目录

前言 1.1
背景 1.2
电脑基础知识 1.3
文件后缀名 1.3.1
路径中包含空格 1.3.2
编程语言基础问题 1.4
变量名非法 1.4.1
中文字符导致出错 1.4.1.1
测试文件名和库同名 1.4.1.2
不要故意用错误语法 1.4.2
属性报错AttributeError 1.4.3
大小写不同 1.4.3.1
上下代码逻辑问题 1.4.3.2
函数参数 1.4.4
代码注释 1.4.5
Python常见问题 1.5
input返回值是字符串 1.5.1
Python的shell不是系统终端 1.5.2
Python的shell和系统的终端 1.5.2.1
字符编码问题 1.5.3
文件顶部注释 1.5.4
文件打开问题 1.5.5
代码缩进问题 1.5.6
如何学会提问 1.6
把问题要素描述清楚 1.6.1
学会贴图和贴代码 1.6.2
csdn的bbs论坛 1.6.2.1
附录 1.7
print字符串格式化语法 1.7.1
参考资料 1.7.2

Python新手小白常见错误和问题

- 最新版本：`v1.9`
- 更新时间：`20210430`

简介

整理Python新手和小白用户在开发期间，常见的一些问题以及现象背后的根源，从而搞清楚最终的解决办法和思路，并附上引申出的举一反三的思考内容，避免其再犯类似错误。主要内容有通用的编程语言基础方面的问题，包括路径中带路径的问题，以及由于中文字符、测试文件名和库同名等原因导致的变量名非法问题，故意写成了错误的语法，由于大小写不同、上下代码逻辑异常导致的属性报错问题，以及函数参数弄错了和代码注释方面的常见问题；并且针对于最常见的Python语言，总结了典型的各种问题，包括input问题、不清楚系统终端和Python的shell以及IDLE之间的区别和联系、字符编码、文件顶部的注释、文件打开、代码缩进等等；最后手把手教大家如何学会提问题，给出问题的基本背景信息，最好加上必要的格式化后的带语法高亮的彩色的代码以及相关的贴图，并以csdn为例详细解释如何操作。最后附录加上了Python的一些常用内容，比如print的字符串格式化的语法。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

Gitbook源码

- crifan/python_newbie_mistakes_questions: Python新手小白常见错误和问题

如何使用此Gitbook源码去生成发布为电子书

详见：crifan/gitbook_template: demo how to use crifan gitbook template and demo

在线浏览

- Python新手小白常见错误和问题 book.crifan.com
- Python新手小白常见错误和问题 crifan.github.io

离线下载阅读

- Python新手小白常见错误和问题 PDF
- Python新手小白常见错误和问题 ePub
- Python新手小白常见错误和问题 Mobi

版权说明

此电子书教程的全部内容，如无特别说明，均为本人原创和整理。其中部分内容参考自网络，均已备注了出处。如有发现侵犯您版权，请通过邮箱联系我 `admin 艾特 crifan.com` ，我会尽快删除。谢谢合作。

鸣谢

感谢我的老婆**陈雪**的包容理解和悉心照料，才使得我 `crifan` 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

更多其他电子书

本人 `crifan` 还写了其他 `100+` 本电子书教程，感兴趣可移步至：

crifan/crifan_ebook_readme: Crifan的电子书的使用说明

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2021-04-30 17:14:33

背景

为何要写此教程

之前看到很多Python的新手，可以算是小白，在学习期间遇到各种问题和很多坑，但是往往没有太多人解释的足够清楚错误的：

- 是什么
- 为什么
- 引申
  - 其他相关（要学的）知识
  - 如何举一反三

导致

- 无法彻底理解问题
  - 也就无法真正解决问题
    - 往往以后还会再犯同样问题或类似问题

此文就尝试整理小白新手常犯的错误，并试图把问题解释透彻，帮助其扫清学习障碍。

常见问题分类

新手和小白用户常犯错误中，根据层次和类型分，可以分为：

- 最底层的：编程语言
  - 中的
    - 基础问题
- 中间的：Python语言
  - 本身常见的一些问题

以及，在学习技术期间要

- 学会如何提问题
  - 否则别人看不懂你的问题
    - 也就没法帮你更好的解决问题
      - 此处给出具体实例去示范，如何使用各种国内外的网站和系统，去提问题。

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:25:54

电脑基础知识

此处整理新手小白用户常犯的一些，和电脑基础知识有关的错误。

以及完整的解释相关的电脑基础知识，供参考。

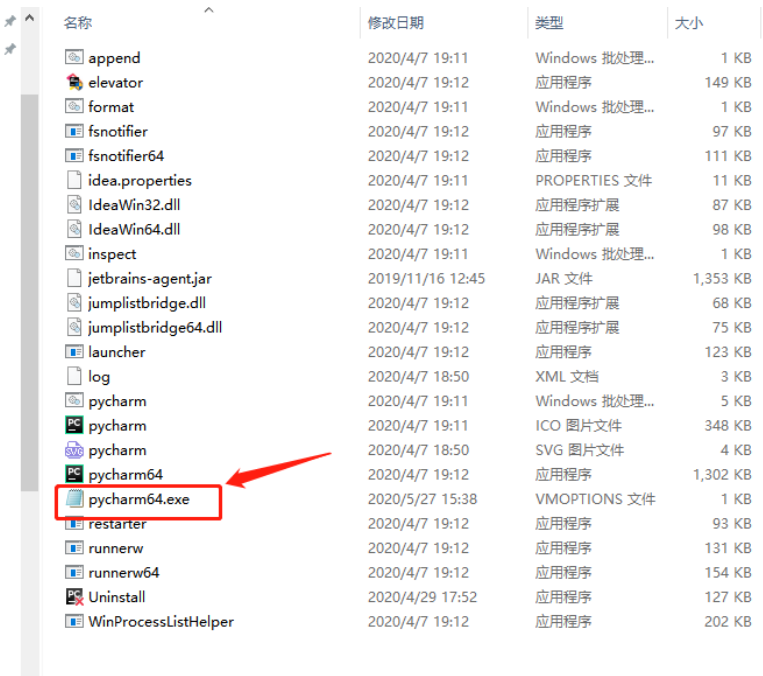
crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-08-06 20:44:27

文件后缀名

举例：缺少文件pycharm64.exe.vmoptions

问题

安装Pycpycharm到破解时bin文件夹只有一个pycharm64.exe.vmoptions怎么办 - CSDN论坛

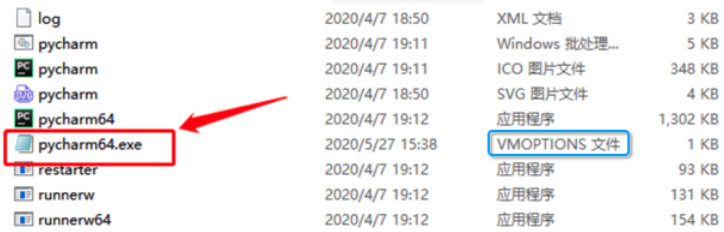


名称	修改日期	类型	大小
append	2020/4/7 19:11	Windows 批处理...	1 KB
elevator	2020/4/7 19:12	应用程序	149 KB
format	2020/4/7 19:11	Windows 批处理...	1 KB
fsnotifier	2020/4/7 19:12	应用程序	97 KB
fsnotifier64	2020/4/7 19:12	应用程序	111 KB
idea.properties	2020/4/7 19:11	PROPERTIES 文件	11 KB
IdeaWin32.dll	2020/4/7 19:12	应用程序扩展	87 KB
IdeaWin64.dll	2020/4/7 19:12	应用程序扩展	98 KB
inspect	2020/4/7 19:11	Windows 批处理...	1 KB
jetbrains-agent.jar	2019/11/16 12:45	JAR 文件	1,353 KB
jumplistbridge.dll	2020/4/7 19:12	应用程序扩展	68 KB
jumplistbridge64.dll	2020/4/7 19:12	应用程序扩展	75 KB
launcher	2020/4/7 19:12	应用程序	123 KB
log	2020/4/7 18:50	XML 文档	3 KB
pycharm	2020/4/7 19:11	Windows 批处理...	5 KB
pycharm	2020/4/7 19:11	ICO 图片文件	348 KB
pycharm	2020/4/7 18:50	SVG 图片文件	4 KB
pycharm64	2020/4/7 19:12	应用程序	1,302 KB
pycharm64.exe	2020/5/27 15:38	VMOPTIONS 文件	1 KB
restarter	2020/4/7 19:12	应用程序	93 KB
runnerw	2020/4/7 19:12	应用程序	131 KB
runnerw64	2020/4/7 19:12	应用程序	154 KB
Uninstall	2020/4/29 17:52	应用程序	127 KB
WinProcessListHelper	2020/4/7 19:12	应用程序	202 KB

为何少了pycharm64.exe.vmoptions文件?

解答

如箭头所示的红色框内文件

- 新手以为：
  - 只看到文件（文本文件的图标的）： `pycharm64.exe`
    - 缺少了（想要的）文件： `pycharm64.exe.vmoptions`
- 实际上是：
  - 其红色框标注的文件本身就是： `pycharm64.exe.vmoptions`
    - 而**不是** `pycharm64.exe`
  - 因为其后面的**类型**写的清清楚楚是： `VMOPTIONS文件`
    - 

log	2020/4/7 18:50	XML 文档	3 KB
pycharm	2020/4/7 19:11	Windows 批处理...	5 KB
pycharm	2020/4/7 19:11	ICO 图片文件	348 KB
pycharm	2020/4/7 18:50	SVG 图片文件	4 KB
pycharm64	2020/4/7 19:12	应用程序	1,302 KB
pycharm64.exe	2020/5/27 15:38	VMOPTIONS 文件	1 KB
restarter	2020/4/7 19:12	应用程序	93 KB
runnerw	2020/4/7 19:12	应用程序	131 KB
runnerw64	2020/4/7 19:12	应用程序	154 KB

      - 表示文件后缀名是： `.vmoptions`

        - 所以**完整的文件名**是： `pycharm64.exe.vmoptions`

引申

所以如果去Windows中设置 `显示文件名后缀` 后，就可以看到完整的文件名 `pycharm64.exe.vmoptions` 了

且其他的文件，也都可以看到完整的文件名了，比如：

- 类型是 `应用程序` 的 `pycharm64`
    - -> `pycharm64.exe`
- 类型是 `Windows批处理` 的 `pycharm`
    - -> `pycharm.bat`
        - 或 `pycharm.cmd` ？
- 类型是 `ICO图片文件` 的 `pycharm`
    - -> `pycharm.ico`
        - 是图标类型文件
- 类型是 `应用程序` 的 `restarter`
    - -> `restarter.exe`

关于如何显示文件的后缀名：

Win10中， `资源管理器` -> `查看` ->勾选 或 取消勾选 `文件扩展名` ，即可。

更多细节详见：显示文件后缀名 · 计算机电脑知识总结

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-08-07 23:02:43

路径中包含空格

很多初学者在使用Python期间，尤其是 `Windows` 环境下，常会遇到：

给命令行或代码参数中传递路径时，**路径中包含了空格**

其不知道路径中的空格，会导致实际上传递的参数，已经被空格分开为多个部分，因而出现找不到子路径等异常情况。

举例：

pyinstaller打包时路径带空格导致异常

python中pyinstaller打包时出现的问题，跪求大神帮助-CSDN论坛

某人用 `PyInstaller` 去打包python程序，用命令：

```
C:\Users\Administrator>pyinstaller -F D:\python VIP\chap16\stusystem
```

结果出错：

```
39 INFO: PyInstaller: 4.1
39 INFO: Python: 3.9.0
39 INFO: Platform: Windows-10-10.0.18362-SP0
40 INFO: wrote C:\Users\Administrator\python.spec
41 INFO: UPX is not available.
script 'C:\Users\Administrator\VIP\chap16\stusystem' not found
```

其中很明显就是：

`-F` 参数所传入的路径 `D:\python VIP\chap16\stusystem` 中间有空格

导致实际结果相当于：

```
C:\Users\Administrator>pyinstaller -F D:\python
```

而此处很明显Windows中只存在目录 `D:\python VIP` ，而（估计）不存在 `D:\python`

所以导致最后报错找不到相关目录：

```
script 'C:\Users\Administrator\VIP\chap16\stusystem' not found
```

根本原因：

各种系统（Windows、Linux、Mac等）中的路径，往往是通过空格去区分参数的

```
your_command parameter1 parameter2
```

不论是：

- 命令行环境

- 代码运行环境

中，所以，如果路径中有空格，往往会导致路径被空格区分开，变成多个参数，导致传入的路径本身不对，且后续其他参数也不正常了，导致结果异常

对于此处的 `pyinstaller` 的命令行参数语法是：

```
~ ◻ pyinstaller --help
usage: pyinstaller [-h] [-v] [-D] [-F] [--specpath DIR] [-n NAME]
                   [--add-data <SRC;DEST or SRC:DEST>]
                   [--add-binary <SRC;DEST or SRC:DEST>] [-p DIR]
                   [--hidden-import MODULENAME]
                   [--additional-hooks-dir HOOKSPATH]
                   [--runtime-hook RUNTIME_HOOKS] [--exclude-module EXCLUDES]
                   [--key KEY] [-d {all,imports,bootloader,noarchive}] [-s]
                   [--noupx] [--upx-exclude FILE] [-c] [-w]
                   [-i <FILE.ico or FILE.exe,ID or FILE.icns>]
                   [--version-file FILE] [-m <FILE or XML>] [-r RESOURCE]
                   [--uac-admin] [--uac-uiaccess] [--win-private-assemblies]
                   [--win-no-prefer-redirects]
                   [--osx-bundle-identifier BUNDLE_IDENTIFIER]
                   [--runtime-tmpdir PATH] [--bootloader-ignore-signals]
                   [--distpath DIR] [--workpath WORKPATH] [-y]
                   [--upx-dir UPX_DIR] [-a] [--clean] [--log-level LEVEL]
                   scriptname [scriptname ...]
...
```

此处如果输入：

```
pyinstaller -F D:\python VIP\chap16\stusystem
```

其实变成了：

- 参数1： `-F D:\python`
- 参数2： `VIP\chap16\stusystem`

对应着：

- `-F` 参数的值是： `D:\python`
- `scriptname` 参数的值是： `VIP\chap16\stusystem`

很明显，不是我们希望的结果了，就会导致异常报错了。

解决办法：尤其是命令行操作时，或者代码调用传入的路径时，要确保传入的路径中不能包含空格

如果路径中包含空格，则可以**用（双）引号括起来**：

```
pyinstaller -F "D:\python VIP\chap16\stusystem"
```

