**python下载安装和基本介绍**

一点说明：本文档作为没有接触过编程，或者有一定编程基础却不熟悉python的同学们的参考，如果你已经比较熟悉python的下载安装和基本的语法规则，那么完全可以忽视这篇文档，直接去敲代码吧少年！

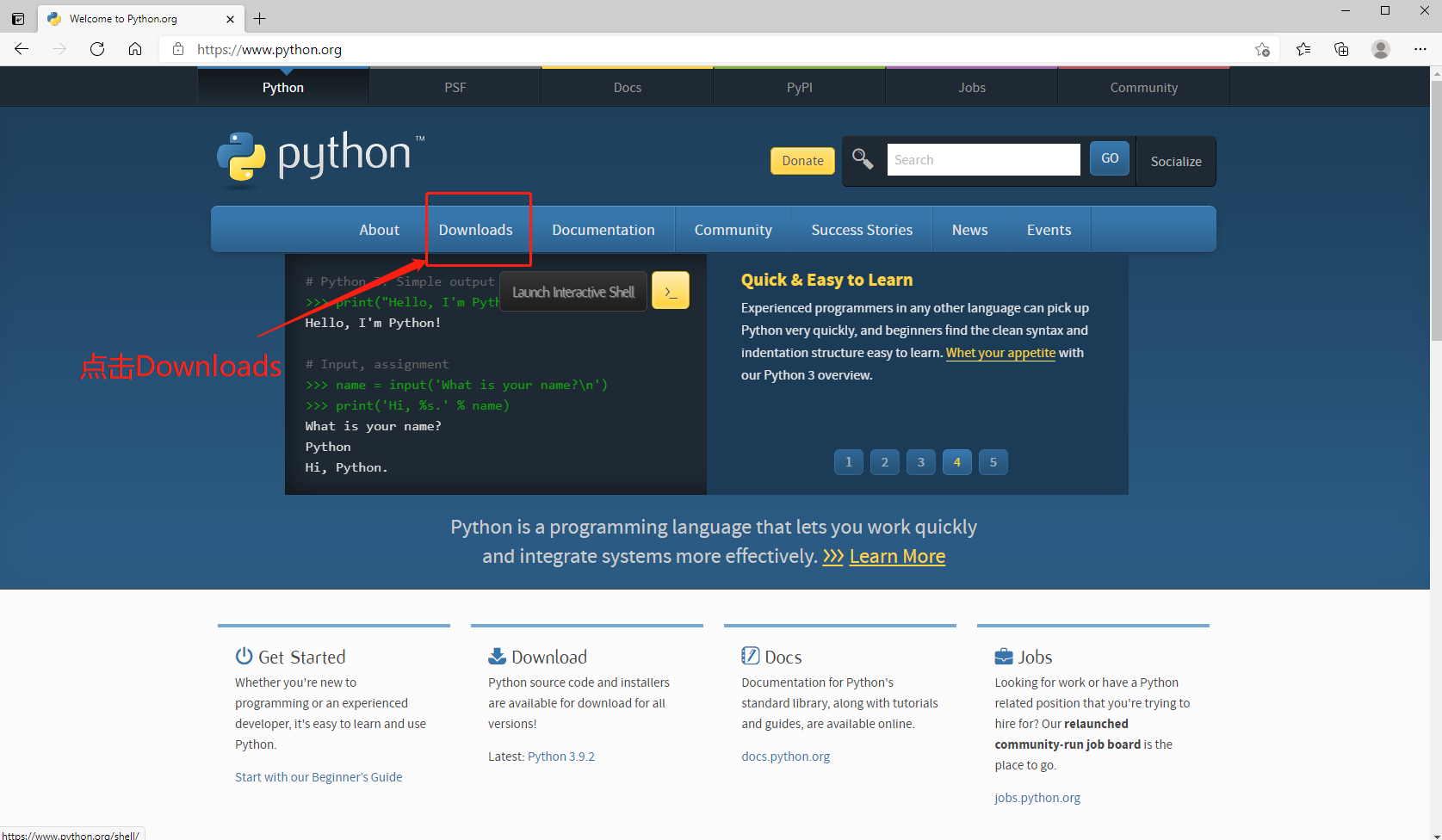
**一.下载和安装：**

**1.python的下载和安装**

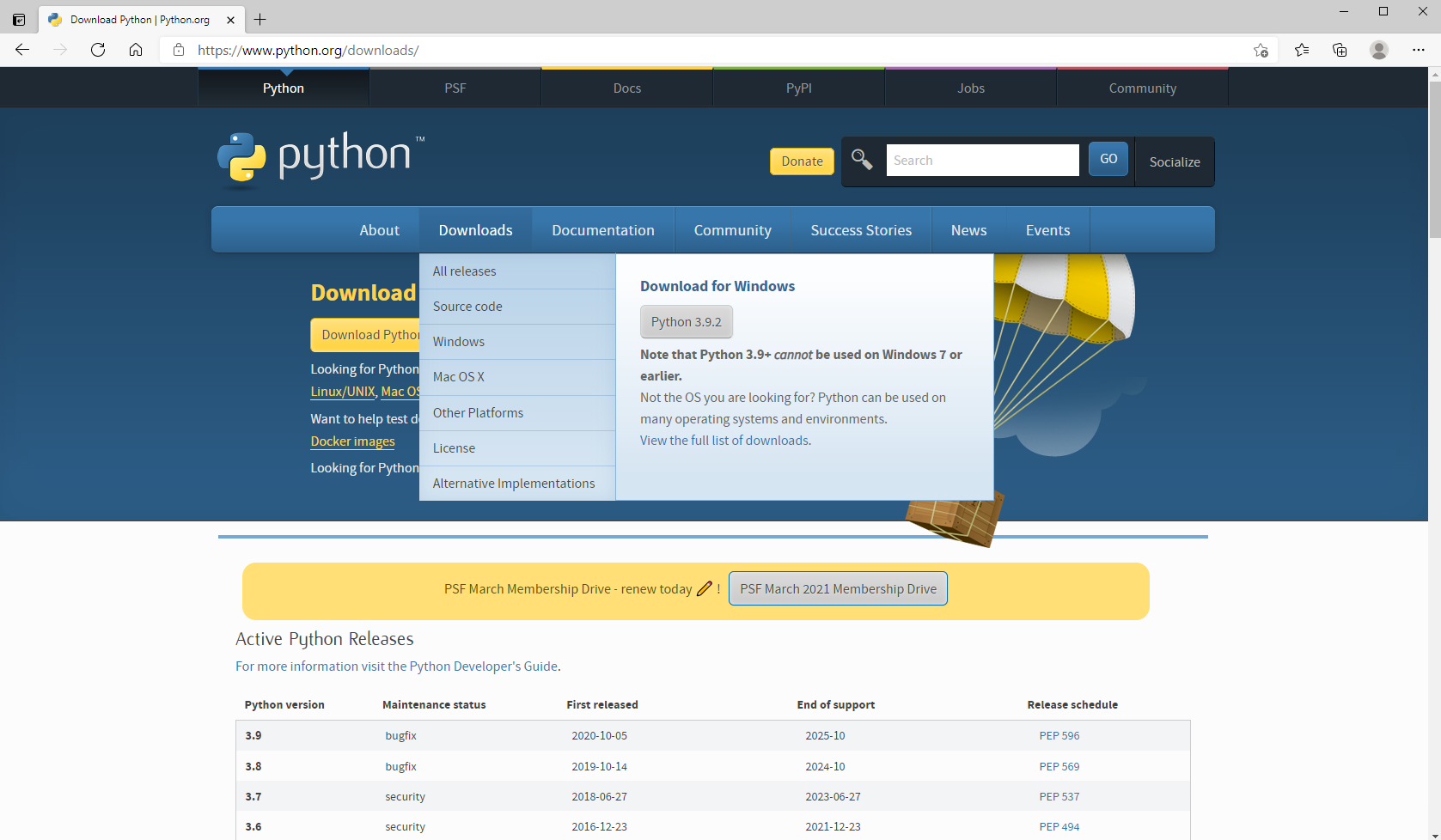
**下载**

我们使用python3.8.6版本，首先进入python的官网（切记不要找非官方渠道下载）：

