向文婷 1813088

2020/6/15 补充

1. Git操作

查看版本：查看历史记录 git log

查看状态：git status

工作区和暂存区：git add将文件添加到暂存区，git commit将暂存区的内容全部提交到当前分支。

版本回退：若修改后想退回到以前版本，命令git reset – hard HEAD^回退到上一个版本。Git reset –hard 版本号，回退到指定版本。

撤销修改：只在工作区修改，未添加到暂存区，git checkout –文件名，将该文件在工作区的所有修改全部撤销。添加到了缓存区git reset HEAD 文件名，将缓存区的修改回退到工作区。

删除文件：rm 文件名，将工作区没有的文件删除。Git rm 文件名，从版本库删除该文件，再rm 工作区的文件，再重新提交。

1. GitHub

分支：创建分支，切换到新分支 git checkout -b 分支名，创建并切换，git branch 查看当前分支，合并指定分支到当前分支 git merge 分支名。切换分支也可用 git switch -c 分支名。删除分支 git branch -d 分支名。

当发生冲突时，Git不能自动合并分支，必须先手动解决冲突再提交。试图合并时失败，会提醒哪些存在冲突。git log –graph可查看分支合并图。

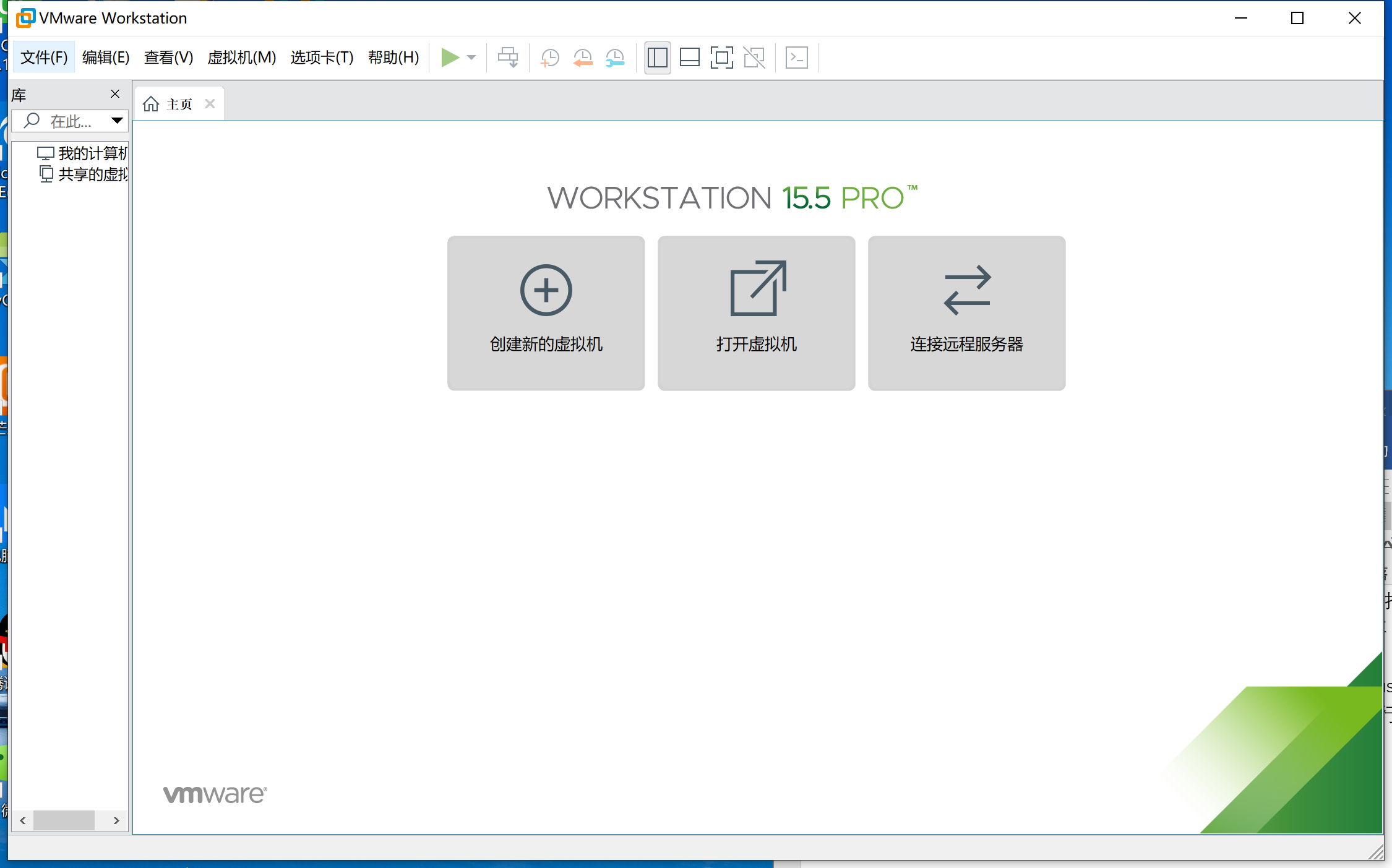
标签：发布一个版本时，通常先在版本库中打一个标签tag。方便取出被打标签的版本。打标签，先切换到需要打标签分支，再git tag 标签名；或git tag 标签名 版本号。可用git tag查看所有标签。git tag -d 标签名。推送标签git push origin 标签名。远端删除标签，先本地删除，再git push origin : refs/tags/标签名。

SourceTree：Git图形界面工具。下载，运行。添加本地Git库。方便提交、分支、推送等操作。

2020/6/16

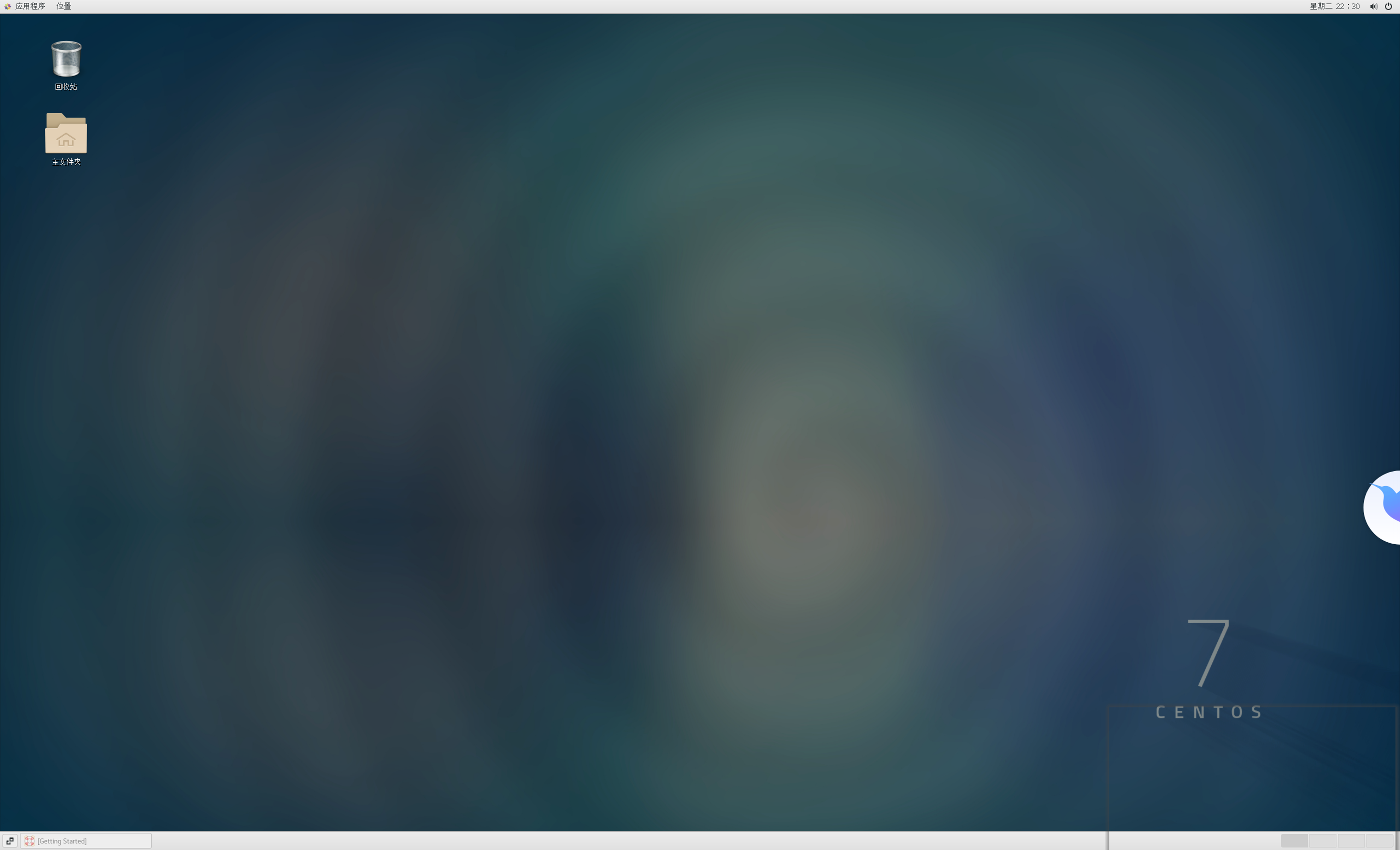
1. 安装虚拟机：

下载安装VMware



下载CentOS 7镜像

打开VM，创建新的虚拟机。



1. Linux：

基于POSIX和UNIX的多用户、多任务、支持多线程和多cpu的操作系统。

Ls列出目录及文件，ls -p查看文件属性、所属用户和组。文件类型；属主权限；属组权限；其他用户权限。d,目录文件；-文件；l链接文档；b可供储存的接口设备；c端口设备。

