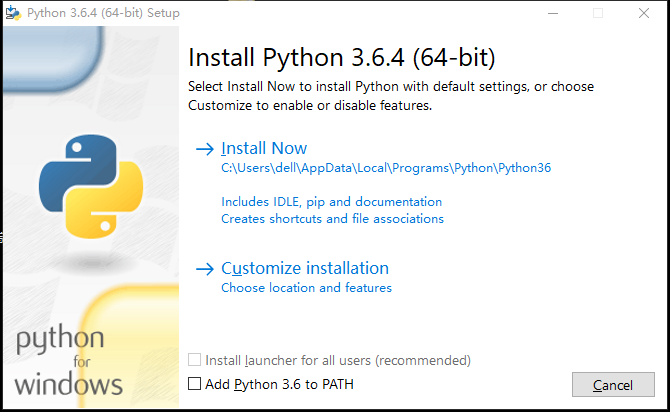
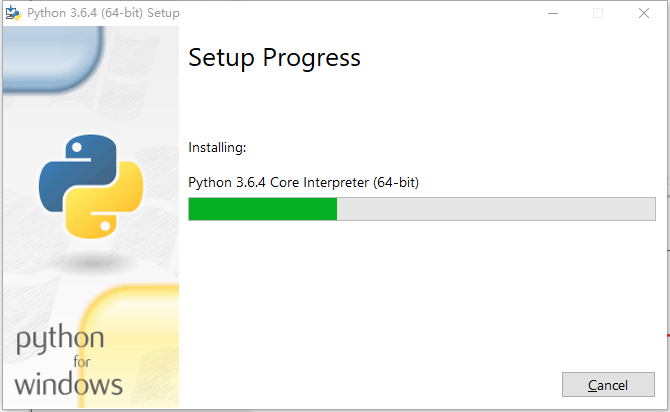
1. 环境配置
   1. 安装Python3

1.1.1 基本步骤

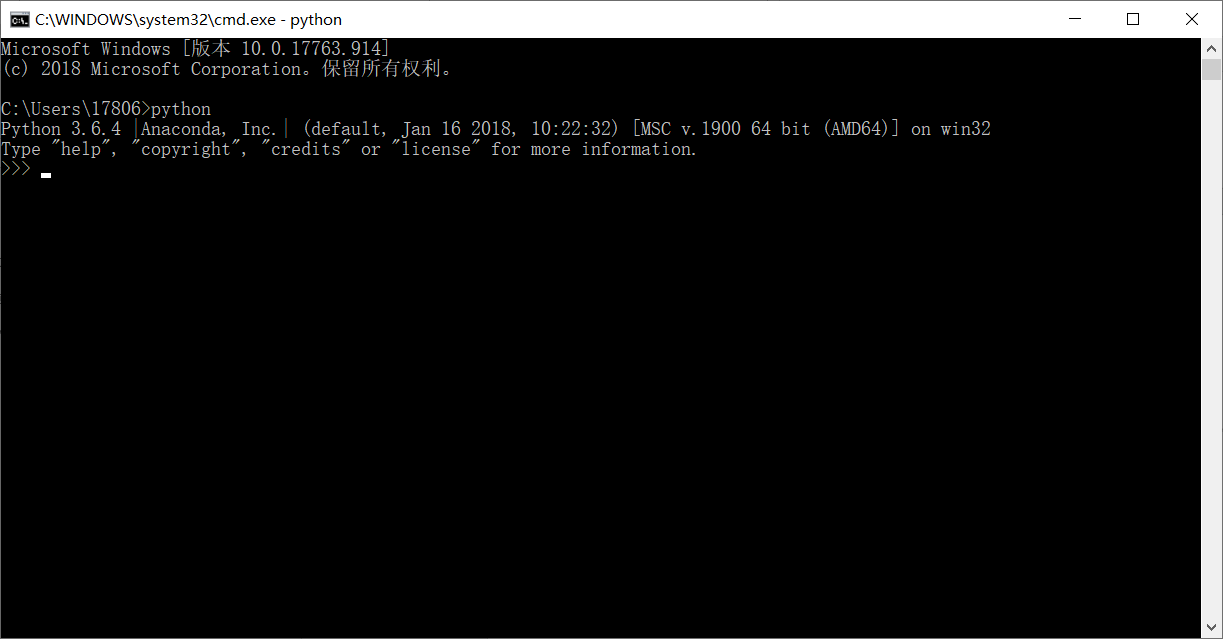
去python官网，找到我们需要的python3版本，进行下载，勾选Add Python 3.6 to Path，这样就能加入系统环境变量，然后点击Install Now。等待安装完成即可。





1.1.2 安装测试

按下快捷键“win+r”,直接输入“cmd”，打开命令管理器后输入python，出现以下界面代表安装成功。



1.1.3 不成功解决方案

若不成功则需要配置环境变量

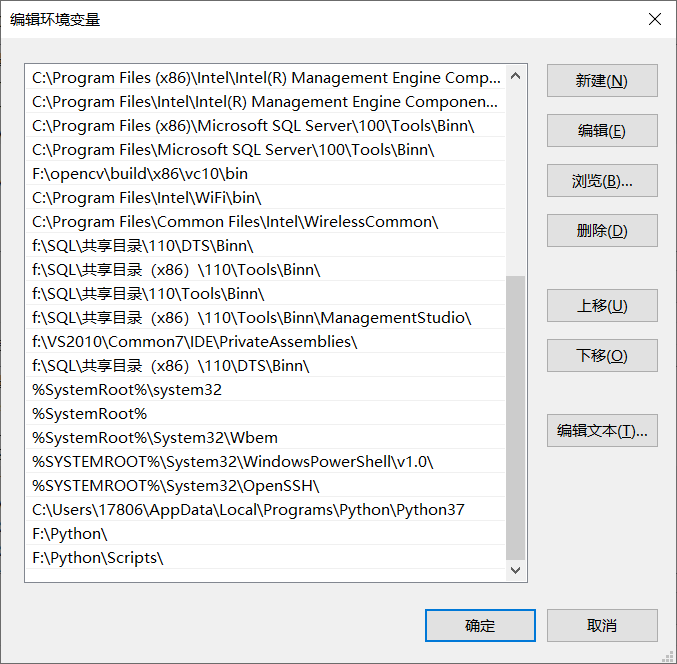
首先，右键单击“此电脑”，选择属性

在左侧选择高级系统设置，之后选择环境变量

找到系统变量中的Path，单击选中，之后点击编辑

点击右侧的新建，将你的python完整安装路径填进去即可，例如我的是 “F:\Python\”（填到文件夹的位置即可）

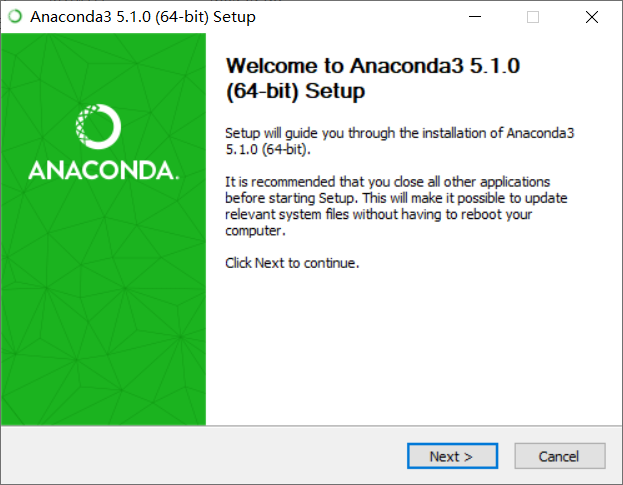
点击新建，复制上一个的路径，在后面加上Scripts即可，例如我的就是 “F:\Python\Scripts\”