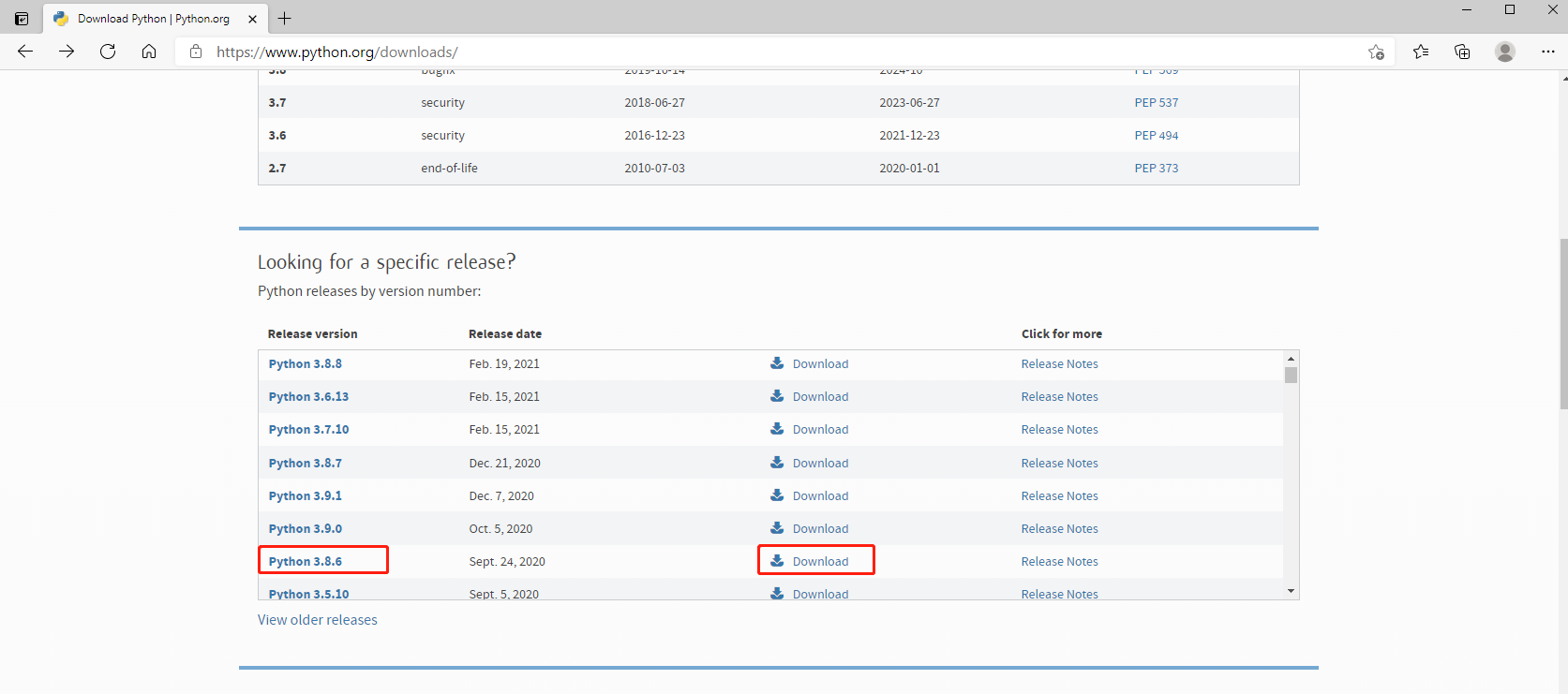
<https://www.python.org>



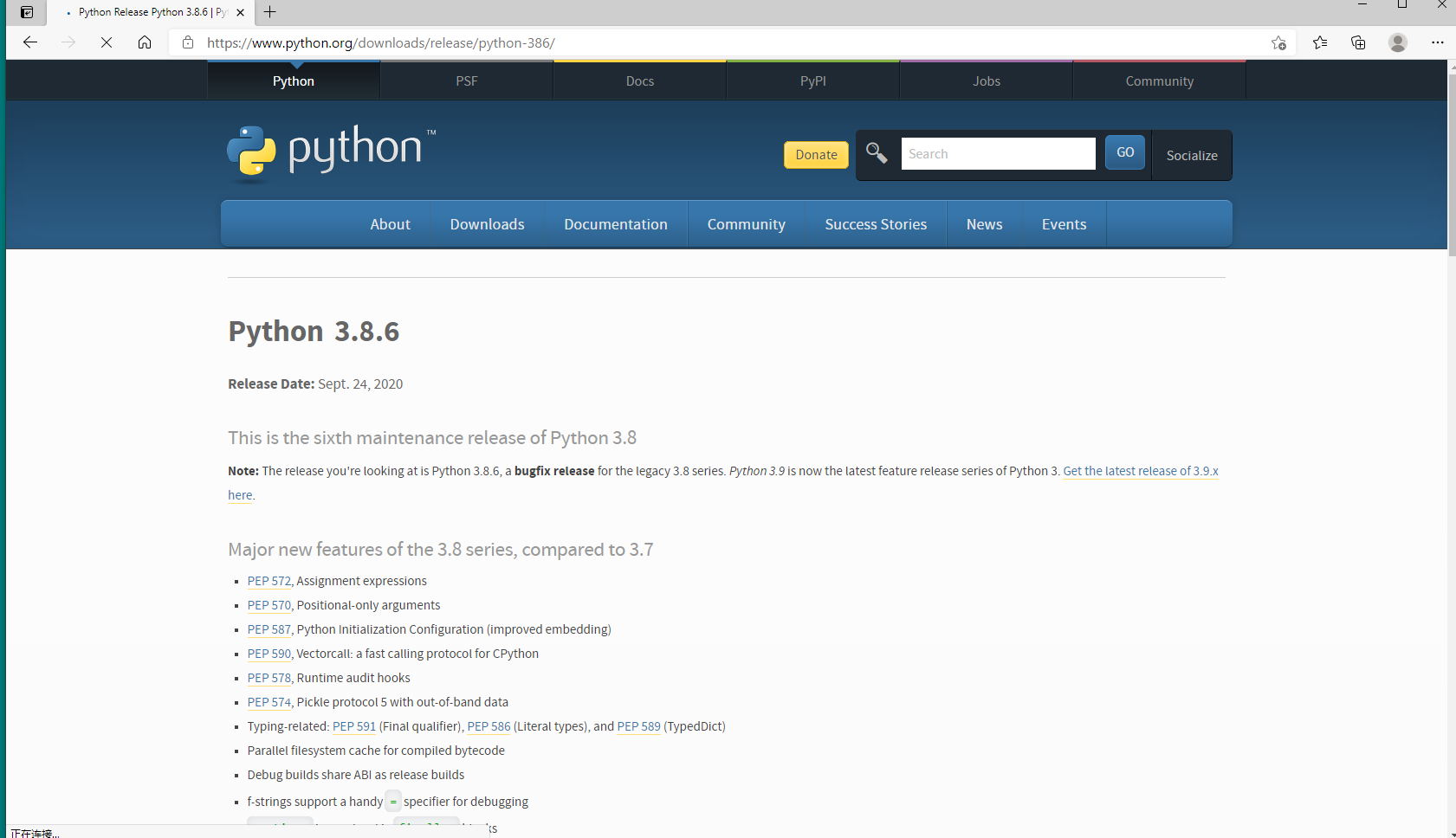
然后点击Downloads进入下载页面



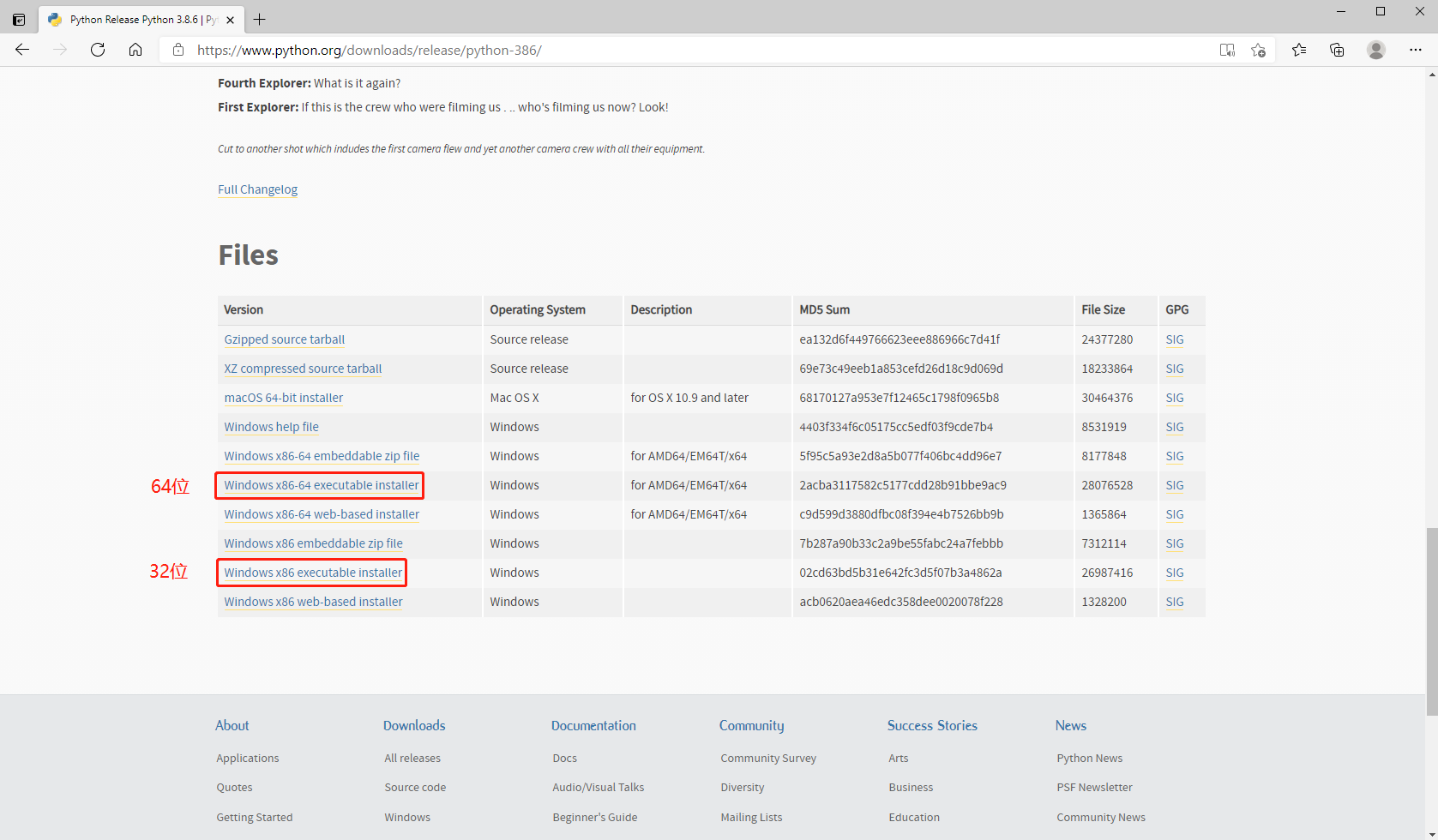
一般来说这个网站会根据你的操作系统是widnows还是MacOS或者Linux自动跳转到对应的系统的下载页面，这里以windows为例：