更改文件属性，chgrp：更改文件属组；chown：更改文件属主，也可以同时更改文件属组；chmod：更改文件9个属性。

cd：切换目录

pwd：显示目前的目录

mkdir：创建一个新的目录

rmdir：删除一个空的目录

cp: 复制文件或目录

rm: 移除文件或目录

mv: 移动文件与目录，或修改文件与目录的名称

用户和用户组管理：

useradd: 添加新的用户账号，useradd 选项 用户名

选项：-c comment 指定一段注释性描述。

-d 目录 指定用户主目录，如果此目录不存在，则同时使用-m选项，可以创建主目录。

-g 用户组 指定用户所属的用户组。

-G 用户组，用户组 指定用户所属的附加组。

-s Shell文件 指定用户的登录Shell。

-u 用户号 指定用户的用户号，如果同时有-o选项，则可以重复使用其他用户的标识号。

userdel：删除账号 userdel 选项 用户名 -r把用户主目录一起删除。

usermod: 修改账号， -l改为新用户名。

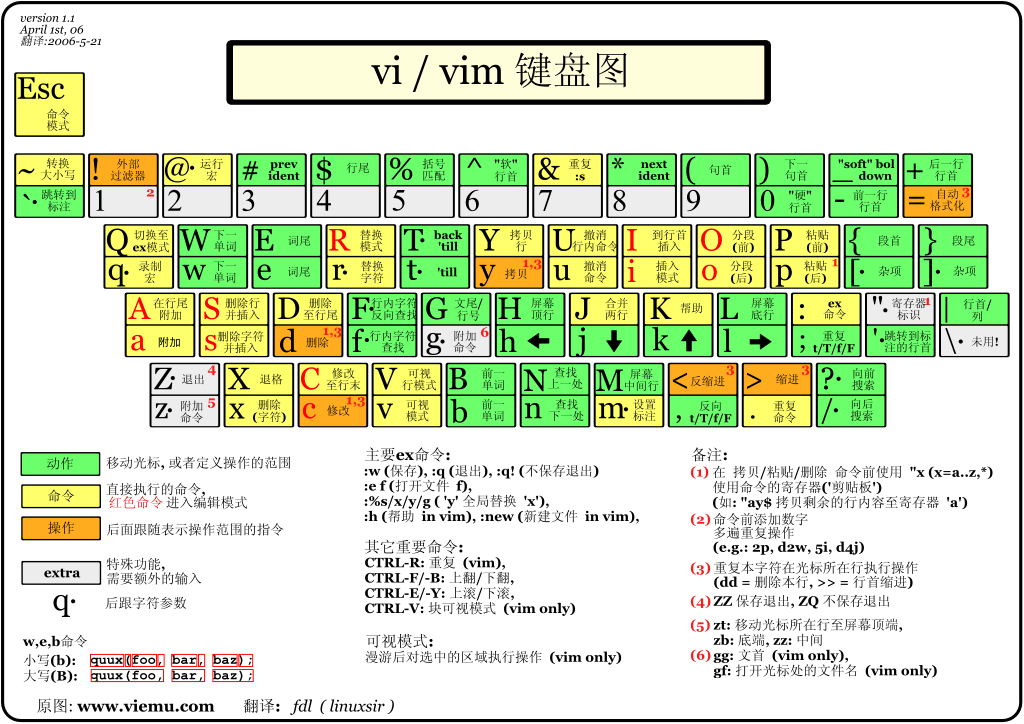
groupadd：添加新的用户组。-g 指定新用户组的组标识号。

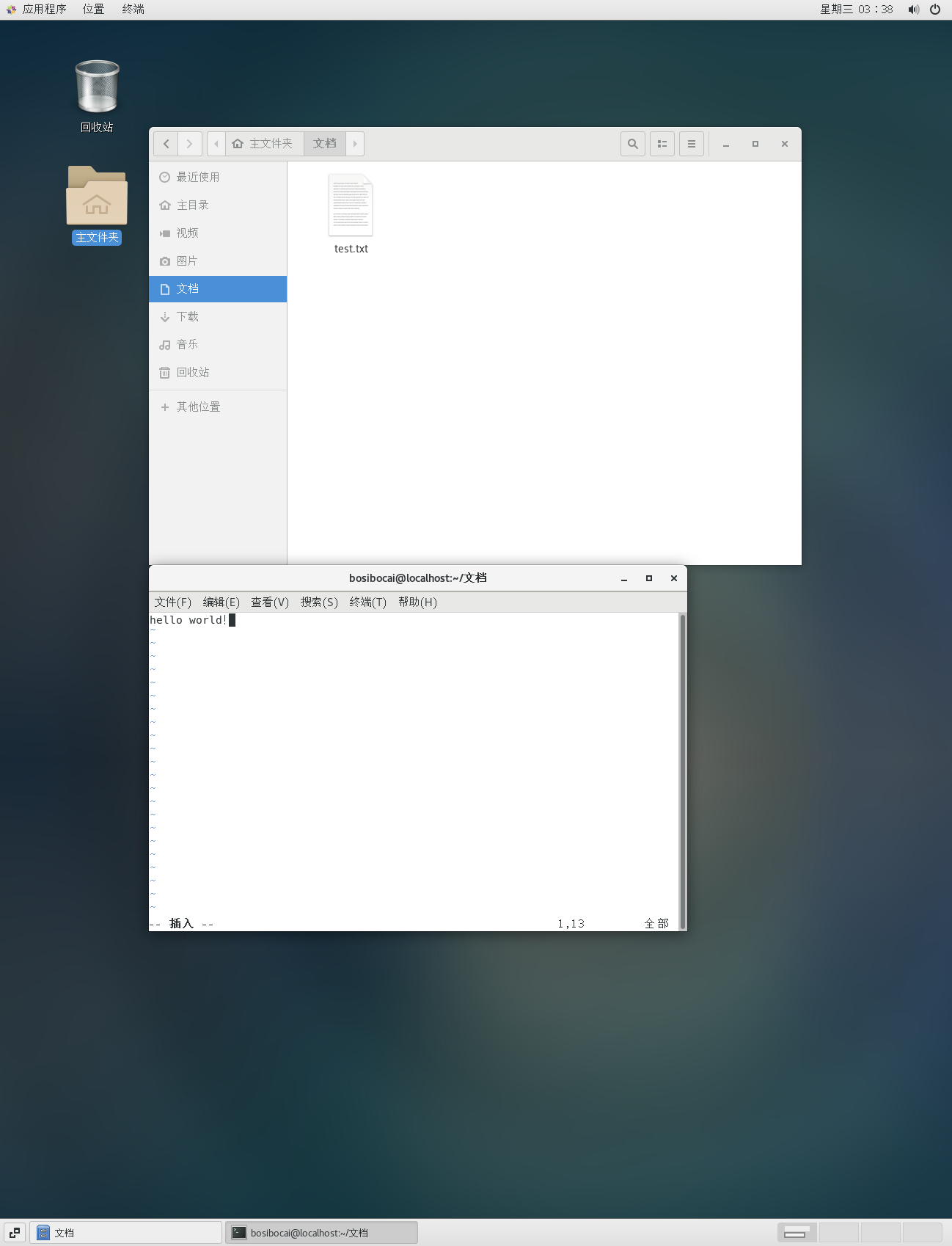
groupdel：删除用户组。

groupmod：修改用户组属性。

1. Vi/Vim编辑器:

Vi三个模式，插入模式、命令模式、底线命令模式。





1. rpm：

rpm软件包的管理工具。

选项：

-a：查询所有套件；

-b<完成阶段><套件档>+或-t <完成阶段><套件档>+：设置包装套件的完成阶段，并指定套件档的文件名称；

-c：只列出组态配置文件，本参数需配合"-l"参数使用；

-d：只列出文本文件，本参数需配合"-l"参数使用；

-e<套件档>或--erase<套件档>：删除指定的套件；

-f<文件>+：查询拥有指定文件的套件；

-h或--hash：套件安装时列出标记；

-i：显示套件的相关信息；

-i<套件档>或--install<套件档>：安装指定的套件档；

-l：显示套件的文件列表；

-p<套件档>+：查询指定的RPM套件档；

-q：使用询问模式，当遇到任何问题时，rpm指令会先询问用户；

-R：显示套件的关联性信息；

-s：显示文件状态，本参数需配合"-l"参数使用；

-U<套件档>或--upgrade<套件档>：升级指定的套件档；

-v：显示指令执行过程；

-vv：详细显示指令执行过程，便于排错。

-ql：查询

1. yum：

是一个在Fedora和RedHat以及SUSE中的Shell前端软件包管理器。提供了查找、安装、删除某一个、一组甚至全部软件包的命令。

yum 可选 操作 操作对象

常用命令：

列出所有可更新的软件清单命令：yum check-update

更新所有软件命令：yum update

仅安装指定的软件命令：yum install <package\_name>

仅更新指定的软件命令：yum update <package\_name>

列出所有可安裝的软件清单命令：yum list

删除软件包命令：yum remove <package\_name>

查找软件包 命令：yum search <keyword>

清除缓存命令:

yum clean packages: 清除缓存目录下的软件包

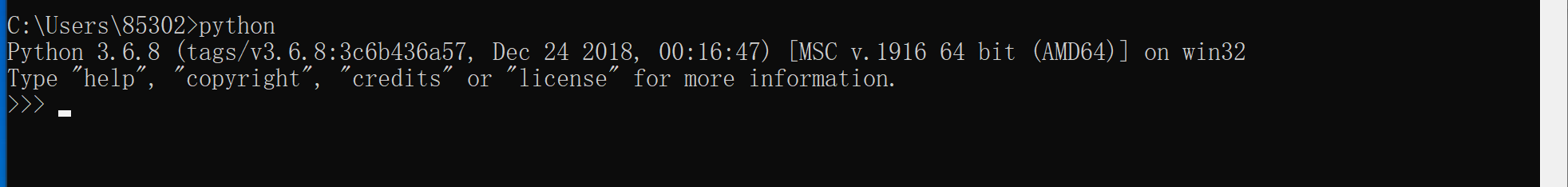
yum clean headers: 清除缓存目录下的 headers

yum clean oldheaders: 清除缓存目录下旧的 headers

yum clean, yum clean all (= yum clean packages; yum clean oldheaders) ：清除缓存目录下的软件包及旧的headers

2020/6/17

1. 下载安装python



1. Python

代码块不使用{}控制类，函数以及其他逻辑。用缩进来写模块。\可将一句语句分为多行。[]、()、{}内也为一条语句。

标识符，\_开头为不可直接访问的类属性 \_\_开头为类的私有成员 \_init\_结尾代表类的构造函数。

变量和数据类型：

变量赋值不需要类型声明。在使用前必须赋值。

五种标准数据类型，Numbers,String,List,Tuple元组,Dictionary字典。del语句删除引用。

Number包括int,long,float,complex。String从左向右从0开始，从右向左从-1开始。截取s[头下标:尾下标]。List标识符[]，切片，l[头下标:尾下标]，复制列表=list[:]而不使用切片时则会指向一个list，添加元素，append()添加到列尾，insert(索引，值)添加到列表中；删除元素del list[索引]，pop()删除列为元素，pop(索引)弹出任意位置元素；sort()永久性排序，sorted()临时排序，如sorted(list)为排序后，list仍为原顺序；reverser()反转list；len()长度，对数值列表range(开始,结束)但不包括结束。min(),max(),sum()进行统计计算。Tuple标识符()，元组不可修改元素，但可重新赋值，相当于只读列表。Dictionary，{}标记，list为有序对象的集合，字典为无序对象的集合。通过键来存取，’key’:’value’,添加/修改 dictionary[‘key’]=’value’，删除del dictionary[‘key’]。类型转化。

条件语句：if 条件判断：，and、or检查多个条件，某元素是否存在在列表中，if 元素 not in/in list，if-elif-else结构。对于列表 if list:若list不为空else：list为空

for语句：for 变量 in 范围： 缩进控制开始和结束 遍历字典 for key,value in dictionary.items()：，遍历字典所有键for key in dictionary.keys()：，遍历字典所有值for key in dictionary.values()：,嵌套{{,,},{,,},{,,},}和字典存储字典，存储list，key=list

while语句：while 条件：break退出循环，continue继续循环。While value in list:list.remove()。

用户输入：input()等待用户输入返回字符串，int()返回数字。

函数：def构造一个函数，形参实参，可以关键字值对传值，def describe\_pet(animal\_type, pet\_name):，describe\_pet(animal\_type='hamster', pet\_name='harry')，默认值def describe\_pet(pet\_name, animal\_type='dog'):，也可返回字典，禁止函数修改列表可以传切片，参数\*trup ，将所有参数存入trup元组中因此可以传递任意数量的参数，参数\*\*trup,未知传入参数类型时。匿名函数 lambda创造 只含有 一个语句 函数名=lambda 参数1，参数2：语句。导入模块，import 模块名 或 from 模块import 函数。as取别名，from 模块 import函数as别名。

类：创建类，如dog类，class dog(): def \_init\_()初始化函数。继承 class bigdog子类(dog父类):，super()父类构造函数，重写父类方法。导入模块，from模块import类，python标准库。

文件和异常：读取文件，with open(‘绝对/相对路径文件名’) as file\_object: contents=file\_object.read() print(contents) 会比原文件多一排空行，若想删除最后一行改为print(contents.rstrip())。逐行阅读for line in file\_object:。使用文件的内容，lines =file\_object.readlines() for line in lines:。写入文件，写入空文件open之后，file\_object.wirte(“”)，添加内容到文件，open时在文件后传入参数‘a’。异常，try-except代码块，else代码块。

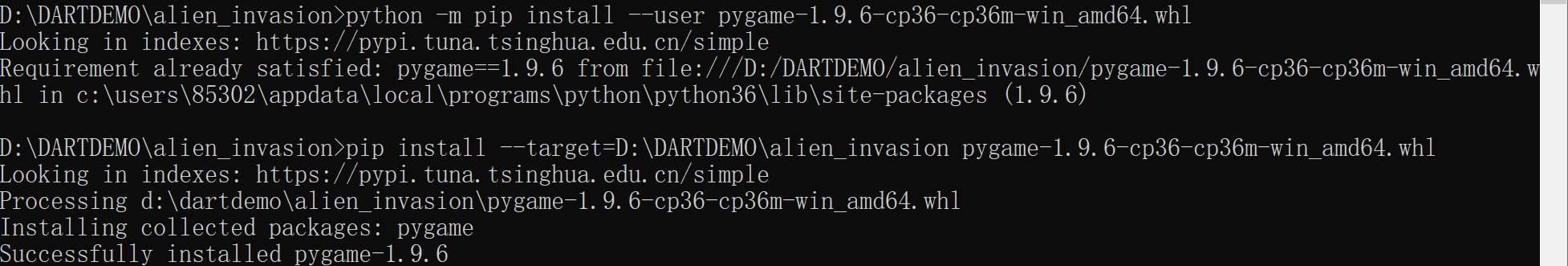
2020/6/18

外星人入侵：

在游戏《外星人入侵》中，玩家控制着一艘最初出现在屏幕底部中央的飞船。玩家可以使用箭头键左右移动飞船，还可使用空格键进行射击。游戏开始时，一群外星人出现在天空中，他们在屏幕中向下移动。玩家的任务是射杀这些外星人。玩家将所有外星人都消灭干净后，将出现一群新的外星人，他们移动的速度更快。只要有外星人撞到了玩家的飞船或到达了屏幕底部，玩家就损失一艘飞船。玩家损失三艘飞船后，游戏结束。

新建文件夹alien\_invasion

下载安装Pygame



1. 武装飞船：将创建一艘可左右移动的飞船，这艘飞船在用户按空格键时能够开火。

创建一个空的Pygame窗口。监听各种事件。设置背景窗口大小。

添加飞船图像。新建文件夹image，将飞船图像文件保存到这个文件夹中，绘制图形。

重构game\_functions，存储运行函数。管理事件check\_events（），更新屏幕update\_screen()