这样就是我们希望的效果了：

- 参数1： `-F "D:\python VIP\chap16\stusystem"`

即：

- `-F` 参数的值是： `D:\python VIP\chap16\stusystem`

即可正常运行。

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-12-23 21:02:42

编程语言基础问题

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:28:25

变量名非法

之前看到很多小白程序员，包括Python的小白，在写（编程语言的）代码时，把别人教程中英文字符，误写成中文的各种标点符号，导致代码报错。

关于新手常见问题中，关于变量的，有不少是：变量名是非法的，主要有以下几类：

- 不能用中文标点符号作为代码本身
- 路径中尽量不要包含空格
- 路径中尽量不要用中文（非 `ASCII` ）

下面通过具体例子来解释。

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:26:59

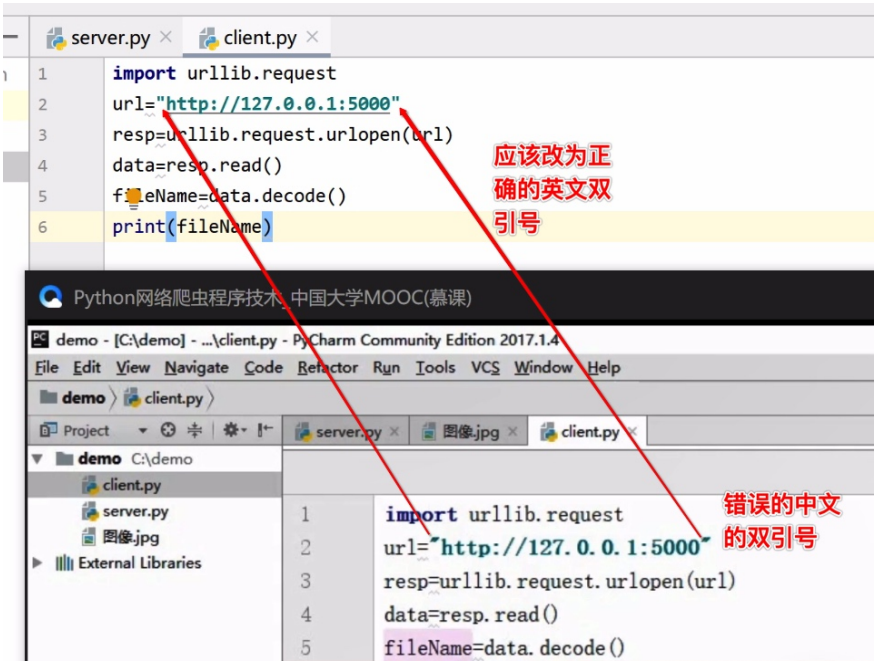
中文字符导致出错

很多初学者，不懂，或者不小心笔误，把中文的一些字符，比如：

- 逗号：`，`
- 引号
  - 单引号
    - 左单引号：`‘`
    - 右单引号：`’`
  - 双引号
    - 左双引号：`“`
    - 右双引号”：`”`
- 括号
  - 圆括号
    - 左圆括号：`（`
    - 右圆括号：`）`

不小心写到了代码里，变成了代码的一部分，导致语法上报错，导致代码无法运行。

截图举例：



对此问题，在此专门解释：

先说：标识符

编程语言，包括Python，中的各种，写在代码里，作为代码和表达式，中的各种符号，都必须是英文的。

否则你的编程语言的编译器，是无法识别，无法解析的，会报错的：

- 代码静态检测时报错：`unexpected token ‘xxx’`
- 代码运行会报错：`SyntaxError invalid character in identifier`

下面来举例说明：

常见的Python代码中（其实其他语言也是）非法的中文的标点符号 以及对应的英文标点符号是：

写在代码中，作为标点符号时：无效的中文字符	错误写法举例	应该改为：正确的英文字符	正确写法举例
单引号： • 左单引号：`‘` • 右单引号：`’`	`name =‘crifan’`	英文单引号（不分左右）：`'`	`name='crifan'`
双引号： • 左双引号：`“` • 右双引号：`”`	`url=“http://xxx”`	英文双引号（不分左右）：`"`	`url="http://xxx"`
括号： • 左括号：`（` • 右括号：`）`	`input（"请输入"）`	英文括号： • 左括号：`(` • 右括号：`)`	`input("请输入")`
逗号：`，`	`isOk，myScore = False，0`	英文逗号：`,`	`isOk, myScore = False, 0`

然后专门写了代码去演示效果：

```python
# Function: 新手小白错误使用中文的标点符号作为代码的一部分，演示如何修改成准确的英文字符
#   初学者，照着老师的编码打的一样，运行错误-CSDN论坛
#   https://bbs.csdn.net/topics/395827505
# Author: Crifan Li
# Update: 20200212

# 下面演示错误的中文字符标点符号，放在代码中，作为标识符的一部分，则：
# 代码无法运行会报错：SyntaxError: invalid character in identifier

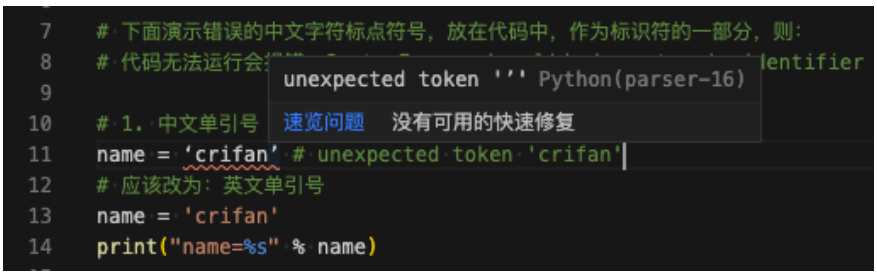
# 1. 中文单引号
# name = ‘crifan’ # unexpected token 'crifan'
# 应该改为：英文单引号
name = 'crifan'
print("name=%s" % name)

# 2. 中文双引号
# url = “http://xxx”
# 应该改为：英文双引号
url = "http://xxx"
print("url=%s" % url)

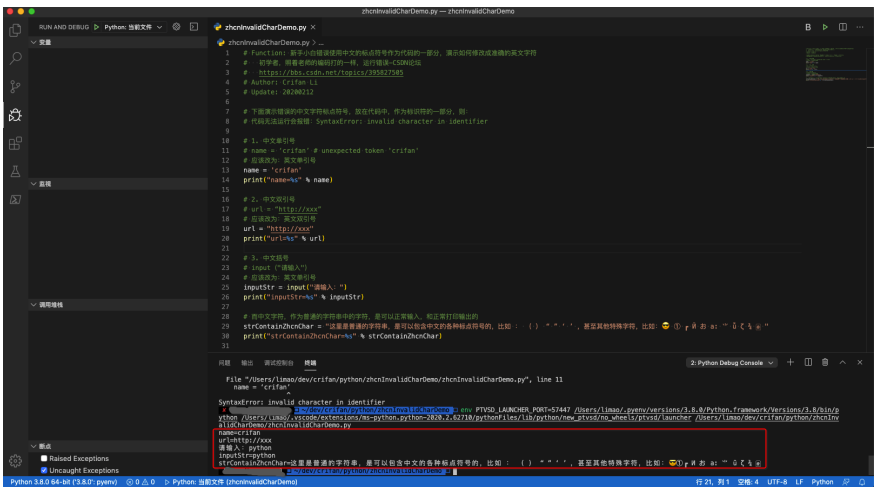
# 3. 中文括号
# input （"请输入"）
# 应该改为：英文单引号
inputStr = input("请输入：")
print("inputStr=%s" % inputStr)

# 而中文字符，作为普通的字符串中的字符，是可以正常输入，和正常打印输出的
strContainZhcnChar = "这里是普通的字符串，是可以包含中文的各种标点符号的，比如 ：  （ ）  “
print("strContainZhcnChar=%s" % strContainZhcnChar)
```

而如果把第一行解除注释，VSCode中就会提示代码错误：



把有问题的，中文字符作为标识符的，都注释掉，才能正常运行：



因此要记得，用于代码和变量和表达式中的字符，都一定要是英文字符，而不能是中文字符（或其他语言的特殊的字符）。

举例：中文右括号错误

问题

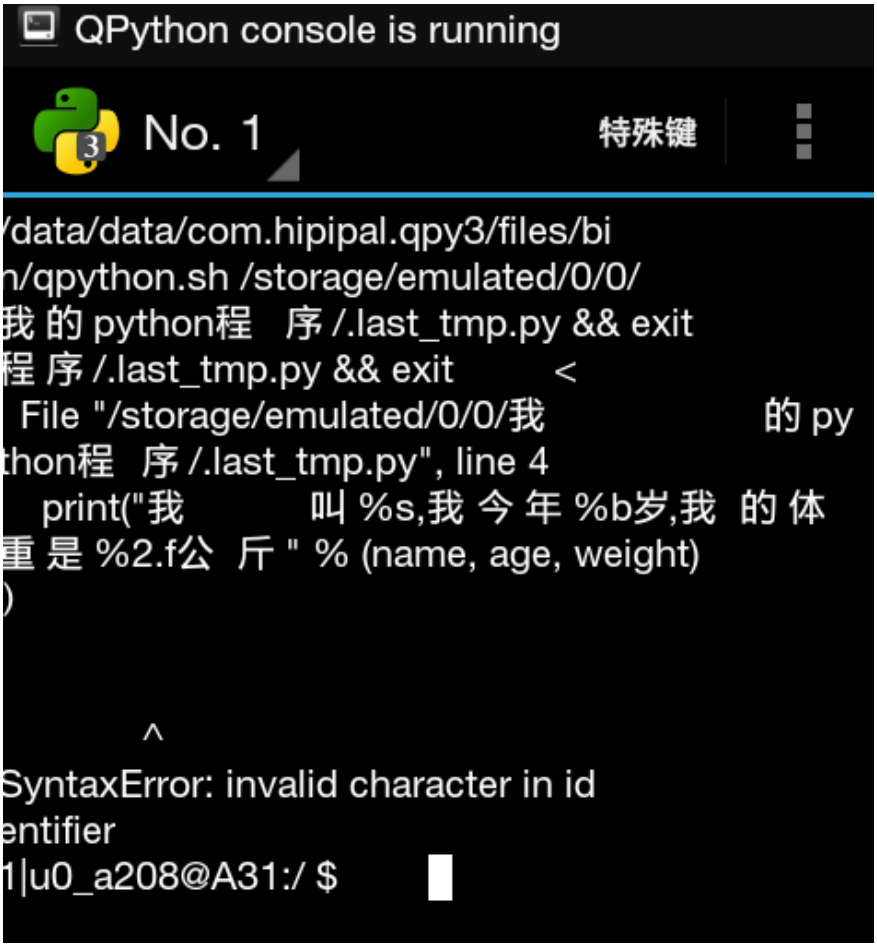
Qpython3-CSDN论坛

解答

先说错误：

不过在具体解释之前，先针对其错误，给出自己的判断：

估计其输入代码时，最后一个括号是中文的括号，所以报错：



而之所以没说是 `%b` 引起的问题，是因为我自己调试了同样的代码后，知道如果是 `%b` 的话，会是另外不同错误提示：

```
ValueError: unsupported format character 'b' (0x62) at index 9
```

而不会是这里的：

```
SyntaxError: invalid character in identifier
```

解决办法：

把最后一个中文括号

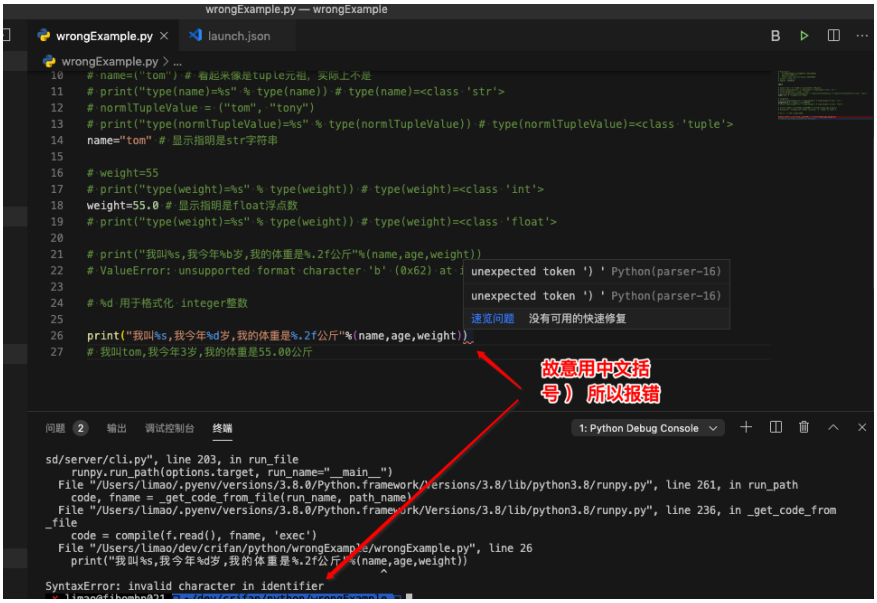
weight)）

改为正常的英文括号

weight))

后记：

后来也故意去把正确的代码，改为错误的，最后用了个中文的括号，所以也会报错的：



且没有运行之前， `pylint` 就可以识别和检测出来，并提醒错误：

```
unexpected token ‘) ’Python(parser-16)
```

以及：运行后也会报错：

```
File "/Users/limao/dev/crifan/python/wrongExample/wrongExample.py", line 26
SyntaxError: invalid character in identifier
```

验证了我第一次的推断是对的：的确是中文括号的问题。

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:27:34

测试文件名和库同名

还有一种新手常见的问题是：把测试文件或变量的名字，和要调用的库名字一样，从而导致各种难以察觉的问题。

下面举例解释：

导入tushare出错

问题

菜鸟求教tushare pro，看不懂的问题-CSDN论坛

解答

刚开始看，以为是其他什么问题呢。

帮其搜索：

tushare

找到：

Tushare -财经数据接口包

> module tushare has no attribute set_token

新人发帖求助！python使用tushare股票分析包方法报错-CSDN论坛

> 估计你的文件名叫tushare.py, 改个名就好了.