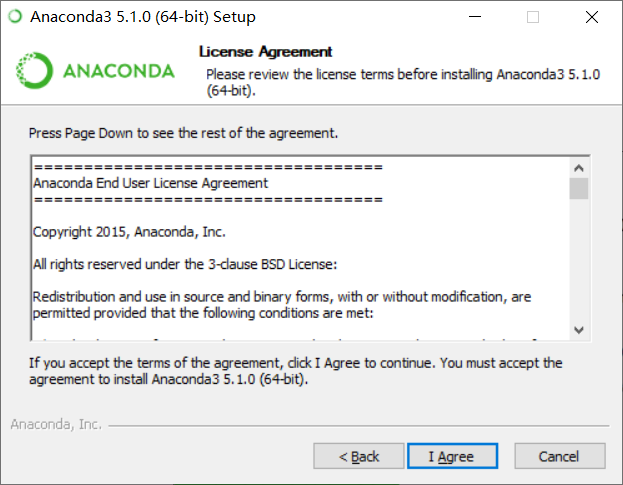


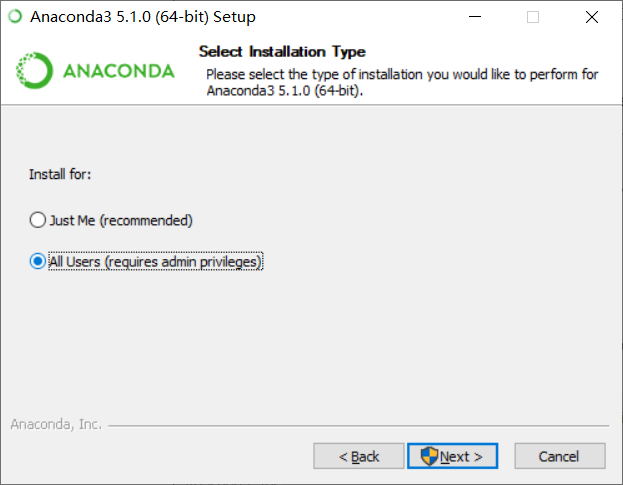
1.2安装anaconda

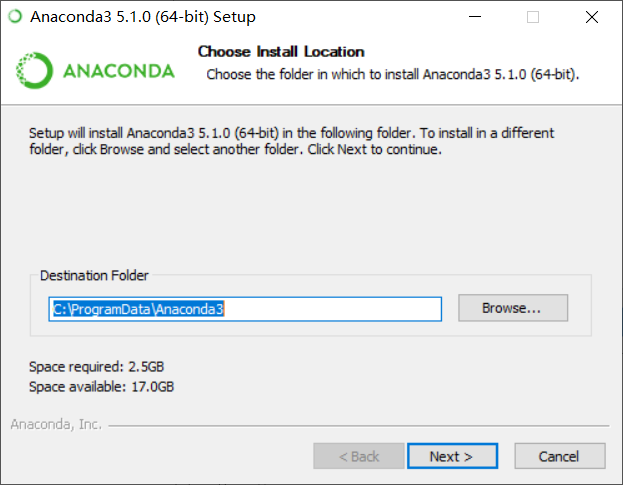
1.2.1 安装步骤

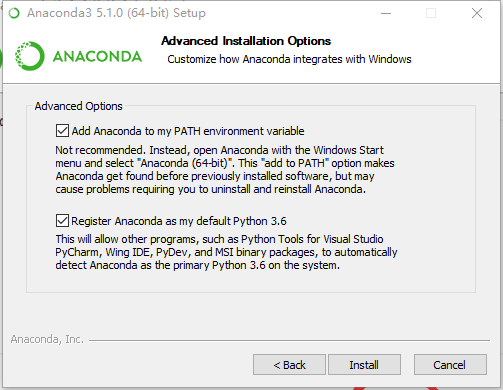
如下图

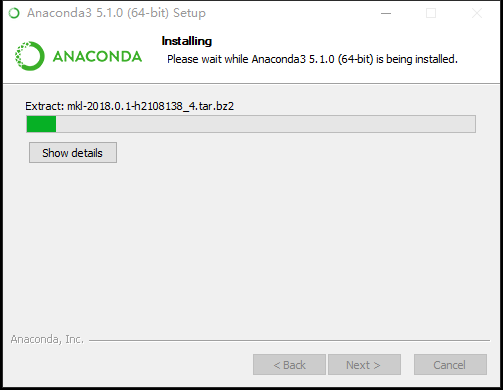












1.2.2 安装测试

打开CMD，输入comda，不报错就代表环境变量已经加入了anaconda。

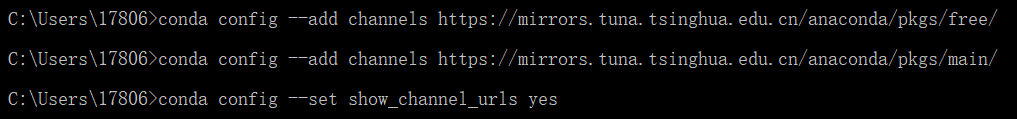
1.2.3 conda的基本配置

首先更改anaconda的下载源，改为清华下载源，在命令行输入：

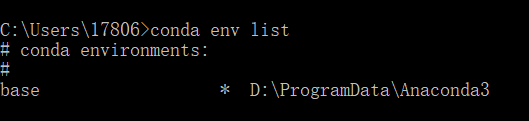
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/

conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/

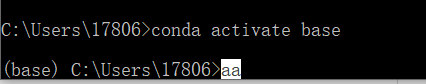
conda config --set show\_channel\_urls yes



输入conda env list，由于第一次安装anaconda，此时的环境只有一个已有的base环境，可以创建一个新的环境也可以在当前环境操作：

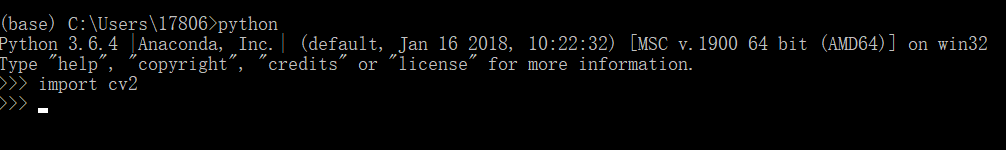


输入：conda activate base启动base环境：



然后输入pip list可以查看已经安装了的python库

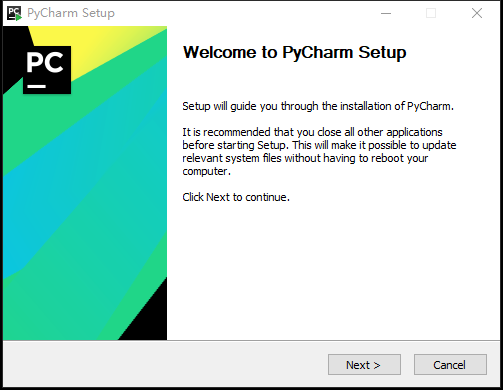
安装opencv-python：在base环境下，输入pip install opencv-python，等待安装完成以后，输入python，进入python编译环境，输入import cv2，不报错就安装成功