找到Python3.8.6版本，点击下载，进入到python3.8.6的下载页面：



我们把页面拉到底部的Files：



直接选择简单的可执行文件的安装包（executable installer）进行安装，根据你的系统位数选择下载64位操作系统的还是32位的。

(国内网络环境下载可能有点慢，大家也可以直接从群文件中下“python-3.8.6-amd64.rar”解压安装)

**安装**

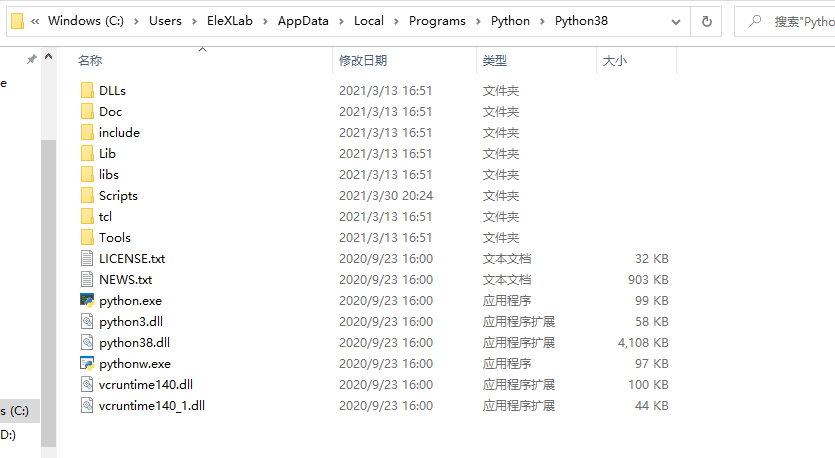
下载之后就像安装普通软件那样，点击exe文件安装，使用默认配置下一步下一步就好。

等待一会儿安装成功后，为了能改在命令窗口中运行python，我们还需要配置一下把python配置到系统环境变量中。

**配置到系统环境变量**

如果你是默认安装，你的python将会被安装在这个目录下：

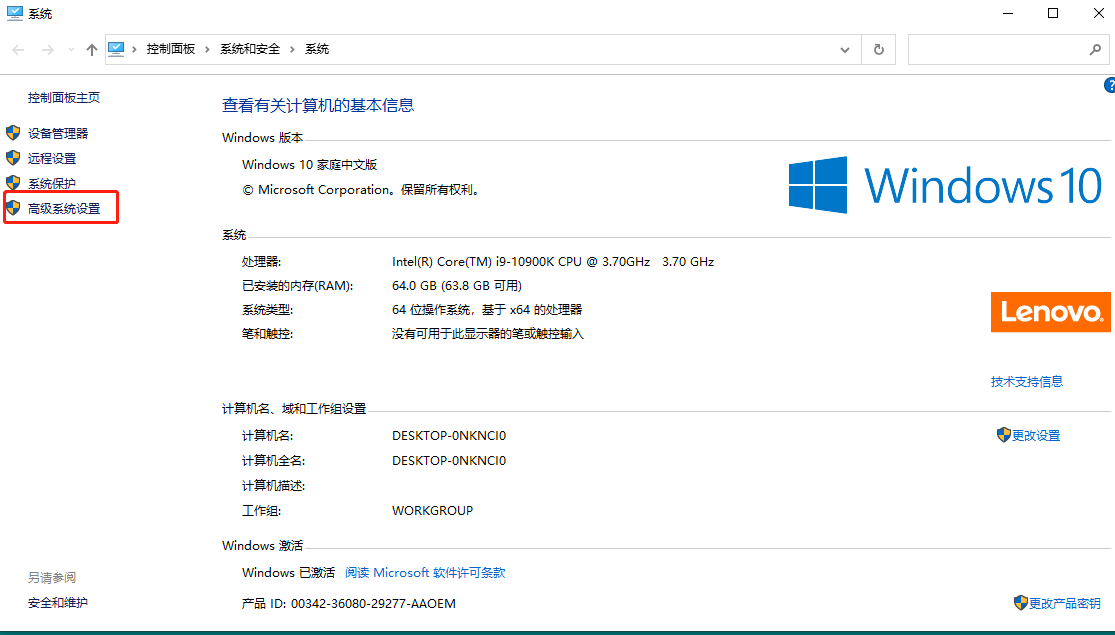
C:\Users\你的用户名字\AppData\Local\Programs\Python\Python38