-> 那估计是这个问题了：小白测试python时，把测试文件，写成了：`tushare.py`

和要测试的库`tushare`重名了。

导致Python解析器去解析代码

```
import tushare as ts
```

变成自己的测试文件`tushare.py`，而不是要测试的库`tushare`了。

也看到了别的类似的例子

pycharm 导入tushare错误，请帮帮忙，百度半天也没搞定-CSDN论坛

```
import tushare as ts
print(ts.__version__)
提示错误##########
D:\Programs\Anaconda3\python.exe D:/PythonWorks/PycharmProjects/Stocks/tushare
Traceback (most recent call last):
 File "D:/PythonWorks/PycharmProjects/Stocks/tushare.py", line 1, in <module>
import tushare as ts
File "D:\PythonWorks\PycharmProjects\Stocks\tushare.py", line 2, in <module>
 print(ts.__version__)
AttributeError: module 'tushare' has no attribute '__version__'
Process finished with exit code 1
```

很明显就是同样的问题。

关于tushare python的问题。-CSDN论坛

> 我下好了tushare 导入正确，但是在调用函数的时候，为什么总是报没有这个函数呢，
>
> `print(help(ts.get_k_data()))`
>
> AttributeError: module ‘tushare’ has no attribute ‘get_k_data’
>
> 或许你将tushare进行了改变，赋值替换了它或自己新建了一个叫tushare的文件

AttributeError: module ‘tushare’ has no attribute ‘**version**’ · Issue #241 · waditu/tushare · GitHub

> 不要以tushare作为文件名

在策略模块定义函数引用tushare pro – VincentZHOU – JoinQuant

> 也有可能你的研究中有一个叫 tushare.py 的文件，导致没有调到真正的 tushare 包，如果有的话可以试着改个名字

此处Mac中去写代码测试效果：

先用pip安装库

【已解决】mac中pip安装Python库tushare

然后去VSCode中写代码测试：

起个正常的，典型的，用于测试的文件名：

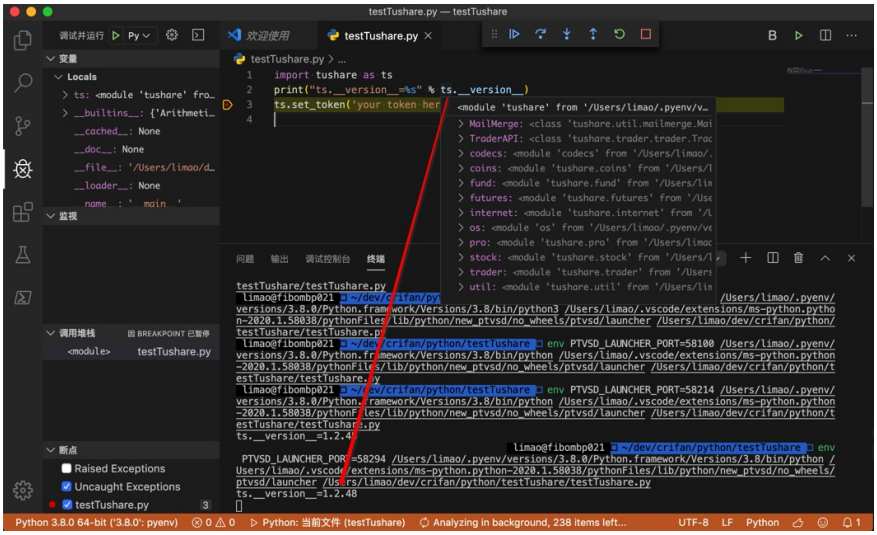
比如：

`testTushare.py`

```
import tushare as ts
print("ts.__version__=%s" % ts.__version__)
ts.set_token('your token here')
```

可以正常输出：

```
ts.__version__=1.2.48
```

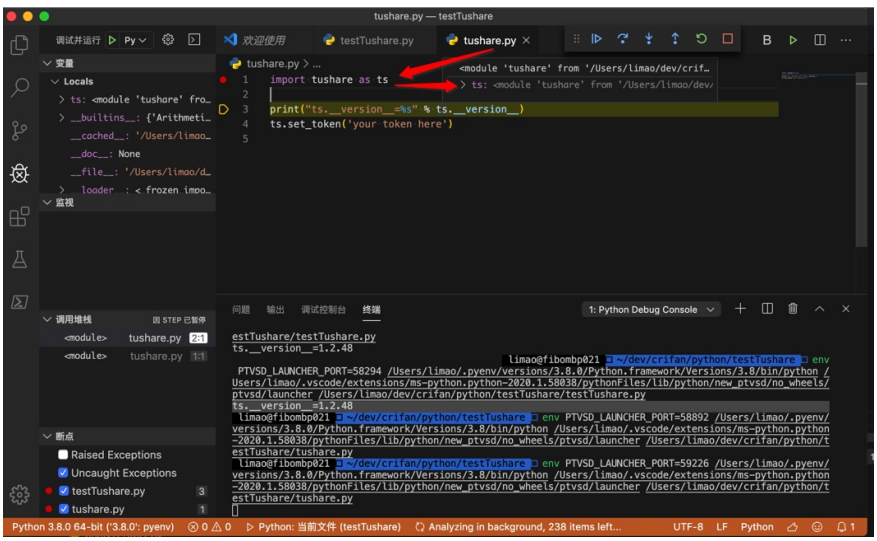


而如果改为：

用于测试的文件名，和要测试的库 `tushrare` 同名，变成： `tushare.py`

```
import tushare as ts
print("ts.__version__=%s" % ts.__version__)
ts.set_token('your token here')
```

加断点调试时就会发现，导入的 `ts` ，其实是空的，啥函数和属性都没有：

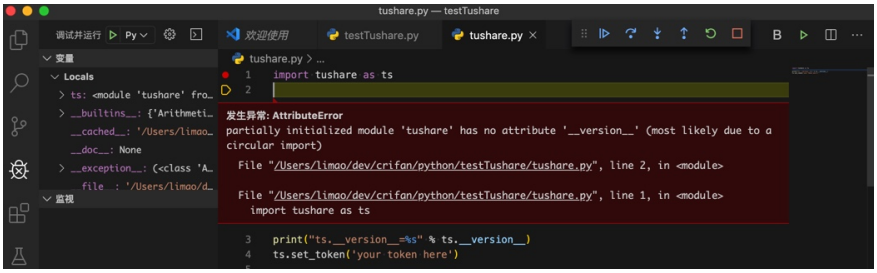


由此后面的代码：

```
print("ts.__version__=%s" % ts.__version__)
```

就会报错了：

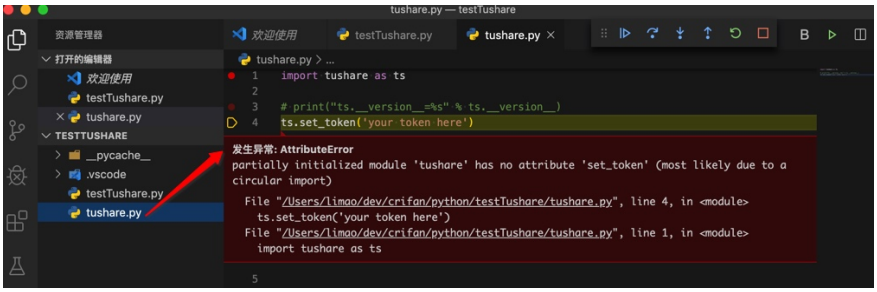
```
发生异常: AttributeError
partially initialized module 'tushare' has no attribute '__version__' (most li
File "/Users/limao/dev/crifan/python/testTushare/tushare.py", line 2, in <modu
```



如果注释掉上面一行，用下面的

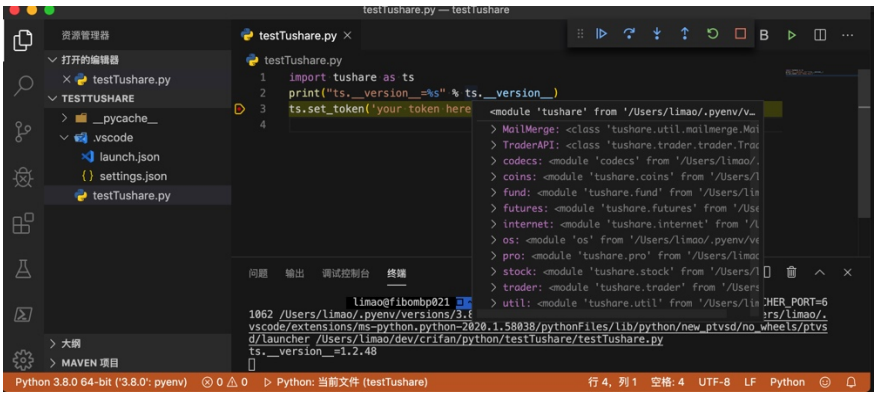
```
ts.set_token('your token here')
```

去测试，也是同样问题：



与之对比：

名字不要是库名，即可正常导入和调试：



结论

写(python)代码用于测试时，创建测试用的(python)文件

- **应该**：起个，见名知意的，更加易懂的，不容易混淆的文件名
  - 常见思路和做法：测试用的，演示用的文件名中，往往包含 `test` ， `demo` ， `example` 等字眼
  - 此处：
    - 举例：比较合适的写法
      - `testTushare.py`
      - `demoTushare.py`
      - `tushareDemo.py`
      - `tushareTest.py`
      - `tushareExample.py`

- **不应该**：和测试的库同名
  - 此处：
    - 用于写测试tushare的Python库的测试文件，不要写成：`tushare.py`
  - 坏处：会和原有的库冲突
    - 导致：import导入时，错误导入了自己的测试文件，而不是原有的库
- -》引申 + 举一反三 + 相关：
  - 也不应该：起个其他简单的，偷懒的名字
    - 典型的有：
      - `1.py`
      - `123.py`
      - `test.py`
      - `demo.py`
      - `a.py`
      - `abc.py`
      - 等等
    - 都是属于：
      - 坏习惯
      - 偷懒的写法
      - 容易和其他文件冲突的写法
      - 也不容易看懂=没法从你文件名看出你要做什么

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:27:23

不要故意用错误语法

遇到一些新手，在写代码时，本身逻辑不是很清晰，又参考了别人一些错误的写法，导致代码无法输出预期结果。

下面举例来解释，什么叫故意写错，用了错误的语法，以及正确的语法是什么样的，应该怎么写。

举例：故意写错变量类型

看到

Qpython3-CSDN论坛

代码中有很多错误的写法，且感觉是，被别人故意设计成这样的。

现在去详细解释如下：

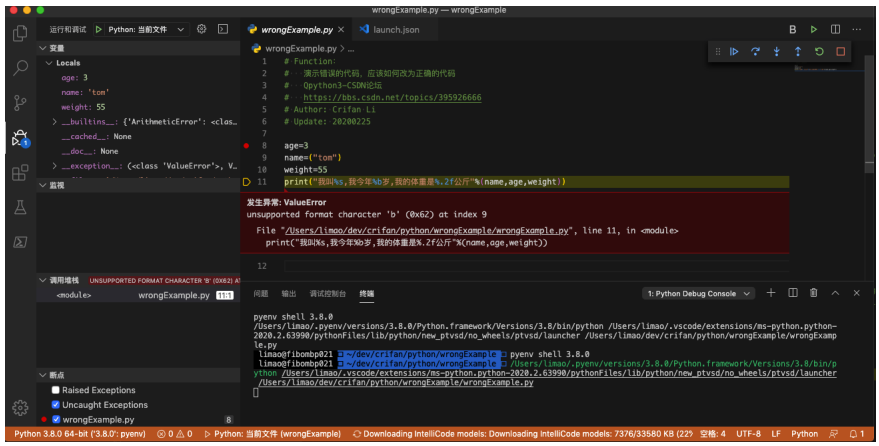
对于代码：

```
# Function:
#   演示错误的代码，应该如何改为正确的代码
#   Qpython3-CSDN论坛
#   https://bbs.csdn.net/topics/395926666
# Author: Crifan Li
# Update: 20200225

age=3
name=("tom")
weight=55
print("我叫%s,我今年%b岁,我的体重是%.2f公斤"%(name,age,weight))
```

去调试运行，肯定会报错：

```
unsupported format character 'b' (0x62) at index 9
```



所以，应该改为：

```
# Function:
#   演示错误的代码，应该如何改为正确的代码
#   Qpython3-CSDN论坛
#   https://bbs.csdn.net/topics/395926666
# Author: Crifan Li
# Update: 20200225

age=3
# name=("tom")
# weight=55
# print("我叫%s,我今年%b岁,我的体重是%.2f公斤"%(name,age,weight))
# ValueError: unsupported format character 'b' (0x62) at index 9

name="tom" # 显示指明是str字符串，而不是 看起来像是tuple元祖，实际上不是
weight=55.0 # 显示指明是float浮点数
# %d 用于格式化 integer整数
print("我叫%s,我今年%d岁,我的体重是%.2f公斤"%(name,age,weight))
# 我叫tom,我今年3岁,我的体重是55.00公斤
```

其中：

不要假装是tuple，但实际却是str

```
name=("tom")
```

是错误的示范和写法

因为 `(xxx, yyy)` 是标准的 `tuple` **元祖**的语法

上述写法很容易让人产生误解，以为name是个tuple元祖变量

但实际上其却等价于：

```
name="tom"
```

name实际上是个 `str` **字符串**变量

所以，本来就应该改为正常的标准的写法：

```
name="tom"
```

才对。

去加上 `type` **变量类型**的打印，就容易对比看出区别了：

```
name=("tom")
print("type(name)=%s" % type(name))
# type(name)=<class 'str'>
# normlTupleValue = ("tom", "tony")
print("type(normlTupleValue)=%s" % type(normlTupleValue))
# type(normlTupleValue)=<class 'tuple'>
```

不要故意用错误的格式化写法

`%d` 是用来格式化显示数字的

`%b` ，是没见过的写法

即，此处的age是3岁，3个是integer整型数字，应该用%d去格式化输出，而不应该是%b

以及：weight=55

本意是：身体体重是55公斤

然后往往是个 `float` 浮点数，即有小数的部分的

所以后续是去用**浮点数**的 `%.2f` 去格式化显示的

但是此处初始化时的写法却是

```
weight=55
```

此时，weight是 `int` = `integer` = `整型`

而不是我们希望的：

`float` = `浮点数`

应该改为：

```
weight=55.0
```

才对。

加上type变量类型的打印，就容易看出区别来了：

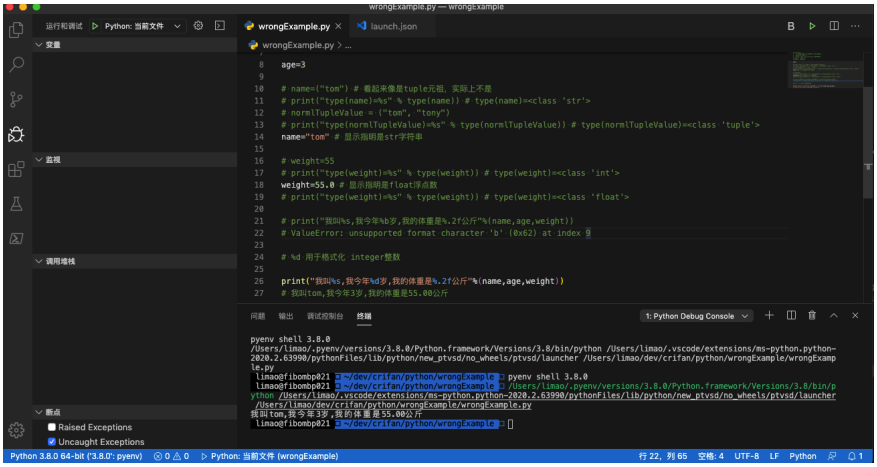
```
weight=55
print("type(weight)=%s" % type(weight))
# type(weight)=class 'int'
weight=55.0 # 显示指明是float浮点数
print("type(weight)=%s" % type(weight))
# type(weight)=class 'float'
```

最终完整代码是：

```
# Function:
#   演示错误的代码，应该如何改为正确的代码
#   Qpython3-CSDN论坛
#   https://bbs.csdn.net/topics/395926666
# Author: Crifan Li
# Update: 20200225

age=3
# name=("tom") # 看起来像是tuple元祖，实际上不是
# print("type(name)=%s" % type(name)) # type(name)=<class 'str'>
# normlTupleValue = ("tom", "tony")
# print("type(normlTupleValue)=%s" % type(normlTupleValue)) # type(normlTupleV
name="tom" # 显示指明是str字符串
# weight=55
# print("type(weight)=%s" % type(weight)) # type(weight)=<class 'int'>
weight=55.0 # 显示指明是float浮点数
# print("type(weight)=%s" % type(weight)) # type(weight)=<class 'float'>
# print("我叫%s,我今年%b岁,我的体重是%.2f公斤"%(name,age,weight))
# ValueError: unsupported format character 'b' (0x62) at index 9
# %d 用于格式化 integer整数
print("我叫%s,我今年%d岁,我的体重是%.2f公斤"%(name,age,weight))
# 我叫tom,我今年3岁,我的体重是55.00公斤
```

