应的console，而是打开的是Python Shell，所以从逻辑上就是：

其在Python Shell中输入的任何代码，本质上和你当前IDE中所打开的py文件，都没有半毛钱关系的。

所以导致的结果就是，虽然在Python Shell中输入了一些代码：

```
color = {"red":0.2, "green":0.4, "blue":0.4}
balanced_color(color);
```

但是结果却出错，找不到对应的balanced\_color函数。

所以，也就在常理之中，预料之内了。

因为，此时其所做的事情是：

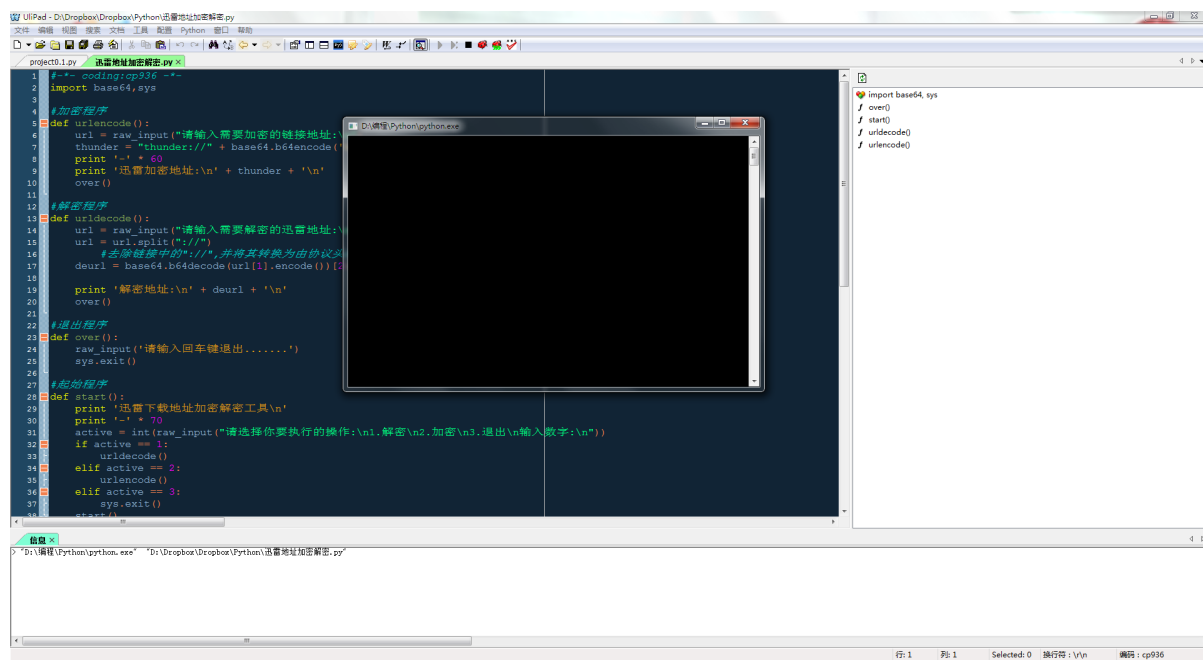
使用了IDE，但是并没有使用IDE中集成的终端调试输出

却另外打开了Python Shell，

而由于此处的IDE中打开的代码，和Python Shell没有半毛钱关系，所以才会运行代码出错，找不到对应的函数的。

#### 4.1.3.4.2. 需要注意，确保有可以运行的Python起始部分的代码

比如[这里](#)<sup>30</sup>遇到的问题，通过IDE中运行Python代码，但是没有响应：



其所遇到的问题，其实本质上，和IDE没有关系，但是不熟悉Python的人，还是容易在IDE中犯这类问题的。

错误原因是：

没有起始代码可以运行；

解决办法是：

把上面最后一行的

```
start();
```

去掉缩进，顶行写，变成：

```
start();
```

就可以了。使得Python可以执行到对应的`start`函数了。

当然，最好的写法是，把：

```
start();
```

再改为：

```
if __name__ == "__main__":
    start();
```

<sup>30</sup> <http://zhidao.baidu.com/question/505568675.html>

具体的解释，参见：

[【整理】Python中的\\_\\_name\\_\\_和\\_\\_main\\_\\_含义详解](#)<sup>31</sup>

## 4.1.4. 总结：到底使用哪种环境去开发Python

一句话：

各取所需。根据自己的需求，决定用什么开发环境。

### 4.1.4.1. 对初学者的建议：如何选用Python的开发环境

再加一句：

针对初学者，我个人倒是建议使用第一种，即windows的cmd下，去运行python脚本

目的很明确：

很多东西的学习，其本质上，都是需要一个循序渐进的过程的，学习Python语言同样如此。

在没有学会走路，即如何搞懂Windows的cmd下运行Python脚本，

就想学会跑了，即直接利用Python的IDE，包括shell和第三方开发环境，

结果就是，很多东西，还是不明白到底是为什么，理解的不透彻。

而当Python的基本知识，基本开发流程熟悉了之后，再建议你去使用第三方的Python的IDE，到时候，才能算是用着很爽。

即：

1. 先：[Windows的cmd](#)<sup>32</sup> + [Notepad++](#)<sup>33</sup>
2. 再：选用某个IDE<sup>34</sup>，比如[PyScripter](#)<sup>35</sup>，[Ulipad](#)<sup>36</sup>，[Eclipse+PyDev](#)<sup>37</sup>等。