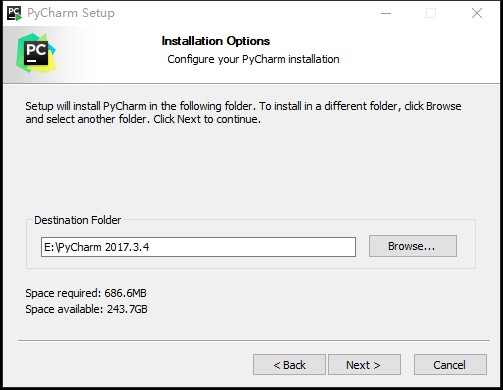


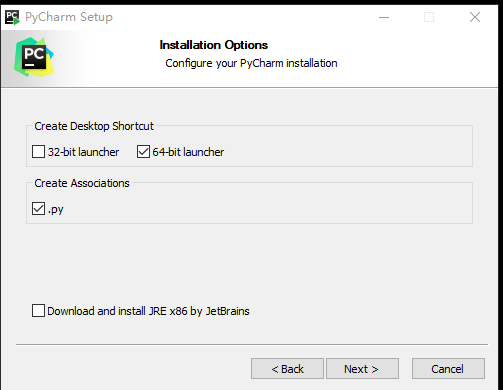
* 1. 安装pycharm

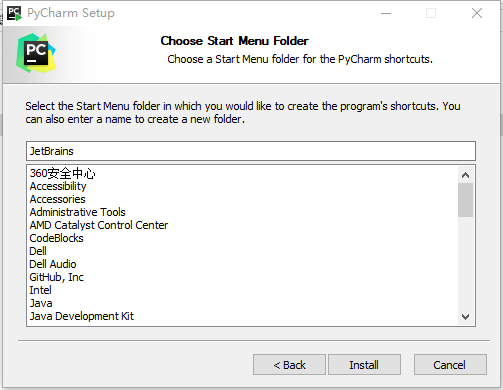
1.3.1安装步骤

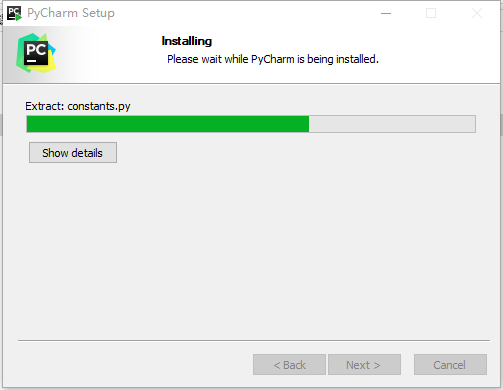