调试输出效果：



另外，关于 `Python` 中 `print` 参数的字符串格式化支持哪些，详见附录：

print字符串格式化语法

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:28:19

属性报错AttributeError

新手常见问题中，很多都是找不到某某属性，报：

`AttributeError: module xxx hs no attribute yyy`

对此AttributeError类型的错误：

- 背景
  - 实际上涉及到的代码中往往是 `xxx.yyy` 类型的写法
    - 而之所以报错，说明是 某某**变量（库）** `xxx` ，没有某个 `yyy` 的**属性** `attritude`
- 而具体错误的原因，典型的有以下几种可能：
  - 多数
    - 都是一些普通的错误
      - 比如
        - 库的名字写错了
        - 属性的名字写错了
        - 库或属性的大小写（没看清，笔误）写错了
  - 个别
    - 是其他原因
      - 比如
        - 库的版本升级了，导致该接口取消（废弃、不用）了，取而代之新的接口了
          - 解决思路：找到官网最新的接口，改用新的接口
        - 上下代码逻辑问题，导致得到的值xxx，不是希望的值

下面通过举例来解释如何解决：

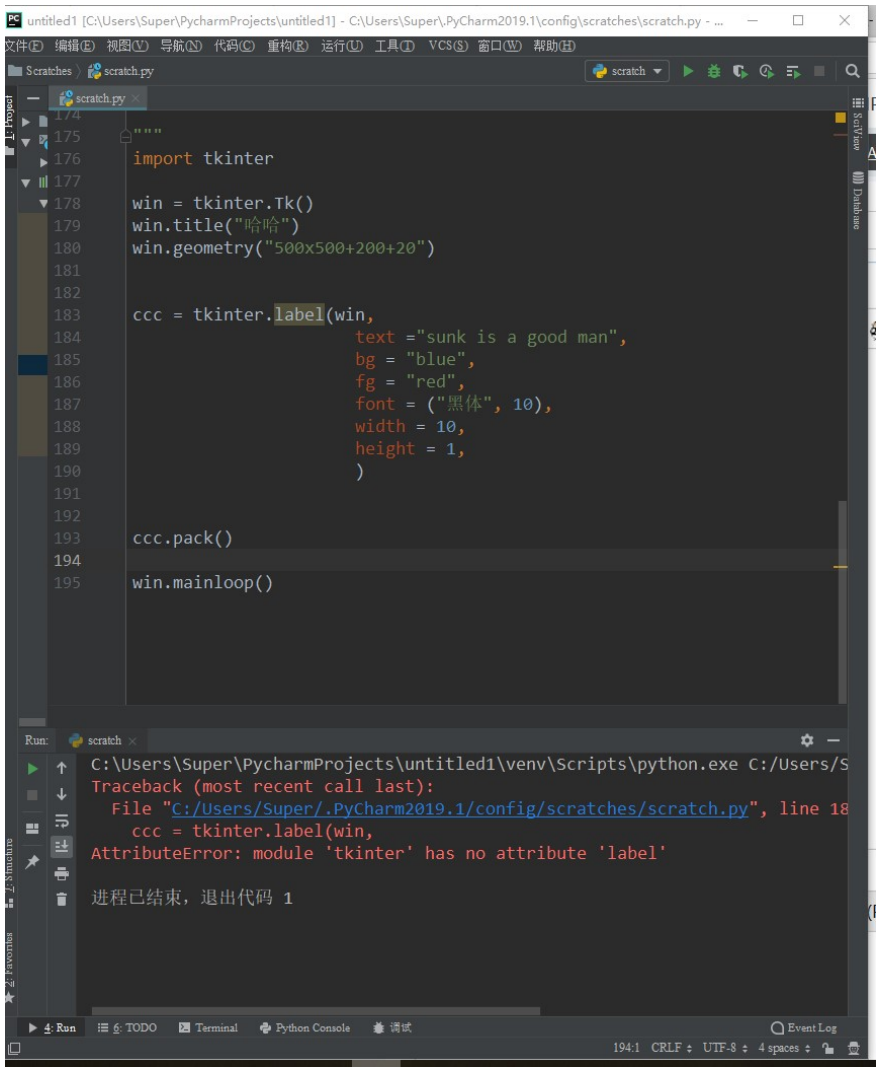
crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:26:31

大小写有误导致属性报错

举例：tkinter的label中l应该大写

问题

小白遇到一个关于Pycharm中 Tkinter问题-CSDN论坛



解答

先从问题入手，看到你的错误提示是：

`AttributeError: module 'tkinter' hs no attribute 'label'`

-》字面意思是：`tikinter` 这个（变量?库？）没有 `label` 这个属性

-》为何没有？虽然我也不熟悉此问题具体的细节，但是我有学习能力和解决问题能力

下面记录解决问题的过程：

首先已知python（好像是）有个（内置的）库叫：`tkinter`

所以去google搜：

`python tkinter`

`python tkinter label`

找到一些资料：

- Tk图形用户界面(GUI) — Python 3.8.2 文档
- tkinter --- Tcl/Tk的Python接口 — Python 3.8.2 文档
- tkinter — Python interface to Tcl/Tk — Python 3.8.2 documentation
- Python GUI 编程(Tkinter) | 菜鸟教程

看到一些资料：

> Python3.x 版本使用的库名为 tkinter,即首写字母 T 为小写。
>
> `import tkinter`
>
> 要使用 Tkinter 通常你只需要一条简单的 import 语句: `import tkinter`
>
> 或者更常用的:
>
> `from tkinter import *`

- Python - Tkinter Label - Tutorialspoint
- Tkinter教程之Label篇*Python*灵蛇舞动-CSDN博客

```
from Tkinter import *
#初始化Tk
root = Tk()


#创建一个label，使用编码，到现在为止还没有使用过直接通过“drag-and-drop”就可以完成的IDE。
label = Label(root,text = 'Hello Tkinter')
```

Label & Button 标签和按钮 - 窗口 Tkinter | 莫烦Python

```
l = tk.Label(window,
    text='OMG! this is TK!',    # 标签的文字
    bg='green',     # 背景颜色
    font=('Arial', 12),     # 字体和字体大小
    width=15, height=2  # 标签长宽
    )
l.pack()    # 固定窗口位置”
```

tkinter.ttk --- Tk主题小部件 — Python 3.8.2 文档

```
l1 = tkinter.Label(text="Test", fg="black", bg="white")
l2 = tkinter.Label(text="Test", fg="black", bg="white")
```

结论

- Python中的tkinter是自带的
  - 表示不用额外安装
- tkinter库的名称
  - Python 2中是大写的T：`Tkinter`

    - Python 3中是小写的T：`tkinter`
- Python 3中导入tkinter的常见写法
    - 有2种：
        - 普通的：`import tkinter`
        - 更常用的：`from tkinter import *`
- tkinter中的Label的L是大写的
    - 是从官网可以查看到的
        - tkinter.ttk --- Tk主题小部件 — Python 3.8.2 文档
- tkinter中使用Label的写法
    - ```
      import tkinter
      l1 = tkinter.Label(text="Test", fg="black", bg="white")
      ```
    - 或：
      ```
      from tkinter import *
      l1 = Label(text="Test", fg="black", bg="white")
      ```

-> 你此处，把：

```
ccc = tkinter.label(win
```

改为：

```
ccc = tkinter.Label(win
```

即可。

引申=举一反三

- 为何你会把大写的Label 写成小写的label?
    - 因为：你没有去参考官网文档
- 为何你找不到官网文档？或者 不知道有官网文档?
    - 因为其实不熟悉正确的：学习思路
        - 好的学习思路是：有问题，尽量找官网技术文档
            - 因为官网技术文档，往往解释问题
                - 最清楚和明白：很多技术，官网资料写的往往都很通俗易懂
                - 最准确：不像别的资料可能会滞后或过期
- 如何学习好的学习思路和方法?
    - 参考我的教程
        - 学习方法思路及技术心得总结
- Python小白如何避免常见的坑?
    - 参考当前教程 Python新手小白常见错误和问题 的其他相关内容

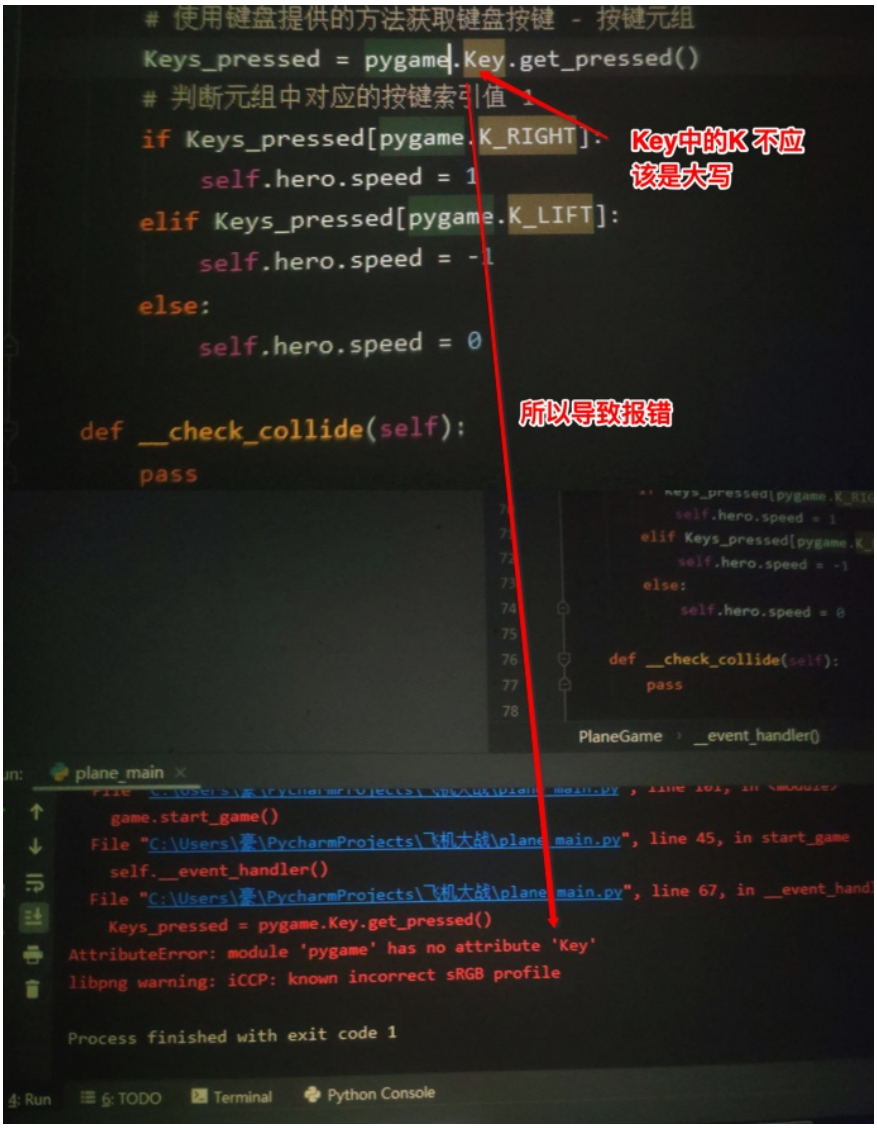
举例：Pygame的Key的K大写是错误的

问题

Pygame has no attribute key-CSDN论坛
```

解答

下图标识出错误位置：




```
pygame.Key.get_pressed()
```

中的 `Key` 中的 `K`，是**大写**，导致报错：

```
AttributeError: module 'pygame' has no attribute 'Key'
```

即：pygame没有Key这个属性

所以报错

-》就像原贴作者说的“按照视频的代码一模一样敲的，却运行出这个错误，看了下别人的，都没出现这个错误”

-》其实问题的原因，人家报错已经告诉你了：pygame没有Ke这个属性

-》那pygame为何没有Key这个属性呢？

-》因为根据官网访问解释：

pygame.key — pygame v2.0.0.dev5 documentation

> `pygame.key.get_pressed()`
>
> get the state of all keyboard buttons
>
> get_pressed() -> bools

可以看出，`pygame.key` 中的 `key` 中的 `k` 是**小写**

-》而你这里大写，不是人家这个pygame的库所支持的属性，所以报错

-》那为何会出现这个错误呢？

-》这就属于典型的情况了：新手在（参考别人代码）敲代码期间，不小心的发生了笔误，把小写的k，写成大写的K了

-》为何其他人，或者说有经验的人，不会出现这个问题呢？

-》那是因为：新手，小白，往往缺乏一些计算机编程语言的基础知识

-》放在此处就是：计算机编程语言（比如Python）中的基本知识是：变量是分大小写的

如果小白新手不注意，就会不小心出现笔误，导致变量属性找不到的问题

-》此处如何解决问题：

根据pygame的官网的要求，把Key改为key：

```
pygame.key.get_pressed()
```

就可以解决问题了。

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:26:48

上下代码逻辑问题

此处介绍，上下代码逻辑问题导致，获取到的值xxx，不是希望的值，导致属性报错的情况。

举例：调用函数之前多了个=等于号

问题

制作数据集时报错AttributeError: 'str' object has no attribute 'write'-CSDN论坛

解答

基本上确定了，就是其自己笔误：多写了个等于号

把：

```
writer= tf.compat.v1.python_io.TFRecordWriter("mask_and_nomask_test.tfrecords"
```

写成：

```
writer= tf.compat.v1.python_io.TFRecordWriter=("mask_and_nomask_test.tfrecords
```

导致此处的 `writer` ：

- 不是：原本希望的TFRecordWriter()所返回的变量
- 而是：一个普通的字符串

因为：

```
writer= tf.compat.v1.python_io.TFRecordWriter=("mask_and_nomask_test.tfrecords
```

等价于：

```
tf.compat.v1.python_io.TFRecordWriter = ("mask_and_nomask_test.tfrecords")
writer = ("mask_and_nomask_test.tfrecords")
```

等价于：

```
tf.compat.v1.python_io.TFRecordWriter = "mask_and_nomask_test.tfrecords"
writer = "mask_and_nomask_test.tfrecords"
```

此时 `writer` 变量只是个str字符串。所以此处才报错：writer（这个str字符串变量）没有（TFRecordWriter才有的）write这个属性

解决办法

去掉你的笔误，即去掉多写的那个**等于号** `=`