## 4.1.5. 如何在Windows环境下使用Python脚本

首先要说明的是，据我目前的了解，对于一般Windows的用户来说，想要使用已有的Python脚本的话，主要有两种形式可以使用。

1. 直接运行文本式的Python脚本文件  
我们所常见的，多数的Python脚本，都是此形式的。而关于Python脚本，其实就是一个文本文件，你可以用任何一个文本编辑器，比如windows的Notepad.exe来打开对应的后缀名为.py的文件，比如我所发布的，用于博客搬家到wordpress的Python脚本BlogsTo Wordpress.py

而对于在Windows的命令行，即cmd下运行Python脚本，用起来，就是这个样子的：

<sup>31</sup> [http://www.crifan.com/python\\_detailed\\_explain\\_about\\_\\_name\\_\\_and\\_\\_main\\_\\_/](http://www.crifan.com/python_detailed_explain_about__name__and__main__/)

<sup>32</sup> [http://www.crifan.com/files/doc/docbook/soft\\_dev\\_basic/release/html/soft\\_dev\\_basic.html#win\\_cmd](http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#win_cmd)

<sup>33</sup> [http://www.crifan.com/files/doc/docbook/rec\\_soft\\_npp/release/html/rec\\_soft\\_npp.html](http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/html/rec_soft_npp.html)

<sup>34</sup> [http://www.crifan.com/summary\\_common\\_python\\_ide\\_pyscripter\\_ulipad\\_eclipse\\_pydev\\_eric](http://www.crifan.com/summary_common_python_ide_pyscripter_ulipad_eclipse_pydev_eric)

<sup>35</sup> [http://www.crifan.com/try\\_with\\_python\\_ide\\_pyscripter](http://www.crifan.com/try_with_python_ide_pyscripter)

<sup>36</sup> [http://www.crifan.com/try\\_with\\_python\\_ide\\_ulipad](http://www.crifan.com/try_with_python_ide_ulipad)

<sup>37</sup> [http://www.crifan.com/try\\_with\\_python\\_ide\\_eclipse\\_pydev](http://www.crifan.com/try_with_python_ide_eclipse_pydev)