驾驶飞船：响应按键；允许不断挪动；左右移动根据移动标记调整飞船的位置；调整飞船的速度每次while循环时飞船最多移动的像素；限制飞船的活动范围；

射击：按空格时发射子弹，向上穿行，抵达上边缘消失。子弹类；将子弹存储到编组中；发射子弹，响应空格键按下；删除已消失子弹；限制子弹数量

1. 外星人：

创建外星人图像。

创建一群外星人，一行容纳多少个，创建多行。

移动外星人群，控制速度，添加方向，检查是否到达边缘。

射杀外星人，判断子弹是否击中外星人。

生成新的外星人群。

提高子弹速度。

结束游戏，判断外星人是否与飞船碰撞或是否到达屏幕底部。发生时游戏结束。

1. 计分：

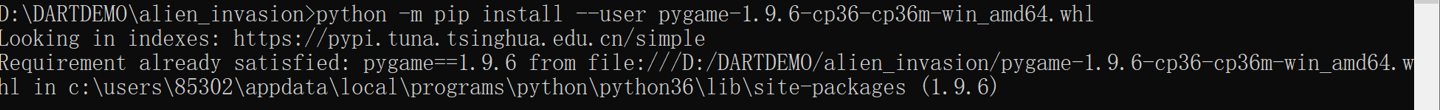
添加play按键，按下时游戏开始，并且按键消失。Button类，绘制按钮。

等级提高，速度增加，每消灭一群外星人后提高等级，加快速度。

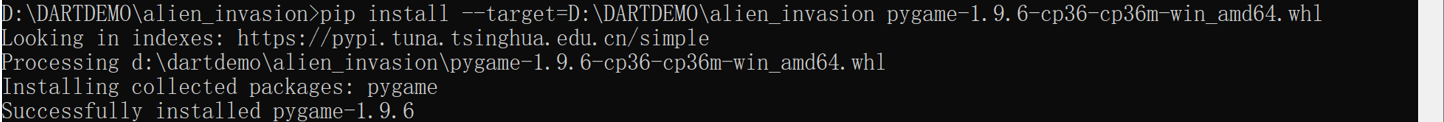
计分，跟踪玩家得分，显示最高得分、当前等级和飞船数。创建记分牌，显示在右上角，每次开始重置为0。每消灭一个外星人计分，随着等级的提高，每消灭一个外星人加分更多。记录所有玩家最高得分，有玩家超过时更新，在上方中间。显示等级，在右上角。显示属于飞船数，在左上角，绘制出来。

1. 问题与解决

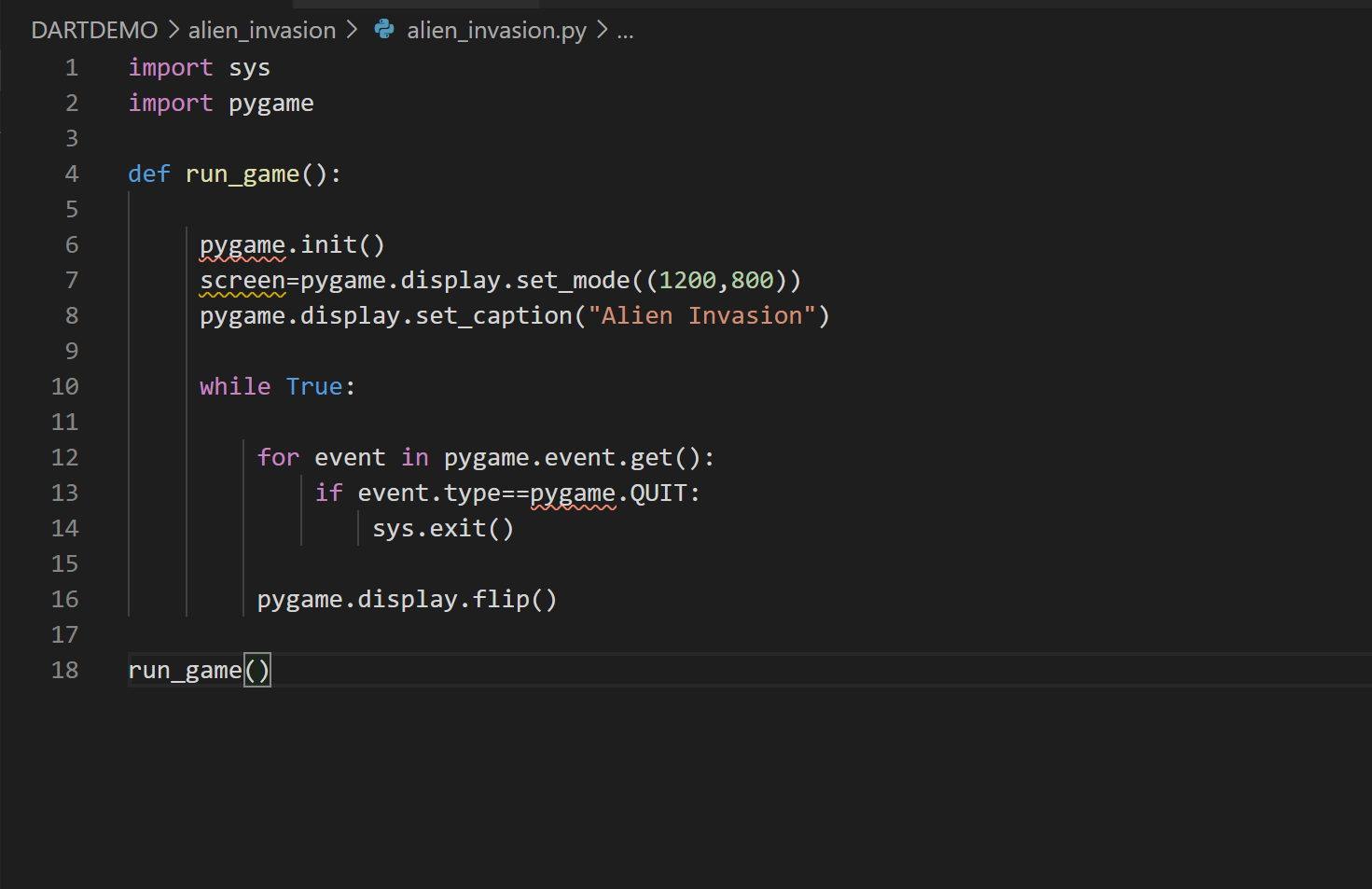
1、在安装pygame时，先打开项目文件



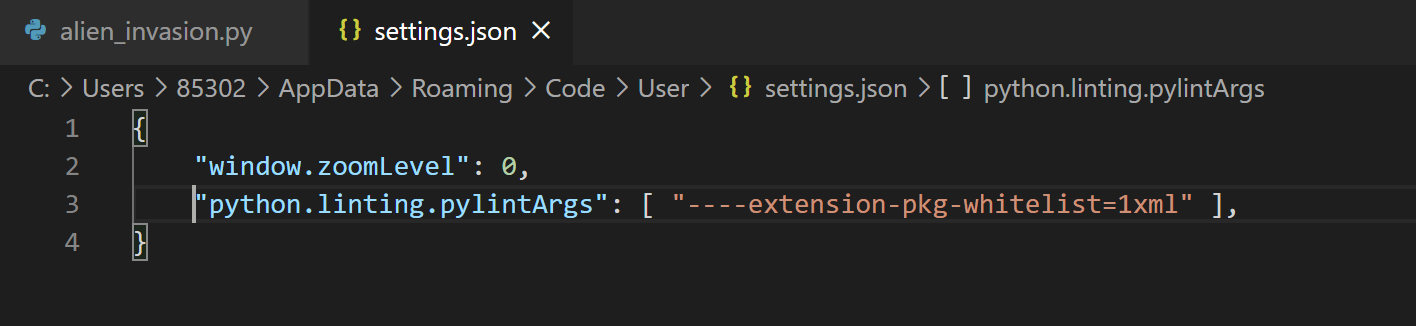
报错，应该指定安装路径



2、在VScord+python导入pygame模板时报错



它与Visual Studio Code按字面意思获取代码的事实有关，因为你无法导入pygame.init（），它认为它不是一个正确的模块。 要解决此问题，请打开settings.json（方法1：按ctrl+shift+p，搜索"setting"，选择“首选项：打开设置(json)”。方法2：文件->首选项->设置->搜索栏搜索"python"->点击“在setting.json中编辑”。）并粘贴"python.linting.pylintArgs": [ "----extension-pkg-whitelist=1xml" ],



错误消失

2、加载图片时出现了错误Couldn't open images\ship.png

将加载语句改为

self.image=pygame.image.load(r'D:\DARTDEMO\alien\_invasion\images\ship.png').convert\_alpha()