```
writer= tf.compat.v1.python_io.TFRecordWriter("mask_and_nomask_test.tfrecords"
```

其含义是：

- tf.data.experimental.TFRecordWriter | TensorFlow Core v2.1.0
- tf.io.TFRecordWriter | TensorFlow Core v2.1.0

调用了：

- 函数： `tf.compat.v1.python_io.TFRecordWriter`
  - 传入的参数是： `"mask_and_nomask_test.tfrecords"`
  - 才会返回
    - 对应的类 `TFRecordWriter`
      - 其才有 `write` 函数
        - 后续的
          - `writer.write(xxx)`
          - 才能正常运行。

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:26:25

函数参数

函数参数相关常见问题，主要分2类：

- 参数个数不匹配
- 参数位置错误

下面举例说明：

参数位置错误=参数顺序搞错了

举例：Python的pyecharts参数

问题

关于Python pyecharts 的问题（已经找资料找了半天了）-CSDN论坛

> ```
> name=['A','B','C','D','E']
> values=[1,2,3,4,5]
> wordcloud.add("",name,values,word_size_range=[20,100],shape= "circle")
> ```
>
> 以上程序会抛出
>
> `TypeError: add() takes 3 positional arguments but 4 positional arguments (and 2 keyword-only arguments) were given`
>
> 难道只能用官网上给的列表嵌套元组的形式吗？但我看到过类似我这样写的。。。。0.0

解释

对于：

```
wordcloud.add("",name,values,word_size_range=[20,100],shape= "circle")
```

中的 `wordcloud` 的 `add` 函数，去找了下，过程如下：

google搜：

`pyecharts add`

找到

30分钟学会pyecharts数据可视化 - 知乎

人家写法是：

```
cloud.add(name = 'utils',attr = words,value = counts,
          shape = "circle",word_size_range = (10,70))
```

-> 也没有你的

`wordcloud.add("",name`

中的第一个，空字符串：`""`

后来找到官网文档

Documentation - pyecharts-en

中是：

> WordCloud
>
> WordCloud.add() signatures
>
> add(name, attr, value, shape="circle", word_gap=20, word_size_range=None, rotate_step=45)

-> 更没有你写的 第一个参数 空字符串 `""`

-> 所以结论很明显：

看起来是，你多传递了一个参数，第一个的空字符串 `""`

后来发现：实际上也不是，而是：

对照官网参数顺序：

name, attr, value, shape="circle", word_gap=20, word_size_range=None, ...

而你的是：

"",name,values,word_size_range=[20,100],shape= "circle"

很明显是（看起来是？）：你把参数的顺序搞错了吧？

（看起来）应该改为：

wordcloud.add(**name,"",values**,word_size_range=[20,100],shape= "circle")

其中的 **""** 对应着第二个参数：`attr`

注：我不是很确定你代码逻辑，需要你自己明确要给

`name, attr, value`

传递具体什么值。

引申=心得

-》不要随便参考别人（可能错误，可能是落后的，没更新的）代码

-》或者自己瞎猜一个（函数的参数，和顺序）

无论如何，都应该是：

- 改为正确的学习思路和方法
  - 核心要点是：去找官网正规资料
    - 关于
      - 如何掌握正确的学习方法和思路
      - 如何利用google查找到自己要的资料
    - 可参考我的：
      - 学习方法思路及技术心得总结

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:28:14

代码注释

疑问：#的注释

- **问**：加 `#` 号后机器能读取这个命令吗？
- **答**：不能

Python中一行的开头是 `#` ，后面的内容就会被（解释器/编译器）忽略了

->专业术语叫做 **注释=注释代码**：仅仅是起解释和说明，供人类看的内容 机器不读，会忽略掉

->不同编程语言的注释代码的语法都不同，比如：

- 普遍采用的（C/C++/Java等）：
  - 单行注释： `// xxx`
  - 多行注释： `/* yyy */`
    - 其中 `yyy` 可以是很多行
- `Python` ： `# xxx`
- `HTML` ： `<!-- xxx -->`
- `ini` 文件： `; xxx`

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:27:51

Python常见问题

此处整理和Python相关的基础方面的，新手和小白用户常犯的问题。

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-08-04 19:43:20

input返回值是字符串

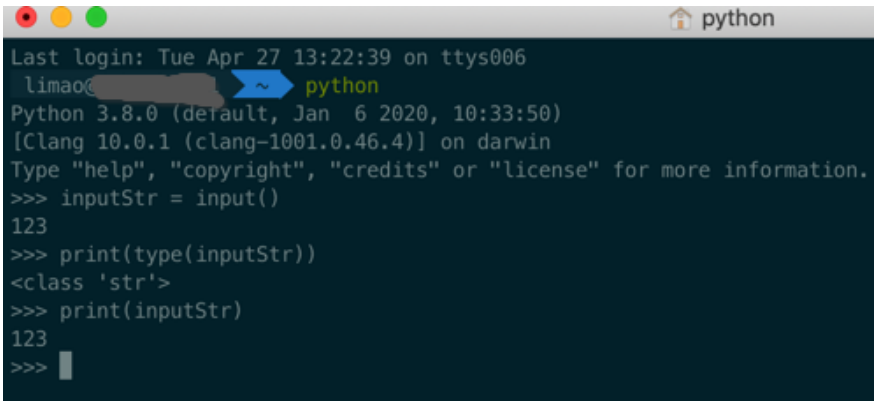
新手常犯的一个问题是：以为input返回的值是 `int` 数字等所希望的类型

但其实：input返回的值，都是 `str` **字符串**类型。

演示代码

此处，用如下简单的代码，演示，input返回值是字符串 `str`

```
limao@xxx □ ~ □ python
Python 3.8.0 (default, Jan  6 2020, 10:33:50)
[Clang 10.0.1 (clang-1001.0.46.4)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> inputStr = input()
123
>>> print(type(inputStr))
<class 'str'>
>>> print(inputStr)
123
>>>
```



其中可见：

- 输入了 `123` ，通过 `input` 返回后的变量类型
  - 是： `str` 字符串 -> 内部表示是 **'123'**
    - 不是，（python小白）以为的： `int` 整数 -> 内部表示是：**123**

常见input相关问题

把input返回值当做int数值

把 `input` 返回的内容是**字符串**而不是以为的变量类型，比如 `int` =整数， `float` =浮点数等

看看哪里错了，快崩溃了-CSDN论坛

类似的：Python求助-CSDN论坛

问题

利用列表实现筛选法求素数

问题描述：编写程序，输入一个大于2的自然数，然后输出小于该数字的所有素数组成的列表

```
def sushu(num):
    lst=[]
    for i in range(2,num):
        flag = 1
        for j in range(2,i):
            if i%j==0:
                flag=0
                break

        if flag:
            lst.append(i)
    print(lst)

num=input("请输入一个大于2的自然数：")
sushu(num)
```

解答

你是把

```
num=input("请输入一个大于2的自然数：")
```

得到的 `num` 以为是 `数字 = int` 类型了

其实是 `字符串 = str` 类型

所以可以通过打印变量类型，更好的理解代码逻辑：

```
numStr = input("请输入一个大于2的自然数：")
print("type(numStr)=%s" % type(numStr))
# <class 'str'>
numInt = int(numStr)
print("type(numInt)=%s" % type(numInt))
# <class 'int'>
```

```
C:\Users\[redacted]>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32

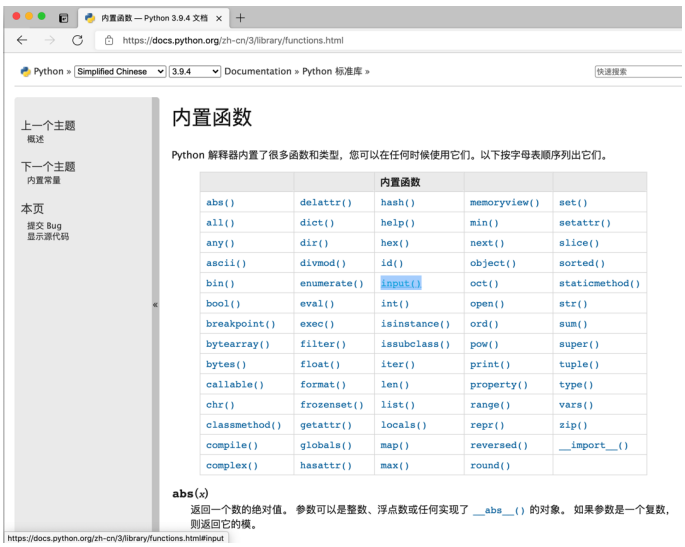
Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated.  Libraries may fail to load.  To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> numStr = input("请输入一个大于2的自然数：")
请输入一个大于2的自然数：5
>>> print("type(numStr)=%s" % type(numStr))
type(numStr)=<class 'str'>
>>> numInt = int(numStr)
>>> print("type(numInt)=%s" % type(numInt))
type(numInt)=<class 'int'>
>>> print(numStr)
5
>>> print(numInt)
5
>>>
```

注意到，input返回的变量类型，通过 `type(numStr)` 查看到，是 `<class 'str'>` ，即是str字符串的类型

Python官网中input的解释

此处再去深究，Python中官网中，关于input，是如何解释的：

- Python中的内置函数
  - 中文
    - https://docs.python.org/zh-cn/3/library/functions.html
    - 

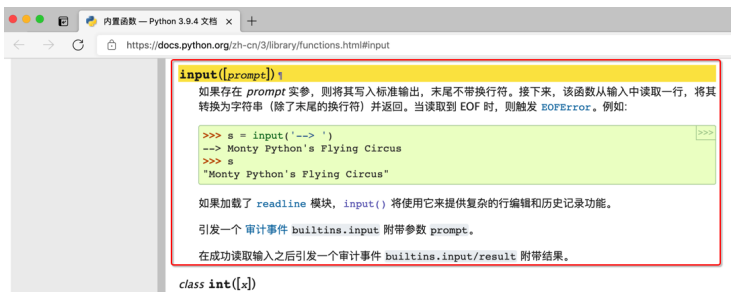
  - 英文
    - https://docs.python.org/3/library/functions.html
- -> 内置函数： `input`
  - 中文
    - https://docs.python.org/zh-cn/3/library/functions.html#input

```
input([prompt])
如果存在 prompt 实参，则将其写入标准输出，末尾不带换行符。接下来，该函数从

>>>
>>> s = input('--> ')
--> Monty Python's Flying Circus
>>> s
"Monty Python's Flying Circus"
如果加载了 readline 模块，input() 将使用它来提供复杂的行编辑和历史记录功

引发一个 审计事件 builtins.input 附带参数 prompt。

在成功读取输入之后引发一个审计事件 builtins.input/result 附带结果。
```



  - 英文
    - https://docs.python.org/3/library/functions.html#input

```
input([prompt])¶
If the prompt argument is present, it is written to standard ou

>>>
>>> s = input('--> ')
--> Monty Python's Flying Circus
>>> s
"Monty Python's Flying Circus"
If the readline module was loaded, then input() will use it to

Raises an auditing event builtins.input with argument prompt be

Raises an auditing event builtins.input/result with the result
```

其中官网中 `input` 的内部处理过程，都已经解释的很明确了：

该函数从输入中读取一行，将其转换为字符串（除了末尾的换行符）并返回

即， `input` 返回的值是字符串类型 `str` ，不是很多人以为的 `int` 、 `float` 等数值类型

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2021-04-30 17:13:13

Python的shell不是系统终端

此处通过例子来说明，使得新手和小白明白，`系统的终端shell` 和 `Python shell` 不是一个东西。

这个概念需要搞清楚，然后才能正常开发。

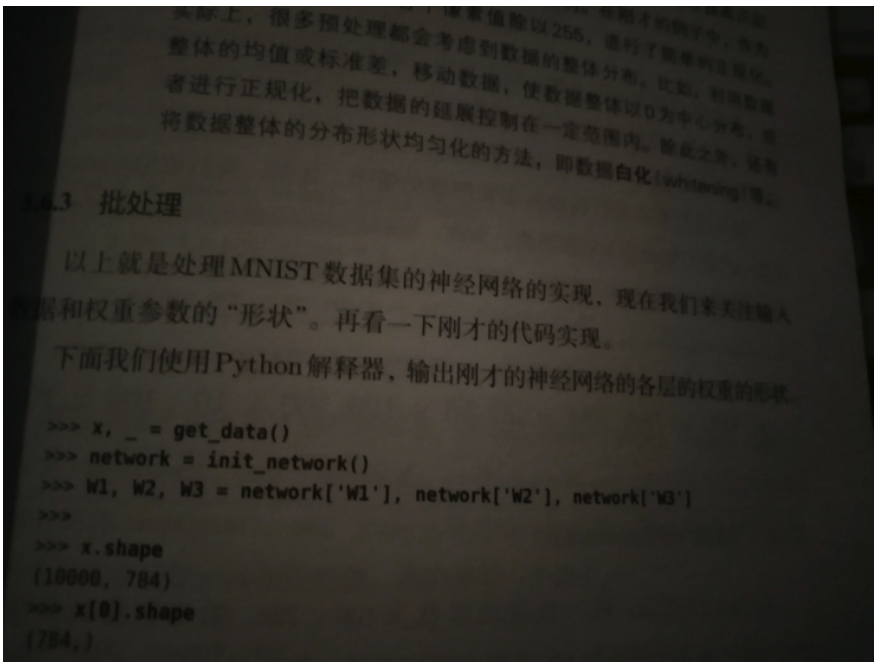
举例

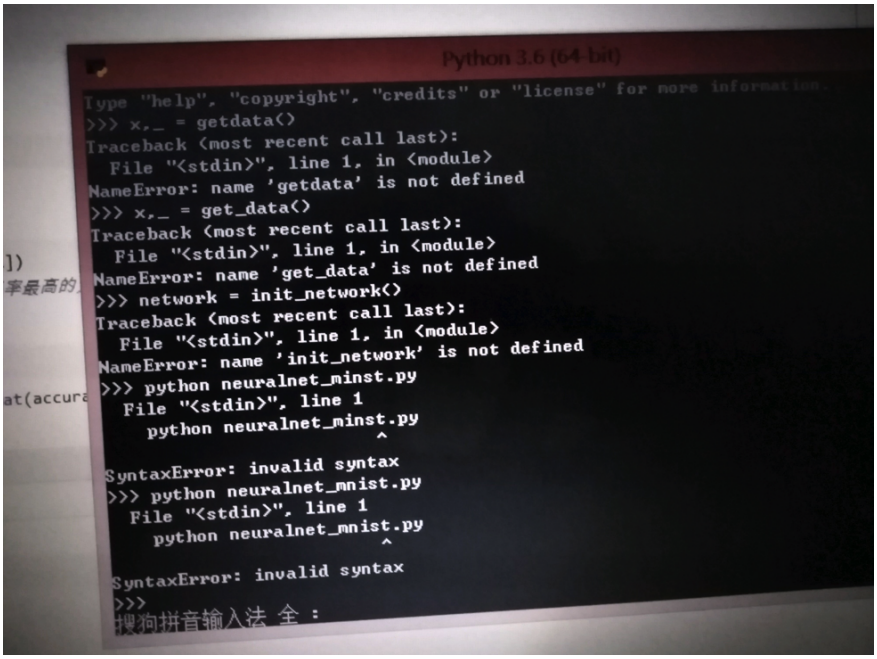
关于python解释器的问题

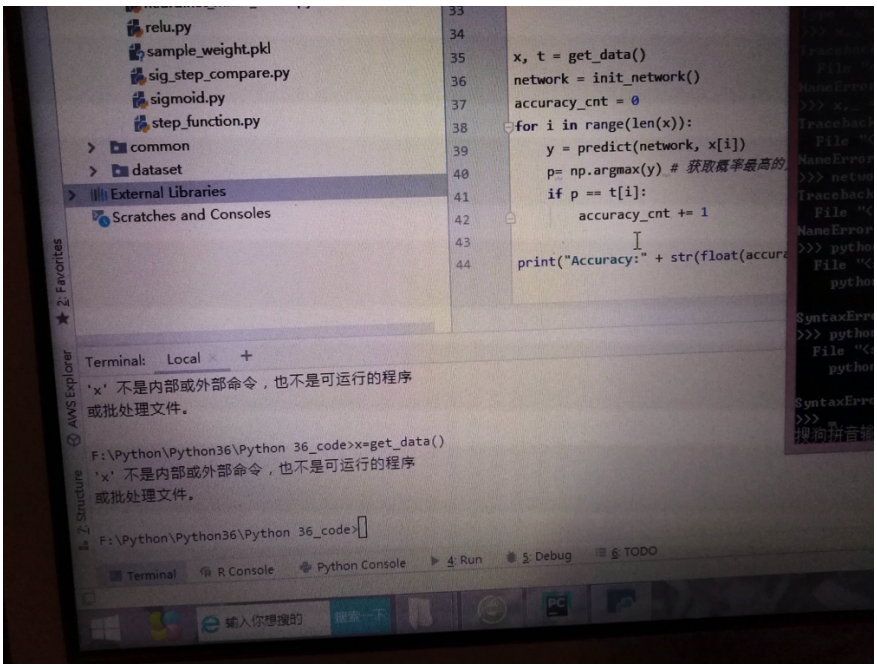
问题

关于python解释器的问题?-CSDN论坛

> 这个教材说用python解释器执行如下代码，我使用的是pycharm，然后用了terminal告诉我没有这个命令
>
> 然后，我用cmd也不行
>
> 然后用python3.6的那个终端，还不行
>
> 之前，也遇到过这种状况，就是terminal必须弄一个虚拟环境才能运行整个库
>
> 我很无奈，就是想问问如何使用终端生成这个东西!



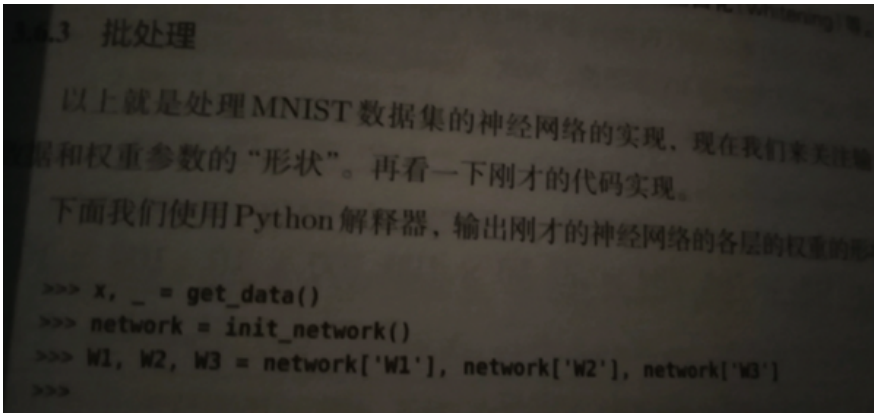




解答

- **简答**：
  - 你
    - 先要搞清楚，书中代码是python shell中运行的
    - 其次还要搞清楚，那段get_date()代码，是需要先输入函数定义，才能继续调用执行的
  - 都搞清楚后
    - 就可以在python shell中输入完整代码，去测试你要的效果了
- **详解**：

对于图片：



很明显是从书中拍的照片

而其中的：