图 4.1. 在Windows下的cmd下面运行Python脚本的样子

```

管理员: C:\Windows\system32\cmd.exe
E:\Dev_Root\svn_dev_root\website\python\BlogsToWordPress\BlogsToWordPress.py -s http://blog.sina.com.cn/lifecoaching
Imported: crifanLib, v1.6
Imported: BlogNetease, v1.4
Imported: BlogBaidu, v1.3
Imported: BlogSina, v1.4
Imported: BlogQQ, v1.7
Imported: BlogCsdn, v1.0
Imported: BlogSohu, v1.3
LINE 1139 : INFO 版本信息: v8.3
LINE 1140 : INFO 1.如发现bug, 请将日志文件和bug截图等信息发至: admin(at)crifan.com
LINE 1141 : INFO 2.如对此脚本使用有任何疑问, 请输入 -h 参数以获得相应的参数说明。
LINE 1142 : INFO 3.关于本程序详细的使用说明和更多相关信息, 请参考:
LINE 1144 : INFO BlogsToWordPress: 将百度空间, 网易163, 新浪Sina, QQ空间, 人人网, CSDN, 搜狐等
博客搬家到WordPress
LINE 1145 : INFO http://www.crifan.com/crifan_released_all/website/python/blogstowordpress/
LINE 191 : INFO -----
LINE 1186 : INFO Your process type of post is: Export post to WXR(WordPress eXtended Rss).
LINE 1230 : INFO Source URL: http://blog.sina.com.cn/lifecoaching
LINE 1379 : INFO Your blog provider : Sina Blog.
LINE 1086 : INFO Extracted Blog user [lifecoaching] from http://blog.sina.com.cn/lifecoaching
LINE 1087 : INFO Blog entry url is http://blog.sina.com.cn/lifecoaching
LINE 1065 : INFO Username and/or password is null, now in un-login mode.
LINE 1235 : INFO Now start to find the permanent link for http://blog.sina.com.cn/lifecoaching
LINE 1241 : ERROR Can not find the first link for http://blog.sina.com.cn/lifecoaching, error=Unk
nown error!
半:

```

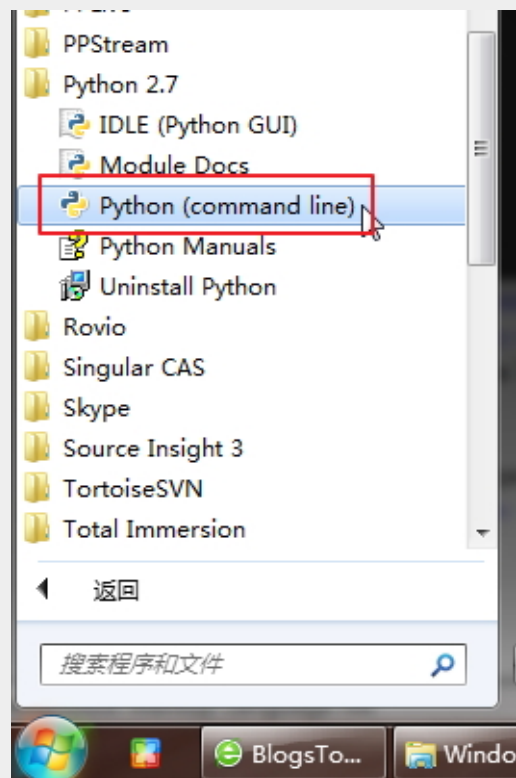


## Windows的cmd下运行Python脚本，不是在Python的Command Line工具下运行Python脚本

对于此用法不太熟悉的新手，需要注意一点的是，不要把，在Windows下的cmd中运行Python脚本，和在Python（安装好后自带的）Command Line去写Python代码，运行Python代码，相混淆了