加r和绝对路径即可加载

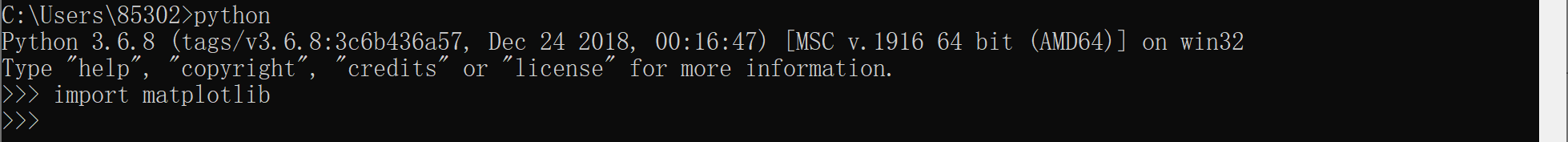
2020/6/19

数据可视化：通过可视化表示来探索数据

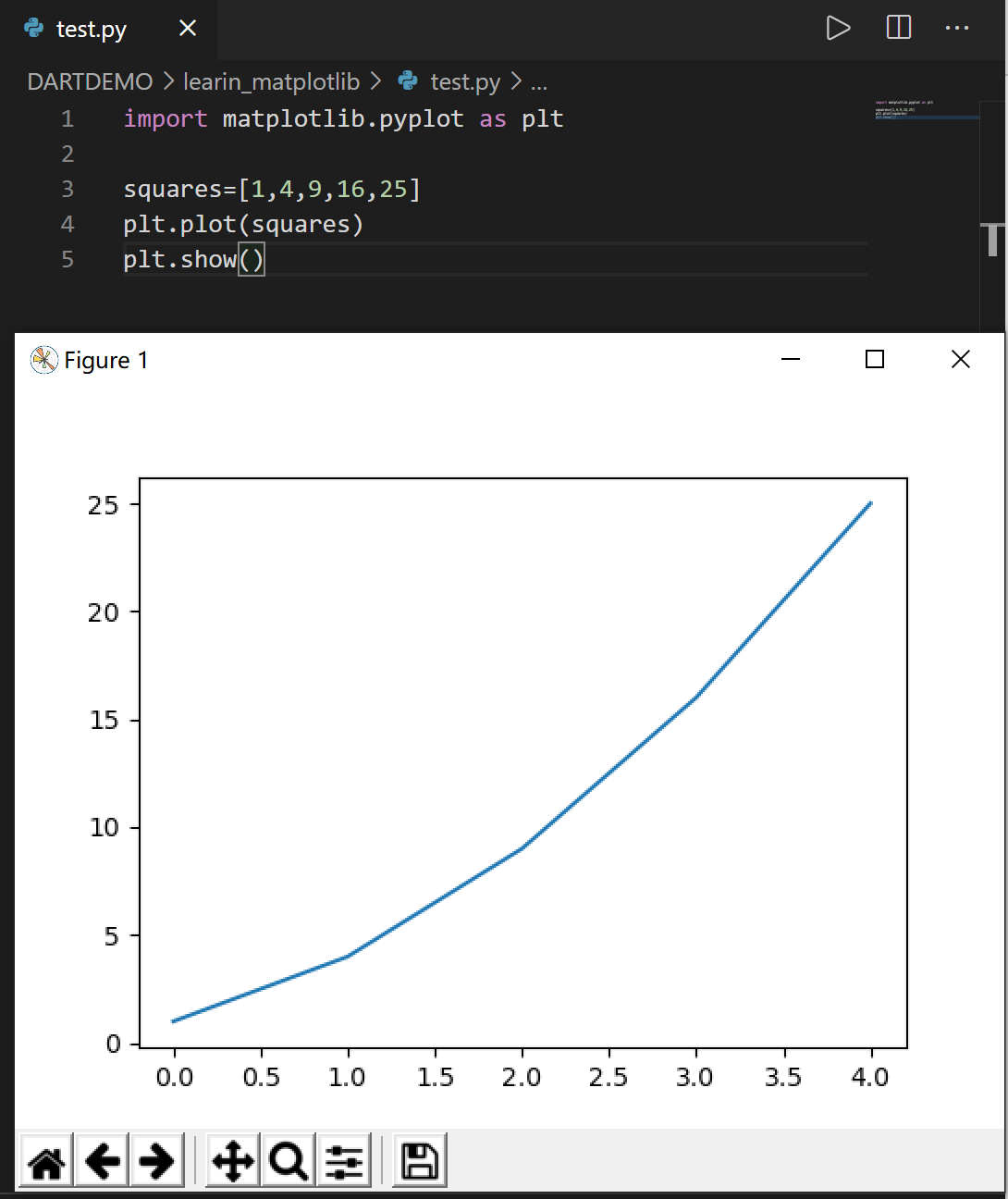
1. 安装matplotlib

进入项目文件夹，运行

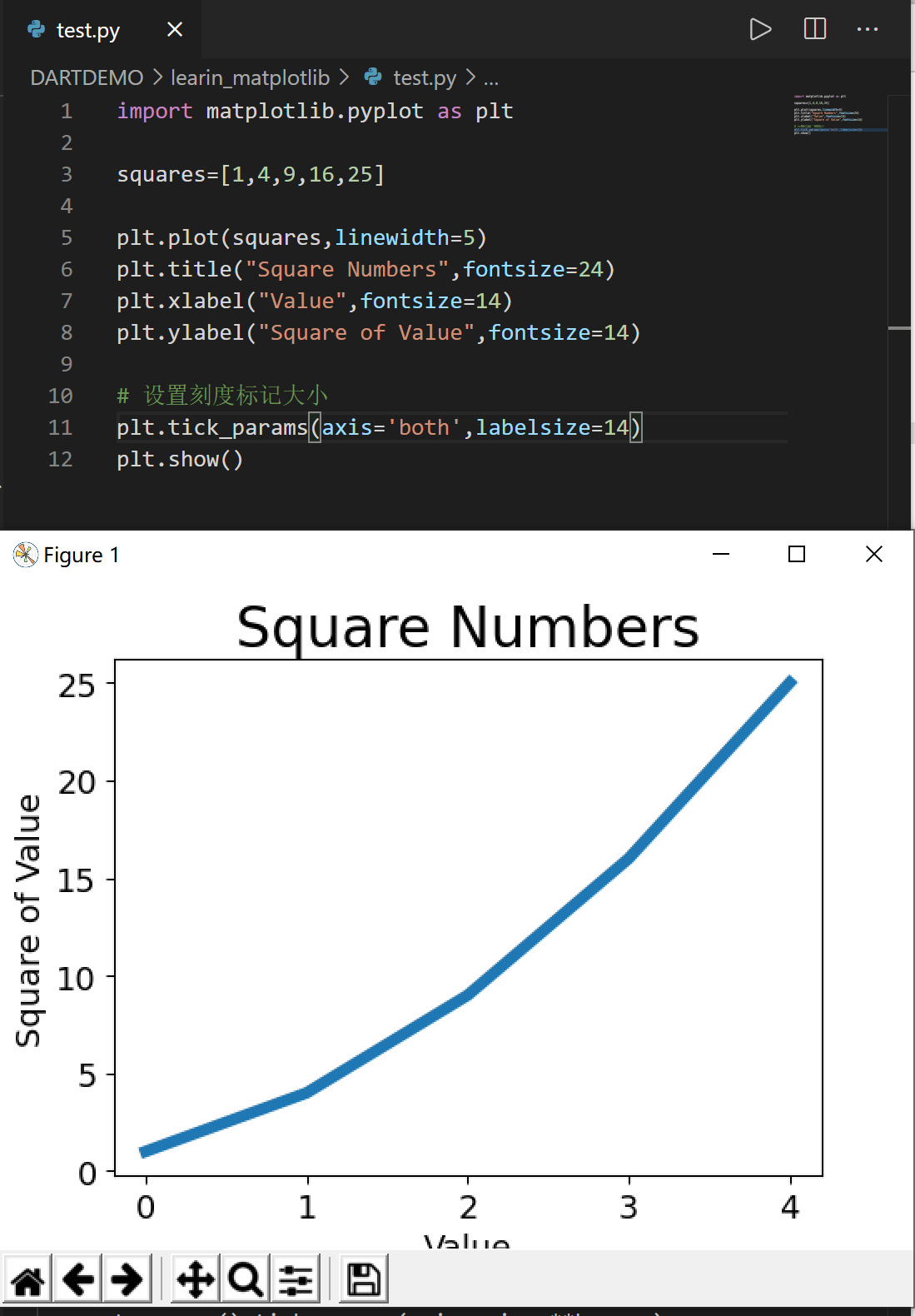
pip install --target=D:\DARTDEMO\learin\_matplotlib matplotlib， 安装matplotlib，安装成功