```
>>> x, _ = get_data()
>>> network = init_network()
...
```

等内容，是：在 `Python的shell` 中去测试的代码

对此先要去搞清楚：什么是Python的shell

你此处：

截图的书中的代码：

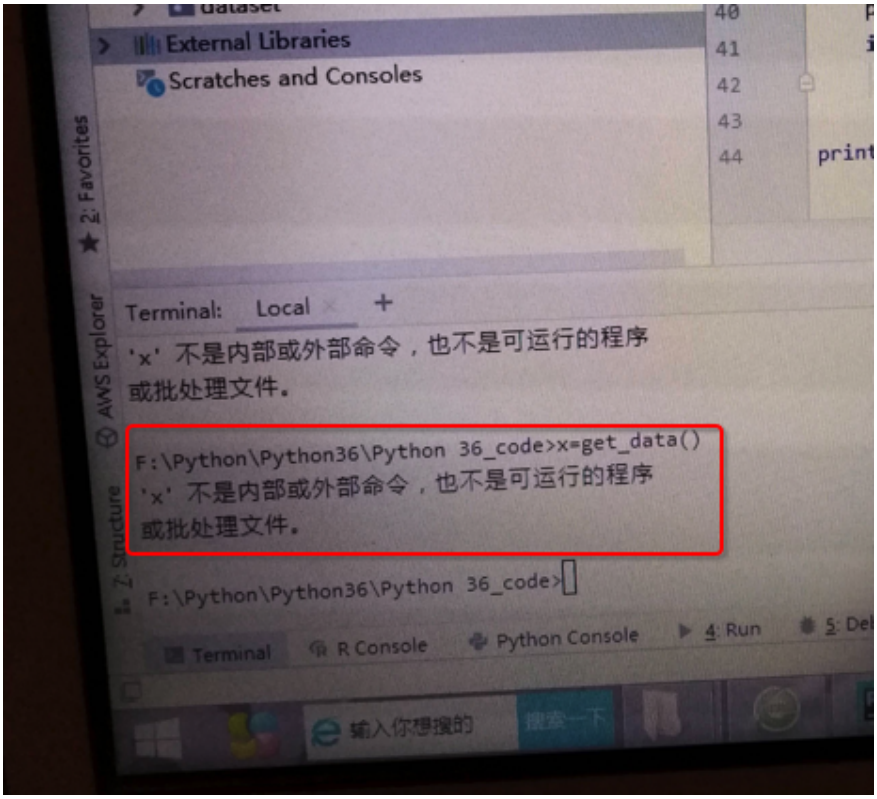
```
x, _ get_data()
network = init_network()
...
```

明显是，用来放到 `Python shell` 中才（可）能运行。

所以你说的：

> 这个教材说用python解释器执行如下代码，我使用的是pycharm，然后用了terminal告诉我没有这个命令

好像对应是这个截图：




```
x=get_data()
'x' 不是内部或外部命令，也不是可运行的程序或批处理文件
```

回答你这句之前，先要给你解释清楚，即你先要明白，要搞清楚：

以及再需要去搞清楚：什么是系统的终端

以及：引申：编辑器和IDE的终端往往就是系统的终端

> 然后，我用cmd也不行

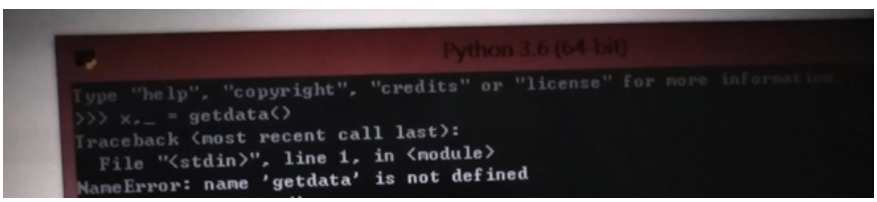
至此你也（应该）明白了，理解了，此 PyCharm的终端=系统的终端 不等于Python的shell终端

也就没法直接运行你的Python代码了。

然后再去搞懂对比：Python的shell对比系统的终端

> 然后用python3.6的那个终端，还不行

你说的应该是：



很明显，此处：

- 你不仅（粗心，笔误）把get_data()误写成getdata()
- 还不理解：Python中代码执行的基本逻辑

- get_data()是个函数的调用
  - 需要你先把函数实现=函数定义 写出来，或者是导入别的实现了此函数的库
    - 然后才能去调用此函数，运行此函数

在具体点说就像是，你能运行：

```
x, _ = get_data()
```

的前提是，前面已有类似get_data()的具体实现，类似于：

```
def get_data()
    ...
    return x, y
```

然后你才能调用：

```
x, _ = get_data()
```

否则，当然，也就会报错：

```
NameError: name 'getdata' is not defined
```

中文翻译为：

名字错误：getdata这个（变量或函数的）名字没有定义

> 之前，也遇到过这种状况，就是terminal必须弄一个虚拟环境才能运行整个库

其实你说的，大意是对的，但是细节上的逻辑还是不严谨的

其实不是 “弄一个虚拟环境” 才能运行整个库

至少对于前面这几行要测试的代码，则只是：

先要确保你的get_data()函数有定义了，你已经在Python的shell中输入了get_data()的定义了，或者导入了相关的库，然后