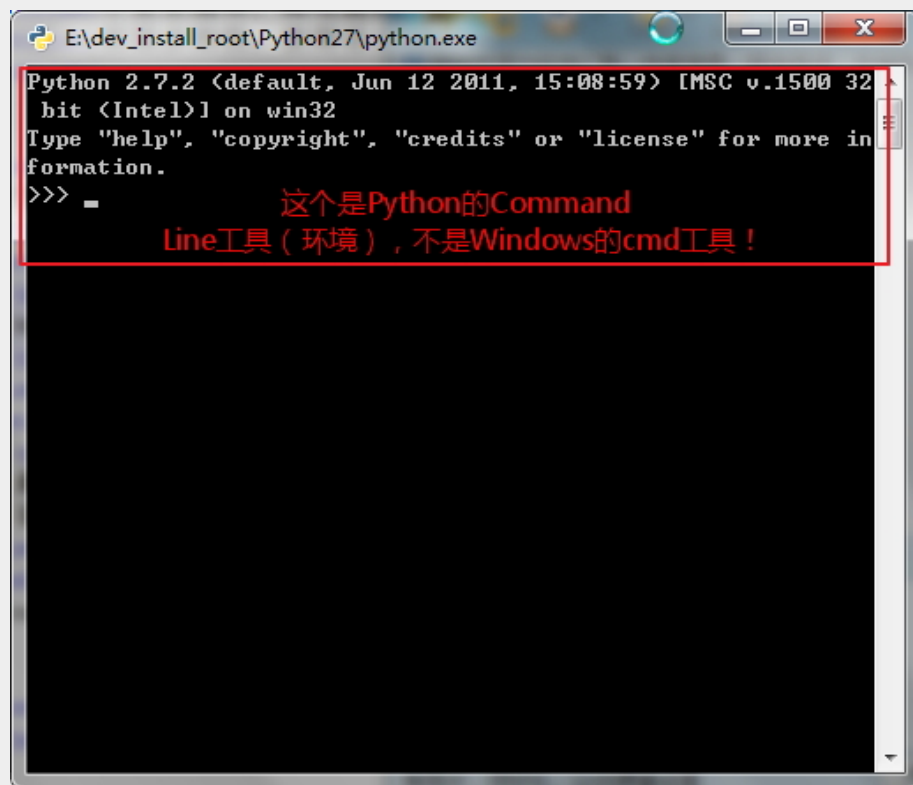
对于Python的Command Line，以我此处Python 2.7为例，是从开始菜单中找到的：开始菜单 ⇒ Python 2.7 ⇒ Python (Command Line)，如图：

图 4.2. 开始菜单中找到的Python (Command Line)