1. 绘制简单折线图：

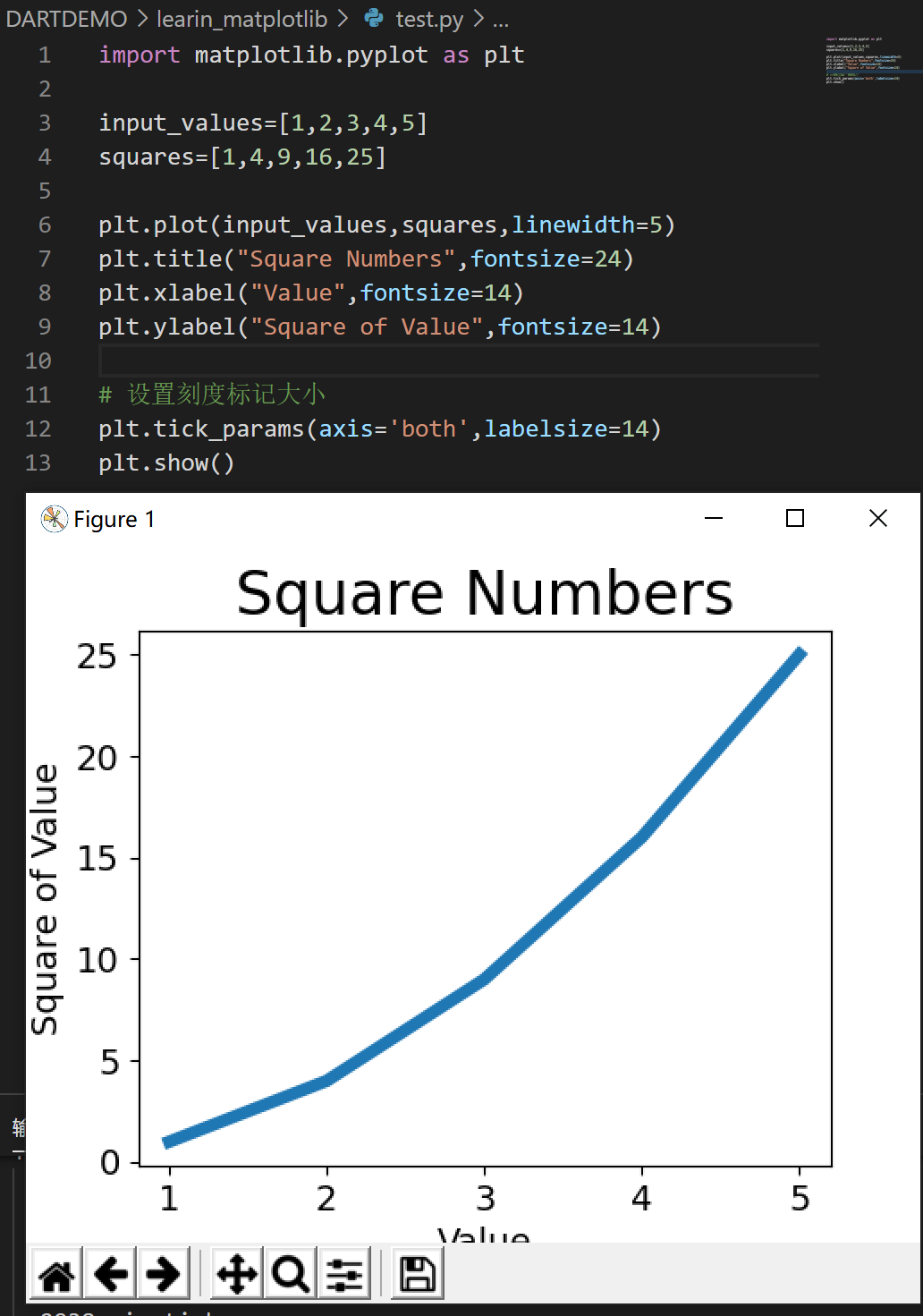


修改标签文字和线条粗细

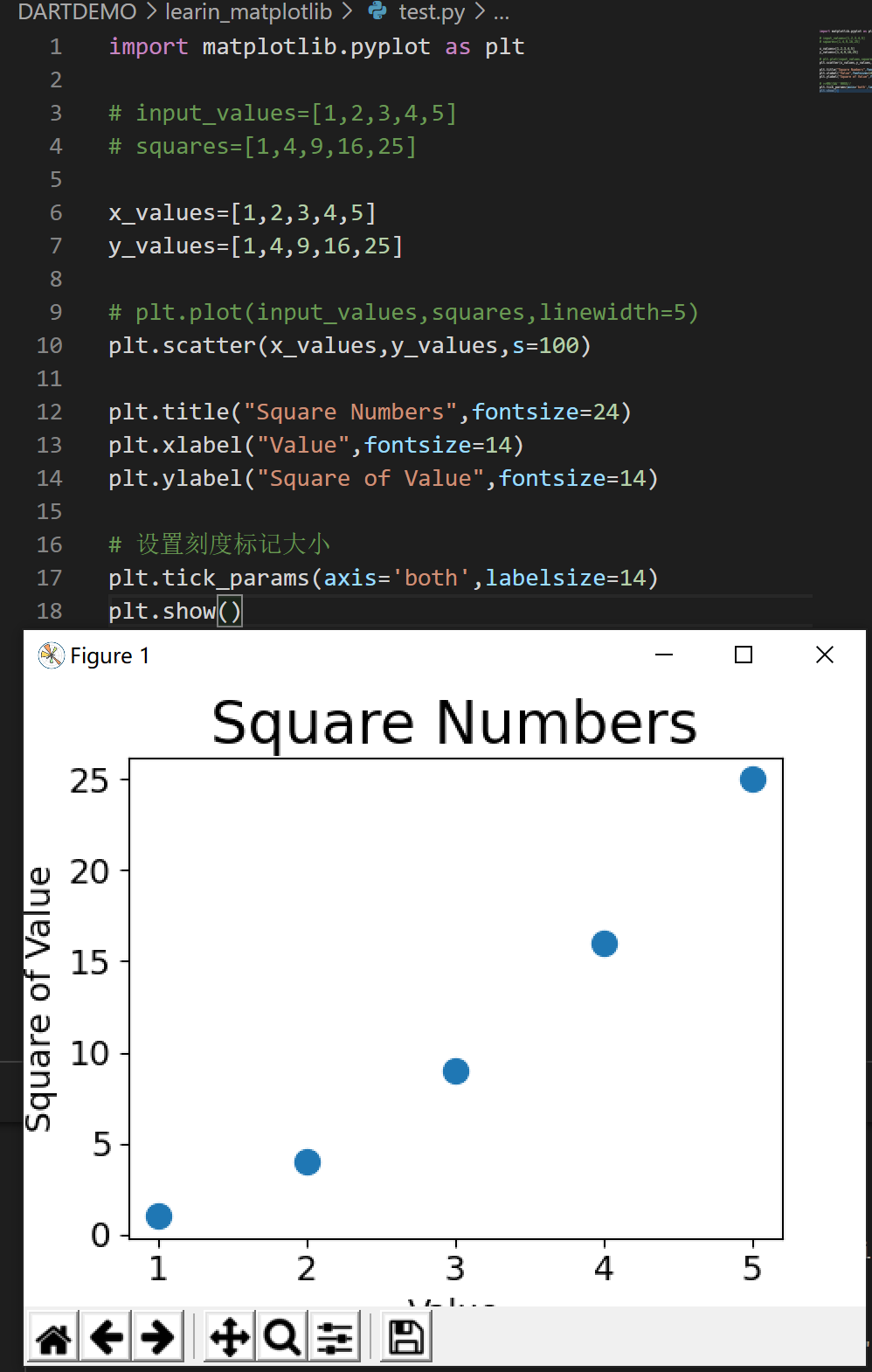


校正图形：

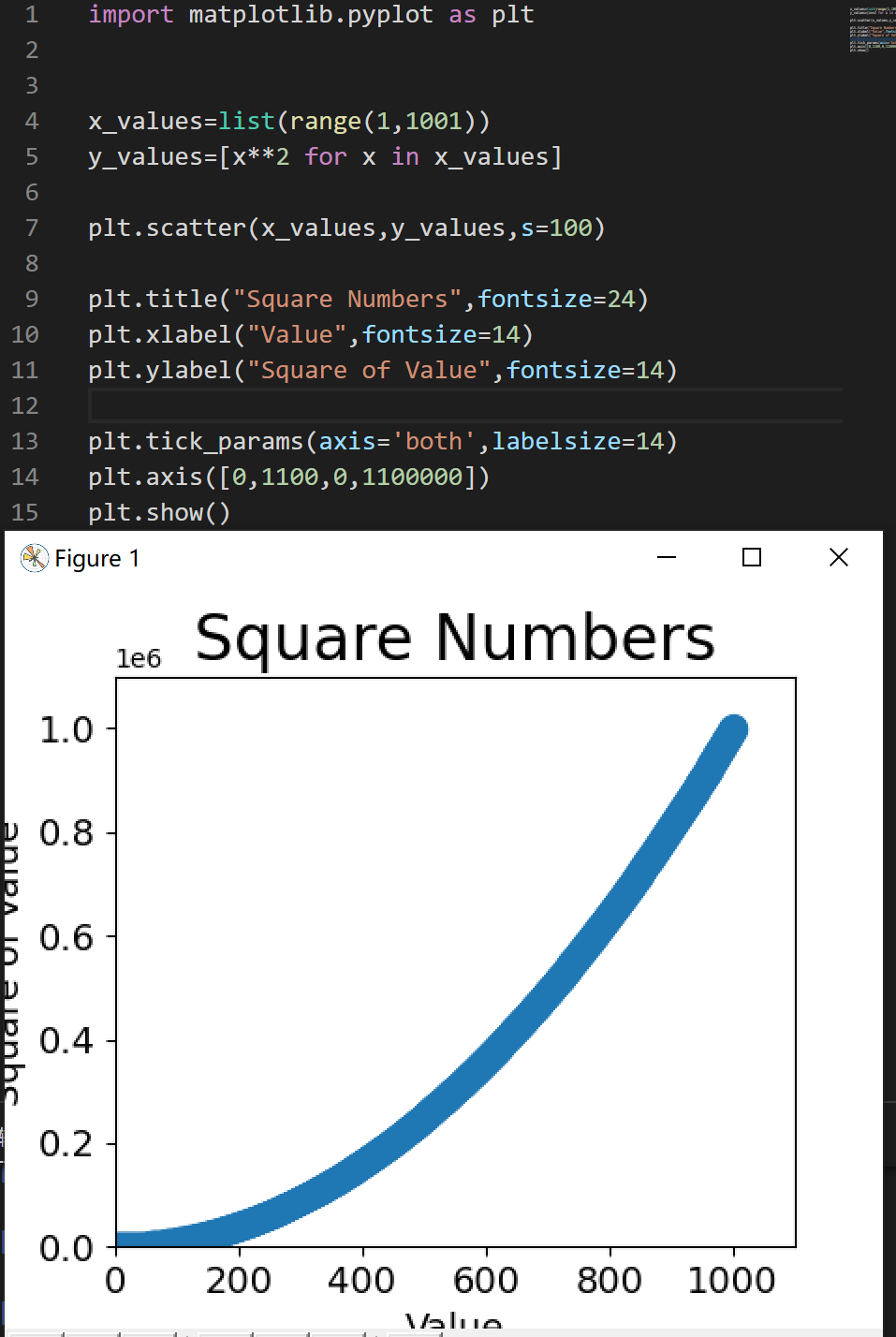
向plot()提供一系列数据时，默认第一个数据点对应x坐标为0，改变默认模式可向plot()提供输入值和输出值



scatter()绘制散点图



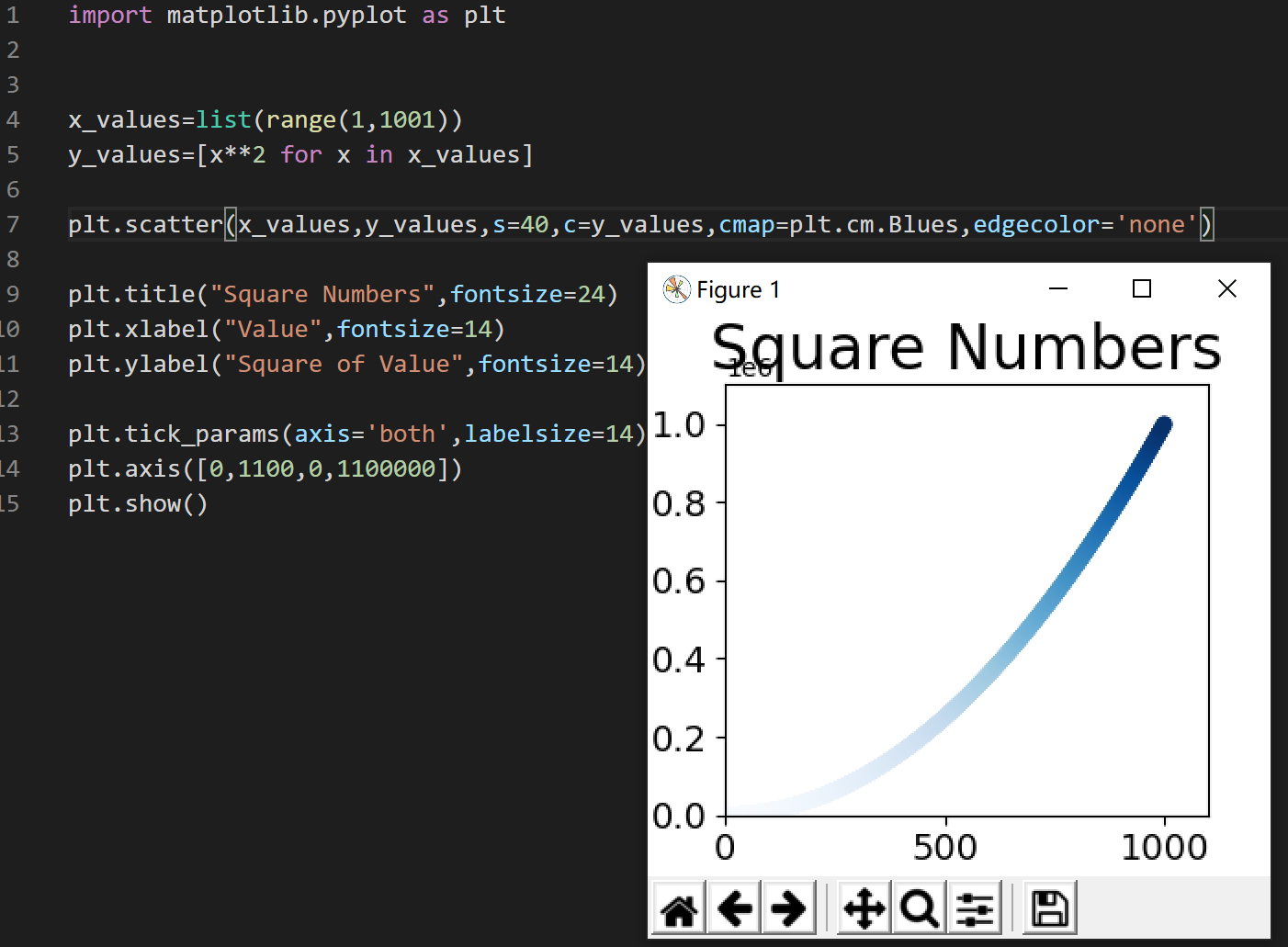
自动计算数据



删除数据点的轮廓，默认为蓝色点黑色轮廓，scatter()传参时传递实参edgecolor='none'，如plt.scatter(x\_values, y\_values, edgecolor='none', s=40)

自定义颜色scatter()传递参数c，如plt.scatter(x\_values, y\_values, c='red', edgecolor='none', s=40)，c也可为RGB模式。

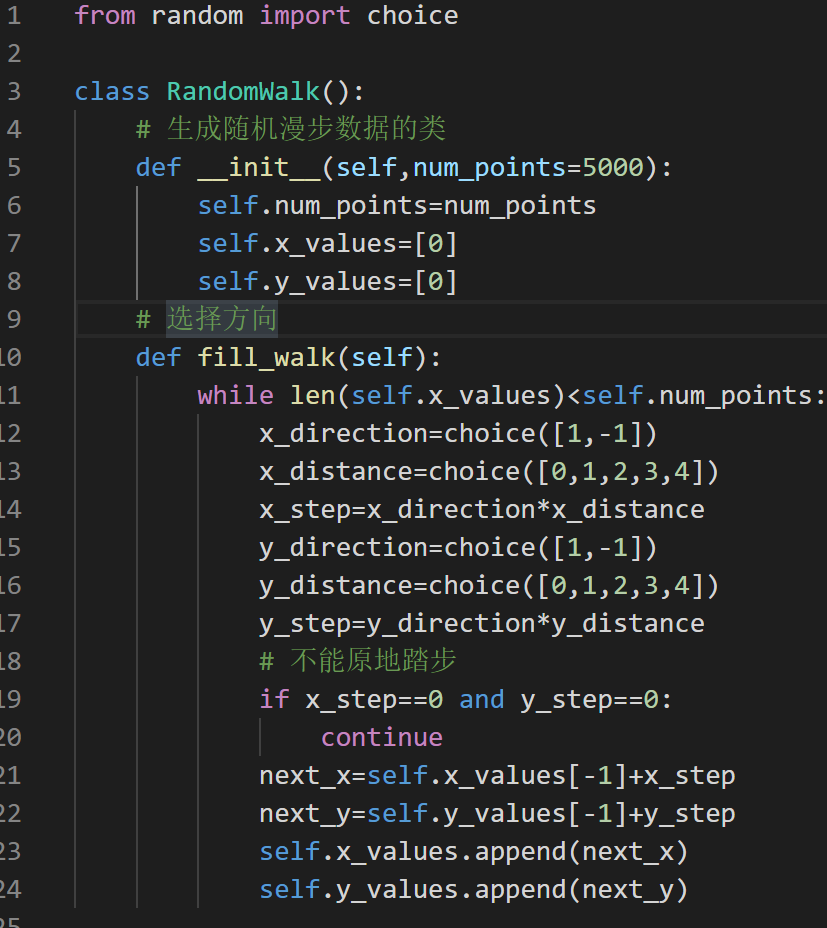
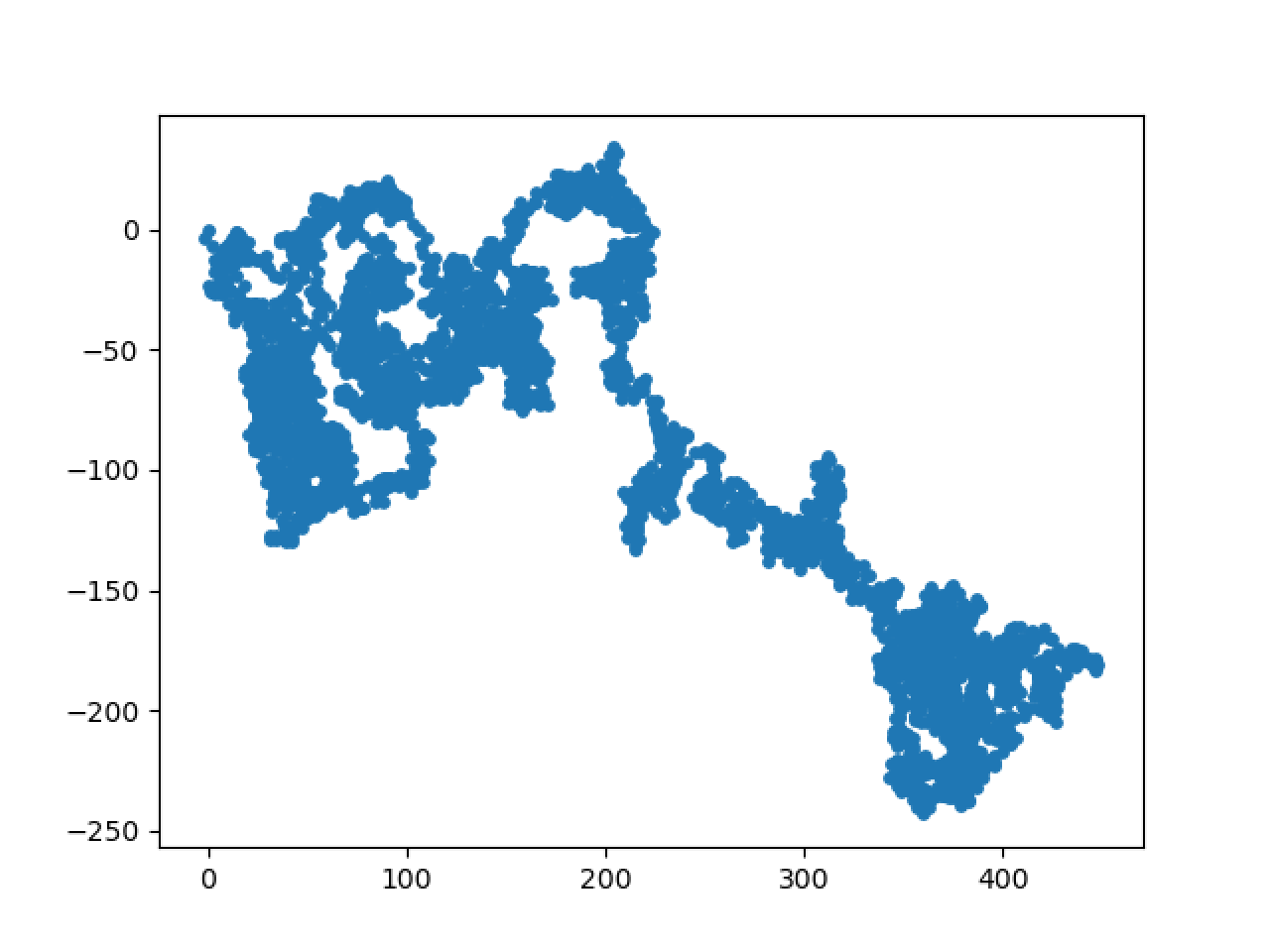
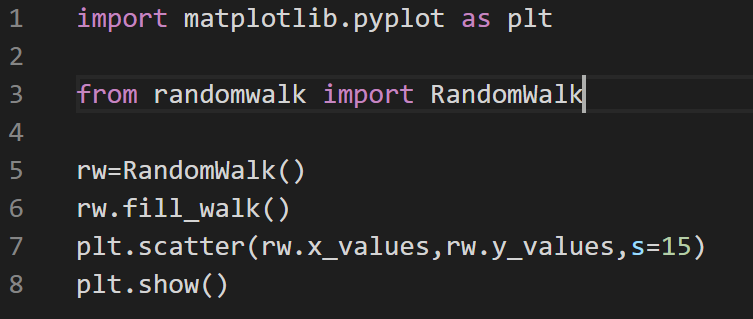
颜色映射，一系列颜色，从开始颜色到结束颜色，方便看出数据规律