搜索下载pycharm，按以下步骤操作即可

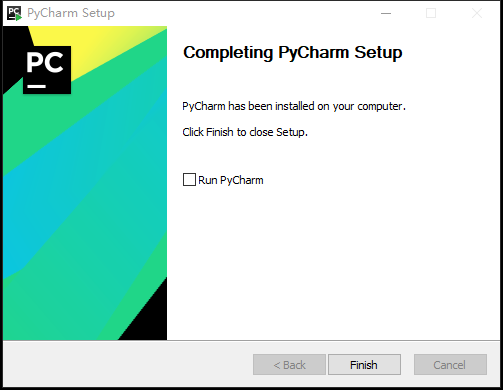




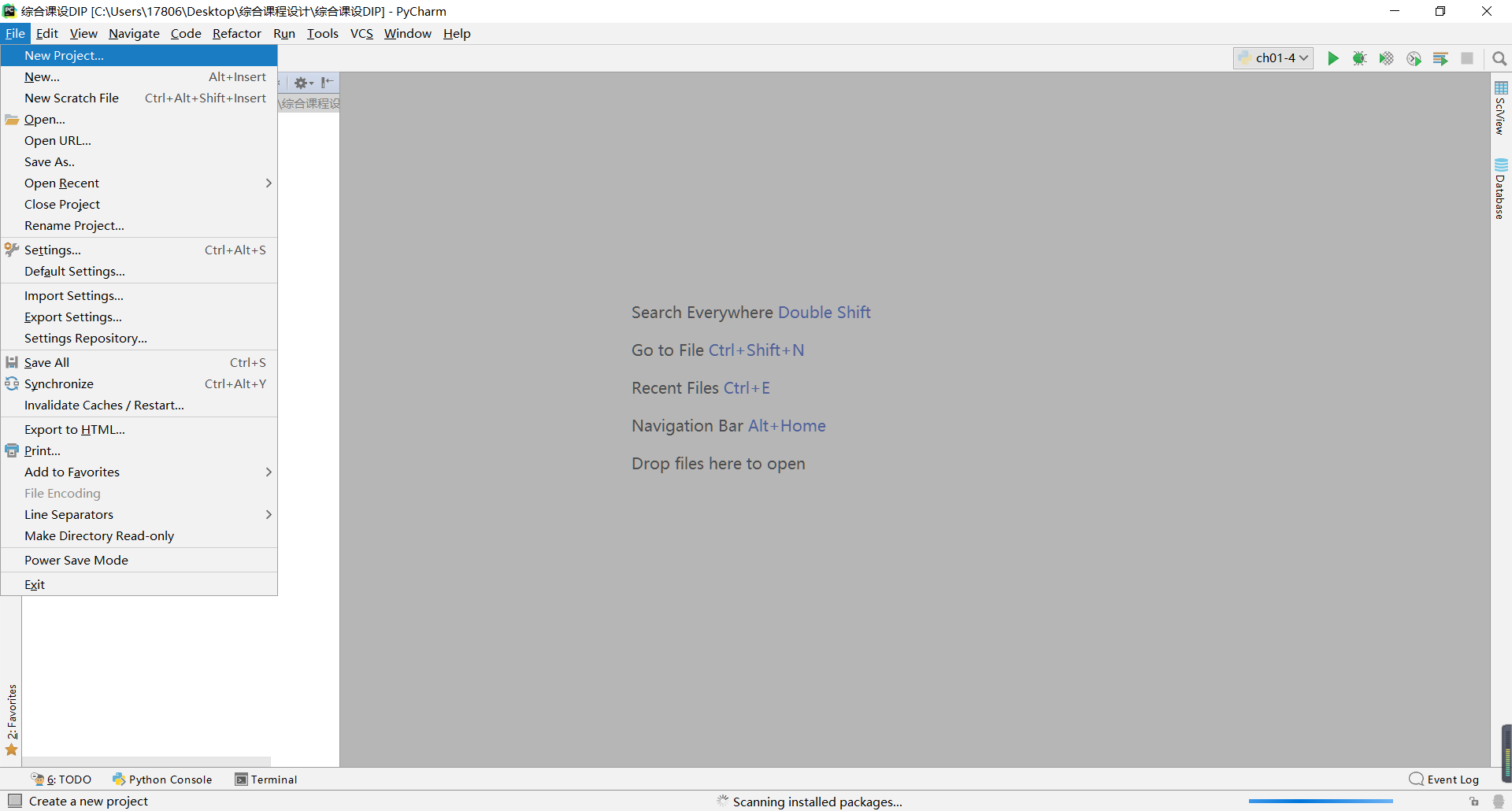


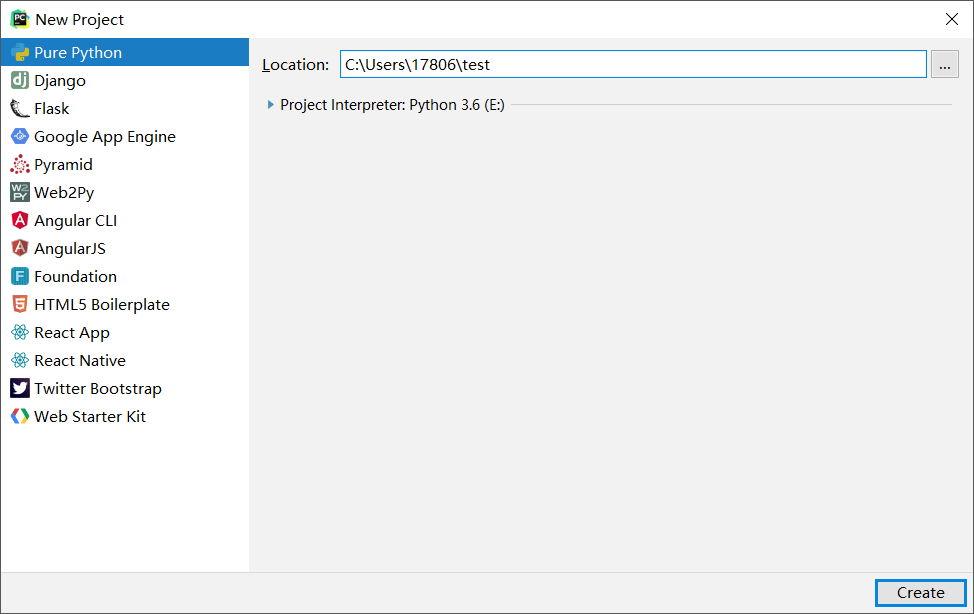


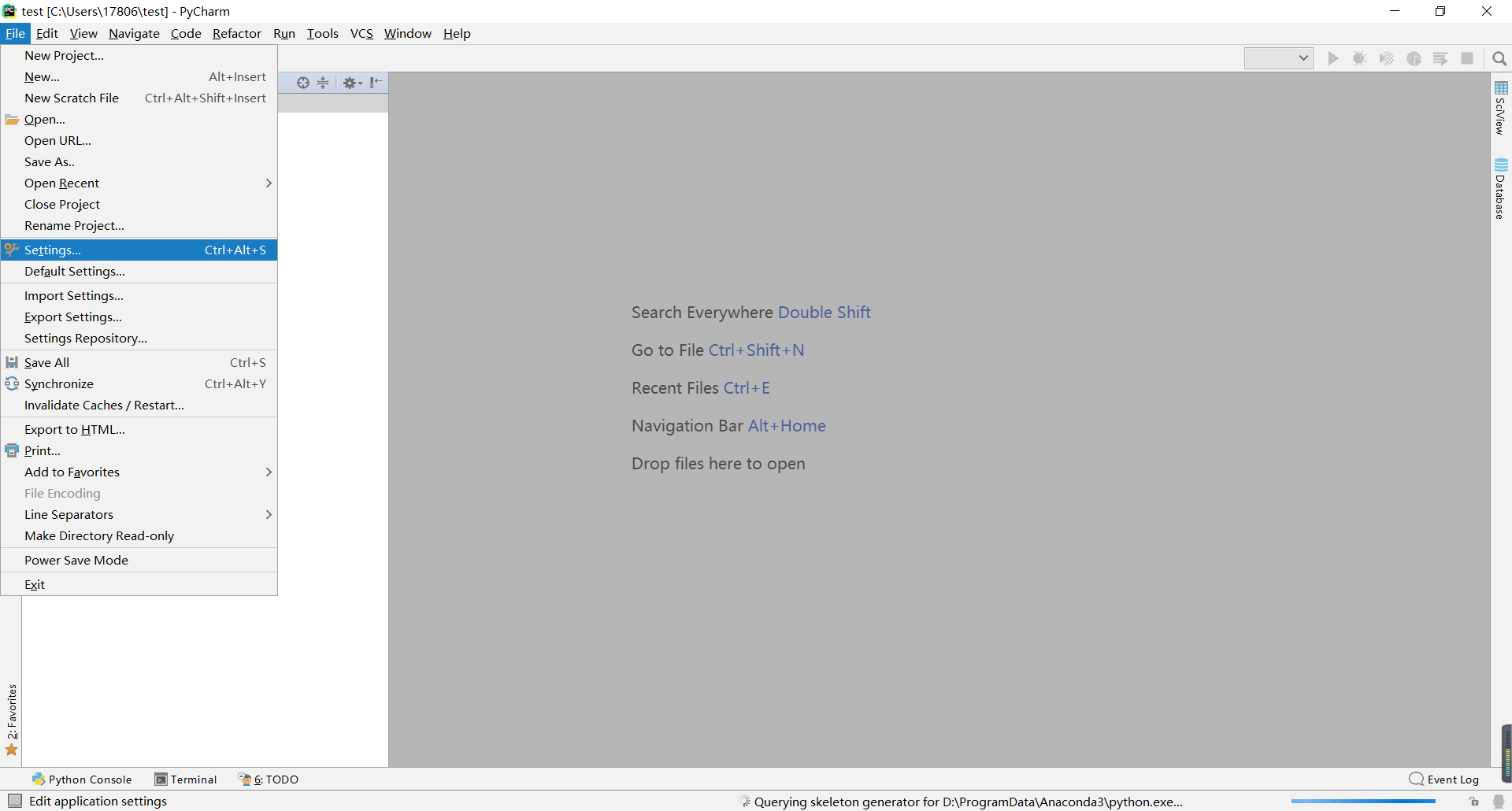