在“此电脑”上右键点击属性：



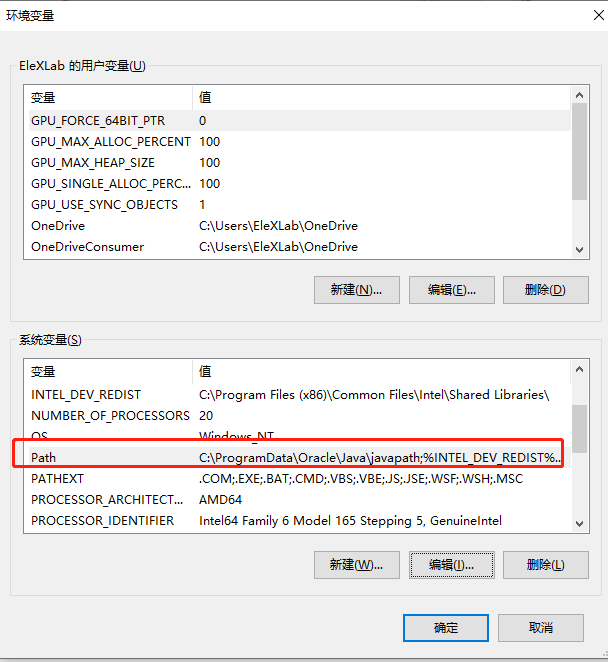
然后点击系统高级设置：



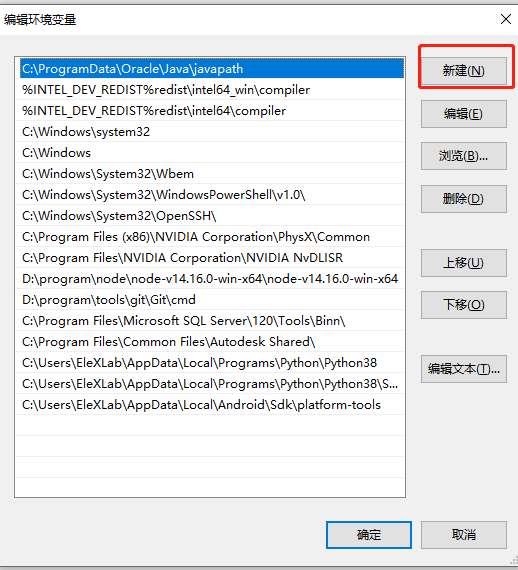
点击环境变量：

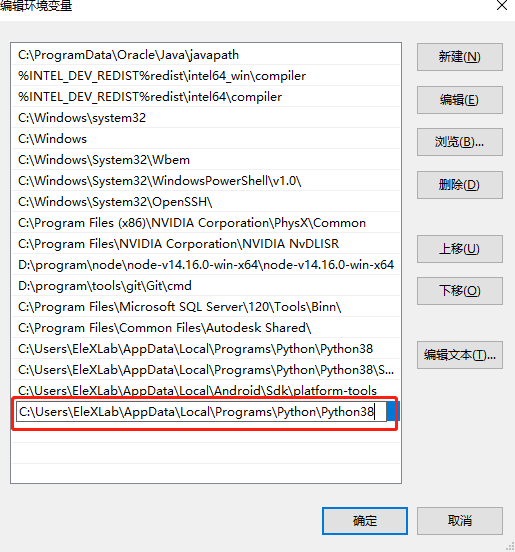


找到系统变量Path这一栏：

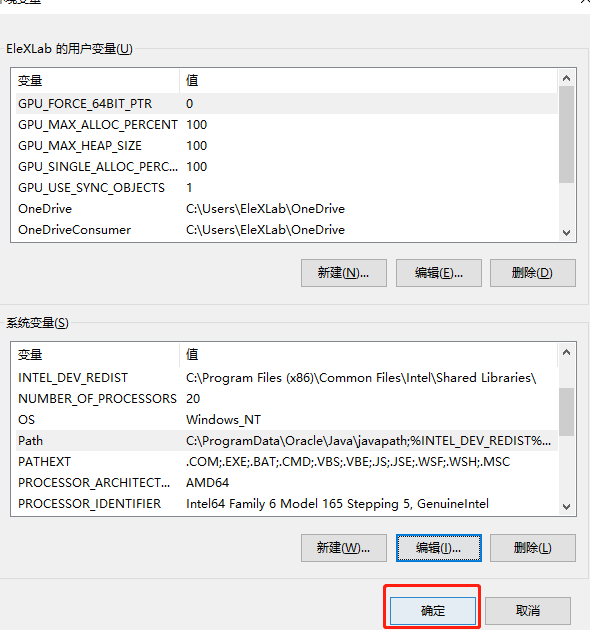
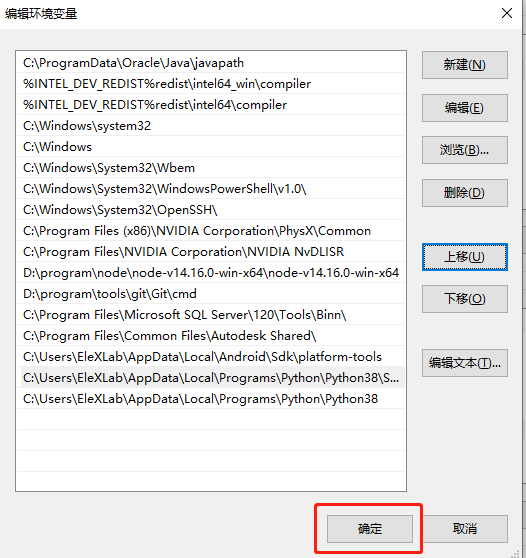