自动保存图表，将对plt.show()的调用替换为plt.savefig()

1. 随机漫步

生成随机漫步数据，RandomWalk()类

模拟多次随机漫步，只要程序处于活动状态，不断模拟随机漫步，while True:

设置随机漫步图的样式，点着色；重绘起点和终点；隐藏坐标轴；增加点数即为增大num\_points值；调整尺寸设置，设置绘图窗口的尺寸。

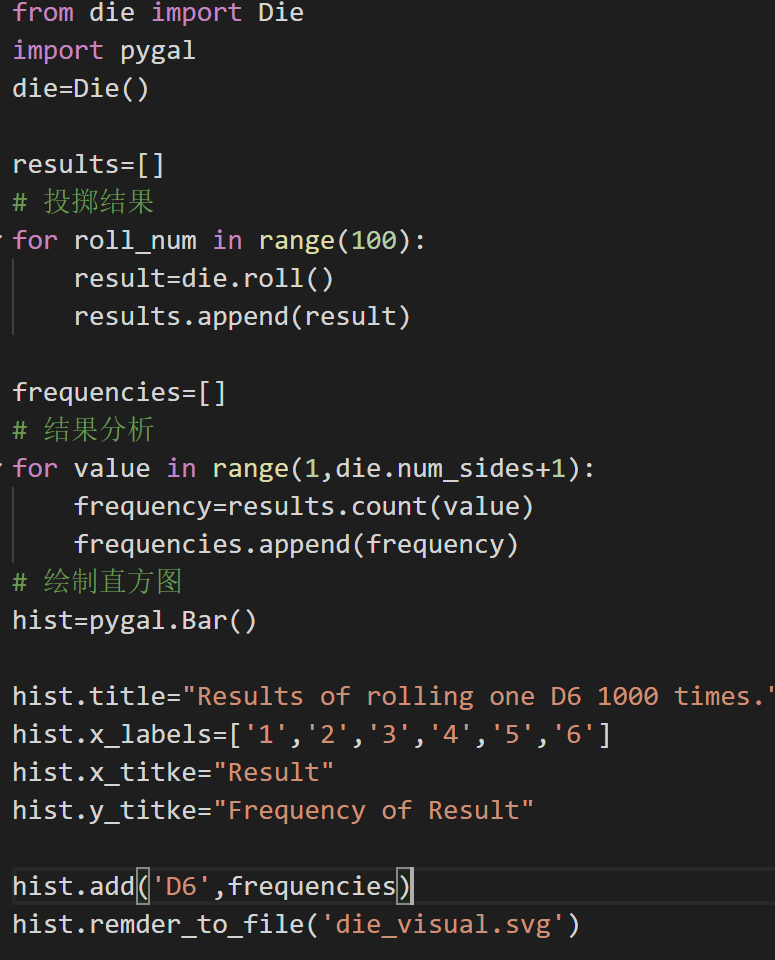
1. 使用Pygal模拟掷骰子

安装Pygal

模拟骰子



掷骰子，分析数据，绘制直方图



还可以同时掷两个骰子，掷两个不同的骰子

1. 下载数据

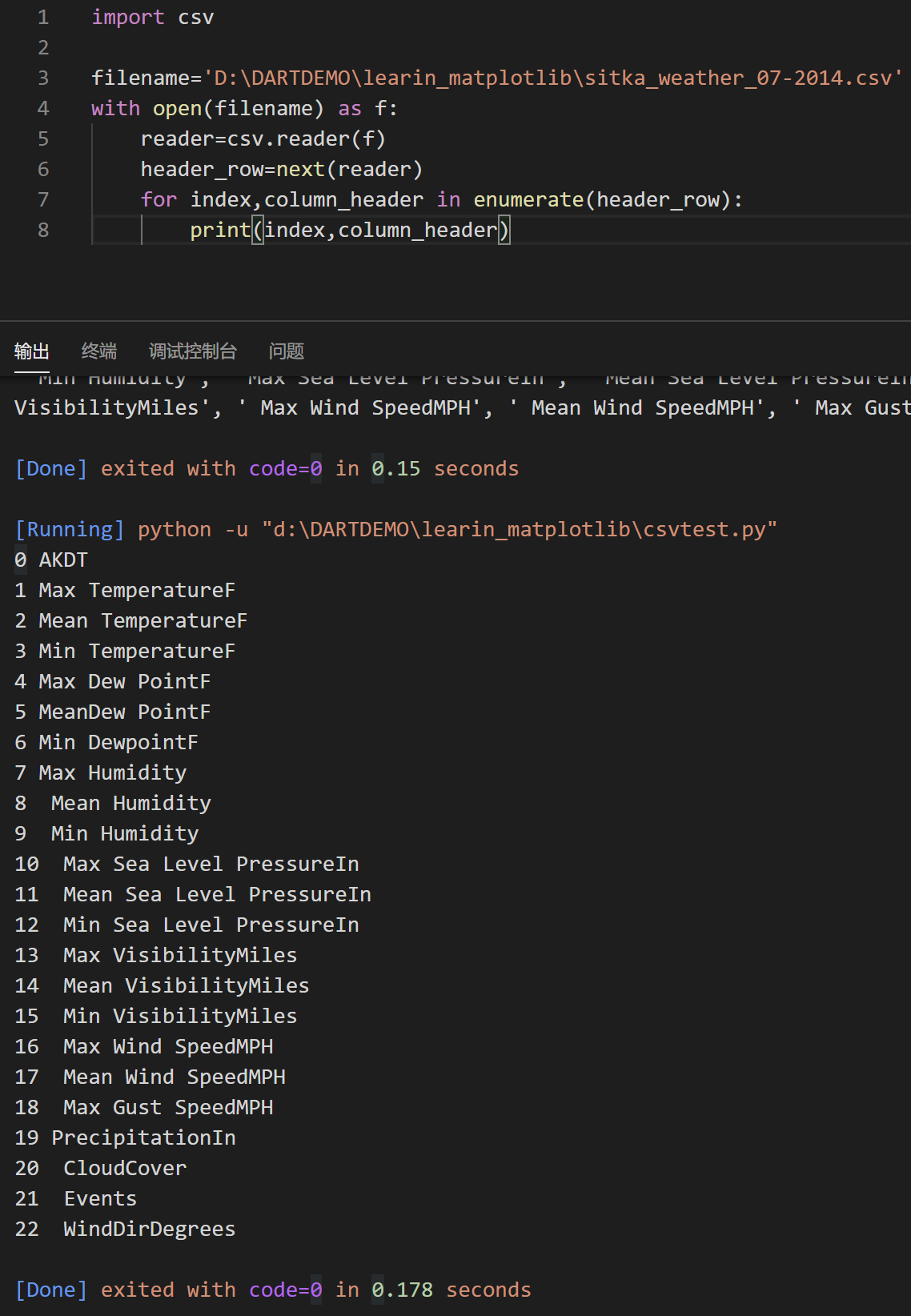
从网上下载数据，并对其进行可视化。两个常见格式存储数据：csv和JSON。

CSV格式：

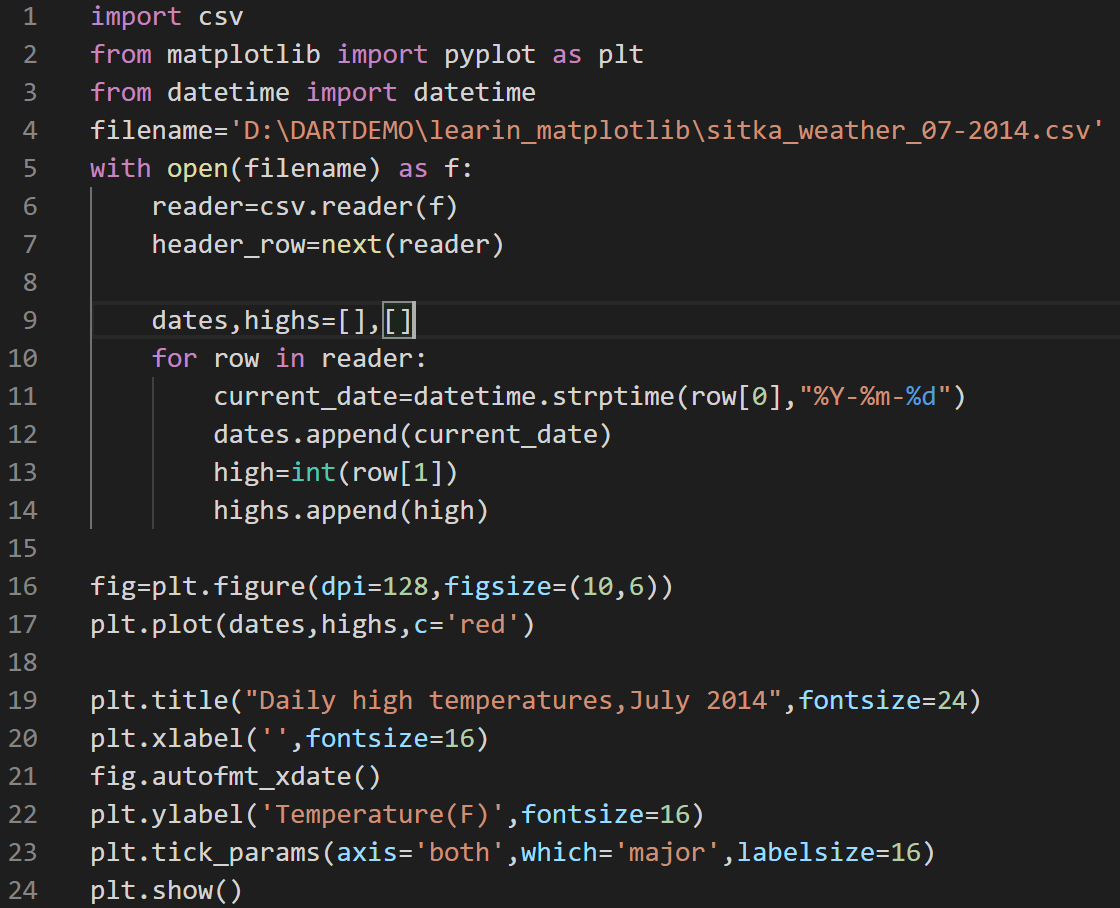
一系列以逗号分隔的值

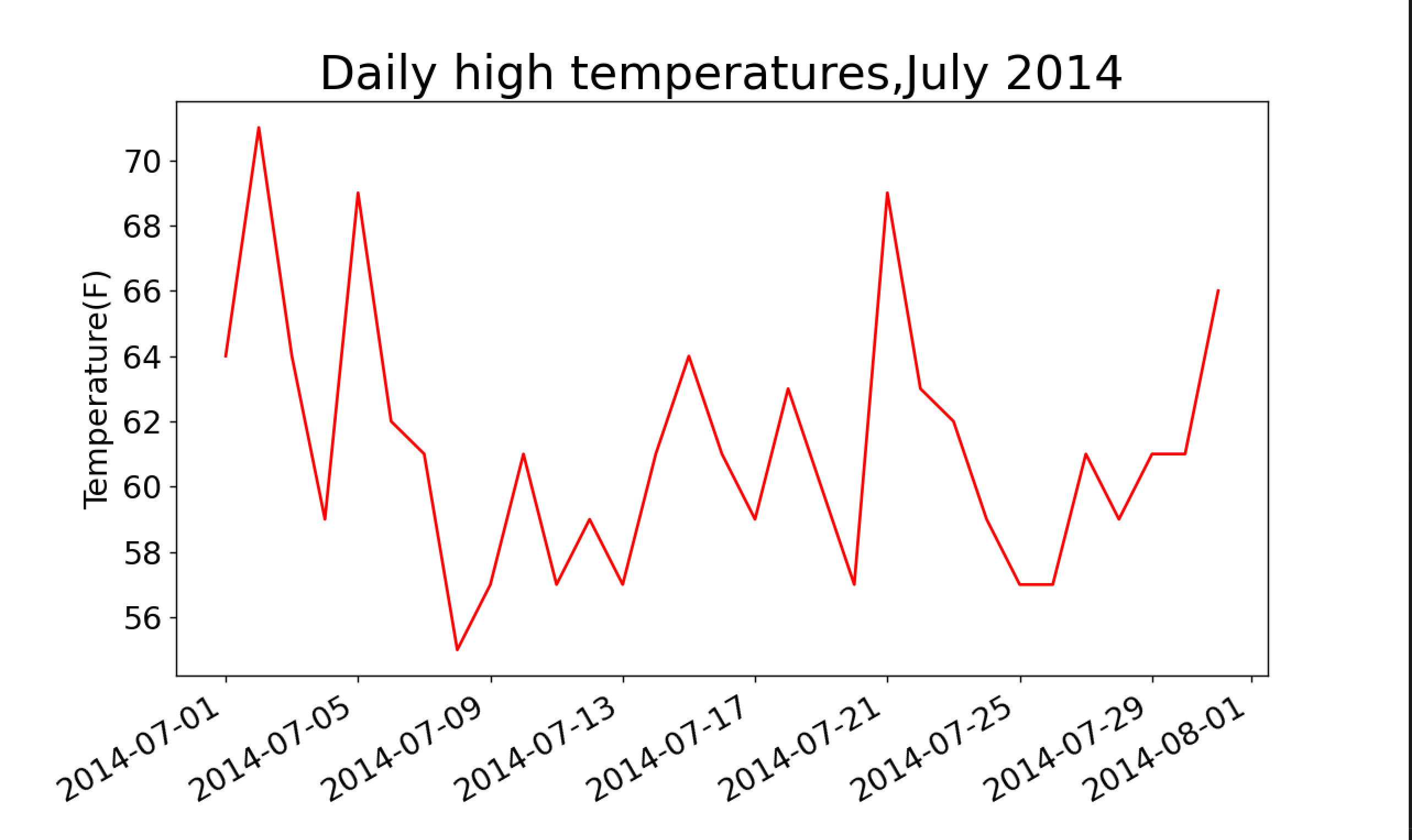
读取csv中的数据

读取每天最高温，将字符串转换为number型，让matplotlib能够读取

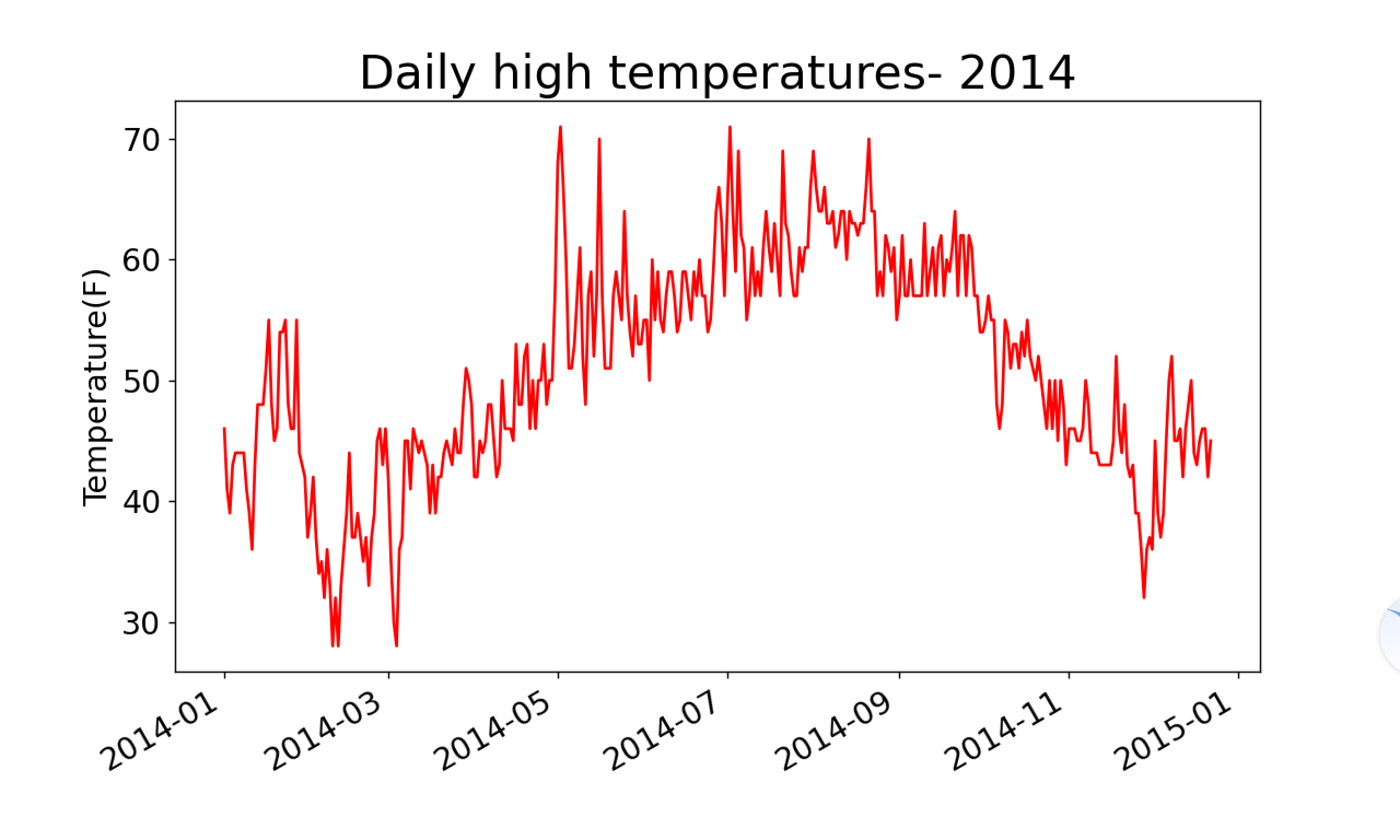


最高气温折线图表示

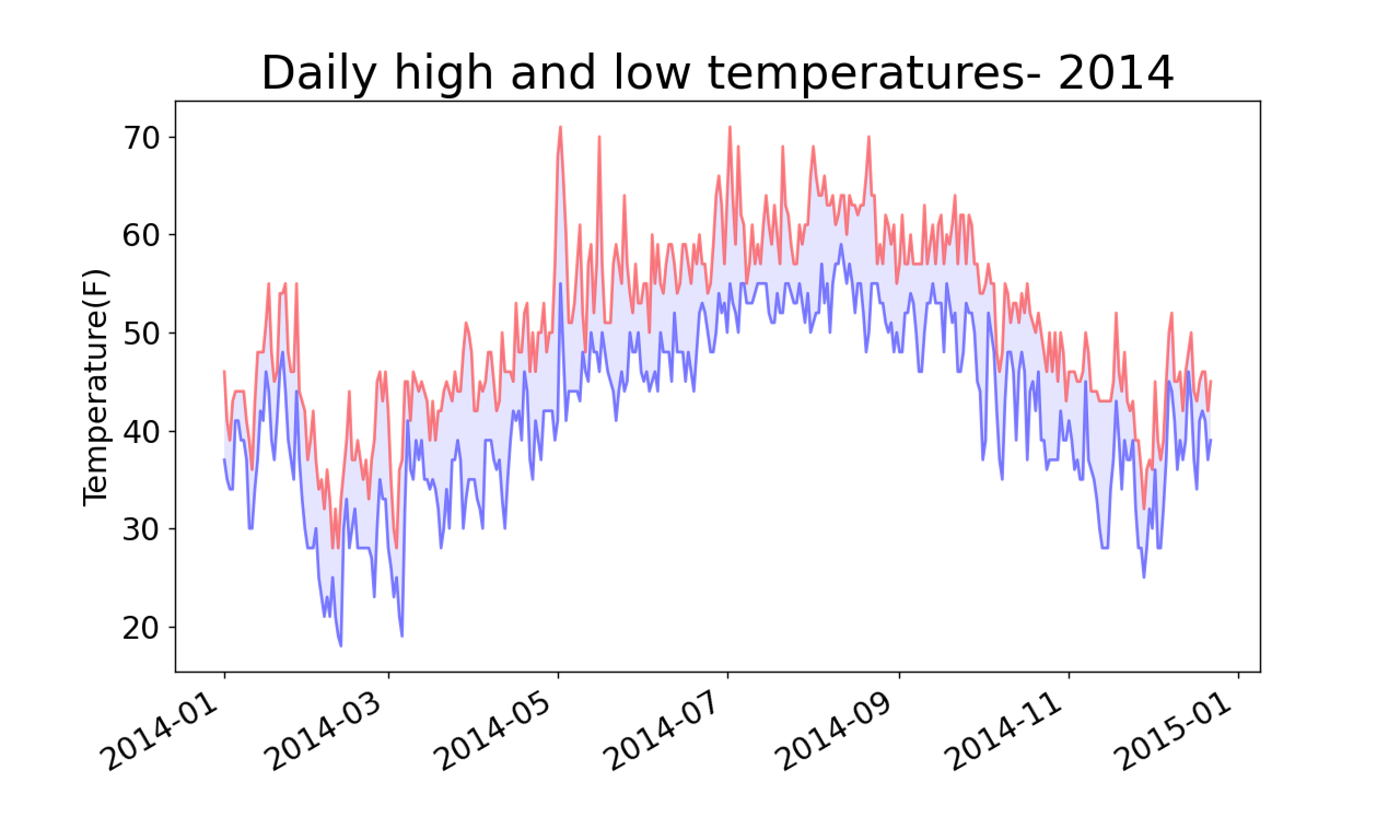




全年



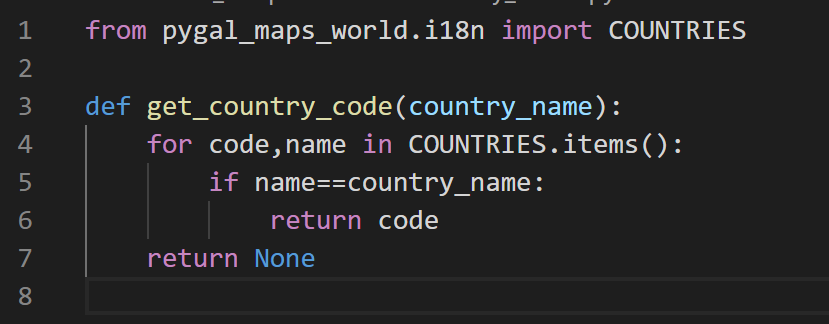
最高温最低温，着色

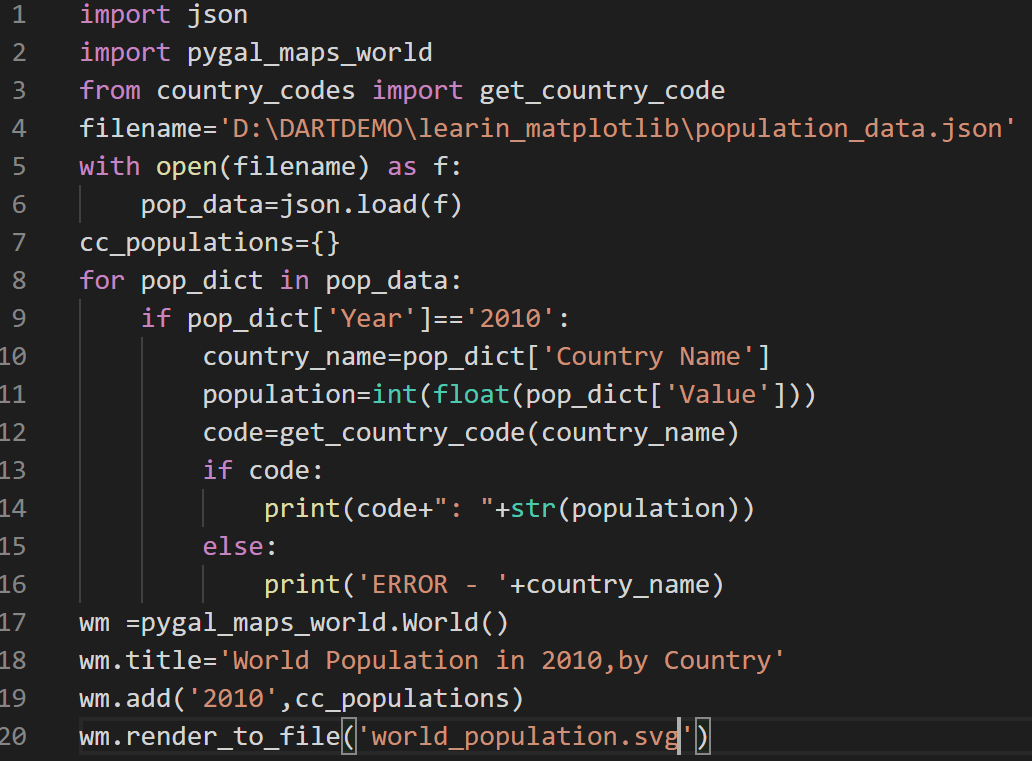


错误检查

JSON格式：

将json中的数据加载到列表中。将字符串转换为数字形式，注意浮点数的转换

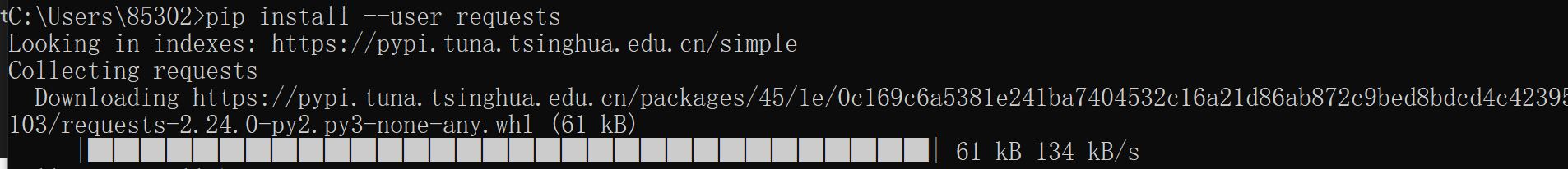




1. 使用API

Web API是网站的一部分，用于与使用非常具体的URL请求特定信息的程序交互。

安装requests



处理API响应，requests.get(url)

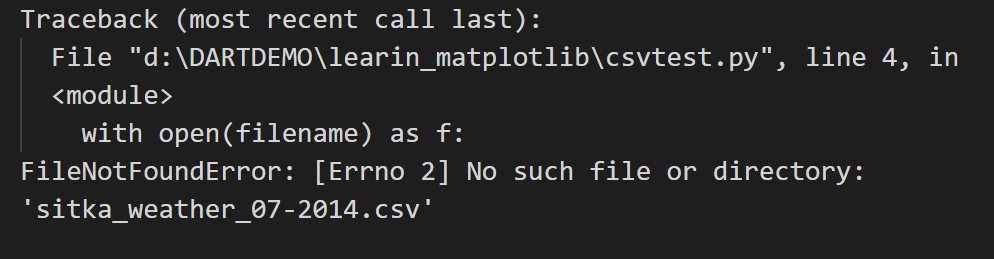
处理响应字典，API调用返回的信息存储到字典中，处理字典中的数据

