Python Tutorial

Created by PhD. Candidate Yuyu Qian

Zhejiang University Hangzhou, China

*You may not use this for any purpose or pass over this without permission. For more information please contact*: [qianyuyulys@gmail.com](mailto:qianyuyulys@gmail.com)

这是有关于Python处理.csv文件的教程。

**介绍**

首先，Python是一个脚本语言，需要运行在python环境下，但是不需要编译，程序会通过解释器直接在系统中执行。而且是一种动态语言，十分方便修改。但是也因为这两个特性，分别造成了python在运行效率上和大规模代码管理上的缺陷，当然，你暂时不需要关心这些性能上的缺点。

**如何构建Python环境**

当然，可以直接去Python官网下载.exe安装，但是我建议你直接下载Anaconda。

理由是，Anaconda是一个Python的版本，包和库管理软件。使用Anaconda，你可以在你的电脑上同时拥有多个版本的Python环境，并且自如切换。

举个例子，首先去官网下载安装Anaconda（无所谓版本，但是建议直接安装最新版，安装过程中请不要修给最后的那些选项，尤其是Path选项，可能会导致奇怪的错误。）

然后在windows栏中搜索



因为如果你没有勾选那个Path选项，只能在这里才能使用Conda指令。

接下来是安装你的Python环境，在你安装完Anaconda以后，你已经在系统里获得了一个Python环境，但是如果你想获得不同的环境，或者你想方便管理你的环境，请按照下面步骤过一遍。

>>conda create -n thePythonEnvironmentName python=3.5 (取一个你喜欢名字，版本可以不是3.5可以是3.6/2.7或者任意你想要的版本)

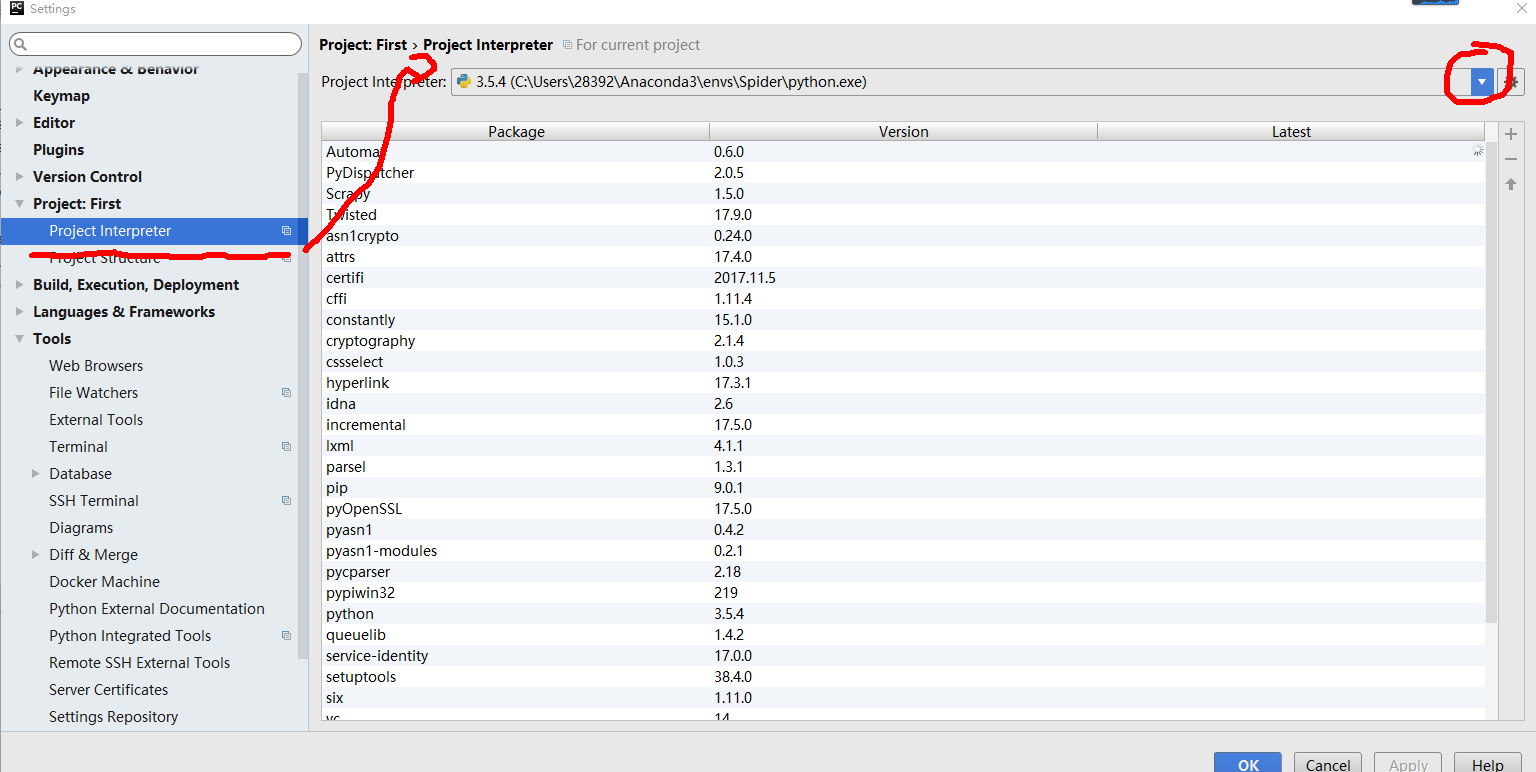
>>activate thePythonEnvironmentName (这会让你接下来的所有操作是在这个环境中的)