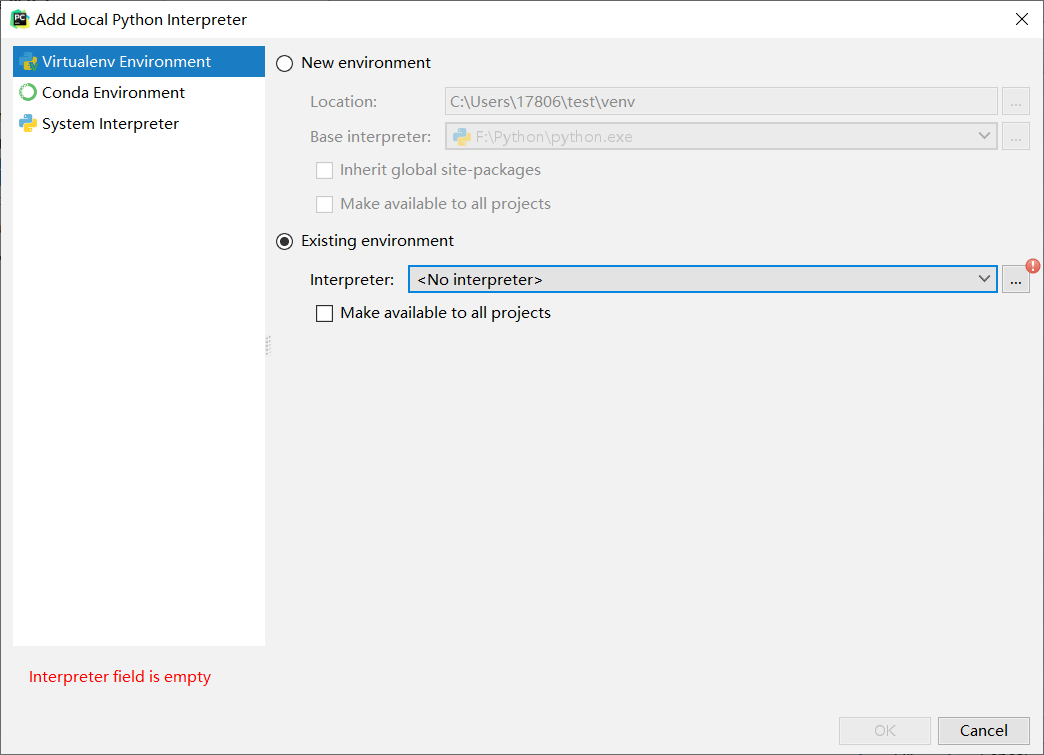


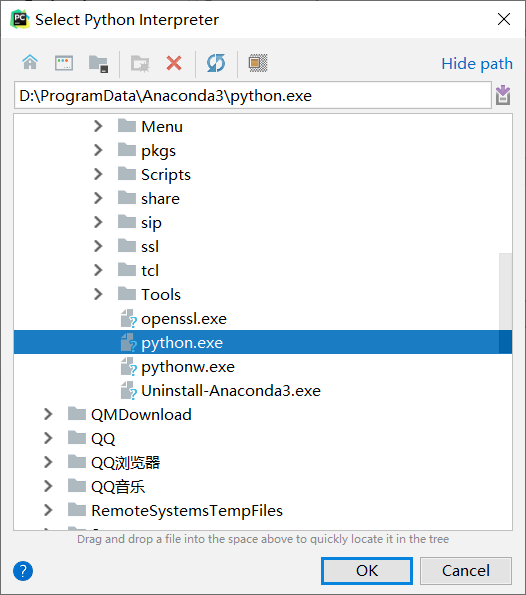
新建工程

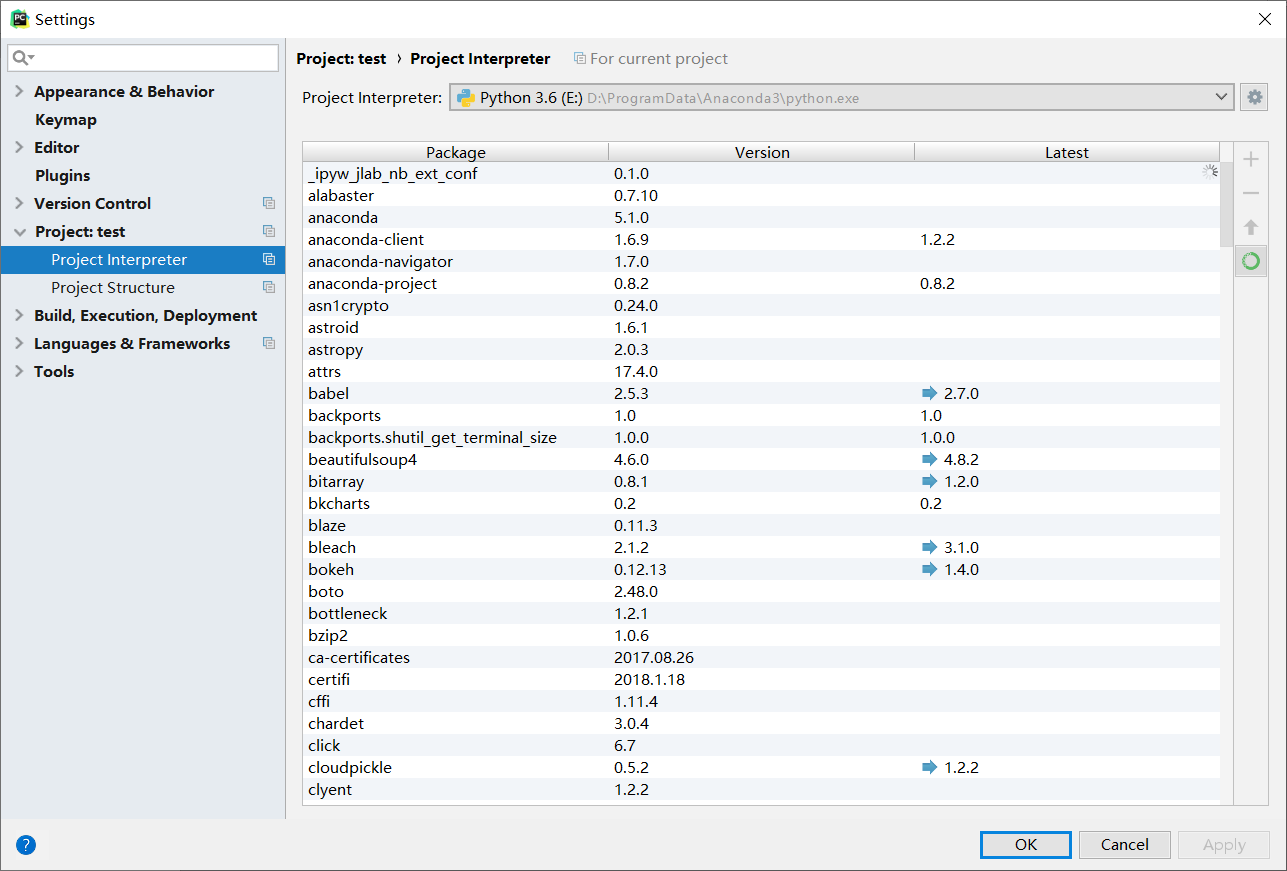


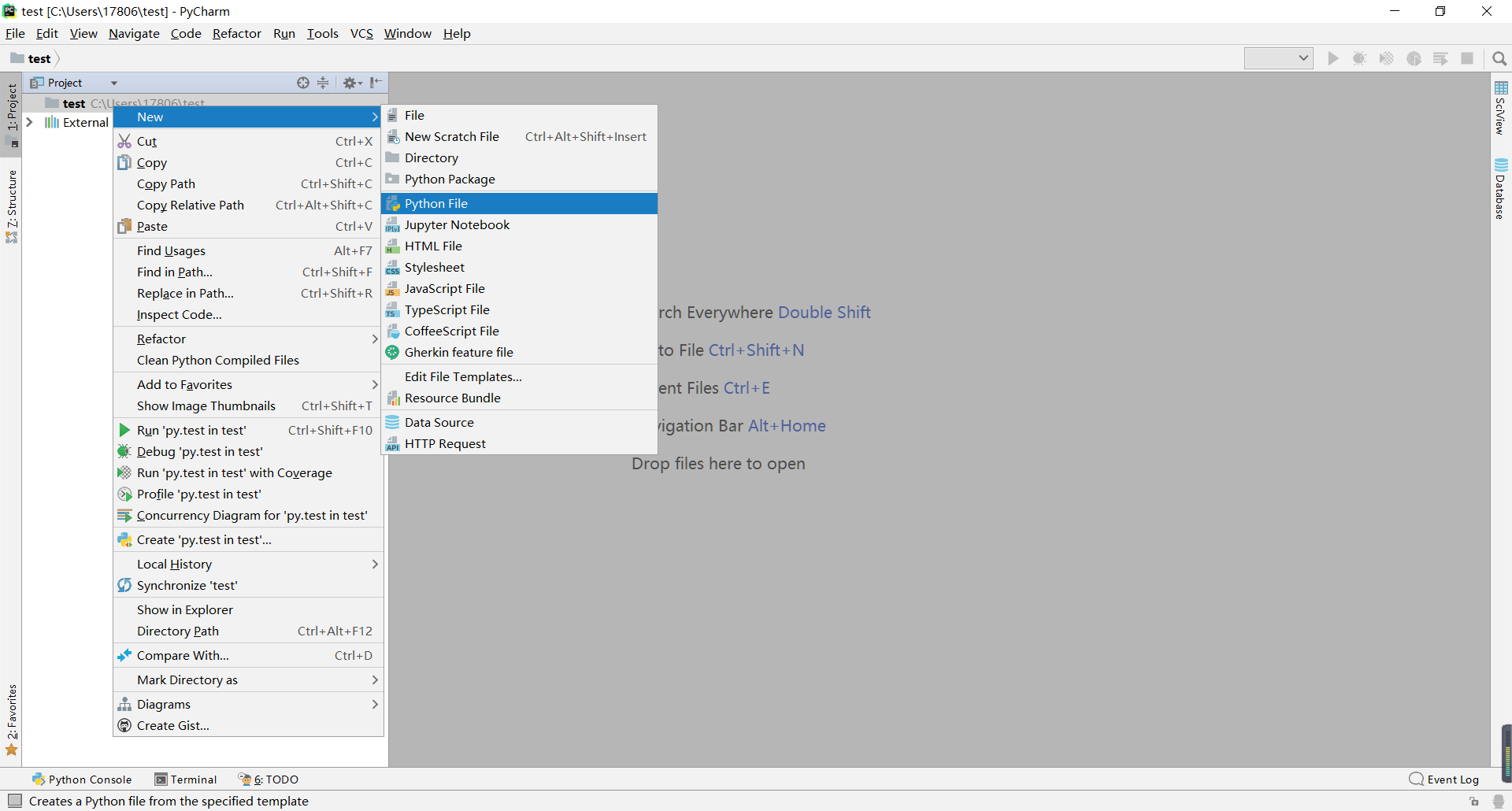