监视API的速度限制，在浏览器中输入<https://api.github.com/rate_limit>查看

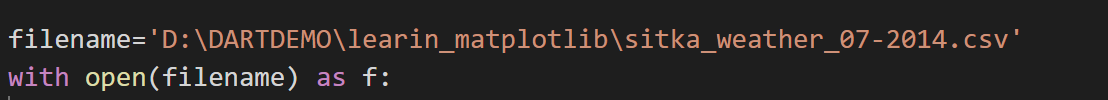
使用pygal可视化仓库，呈现GitHub上python项目受欢迎的程度。

1. 问题和解决方法
2. 运行打开csv文件时，出错



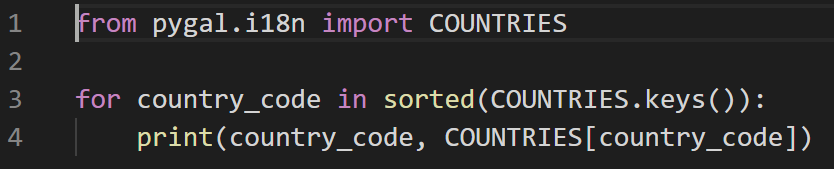


改成

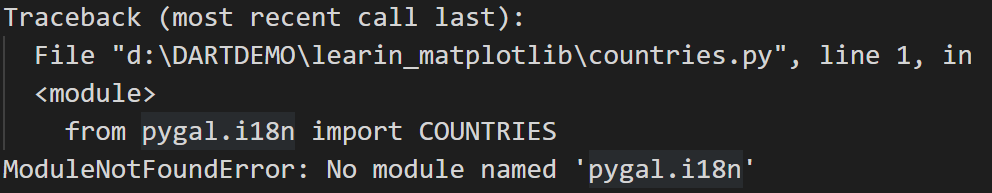


成功运行

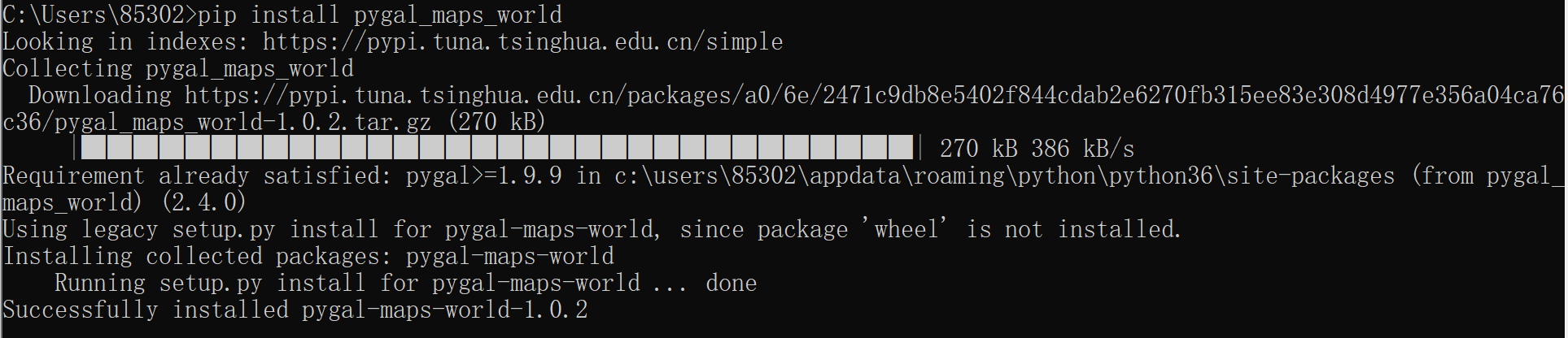
1. Pygal.i18n



报错

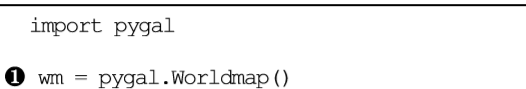


事实上，pygal.i18n已经不存在了，改为了pygal\_maps\_world，需单独通过pip下载



对应的在制作世界地图时

将



改为



2020/6/22

1. Hadoop

Hadoop，一个分布式系统基础架构。实现了分布式文件系统HDFS，存储海量数据集。MapReduce对这些数据集运行分布式计算。

Hadoop最底部是HDFS，上一层为MapReduce引擎。

Hadoop得以在大数据处理应用中广泛应用得益于其自身在数据提取、变形和加载(ETL)方面上的天然优势。Hadoop的分布式架构，将大数据处理引擎尽可能的靠近存储，对例如像ETL这样的批处理操作相对合适，因为类似这样操作的批处理结果可以直接走向存储。Hadoop的MapReduce功能实现了将单个任务打碎，并将碎片任务(Map)发送到多个节点上，之后再以单个数据集的形式加载(Reduce)到数据仓库里

1. 安装
   1. 下载jdk1.8，Linux，64位jdk-8u144-linux-x64.tar.gz。下载Hadoop
   2. 更换主机名，关闭防火墙，时间设置。
   3. 将jdk，hadoop解压包放到home目录下，并新建java，hadoop文件夹

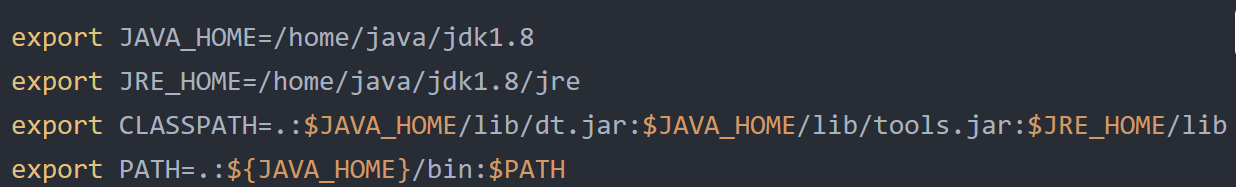
解压文件tar -xvf jdk-8u144-linux-x64.tar.gz

tar -xvf hadoop-3.2.0.tar.gz

并移动到文件夹java，hadoop下，并将文件夹重命名为jdk1.8和hadoop2.8