```
x, _ = get_data()
```

这行代码才能正常运行

类似的后续代码，也都要：对应函数有定义，才能正常运行。

> 我很无奈，就是想问问如何使用终端生成这个东西!

就是按照我前面所说的：

找到get_data()等函数的定义，输入到终端中

（且确保当前Python环境是OK的，即相关依赖的库已安装等）

然后就可以在终端中（输入代码，运行代码）生成（你要的）这个东西了。

不过话说关于Python入门开发的IDE的选择，我多年前就不推荐用IDLE：

小白用户，**不要用python的shell或IDLE**，而是去**用编辑器或IDE去开发Python**

详见：

- 4.1.4.1. 对初学者的建议：如何选用Python的开发环境
  - 注
    1. 可以把
       - > Windows的cmd + Notepad++
       - 改为：**Windows的cmd + VSCode**
    2. 其中关于VSCode开发Python，可以参考最新的：
       - 调试Python · 史上最好用的编辑器：VSCode

以及不了解开发方式的小白，可参考我之前写的：

python初级教程：入门详解

中的：

- 4.1.1. Python的最原始的开发方式是什么样的
- 4.1.2. 利用Python的shell进行交互式开发又是什么样的
- 4.1.3. 利用第三方Python的IDE进行Python开发又是怎么回事

看完后，相信你会对Python开发的编辑器，IDE等概念有个更加充分的认识。

之后就不会出现此提问者这种对于概念混淆的问题了。

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-08-04 19:44:10

Python的shell和系统的终端

然后总结关于 `系统终端` 和 `Python shell` 的内容。

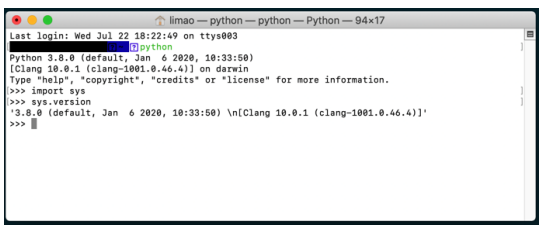
希望看完后，小白新手能搞清楚：

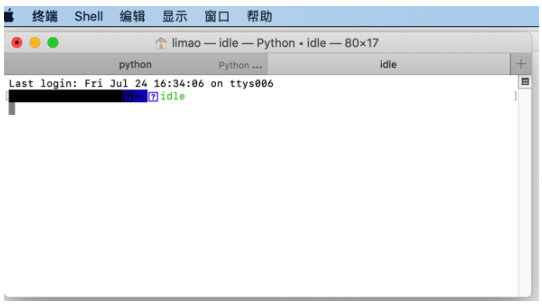
- **Python shell 不是 系统终端**

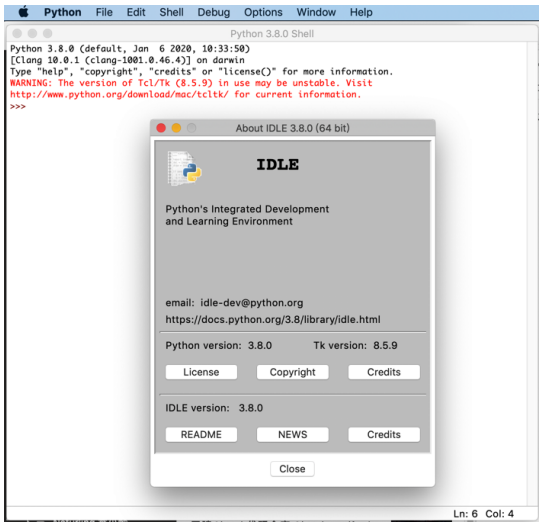
以及相关的概念和逻辑：

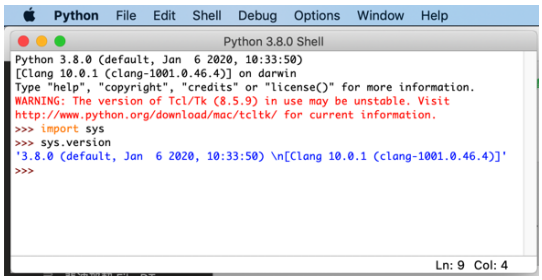
- 什么是 系统的终端中运行Python shell
- 不能在系统中的终端中运行Python代码
  - 应该在Python shell中运行Python代码
- 不能在Python shell中运行系统命令
  - 应该在系统终端中运行系统命令

什么是Python的shell

- Python的shell
  - 是什么：交互式的，一个命令行的界面
  - 干什么：
    - 供你输入代码，测试代码用的
      - 你所输入的代码，往往是很少的代码片段
        - 比如：一行或几行的Python代码
        - -》很少有一次性输入很多很多行的Python代码
          - 对于多行的、非常多的、大型的的Python代码，则往往保存到独立的（后缀是.py的）Python文件中，再去用专业的开发工具去测试和运行
            - 比如PyCharm或VSCode等Python的编辑器或IDE
        - 所以：往往第三方其他的库，才用Python的shell去演示基本的用法
          - 比如：你这里的人工智能方面的Python测试代码，
  - 特点
    - 安装了Python提供的，即安装了Python就自带的
  - 如何启动和长什么样
    - 先说版本：
      - 不带图形界面的，纯文字的，命令行的效果，叫：`python shell`
      - 基于图形界面的，叫做：`IDLE`
    - 再说如何启动：
      - `Python shell`
        - 直接在终端中运行python，即可进入 `Python shell`
          - Mac
            - 

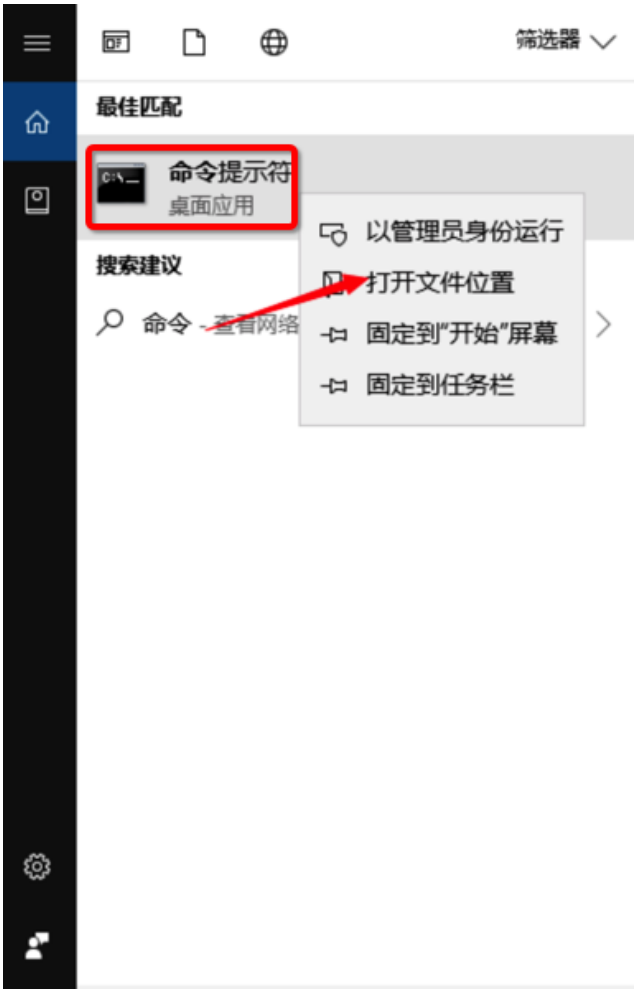
- `IDLE`
  - Mac
    - 通过点击或运行IDLE才能启动
      - 

      - ->
      - 

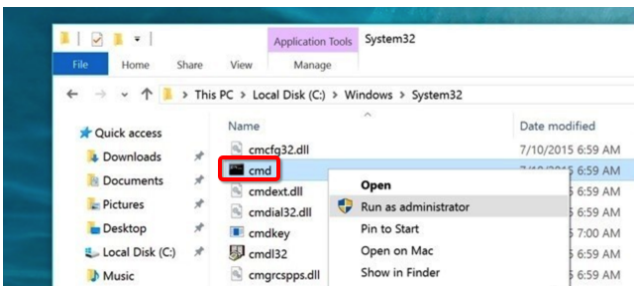
      - 

什么是系统的终端

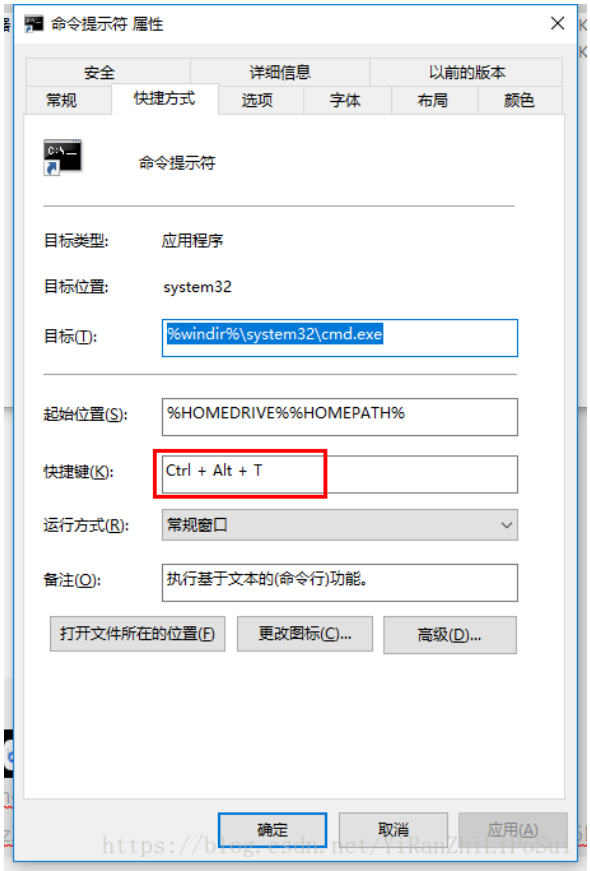
- 系统的终端：
  - 针对不同系统
    - Windows中，默认的是cmd=命令行=命令提示符
      - 长这样

- 

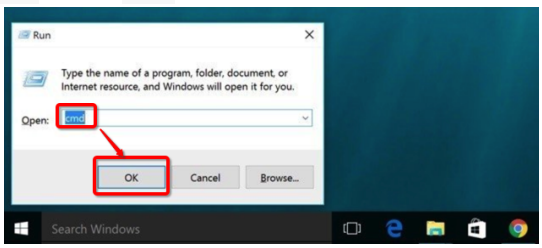
- 英文版
-

- 对应cmd.exe这个文件：
- 

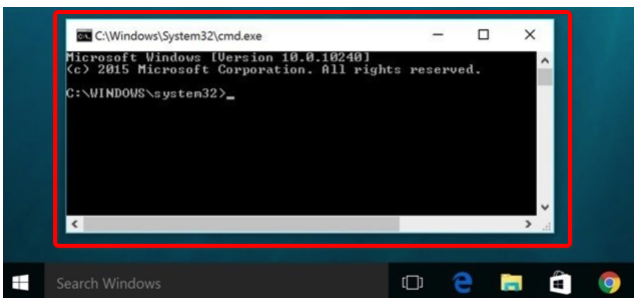
- 文件属性：



- 如何启动=运行：
  - 2种方式
    - 点击上述 命令提示符
    - `Wind+R` -> 输入 `cmd` -> 回车



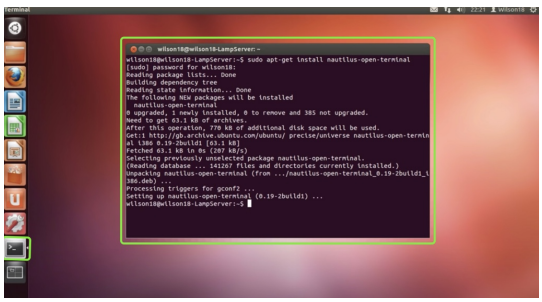
- 启动=运行后长这样



- Mac中的Terminal
  - 长这样

- 启动=运行后长这样

- Linux中的shell
  - 各种发行版
    - Ubuntu
      - 启动后，长这样：



引申：编辑器和IDE的终端往往就是系统的终端

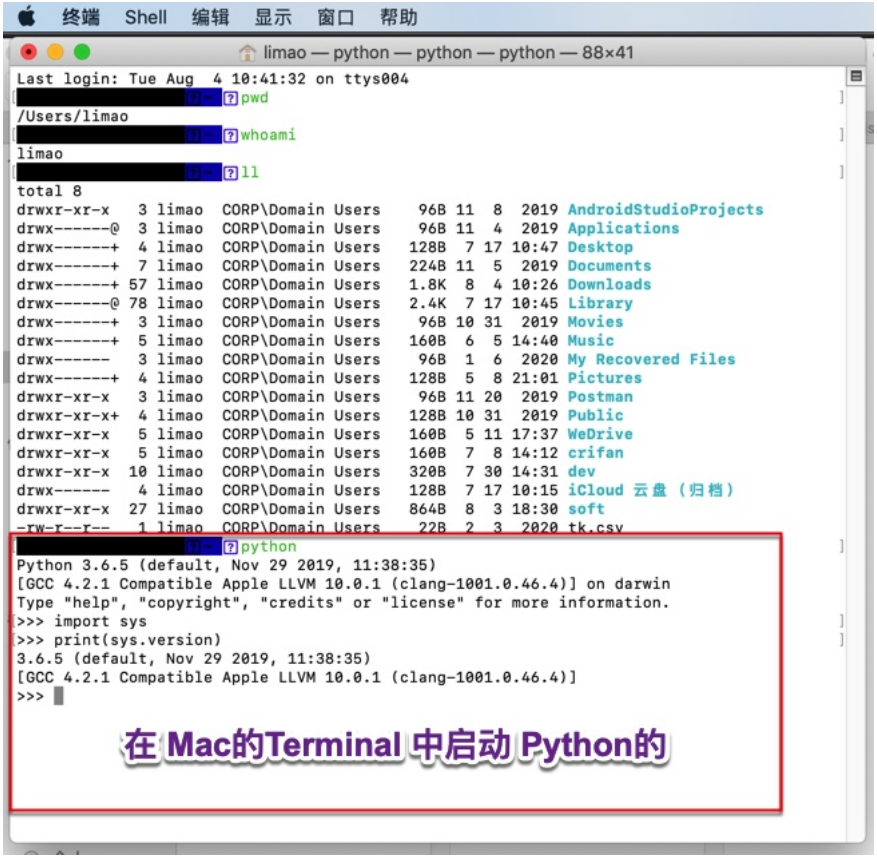
对应的，很多编辑器和IDE，比如 `VSCode` 、 `PyCharm` 等，其中的 `终端` 往往是用的就是系统的终端。

举例：

- VSCode中的终端
- PyCharm中的终端
  - PyCharm中的Terminal（往往）是系统的终端
    - 往往就是系统自带的终端，即win的cmd或Mac的Terminal等

引申：在系统终端中启动Python shell

所以一种常见的启动Python shell的方式就是，在系统终端中输入 `python` ，就可以方便的启动python shell：



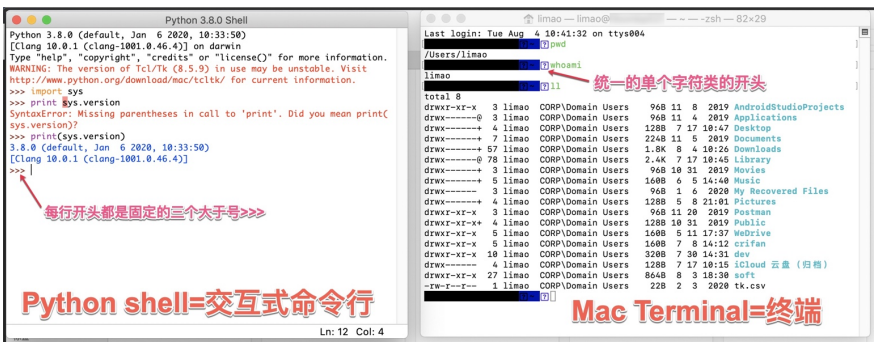
然后继续在 `python shell` 中写（一般是小段的）代码，去测试代码逻辑是否运行正常。

Python的shell对比系统的终端

关于：

- Python的shell
- 系统的终端

先贴出截图对比说明：



再详细解释：

- `Python的shell` ，是Python给你提供了一个**环境**，可以用来执行，运行（你所输入的）Python代码的环境
  - 即：其底层已经有个Python的**解释器**在运行

        - 因此才能实现，你输入了Python代码，回车后，就可以运行代码，看到运行后的结果了
    - 长什么样：每一行的开头往往是 三个大于号 `>>>`
- （`Win` / `Mac` / `Linux` 等）系统的终端，只支持系统的命令，而无法直接支持，直接能运行你所输入的Python代码
    - 长什么样：每一行的开头，往往是 单个字符或符号：井号 `#` / 大于号 `>`
    - 所以如果输入系统不存的命令：
        - 举例
            - `x=get_data()`
            - 就会报错：
            - `'x' 不是内部或外部命令，也不是可运行的程序或批处理文件`
                - 而这种错误，其实对于初学者也很常见
                - 就是因为连基本的系统终端和系统支持的命令等基础概念都不清楚，所以常会犯此初级错误
    - 那哪些才是系统支持的命令呢?
        - 比如
            - `Win` 中的
                - `dir` ：列出当前目录中的内容
                - `cd xxx` ：切换到xxx目录
                - 等等
            - `Mac` / `Linux` 等 中的
                - `ls` ：列出当前目录中的内容
                - `pwd` ：显示当前目录所在路径
                - 等等

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-08-04 19:44:26

字符编码问题

疑问：既然UTF-8编码更省空间 为什么又要换成unicode 换来换去

解答：没有换来换去，现在就是用的UTF-8。而UTF-8就是Unicode的其中一种。最通用。

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:30:01

文件顶部注释

对于Python文件顶部的注释，此处整理相关常见疑问和问题。

疑问：coding utf-8 注释

- **问**：
  - 看到很多Python代码中的第一行（或第二行）都有个
    - `# -*- coding: utf-8 -*-`
    - 这一行啥意思 是每个程序都要用吗？
  - 既然#号后面内容不能读取 那么这个指令岂不是没用
- **答**
  - 含义
    - 简答
      - 指定当前（文件）的（字符）编码
      - python解析器才能知道以哪种编码方式去识别你输入的文字 字符
      - 这样 如果是你代码中包含了普通英文字符（ASCII编码）之外的，比如中文字符等，就能准确识别了
      - 否则就会报错了
        - 细节
      - python解释器会专门解析#开头的第一行和第二行，比如如下这种最常见的格式：

        ```
        #!/usr/bin/python
        # -*- coding: utf-8 -*-
        ```

      - 就会解析得到
        - python解释器的位置是：`/usr/bin/python`
          - 就会去用`/usr/bin/python`这个解释器去解析代码
        - （当前）文件编码是：`utf-8`
          - 以`utf-8`的编码方式去加载python文件
    - 详见：
      - 【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 – 在路上
  - 是否一定要有：不一定
    - -》但是往往都最好有
      - -》尤其是你代码包含非ASCII字符，即除了英文外还有其他的如中文，日文，拉丁文等等等等的字符，一定要有

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:29:55

文件打开问题

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:29:51

代码缩进问题

代码缩进问题，导致代码运行了，以为没运行

问题

求助！！！正则表达式是正确的，但是程序无法运行-CSDN论坛

```
import requests
import json
import re
import time
from requests.exceptions import RequestException
url ='http://www.24timemap.com/'
def get_one_page(url):
    try:
        headers={
            'User-Agent':'Mozilla/5.0(Macintosh;Intel Mac OS X 10_13_3)AppleWe
        }
        response = requests.get(url,headers=headers)
        if response.status_code ==200:
            return response.text
        return None
    except RequestException:
        return None
def parse_one_page(html):
    pattern = re.compile('<li.*?bg.*?title.*?>(.*?)</a>(.*?)</li>',re.S)
    items = re.findall(pattern,html)
    for item in items:
        yield {
            'locantion': item[0],
            'time': item[1]
        }


def write_to_file(content):
        with open('slw.txt', 'a', encoding='utf-8')as f:
            f.write(json.dumps(content, ensure_ascii=False) + '\n')
```

解答

简答：不懂代码缩进问题，以为代码没运行，其实正常运行了

详解

> 正则表达式是正确的，但是程序无法运行

的确，你的 代码 中 正则（或许）是正确的

而你说的 无法运行，其实：已经运行了

-》已经按照你给的Python的代码 运行了

-》运行结果就是：啥都没干

-》为何啥都没干，是因为你没写代码让其干活

-》具体说就是：你只定义了3个函数（其中一个函数中有你说的正则），但是却没调用（其中任何一个函数）

所以变成你说的：`代码无法运行`

如何让其运行?

比如代码最后加上