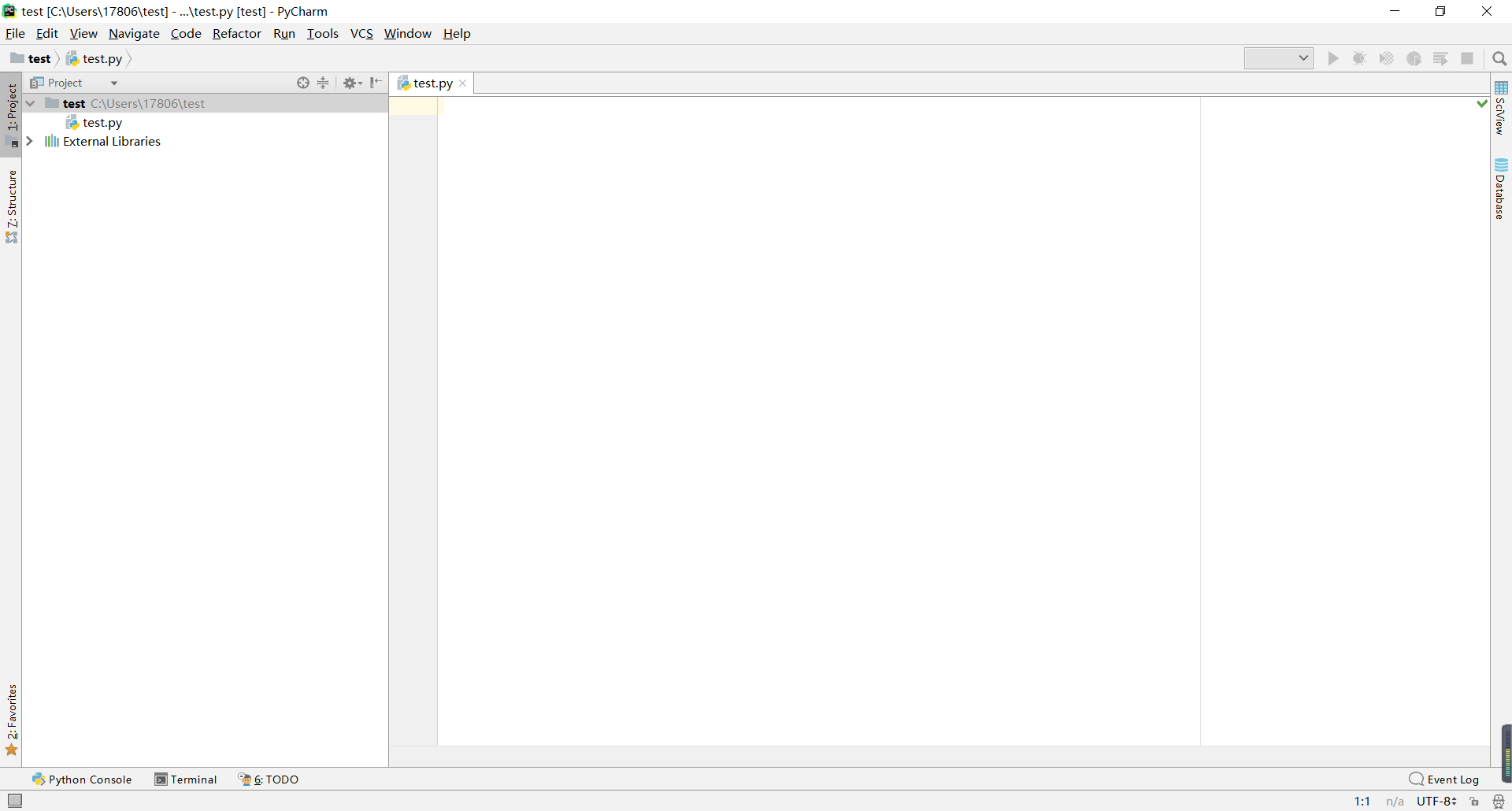












二、

实验1 图像加载与显示（验证性）

## 实验目的

1. 理解并掌握OpenCV开发库的功能和使用方法，学会在Python3环境下安装和配置OpenCV。
2. 能运用OpenCV的图像处理基本函数，开发图像加载与显示的程序，实现指定功能。
3. 能对实验过程和结果正确分析，找出问题，得出结论。