(thePythonEnvironmentName)>>pip install PackageName (这个命令可以让你在该环境中安装名字为PackageName的包)

然后你就可以在C:/User/你的用户名字/Anaconda3/envs/下看到你明明的环境名称

搞定了环境，你就需要一个IDE来帮忙了，IDE全称整合开发环境，也就说这个包含了你开发所需要的一切。PyCharm是一款十分优秀的IDE，下载安装，可以使用edu版本，免费哦。

然后你需要做的就是在PyCharm里



在设置中设定好Interpreter，然后新建工程，新建Python文件，编写代码就可以运行Python了！

Python 处理.csv文件

**注释**

在python的语法中，#开头之后的语句都表示注释，同时用``` ```或者`````` ``````包括起来的多行也是注释

#-\*- coding:utf-8 -\*-

这句话看起来是注释，但是一般放在文件最前面，其实是申明这个python文件使用的是utf-8字符的格式，一般来说，如果不加入这一句，只能识别英文字符，而加入这一行，可以识别utf-8字符格式中的所有字符，包括中文。（所以如果只使用英文可以忽略，但是如果有中文，或者为了确保以后万一使用其它非英文字符，最好加上这一行）

**Import**

接着是import语句，import语句顾名思义就是导入相关的python包，当然也可以从你自己写的python文件中导入相关的函数调用哦。而导入的包可以是python环境中自带的默认包，例如os,csv等，也可以是之前我叙述的conda命令中用户自己安装的包哦！

接着，就是定义系列函数，作为程序的主要入口。但是在介绍怎么编写这些函数实现数据处理功能之前，我们先跳转到程序的最后：

**程序入口以及import的关系**

if \_\_name\_\_ == '\_\_main\_\_':

这一行语句才是这个python文件在执行时的真正入口。此语句的意思是，如果\_\_name\_\_这个系统预留变量为”\_\_main\_\_”的话，换句话说如果是文件自己单独运行的话，那么执行这个if语句中的语句。

于是解释器会调用在这个if语句中包含的，在这条if语句之前的定义的函数或者一些其它语句。

你可能会问，为什么要这样做。因为在工程中，你编写的函数和功能常常会进行复用，为了避免重复劳作，又便于管理，你可以将一系列相关功能定义在一个python文件中；同时，为了是的python文件自己又具有单独进行一些子任务的能力，才设置这样一个接口。

所以，如果你确定你的程序不需要复用，或者不想这么麻烦，那么你在python文件中直接编写语句也是可以运行的。可以不使用这个接口。换句话说，你去掉这个if语句，仅仅保留这之后的语句，这个程序也能运行。

好了，接下来是真正讲解数据处理的部分内容了。

**空格还是tab，这是一个问题**

Python与之前你接触过的c语言不同，它不是依靠特殊预留字符在进行段落分隔的，虽然也会用到括号来进行范围申明，但是大多数情况下，你接触最多的就是缩进！缩进就是python用来识别层级关系的标志，也就是说，程序员在编写python的过程中缩进不是像c语言那样为了美观，而是必须如此。c语言程序可以去掉中间的缩进依然可以运行，但是python不可以。当然，这样天然的缩进风格其实提高了编写程序的效率，也增加了可读性。所以有利有弊，但我认为利大于弊。