双击这一栏弹出编辑环境变量，点击新建：





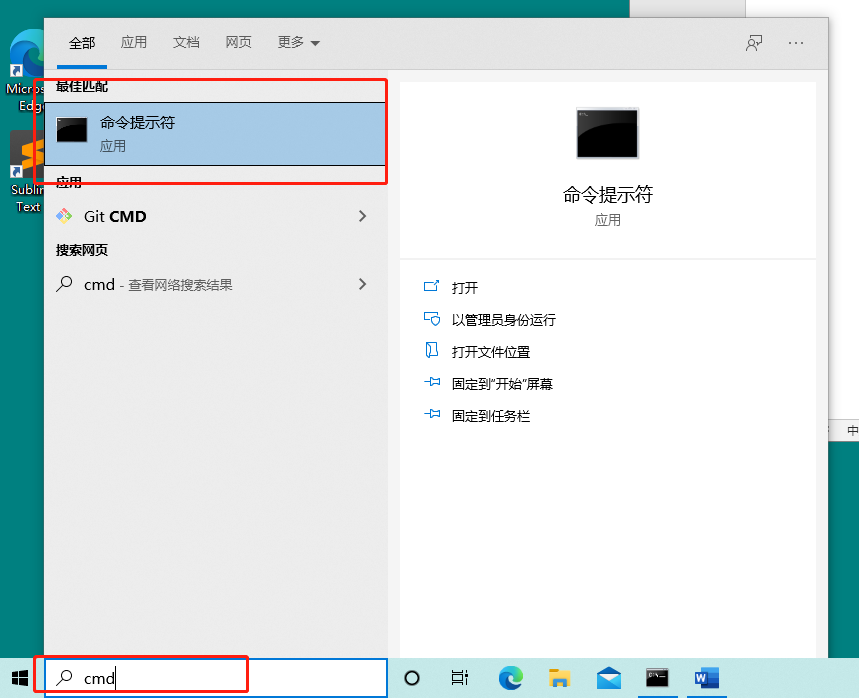
把你的python安装路径输入进去，然后一路确定退出配置



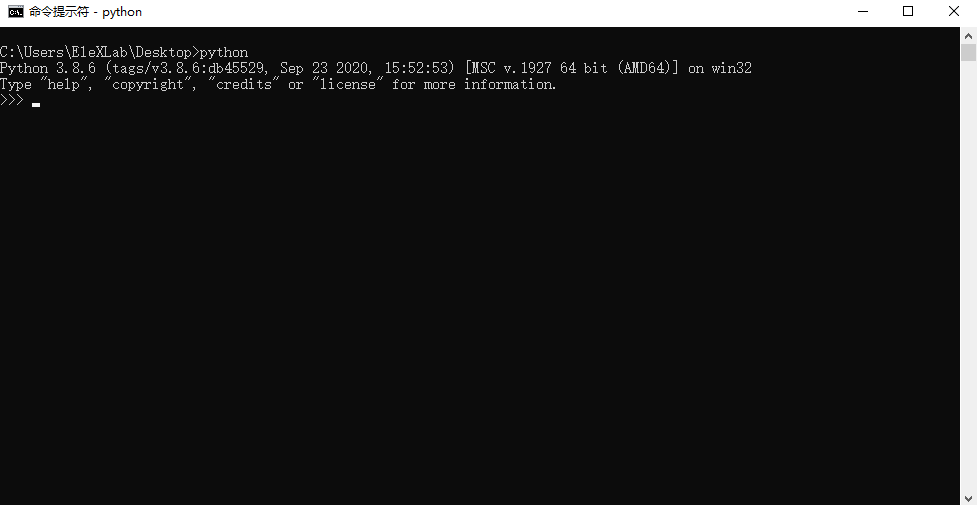
这样你的python环境就彻底安装配置好了

我们测试一下。

在搜索栏中输入cmd，打开命令窗口。

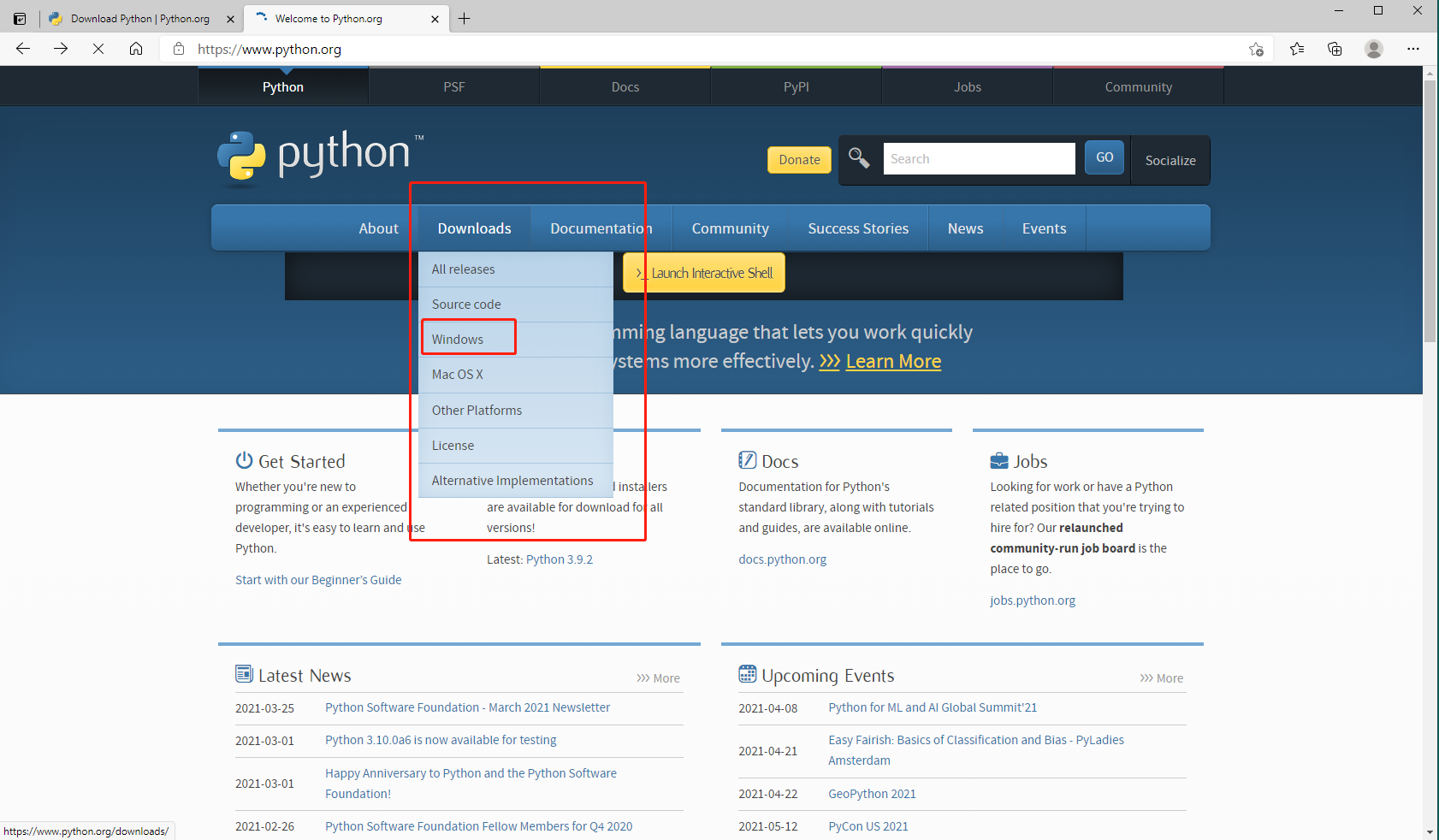


打开windows的命令窗口，输入python：



看到进入python3.8.6就证明安装成功啦！

Ps：如果没有正确跳转到你的系统对于的页面，也可以在把鼠标放在Download选项上，会显示各个系统，然后选择你的系统类型：

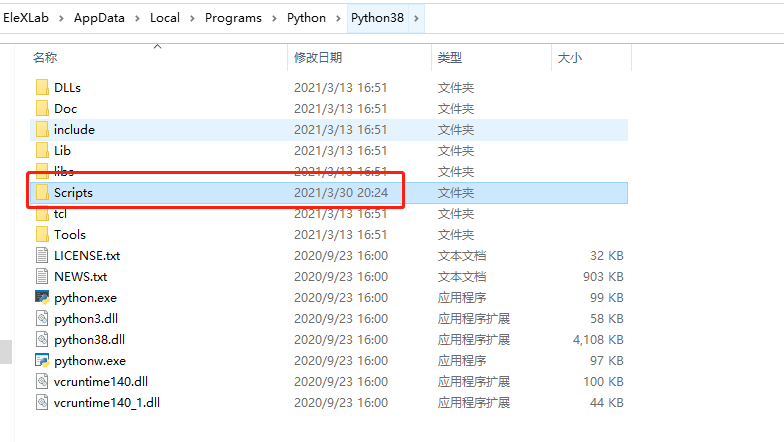


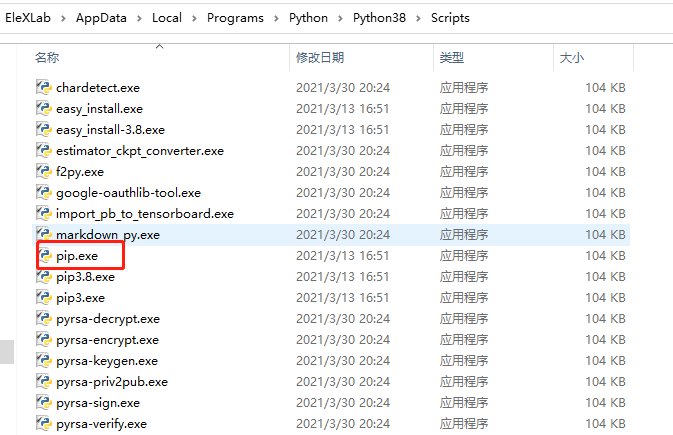
**2.关于第三方库的下载和安装**

Python中有许多的第三方库，站在巨人的肩膀上给我们的编程带来了很多的方便，比如第一课中我们使用的numpy库做数学计算，但是这些库并不是直接伴随python默认安装的，我们需要手动的去安装。

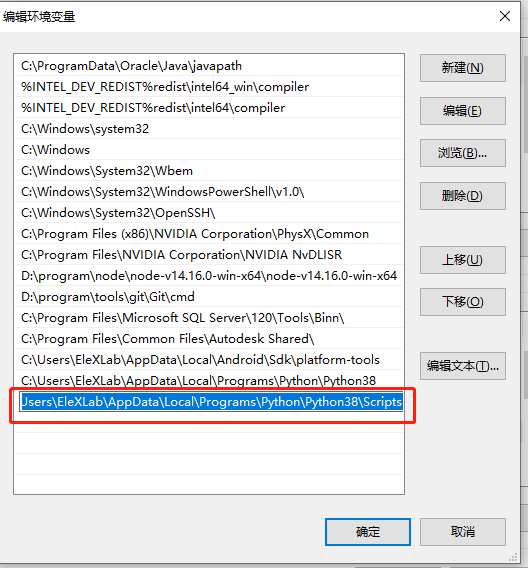
当然，python作为一个如此拉风的编程语言，像安装第三方库自然也是提供了很好的解决方案。

在python的安装目录下有一个Scripts的子目录，里面提供了一个叫做pip的工具程序：





这个pip就是python专门用来安装第三方库的工具，首先我们同样把这个工具配置到系统环境变量中以方便使用：

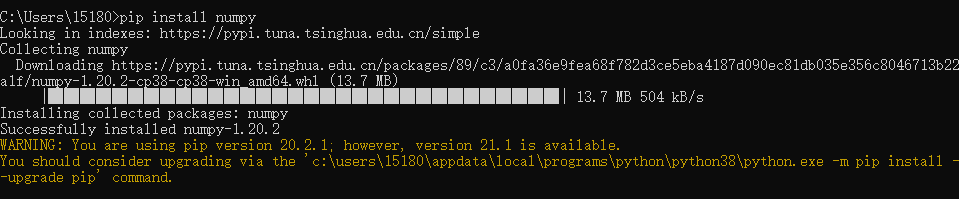


**numpy库安装**

现在我们就可以使用用pip安装第三方库了，比如安装numpy库，只需要输入命令：

pip install numpy

回车等待就万事大吉了，出现successful字样我们就表示成功安装了这个库，非常方便。



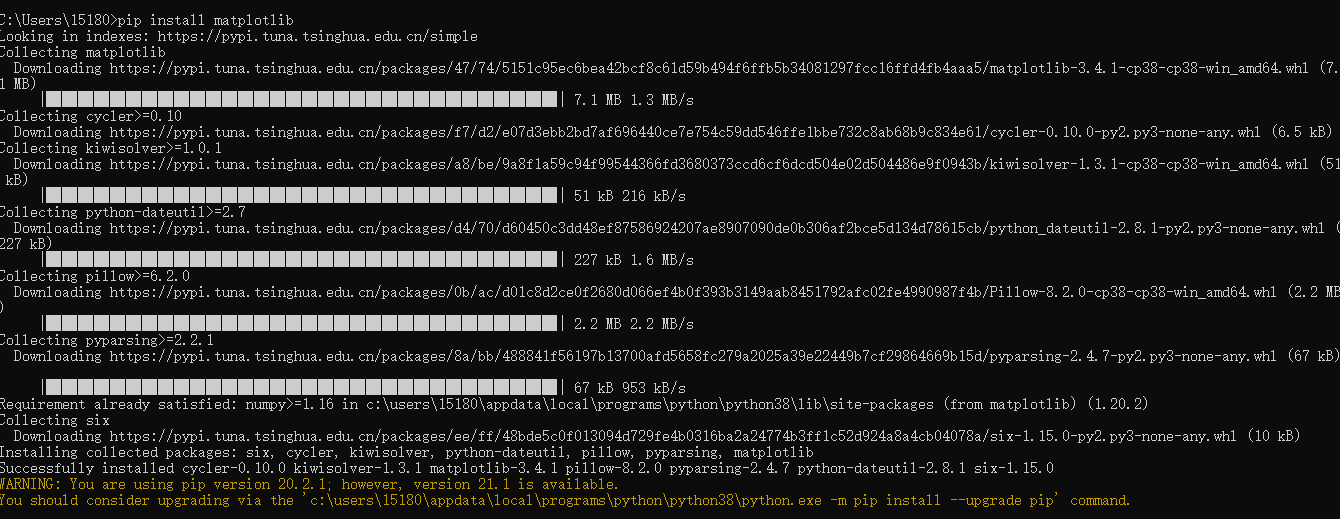
当然在国内的网络环境下这个下载安装的过程可能有点费劲，如果你没有耐心可以给pip配置一个镜像源加速，当然如果你自己能解决网络问题也可以不用配置。