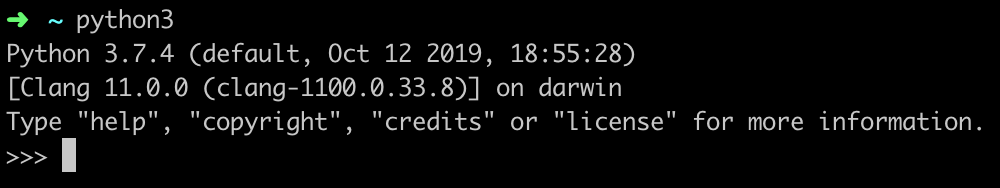
## 实验内容

1. 编写程序掌握OpenCV加载及显示图像的方法。
2. 查看、理解并掌握图像的属性（包括类型、大小、分辨率等）。

## 实验步骤（验证性实验）

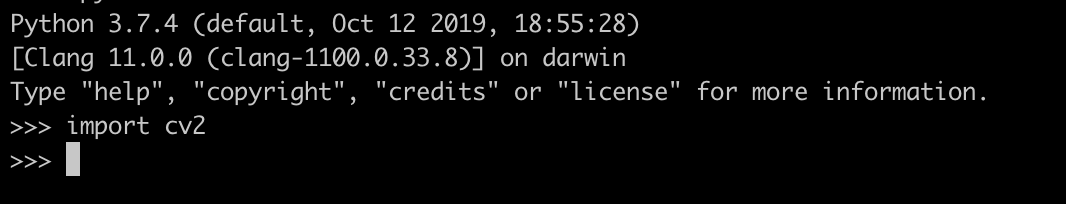
1. 检查所用的计算机系统是否已安装了Python3。

命令行输入：python3



1. 检查计算机是否已经安装opencv-python库。

输入import cv2，若没有报错，则说明已经安装



1. 编写代码并配置opencv程序，完成程序的编译、链接和运行。步骤如下：

1. **在桌面上新建文件夹，以存放将要写的程序，如：ex1。**

****

1. **在pycharm里新建项目**

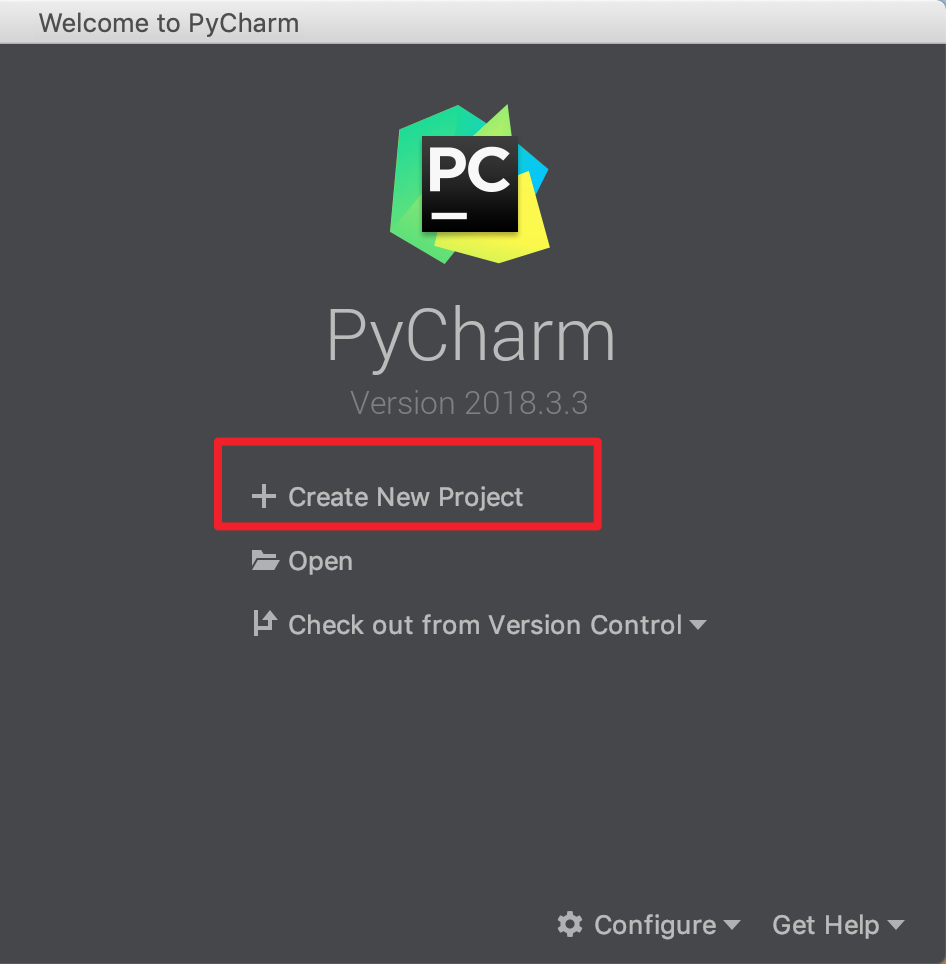
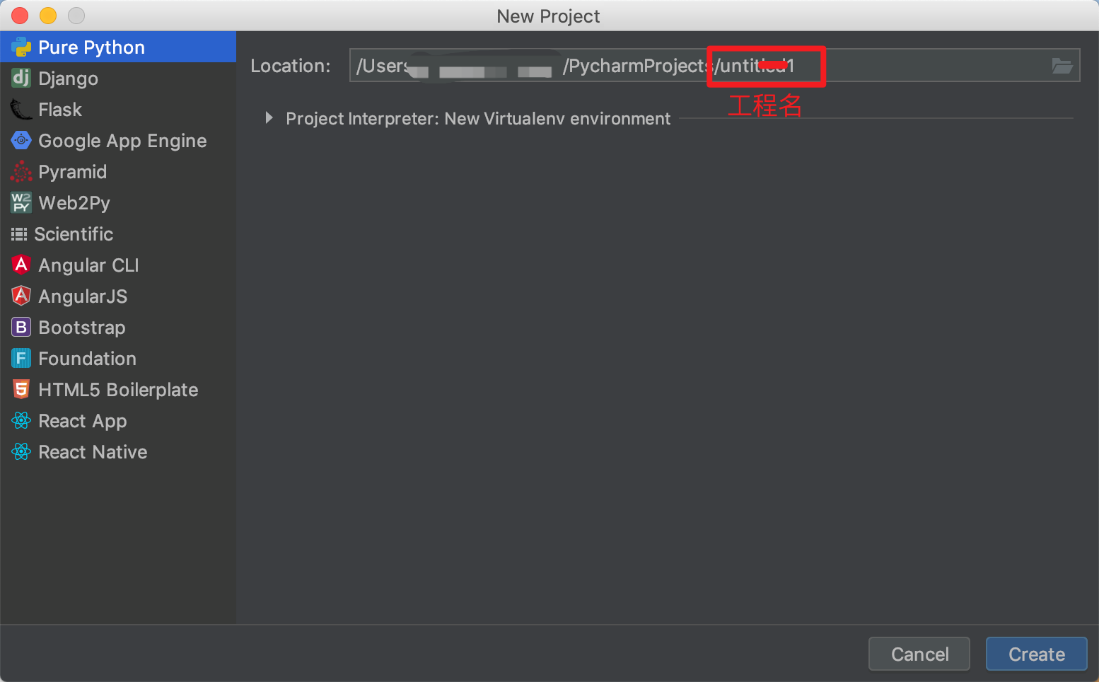
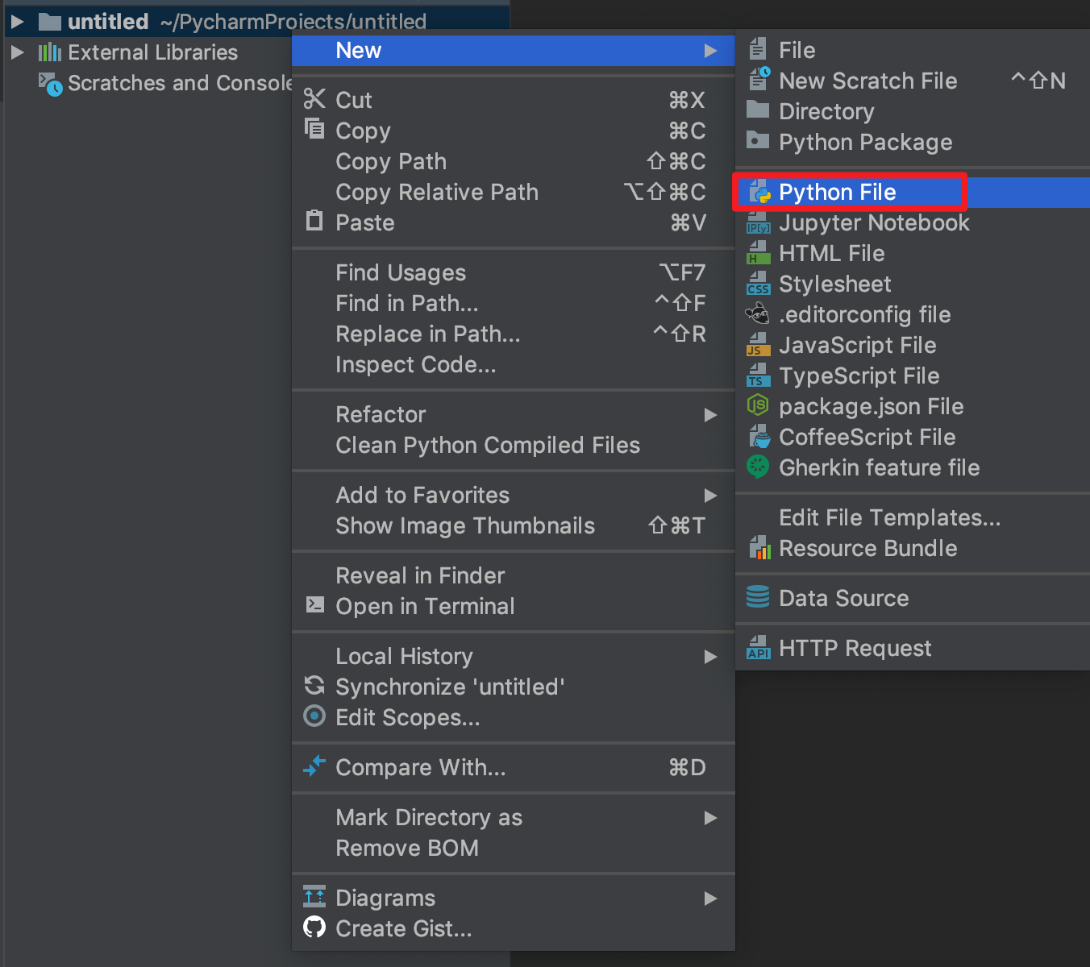
****

图1 在pycharm中新建项目

****

右键创建Python文件

****

1. **编写程序并运行**

调通并实现以下三个函数

主要函数：cv2.imread(filename, flags)，cv2.imshow(winname, mat)，cv2.waitKey(delay)

参数解释：

1. filename：读取文件的名字，或者是路径加名字，例如/user/desktop/cat.jpg
2. flags：imread函数的参数标志，默认为1，0为灰度读图
3. winname：显示窗口的名字
4. mat：图像变量的名称，由于python不需要去定义类型，例如

image = cv2.imread(‘cat.jpg’)，image就是mat类型的引用

1. delay：延迟或者等待的时间，0为一直等待

给出部分代码

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

程序名：ch01-1.py

功 能：读入工程目录下指定文件，并在屏幕上显示

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

imgae = cv2.imread("cat.jpg")

cv2.imshow("windowsName", imgae)

注释：该方法需要在工程内加入cat.jpg文件

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

程序名：ch01-2.py