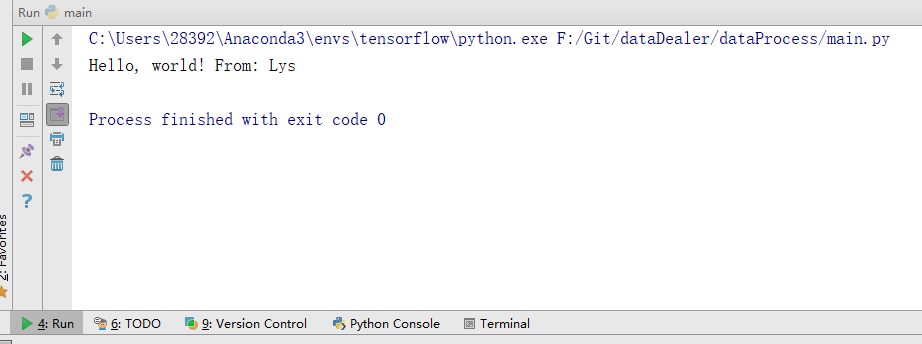
在python中，如果进行缩进呢？其实有多种方法，为了方便，我建议你直接使用tab，当然也有用两个空格来缩进的。不过解释器都是可以识别的，但只能是使用统一的一种。建议用tab，这样更加方便。

**让我们遇见世界**

如同学习任何一门语言的入门教程一样，我在教你任何东西之前，先带给你最快乐的事情，就是带你走进新世界的大门。那么，请和我一起来创造一个HelloWorld程序作为开胃小菜吧。

**def** helloWorld(yourname):  
 print(**"Hello, world! From:"**, yourname)  
  
helloWorld(**'Lys'**)

**如果你在python文件中写入这样三行代码，那么就应该可以运行出一个像下面这样的结果**



**让我来解释一下这之中发生的故事。首先是你定义了一个名字叫做helloWorld的函数，这样的命名的方式比较常见，叫做驼峰命名法则，也就是说，在函数名或者变量名中有多个单词，那么除了第一个单词之外的每一个单词首字母都大写。当然，你也可以用\_来连接这些单词，都是可以的。不能用数字作为开头！**

**接着，你让这个函数能够接受yourname这个参数。这是一个形参，所以在helloWorld函数内部，你可以使用这个yourname参数。**

**然后，在函数内部（所以你需要缩进一次）,你使用print这个预定义函数，print函数接受了一系列参数，分别是字符串"Hello, world! From:" 以及形参yourname，用逗号分隔。**

**最后，你调用这个helloWorld函数并传入参数字符串'Lys'。所以你根据结果也自然明白了这个print函数的使用过程，没错，就是将一系列你需要打印的内容用逗号分割呈现。这里提一句，用逗号分割会自动添加一个空格，并且在每一个print函数输出内容的最后会自动添加一个换行符，当然这些都是可以更改的，你有兴趣可以尝试一下，可以满足许多不同的输出要求。**

**单眼皮和双眼皮**

**这里有一个小细节，你可能会注意到。就是两次的字符串的表示形式是不同的，一次是用双引号，一次是单引号。其实两者在python中都是可以的，但是一般约定，需要输出让用户看见的内容用双引号，否则用单引号，你可以用你喜欢的。**

**让我们一展身手吧**

**好了，我相信你已经迫不及待跃跃欲试了，但是我要提醒你的是，在看似简单的一切背后，其实都有着复杂的原理，如果你像我一样有好奇心，那么，在课余一定要多思考，多探索。记住，Google和StackOverflow可以解答你的几乎所有疑惑。**

**Ok，在这个有针对的教程里，我会带领你领略python对于处理一些繁琐的事情的强大能力。通过对csv文件进行操作，我相信，你通过这个教程能学到很多知识，同时也希望你能够将这个知识和思维继续应用到下一个你觉得棘手的事情中去。**

**首先，我们需要导入两个额外的默认包:**

**import** csv  
**import** os

**接着，定义一个函数，我们暂且命名为main()好了，当然可以是其它。**

**那么接下来我们要思考一下处理csv文件的步骤了。我们先将我们的目的理清。我们需要将一系列有序排列且命名相关的每固定个数个csv文件中特定一些行对应求均值并且返回csv结果文件。**

**为了便于你理解，我准备将分解进行，首先我们来进行对一个单独的csv文件的操作。**

**在main()函数中输入**