清华pip镜像源的配置方法：https://mirrors.tuna.tsinghua.edu.cn/help/pypi/



**matplotlib库安装**

pip install matplotlib



**二.python介绍**

Python诞生之初就是为了“让编程更加简单”，所以它有着极简的设计理念，这对初学者是相当美好的事情。

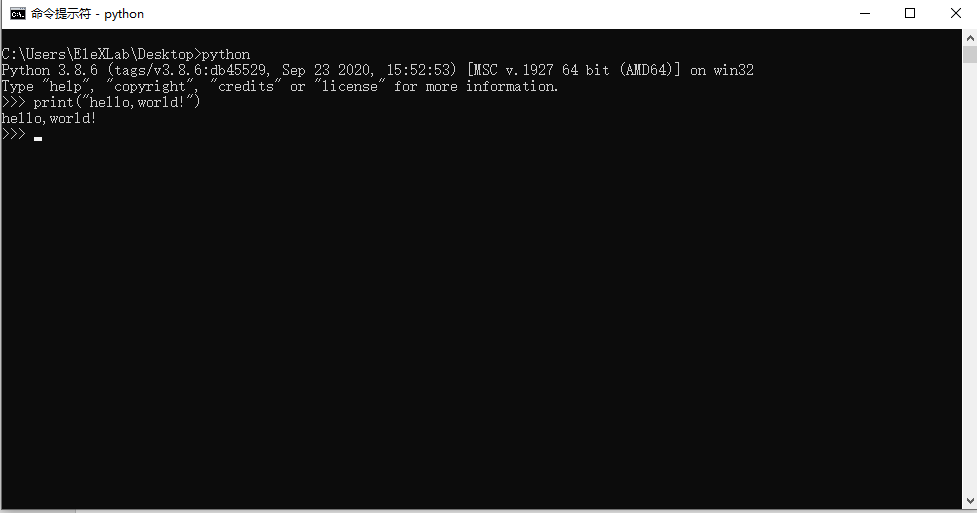
**1.Python程序执行方式**

执行python程序有两种方式，我们先看所有**脚本语言**都会支持的一种简单的执行方法：

**交互模式**

打开cmd后，输入python回车，也就进入了**python的执行环境**，我们可以在这里直接编写并运行，比如经典的“hello，world”：

print("hello,world!")