* 1. jdk环境配置 java -version 查看是否安装了jdk。

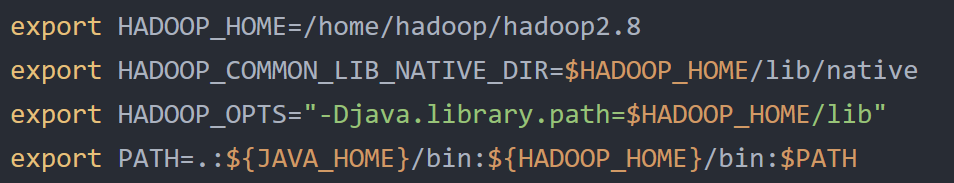
profile文件更改 vim /etc/profile 整体配置文件



source /etc/profile配置生效

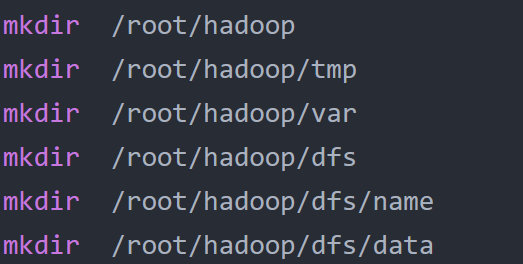
* 1. Hadoop环境配置

Profile文件修改 vim /etc/profile 整体配置文件



source /etc/profile配置生效

新建文件夹，在现在的root目录下建立文件夹



切换到 /home/hadoop/hadoop3.2/etc/hadoop/ 目录下

修改core-site.xml

vim core-site.xml



修改hadoop-env.sh export JAVA\_HOME=/home/java/jdk1.8

修改hdfs-site.xml

修改mapred-site.xml

* 1. 启动Hadoop

初始化，在目录/home/Hadoop/hadoop3.2/bin目录下输入 ./Hadoop namenode -format

启动HDFS

启动YARN

1. 运行wordcount

统计单词出现次数。

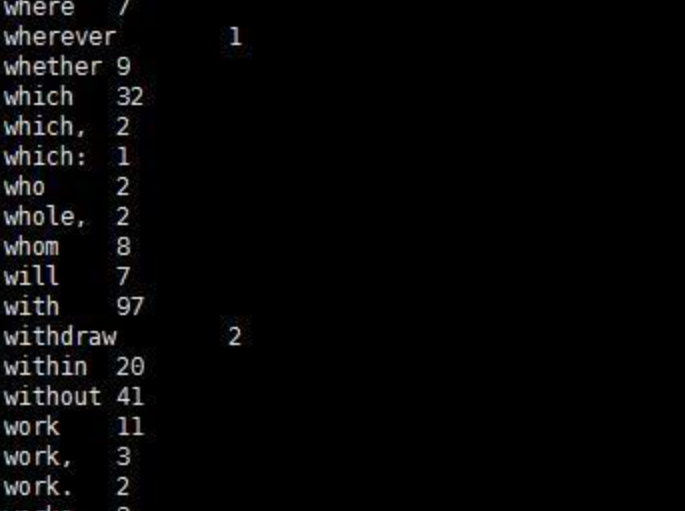
启动hadoop，进入hadoop安装目录cd /home/hadoop/hadoop/ 。查看目录情况$ ll，以文件LICENSE.txt为输入源文件。

在hadoop 目录下，输入命令:$hadoop fs -mkdir /input 创建/input目录。

将输入源文件放入input目录下，$hadoop fs -put LICENSE.txt /input

将/input通过wordcount统计每个单词出现次数。$hadoop jar /home/hadoop/hadoop/share/mapreduce/hadoop-mapreduce-examples-2.7.3.jar wordcount /input /output

打开文件part-r-00000查看结果



1. 问题与解决方法

在虚拟机上安装Hadoop比较困难，主要参考网上教程。