其打开后的效果如下：

图 4.3. Python (Command Line)的界面

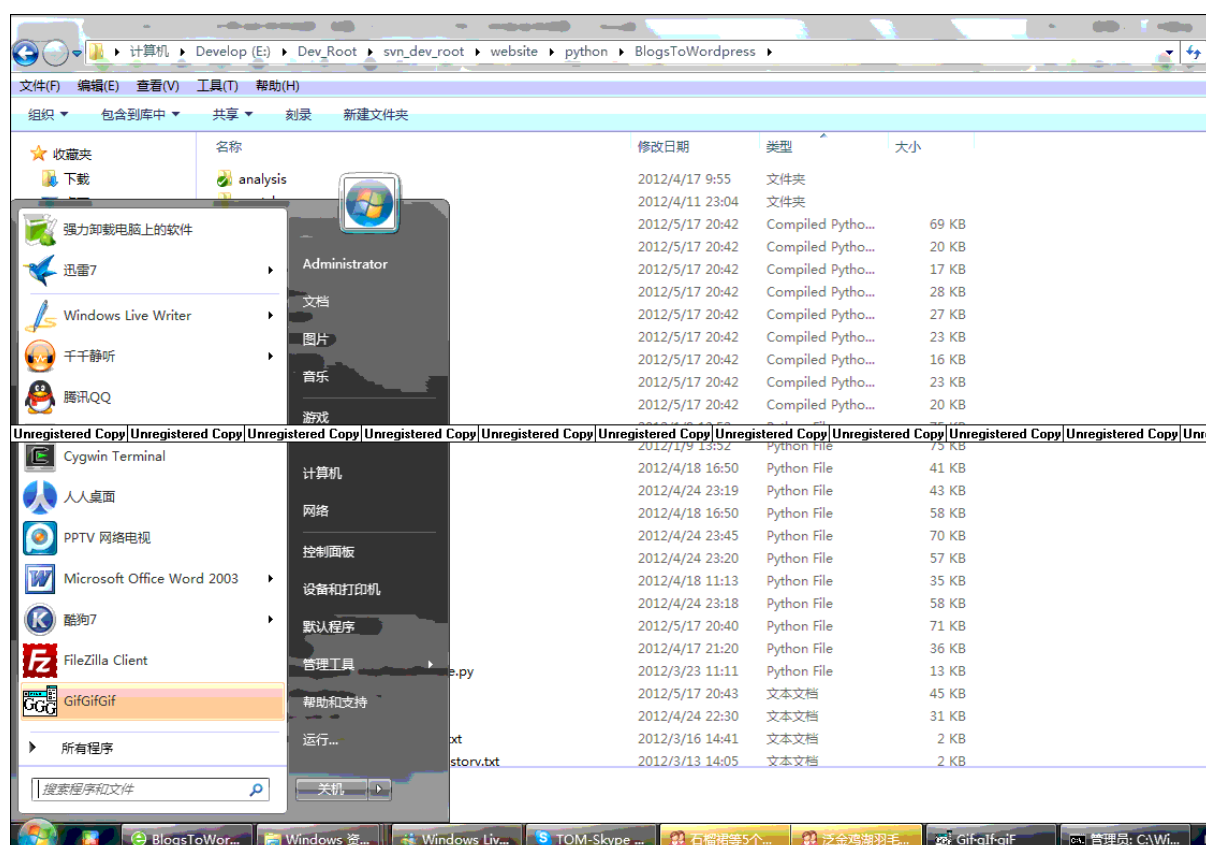


2. 使用Python脚本发布者封装好的，可以独立运行的Python脚本对应的可执行文件  
此用法，必须要Python脚本发布者，专门在写完Python脚本的时候，专门去用相应的工具，将Python的文本文件，封装成可以脱离Python运行环境，可以独立运行的可执行文件。说白了，就是一个exe可执行文件，而运行此exe文件，不需要安装Python，即可运行。

### 4.1.5.1. 如何在Windows下的cmd中运行BlogsToWordpress.py

下面，以Win7为例，通过专门录制的gif动画，来说明，如何在Windows的cmd中，运行我的一个Python脚本：BlogsToWordpress.py

图 4.4. 动画演示如何在Windows的cmd中运行Python脚本BlogsToWordpress.py



### 提示

上述gif动画演示，只支持HTML在浏览器中的显示。其他格式输出中，比如PDF中，不支持此gif动画。所以下面再用文字解释一下大概流程：[用文字解释如何在Windows的cmd中运行Python脚本BlogsToWordpress.py](#)

如果你是在非HTML格式，比如PDF等，中查看此文，想要看动画的话，猛击[动画演示如何在Windows的cmd中运行Python脚本BlogsToWordpress.py](#)<sup>38</sup>

用文字简述就是：

**用文字解释如何在Windows的cmd中运行Python脚本BlogsToWordpress.py.**

1. 打开Windows的命令行环境cmd  
开始 ⇒ 在"搜索程序和文件"的位置，输入cmd，win7会自动搜索到cmd，点击其以打开cmd



2. 切换到对应的你的Python脚本所在位置  
可以先去拷贝你的Python脚本所在的路径，此处为：

E:\Dev\_Root\svn\_dev\_root\website\python\BlogsToWordpress

，然后在cmd中输入

```
E:
cd E:\Dev_Root\svn_dev_root\website\python\BlogsToWordpress
```