print是python的打印函数，给函数传入一个字符串类型的变量，比如“hello，world”，就可以用在控制台中打印出来

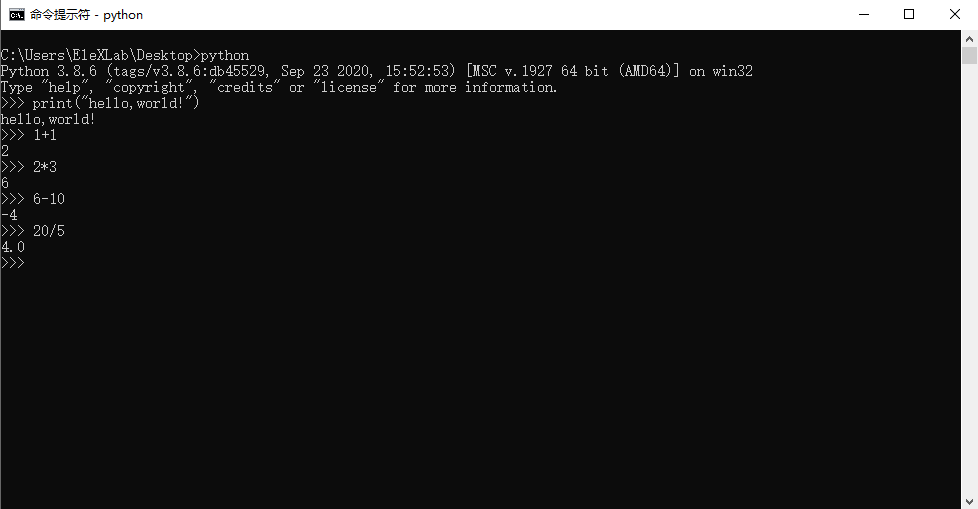
同样，我们也可以直接做数学计算：

1+1

2\*3

6-10

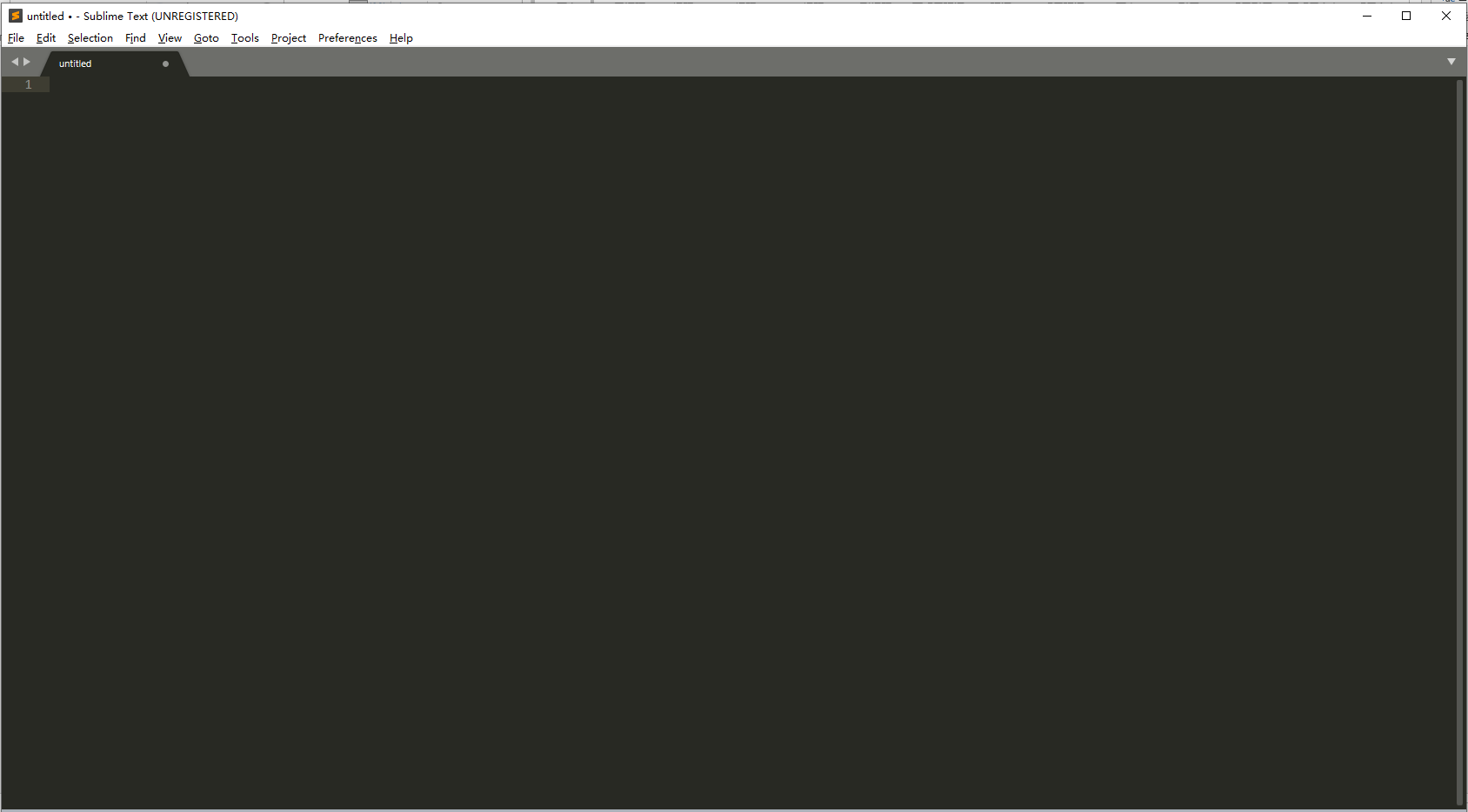
20/5



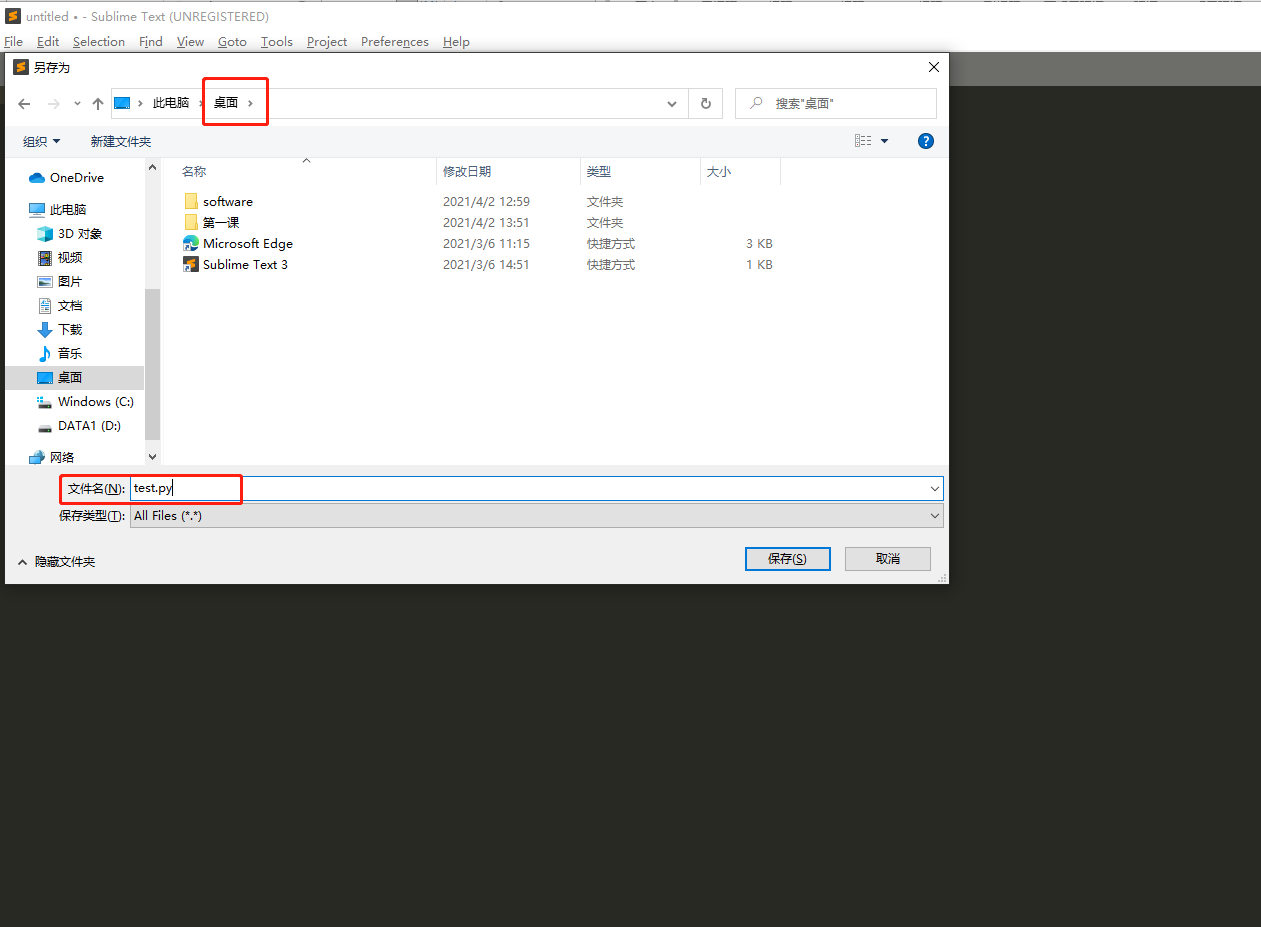
当然随着我们要编写的程序代码越来越多在这个控制台窗口里显然不是很方便，我们一般使用第二种执行方式

**命令模式**

我们打开一个编辑器，这里推荐使用简洁的Sublime Text3软件：

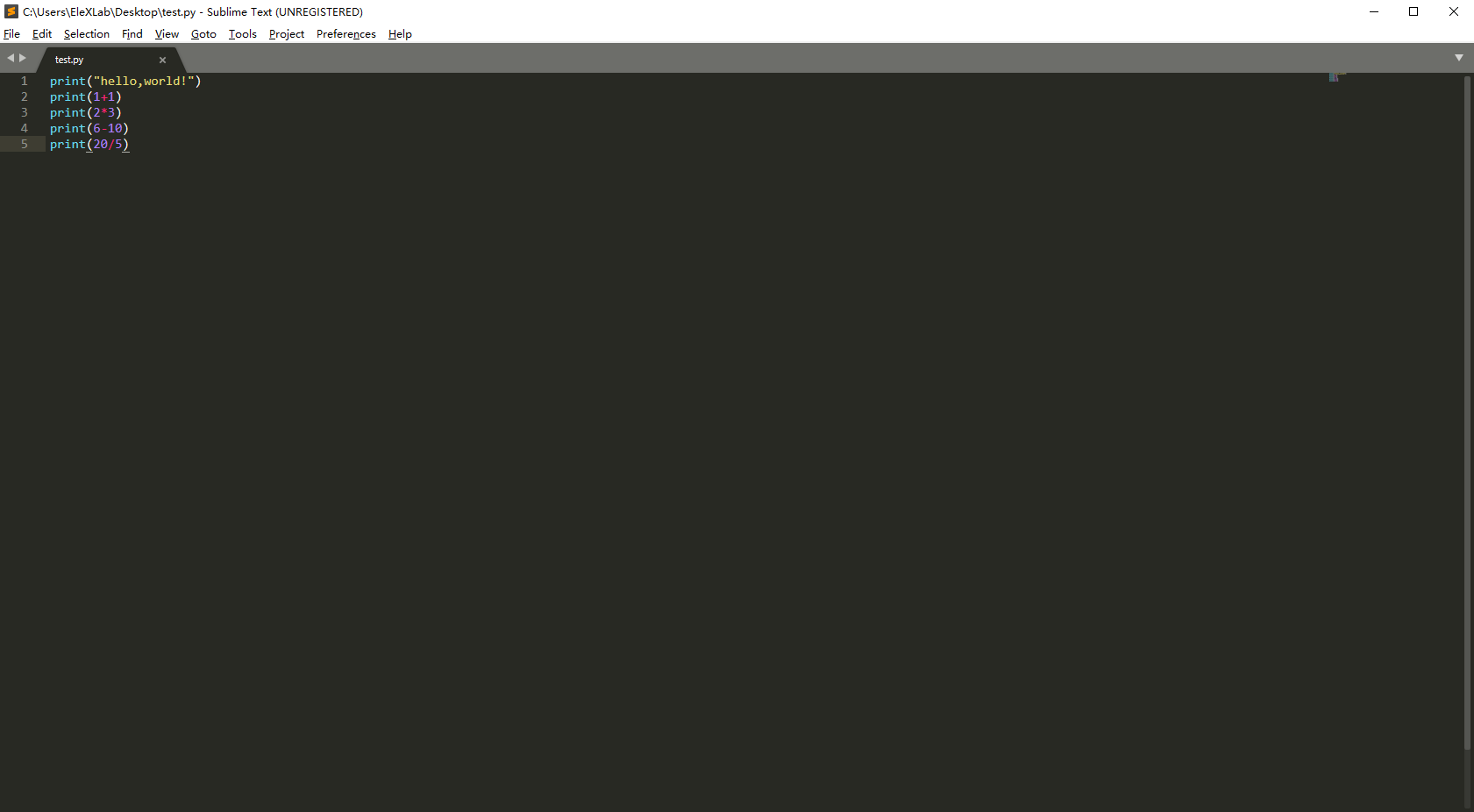


先用Ctrl+S（MacOS系统是CMD+S）把当前编辑的内容保存到电脑上，比如我们直接放在桌面上，命名为test.py:





在test.py源码文件里编写我们的程序：



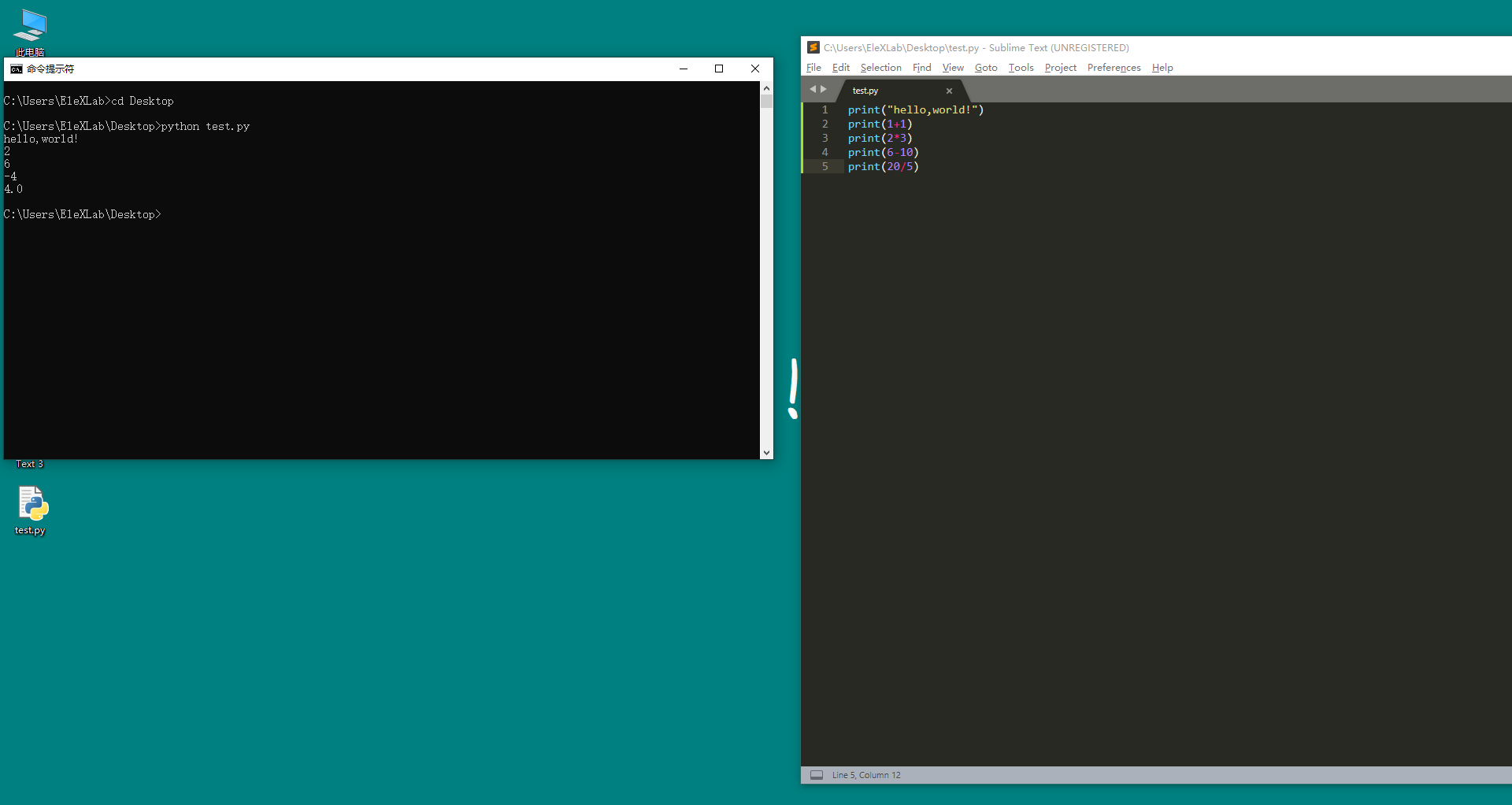
写完后记得按Ctrl+S保存。

然后打开CMD命令窗口，通过cd命令进入test.py文件所在目录



用python命令执行这个程序：

python test.py



**2.Python基本语法介绍**

说明：

Python语法的全部细节还是非常的庞杂的，在这里我们只**简略的**介绍一些课上常用的语法规则，更全面和详细的语法介绍可以查阅python3的官方文档：

<https://docs.python.org/zh-cn/3.8/>

当然，国内的“菜鸟教程网“也提供了十分优秀的python3入门文档：

<https://www.runoob.com/python3/python3-tutorial.html>

一点说明： 如果你已经有了其他语言的编程基础但是没有编写过python程序，这里强烈建议你不看本文档后面的内容而是直接去选择浏览上述教程/文档，那里有更加详细和系统的语法介绍。

**变量和基本数据类型**

扩展阅读：[https://www.runoob.com/python3/python3-data-type.html](https://www.runoob.com/python3/python3-data-type.html" \t "_blank)

python非常简单，以至于变量完全不用申明，在需要使用的地方直接赋值就可以创建，同样，我们创建的时候也不用考虑它的类型，虽然python是有类型的，但是，你不必去管，直接去写就好。

我们用“**=**“来给一个变量赋值：（在python中**#**后属于代码注释）

i= 100 # 整型变量

f= 1000.0 # 浮点型变量

s= "hello" # 字符串

b = True #布尔变量

print (i)

print (f)

print (s)

print (b)

输出：

100

1000.0

Hello

True

上面的代码中我们创建了4种常见类型的变量：整型，浮点型，字符串和布尔型，并把他们打印了出来。

**列表**

扩展阅读：[https://www.runoob.com/python3/python3-list.html](https://www.runoob.com/python3/python3-list.html" \t "_blank)

在实际编程中几乎到处都有“处理一组数据“的情况，这时候我们就需要一种承载一组数据的方法，在其他语言中（比如c/c++,java等）它们称之为数组，而在python中叫做列表。

比如，我们用一个列表来装5个整型数据：

a=[1,2,3,4,5]

print(a)

输出：

[1, 2, 3, 4, 5]

我们可以用索引值来取出列表中的某个元素，比如取出第3个（注意，索引从0开始）

print(a[3])

输出：

4

我们也可以通过索引去替换某个位置的数据：

a[3]=18

print(a[3])

输出：

18

**条件**

同样，我们在实际编程中几乎到处都有“条件判断”的情况，比如一个游戏程序中“如果角色的血量小于0，那么角色就死亡”，我们用if（如果） else（否则）语法来实现条件判断。

blood=100 #血量变量，值为100

if blood<0: #如果血量小于100

print("dead")#打印“死亡”

else: #否则

print("live")#打印“或者”

输出

Live

当然我们也可以用elif（否则如果）判断多种情况，比如“角色血量小于50，属于受伤状态”。

blood=29 #血量变量，值为100

if blood<0: #如果血量小于100

print("dead")#打印“死亡”

elif blood<50:#如果血量小于50

print("hurt")#打印“受伤”

else: #否则

print("live")#打印“或者”

输出

hurt

**循环**

扩展阅读：<https://www.runoob.com/python3/python3-loop.html>

我们在编程的时候会经常遇到这样的一种情况：不断的重复执行某些动作。也就是循环。

**while循环**

while的意思为当,当一个条件成立的时候，就执行。

while 条件: #当条件

执行... #就不断的执行这里的代码

比如:

while 2>1:

print("hello")

输出：

hello

hello

hello

hello

hello

hello

hello

hello

hello

.

.

.

当然，我们也可以用一个变量来控制循环的次数

c=5

while c>0: #如果c大于0

print(c)

c=c-1 #每次把c减1

输出

5

4

3

2

1

我们在循环内每次把c的值减1，这样最后循环条件c>0在c减为0的时候不成立，跳出循环结束。

**for 循环**

除了while循环以外还有常用的for循环语法，这个for是forearch（遍历）的意思，用来遍历一个列表数据十分方便，他是这样的格式：

for 元素 in 列表：

执行...

举例：

a=[1,2,3,4,5] #数组a

for i in a: #从数组中遍历元素，每次得到的元素用i表示

print(i) #打印这次取出的元素

输出

1

2

3

4

5

for  in语法可以依次从列表中取出元素，并使用这个元素，非常适合去遍历一个列表。

**range() 函数**

我们发现有时候我们经常需要按照实际需求进行某些次数的循环，遍历，当然我们可以使用while循环+控制次数的变量来做，但是有点麻烦。python中有个range()函数，配合for in语法可以很方便的实现这一点。

for i in range(5):

print(i)

输出

0

1

2

3

4

range()函数生成了5个数

当然我们也可以指定range生成的起始和结束范围

for i in range(5,9): #从5开始到9结束(不包括9)

print(i)

输出

5

6

7

8

当然我们也可以指定range生成的"步长"

for i in range(10,21,3): #从10开始到21结束，每次递增3

print(i)

输出

10

13

16

19

**函数**

扩展阅读：<https://www.runoob.com/python3/python3-function.html>

我们在实际编码的时候会经常会遇到“某个功能的代码”反复使用的情况，比如下面这个“取出列表中奇数索引的元素”的代码

a=[1,2,3,4,5]

for i in range(1,len(a),2):

print(a[i])

输出

2

4

如果代码中很多地方都需要“取出列表中奇数索引的元素”，那么重复的去编写这些代码有点不方便，我们使用函数来封装一个固定功能的代码块。

def pick(a):

b = []

for i in range(1,len(a),2):

b.append(a[i])#append是向列表中添加元素

return b

print(pick([1,2,3,4,5]))

print(pick([11,24,67,46,59]))

输出

[2, 4]

[24, 46]

一个函数用def（定义）关键词开头，后面的()用来接收调用时候传递的参数，这里接受一个叫做a的数组参数，最后的return表示函数处理之后的返回值，这里我们返回从a中选取的奇数元素构成的新数组b。

**模块**

扩展阅读：[https://www.runoob.com/python3/python3-module.html](https://www.runoob.com/python3/python3-module.html" \t "_blank)

正所谓大树底下好乘凉，巨人肩膀上能飞翔，python自诞生之日起经过这么多年的发展累计了一浪又一浪的众多开发者贡献的代码，不出意外，我们能想到的功能基本上都会有别人已经实现好的现成代码，那把它们直接拿过来使用就是一件很重要的事情了。

在python中通过import关键字导入一个模块库，比如我们第一节课中就导入了两个模块:

import numpy as np

import matplotlib.plot as plt

通过import导入后我们一般习惯用as关键字来给这个库一个简单的别名以方便下面使用的时候少打几个字母。

**numpy库**

numpy库在我们的机器学习课程中十分重要，基本上任何一个地方都需要用到它来做数学计算，这里强烈推荐你直接去阅读numpy的官方中文文档，可以不那么仔细的浏览一遍，知道它大致功能，在实际编程中再去详细的查阅细节：

numpy中文文档：

<https://www.numpy.org.cn/user/>