```
pageHtml = get_one_page(url)
respGenerator = parse_one_page(pageHtml)
for eachItemDict in respGenerator:
  write_to_file(eachItemDict)
```

就可以实现：

真正调用函数 执行内部的代码和逻辑 -》变成你说希望的：代码真的运行了。

另外，关于代码缩进 导致你代码运行了 但你以为没运行，详见：

- 【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 – 在路上
- 5.4. Python中的缩进 - python初级教程：入门详解

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:29:48

如何学会提问

学习技术的期间，肯定会遇到很多问题，以及去网上各种论坛等地方去问问题。

比如国内常见的很多技术论坛，尤其是做的相对还算不错的（但其实都不太好），比如 `csdn`

然后作为初学者，想要别人帮你，回答你的问题，首先要学会如何提问。

鉴于很多国内初学者，连基本的提问都不会，所以下面详细解释如何学会提问。

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:29:42

把问题要素描述清楚

如何学会提问？其实是个很大的话题

此处简略概述一下：

1. 文字描述出问题的背景和现象
    - 主要是，除了问题的具体现象外，还要说一下自己的当前的环境，包括但不限于：
        - 电脑类型：windows还是mac，还是Linux?
        - 所用语言和库的版本：Python 2还是3？具体是哪个版本？用的第三方库的版本是多少?
2. 对于代码的话，要贴出来出错的相关的代码，最好是完整的代码
3. 最好给出附带信息，尤其是截图

即提问之前，先要把自己的问题的要素描述清楚，完整的信息包括：

- 背景信息
    - 系统类型
        - Win
        - Mac
        - Linux
            - 具体哪个发行版?
                - Ubuntu
                - CentOS
                - 其他?
    - 什么开发语言
        - Python
        - Java
        - 其他?
    - 某个开发语言的某个库?
        - 举例：
            - Python的网络库Requests
    - 具体的版本
        - 举例
            - CentOS
                - CentOS 6.4
                - CentOS 7
            - Python
                - Python 2.7
                - Python 3.8
- 具体信息
    - 错误：
        - 最基本要有：文字描述
            - 且要描述清楚，不要模糊不清
        - 最好也有：
            - 截图
            - 相关代码

- 且带语法高亮的代码：更易读

关于贴出格式化后高亮的代码

比如，作为对于问题提问者贴出来的代码：

和：

- 问：你作为想要回答问题的人，你会喜欢哪个？

- 答：很明显是第二个
  - 因为代码加了语法高亮
    - 更容易看清楚代码的格式和逻辑
      - 否则还要回答的人去花精力去看懂和研究你的代码
        - 别人回答你的问题的意愿就降低了
          - 即：浪费别人的时间就是浪费你自己的时间

另外，对于代码缩进：

尤其是对于Python语言来说，Python代码中的缩进，不仅是美观问题，还影响代码层次和逻辑。缩进错了，代码逻辑可能也错了。

所以结论就是：

- 帮助别人就是帮助自己
  - 提问时给出问题的详细信息
    - 包括
      - 截图
      - 格式化好（高亮）的代码
      - 等
  - 就是给别人省时间
    - 也就是给自己省时间

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:29:38

学会贴图和贴代码

下面简要列出核心内容：

- 国内主流技术论坛：如何贴图和贴代码
  - csdn
    - bbs论坛
      - https://bbs.csdn.net
  - 博客园
    - 博问
      - https://q.cnblogs.com
  - TODO：找到更多国内技术论坛
- 国外的
  - SegmentFault
    - https://segmentfault.com/questions
  - GitHub的issue

下面分别详细介绍。

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:29:10

csdn的bbs论坛

csdn的bbs论坛中提问题时如何贴代码

而关于贴代码到论坛中的问题，很多小白不会用论坛中的编辑器，不会贴代码。

下面把：

【求助】在学Python编程从入门到实践，继承这章添加Battery类后原来的子类跑不了了-CSDN论坛

回复别人的内容整理如下，作为解释：csdn的bbs论坛中提问题时如何贴代码 的回答：

csdn中粘贴代码的核心逻辑是：

确保输入的内容是符合下面这种格式：

```
[code=python]此处粘贴你的代码即可[/code]
```

即可，此时

- 发布后的回复中
- 发布出来的问题帖子中
- 发布之前，点击预览

就是带彩色的，代码高亮后的，效果好看的代码了

对于如何让代码变成上面这种格式，有多种操作方式：

方式1：自己手动输入

其实就是普通的文字，文本，自己输入：

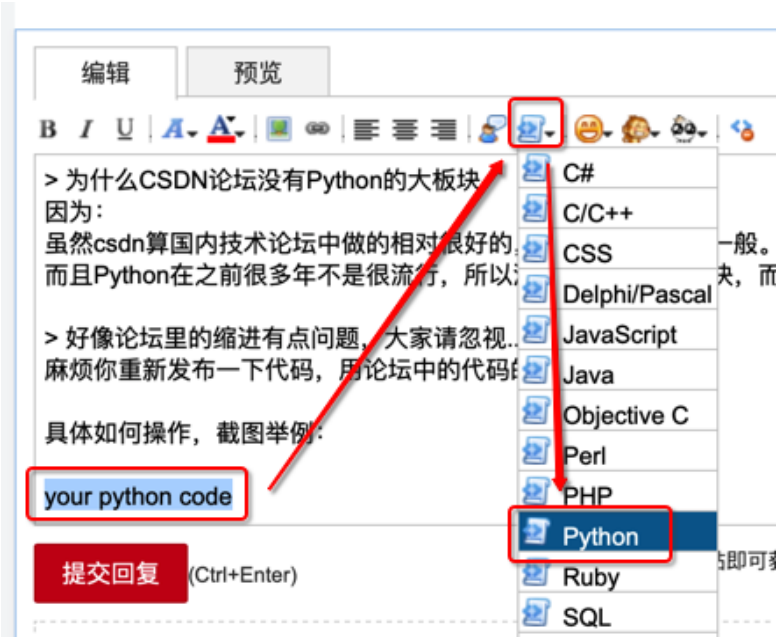
```
[code=python][/code]
```

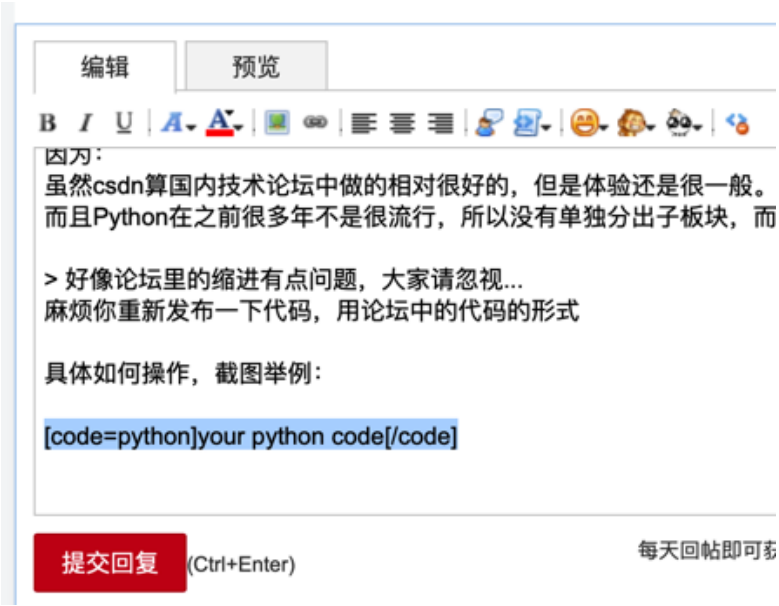
然后把你要代码放进去，即可。

通过编辑器自动插入

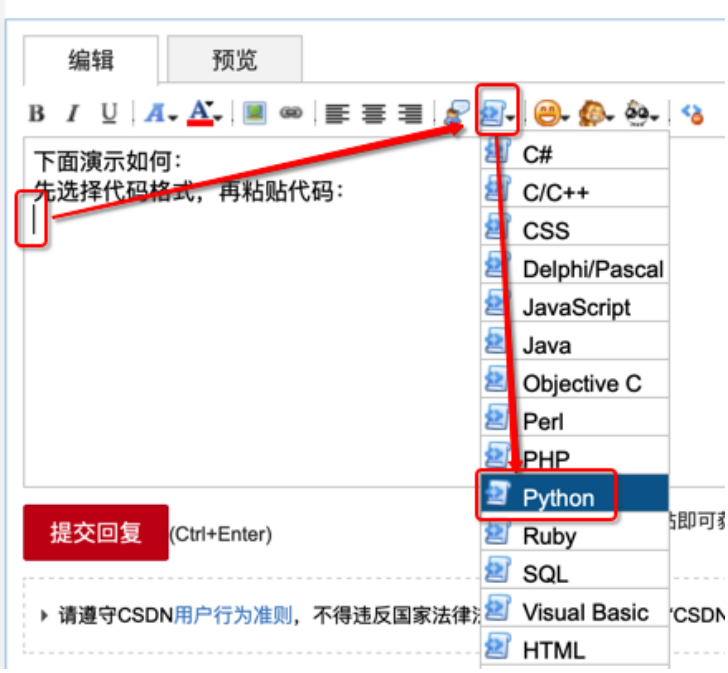
又分2种：

- 方式1：选中代码 -> 选择格式 -> 选择某个语言（比如Python）





- 方式2：鼠标点击要插入代码的位置 -> 选择格式 -> 选择某个语言（比如Python）->再粘贴代码进去
  - 鼠标定位到要粘贴代码的位置，然后去选择：代码->Python



- 会自动输入：
  - `[code=python][/code]`
- 再粘贴代码到里面：
  - 比如：

```
[code=python]
import sys
print("sys.executable=%s" % sys.executable)
[/code]
```

至此，就能看到代码高亮的效果了：

- 发布前的预览

-

- 真正发布后
  -

这样就可以保留原始代码的格式

而带缩进的代码也就不会乱了。

且显示起来带颜色，即代码高亮，更易读。

别人（想要帮你，回答你问题的人）也容易**拷贝出完整代码**，帮你测试代码，找到问题原因。

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-08-04 09:57:52

附录

下面列出相关参考资料。

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:25:26

print字符串格式化语法

此处整理关于 `print` 中变量的格式化的语法：

Python官网

内置类型 — Python 3.8.2rc2 文档

转换符	含义	注释
'd'	有符号十进制整数	
'i'	有符号十进制整数	
'o'	有符号八进制数	(1)
'u'	过时类型 -- 等价于 'd'	(6)
'x'	有符号十六进制数（小写）	(2)
'X'	有符号十六进制数（大写）	(2)
'e'	浮点指数格式（小写）	(3)
'E'	浮点指数格式（大写）	(3)
'f'	浮点十进制格式	(3)
'F'	浮点十进制格式	(3)
'g'	浮点格式。 如果指数小于 -4 或不小于精度则使用小写指数格式，否则使用十进制格式	(4)
'G'	浮点格式。 如果指数小于 -4 或不小于精度则使用大写指数格式，否则使用十进制格式	(4)
'c'	单个字符（接受整数或单个字符的字符串）	
'r'	字符串（使用 repr() 转换任何 Python 对象）	(5)
's'	字符串（使用 str() 转换任何 Python 对象）	(5)
'a'	字符串（使用 ascii() 转换任何 Python 对象）	(5)
'%'	不转换参数，在结果中输出一个 '%' 字符。	

- 注释
  1. 此替代形式会在第一个数码之前插入标示八进制数的前缀 ('0o')。
  2. 此替代形式会在第一个数码之前插入 '0x' 或 '0X' 前缀（取决于是使用 'x' 还是 'X' 格式）。
  3. 此替代形式总是会在结果中包含一个小数点，即使其后并没有数码。
     - 小数点后的数码位数由精度决定，默认为 6。

4. 此替代形式总是会在结果中包含一个小数点，末尾各位的零不会如其他情况下那样被移除。
    - 小数点前后的有效数码位数由精度决定，默认为 6。
5. 如果精度为 N，输出将截短为 N 个字符。
6. 参见 PEP 237。

菜鸟教程

Python3 print 函数用法总结 | 菜鸟教程

python字符串格式化符号:

符号	描述
%c	格式化字符及其ASCII码
%s	格式化字符串
%d	格式化整数
%u	格式化无符号整型
%o	格式化无符号八进制数
%x	格式化无符号十六进制数
%X	格式化无符号十六进制数（大写）
%f	格式化浮点数字，可指定小数点后的精度
%e	用科学计数法格式化浮点数
%E	作用同%e，用科学计数法格式化浮点数
%g	%f和%e的简写
%G	%f 和 %E 的简写
%p	用十六进制数格式化变量的地址

格式化操作符辅助指令:

符号	功能
*	定义宽度或者小数点精度
–	用做左对齐
+	在正数前面显示加号(+)
	在正数前面显示空格
#	在八进制数前面显示零('0')，在十六进制前面显示'0x'或者'0X'(取决于用的是'x'还是'X')
0	显示的数字前面填充'0'而不是默认的空格
%	'%%'输出一个单一的'%'
(var)	映射变量(字典参数)
m.n.	m 是显示的最小总宽度,n 是小数点后的位数(如果可用的话)

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-07-09 19:25:20

参考资料

- 学习方法思路及技术心得总结
- 【提醒】Python新手开发人员注意事项：不要故意用错误的写法而应该用正确标准的写法
- 【提醒】Python新手开发人员注意事项：不要误输入中文标点符号 – 在路上
- 【提醒】Python新手开发人员注意事项：测试文件名不要和导入的库同名 – 在路上
- 【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 – 在路上
-
- 【已解决】mac中pip安装Python库tushare
-
- 关于Python pyecharts 的问题（已经找资料找了半天了）-CSDN论坛
- 30分钟学会pyecharts数据可视化 - 知乎
- Documentation - pyecharts-en
- 小白遇到一个关于Pycharm中 Tkinter问题-CSDN论坛
- Tk图形用户界面(GUI) — Python 3.8.2 文档
- tkinter --- Tcl/Tk的Python接口 — Python 3.8.2 文档
- tkinter — Python interface to Tcl/Tk — Python 3.8.2 documentation
- Python GUI 编程(Tkinter) | 菜鸟教程
- Python - Tkinter Label - Tutorialspoint
- Tkinter教程之Label篇*Python*灵蛇舞动-CSDN博客
- Label & Button 标签和按钮 - 窗口 Tkinter | 莫烦Python
- tkinter.ttk --- Tk主题小部件 — Python 3.8.2 文档
- 求助！！！正则表达式是正确的，但是程序无法运行-CSDN论坛
- 初学者，照着老师的编码打的一样，运行错误-CSDN论坛
- Qpython3-CSDN论坛
- 内置类型 — Python 3.8.2rc2 文档
- Tushare -财经数据接口包
- pycharm 导入tushare错误，请帮帮忙，百度半天也没搞定-CSDN论坛
- 关于tushare python的问题。-CSDN论坛
- AttributeError: module ‘tushare’ has no attribute ‘**version**’ · Issue #241 · waditu/tushare · GitHub
- 在策略模块定义函数引用tushare pro – VincentZHOU – JoinQuant
- Pygame has no attribute key-CSDN论坛
- 制作数据集时报错AttributeError: 'str' object has no attribute 'write'-CSDN论坛
- tf.data.experimental.TFRecordWriter | TensorFlow Core v2.1.0
- tf.io.TFRecordWriter | TensorFlow Core v2.1.0
- 大家觉得国内比较好的技术社区有哪些，优缺点是什么？ - 知乎
- 码农10年，我常去的一些技术社区 - 简书
- window中的cmd中设置别名(alias)及设置快捷键打开cmd_waywaywayw的博客-CSDN博客_windows alias
- 在 Windows 10中，打开命令提示的10方法*苹果iPhone*酷徒
- Windows 7系统下如何现实文件扩展名
-

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新： 2020-08-06 20:45:39