3. 输入脚本名（及参数），以运行Python脚本  
输入相应的python脚本命令：

BlogsToWordpress.py -s http://blog.sina.com.cn/lifecoaching

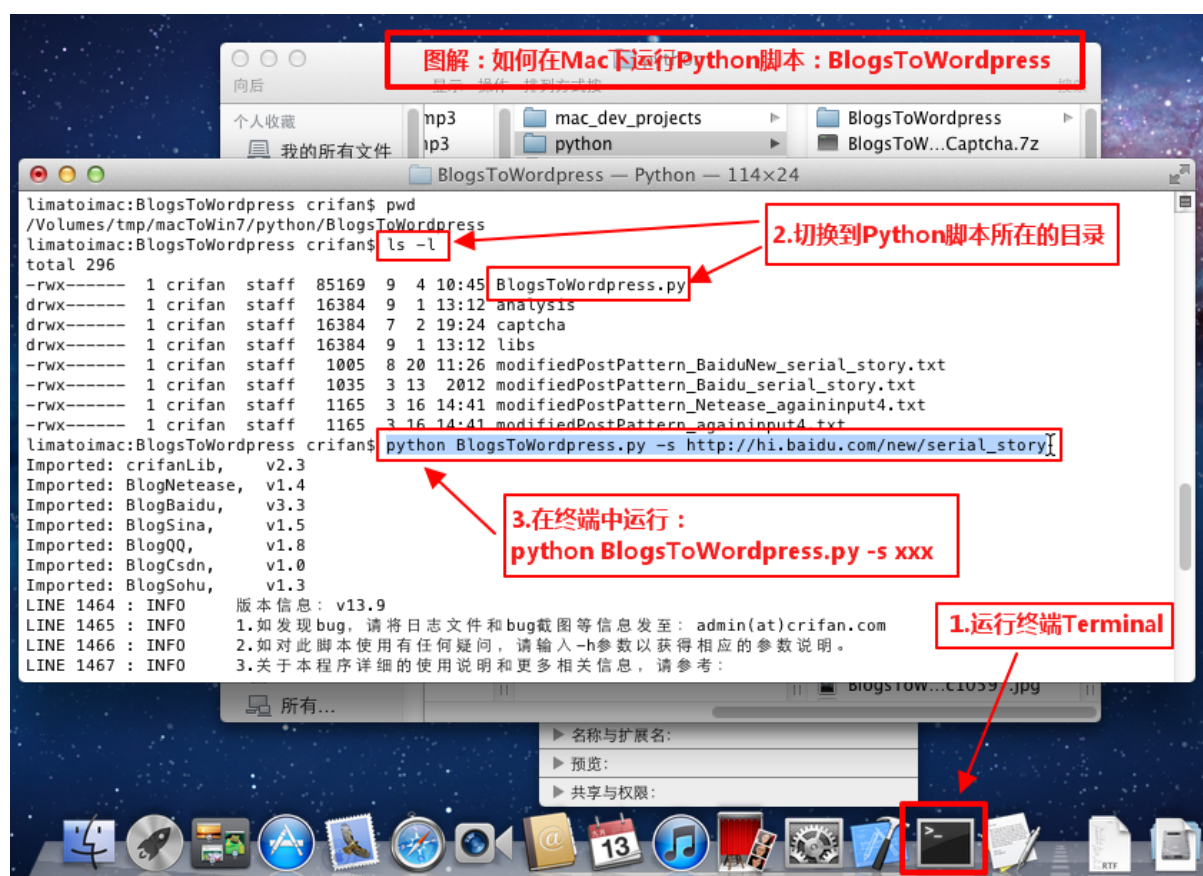
然后回车运行该脚本

## 4.2. 如何在Mac环境下运行Python脚本

Mac下本身已经安装了Python，所以不需要额外安装Python，就可以直接在终端Terminal下去运行Python了。

在Mac下运行Python脚本的方法如下：

图 4.5. 在Mac下的Terminal中运行Python脚本：BlogsToWordpress





---

# 第 5 章 Python的语法和基础知识

此处只介绍，最基本的一些Python的语法，和其他一些基础知识。



## 相关旧帖

[【教程】详解Python中代码缩进（Indent）：影响代码的内在逻辑关系和执行结果](#) <sup>1</sup>

[【整理】关于Python脚本开头两行的：#!/usr/bin/python和# -\\*- coding: utf-8 -\\*-的作用 – 指定文件编码类型](#) <sup>2</sup>

[【整理】Python中：self和\\_\\_init\\_\\_ 的含义 + 为何要有self和 \\_\\_init\\_\\_](#) <sup>3</sup>

[【整理】Python中变量的作用域\(variable scope\)](#) <sup>4</sup>

[【整理】Python中的module,library,package之间的区别](#) <sup>5</sup>

[【整理】Python中的 \\_\\_name\\_\\_ 和 \\_\\_main\\_\\_ 含义详解](#) <sup>6</sup>

---

<sup>1</sup> [http://www.crifan.com/tutorial\\_python\\_indent](http://www.crifan.com/tutorial_python_indent)

<sup>2</sup> [http://www.crifan.com/python\\_head\\_meaning\\_for\\_usr\\_bin\\_python\\_coding\\_utf-8](http://www.crifan.com/python_head_meaning_for_usr_bin_python_coding_utf-8)

<sup>3</sup> [http://www.crifan.com/summary\\_the\\_meaning\\_of\\_self\\_and\\_\\_init\\_\\_in\\_python\\_and\\_why\\_need\\_them](http://www.crifan.com/summary_the_meaning_of_self_and__init__in_python_and_why_need_them)

<sup>4</sup> [http://www.crifan.com/summary\\_python\\_variable\\_effective\\_scope](http://www.crifan.com/summary_python_variable_effective_scope)

<sup>5</sup> [http://www.crifan.com/python\\_module\\_vs\\_library\\_vs\\_package](http://www.crifan.com/python_module_vs_library_vs_package)

<sup>6</sup> [http://www.crifan.com/python\\_detailed\\_explain\\_about\\_\\_name\\_\\_and\\_\\_main\\_\\_](http://www.crifan.com/python_detailed_explain_about__name__and__main__)

---

## 第 6 章 Python的学习方法

此处只介绍，在了解了基本的Python的开发知识之后，接下来，想要进一步开发，如何去做，如何学习。

目的在于，掌握了基本的学习方法后，就可以自己去一点点学习更多细节的东西了。



### 相关旧帖

[【整理】如何学习Python + 如何有效利用Python有关的网络资源 + 如何利用Python自带手册\(Python Manual\)](#)<sup>1</sup>

---

<sup>1</sup> [http://www.crifan.com/howto\\_learn\\_python\\_utilize\\_network\\_resource\\_manual](http://www.crifan.com/howto_learn_python_utilize_network_resource_manual)

---

## 第 7 章 Python的相关学习资料

此处记录一些不错的Python相关的资料，供需要时参考：

1. [简明Python教程](#)<sup>1</sup>

Swaroop, C. H. 著，>沈洁元 译

特点：内容很全。

2. [关于python的中文文档](#)

这里：<http://www.elias.cn/Python/HomePage>介绍了很多的资源。其中就有[python教程的中文版](#)<sup>2</sup>

---

<sup>1</sup> [http://woodpecker.org.cn/abyteofpython\\_cn/chinese/](http://woodpecker.org.cn/abyteofpython_cn/chinese/)

<sup>2</sup> [http://wiki.woodpecker.org.cn/moin/March\\_Liu/PyTutorial](http://wiki.woodpecker.org.cn/moin/March_Liu/PyTutorial)

---

# 参考书目

- [1] [Python基础篇](#)<sup>1</sup>
- [2] [python编辑器对比和推荐](#)<sup>2</sup>
- [3] [【整理】【多图详解】如何在Windows下开发Python：在cmd下运行Python脚本，如何使用Python Shell（command line模式和GUI模式），如何使用Python IDE](#)<sup>3</sup>
- [4] [【整理】计算机语言基础知识介绍](#)<sup>4</sup>
- [5] [【整理】Python语言简介](#)<sup>5</sup>
- [6] [【整理】总结Python2\(Python 2.x版本\)和Python3\(Python 3.x版本\)之间的区别](#)<sup>6</sup>
- [7] [【crifan推荐】轻量级文本编辑器，Notepad最佳替代品：Notepad++](#)<sup>7</sup>
- [8] [Windows的命令行工具: cmd](#)<sup>8</sup>
- [9] [【整理】Python中的图形库](#)<sup>9</sup>
- [10] [【已解决】在用google搜索出来的链接无法打开的情况下，如何找到该链接的真实地址](#)<sup>10</sup>
- [11] [【整理】什么是IDE](#)<sup>11</sup>
- [12] [【整理】各种Python的IDE\(集成开发环境\)的总结和对比](#)<sup>12</sup>
- [13] [【记录】折腾Python中的Tkinter](#)<sup>13</sup>
- [14] [【记录】使用Python的IDE：PyScripter](#)<sup>14</sup>
- [15] [【整理】Python中的 name 和 main 含义详解](#)<sup>15</sup>
- [16] [Eclipse+PyDev](#)<sup>16</sup>

---

<sup>1</sup> [http://www.tsnc.edu.cn/default/tsnc\\_wgrj/doc/python/basic.htm](http://www.tsnc.edu.cn/default/tsnc_wgrj/doc/python/basic.htm)

<sup>2</sup> <http://blog.csdn.net/cserchen/article/details/7036435>

<sup>3</sup> [http://www.crifan.com/how\\_to\\_do\\_python\\_development\\_under\\_windows\\_environment](http://www.crifan.com/how_to_do_python_development_under_windows_environment)

<sup>4</sup> [http://www.crifan.com/computer\\_language\\_basic\\_knowledge\\_introduction](http://www.crifan.com/computer_language_basic_knowledge_introduction)

<sup>5</sup> [http://www.crifan.com/simple\\_intro\\_what\\_is\\_python\\_and\\_how\\_to\\_run\\_python\\_script](http://www.crifan.com/simple_intro_what_is_python_and_how_to_run_python_script)

<sup>6</sup> [http://www.crifan.com/summary\\_the\\_difference\\_between\\_python2\\_and\\_python3](http://www.crifan.com/summary_the_difference_between_python2_and_python3)

<sup>7</sup> [http://www.crifan.com/files/doc/docbook/rec\\_soft\\_npp/release/html/rec\\_soft\\_npp.html](http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/html/rec_soft_npp.html)

<sup>8</sup> [http://www.crifan.com/files/doc/docbook/soft\\_dev\\_basic/release/html/soft\\_dev\\_basic.html#win\\_cmd](http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#win_cmd)

<sup>9</sup> [http://www.crifan.com/summary\\_python\\_graphics\\_gui\\_libs\\_packages/](http://www.crifan.com/summary_python_graphics_gui_libs_packages/)

<sup>10</sup> [http://www.crifan.com/find\\_real\\_link\\_from\\_google\\_link\\_when\\_failed\\_open\\_via\\_google/](http://www.crifan.com/find_real_link_from_google_link_when_failed_open_via_google/)

<sup>11</sup> [http://www.crifan.com/what\\_is\\_ide](http://www.crifan.com/what_is_ide)

<sup>12</sup> [http://www.crifan.com/summary\\_common\\_python\\_ide\\_pyscripter\\_ulipad\\_eclipse\\_pydev\\_eric](http://www.crifan.com/summary_common_python_ide_pyscripter_ulipad_eclipse_pydev_eric)

<sup>13</sup> [http://www.crifan.com/try\\_python\\_tkinter\\_module/](http://www.crifan.com/try_python_tkinter_module/)

<sup>14</sup> [http://www.crifan.com/try\\_with\\_python\\_ide\\_pyscripter/](http://www.crifan.com/try_with_python_ide_pyscripter/)

<sup>15</sup> [http://www.crifan.com/python\\_detailed\\_explain\\_about\\_\\_name\\_\\_and\\_\\_main\\_\\_/](http://www.crifan.com/python_detailed_explain_about__name__and__main__/)

<sup>16</sup> [http://www.crifan.com/try\\_with\\_python\\_ide\\_eclipse\\_pydev](http://www.crifan.com/try_with_python_ide_eclipse_pydev)