功 能：读入相对路径下指定文件，并在屏幕上显示

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#获取脚本所在位置

root\_path = os.getcwd()

#获取cat.jpg所在位置，即资源文件位置

cat\_path = root\_path + '/../source/'

#获取cat.jpg名字

imageName = os.path.abspath(cat\_path + 'cat.jpg')

imgae = cv2.imread(imageName)

cv2.imshow("windowsName", imgae)

注释：该方法每个机器都会不同，需要自己调通

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

程序名：ch01-3.py

功 能：读入绝对路径下指定文件，并在屏幕上显示

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

image\_path = '路径/cat.jpg'

imgae = cv2.imread(image\_path)

cv2.imshow("windowsName", imgae)

注释：该方法获取的是cat.jpg的绝对路径

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

程序名：ch01-4.py

功 能：采用不定参数的形式，实现多图读取

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import cv2

def imageLoad(\*args):

num = 0

for image in args:

imageName = 'srcImage' + str(num)

num = num + 1

srcImage = cv2.imread(image)

cv2.imshow(imageName, srcImage)

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

imageLoad("cat.jpg", "football.jpg")

cv2.waitKey(0)

## 填空及简答

1. OpenCV 2.0及以上版本采用\_\_\_\_\_\_\_结构体作为图像数据结构。
2. os.getcwd()函数的意义，获取的是什么位置
3. 调通ch01-2.py时，遇到的问题是什么？怎么解决的？收获了什么？

## 编程练习

【基础练习】试着不用参考给定例程，自己写出本次实验的代码，掌握OpenCV图像处理程序的基本流程和框架。

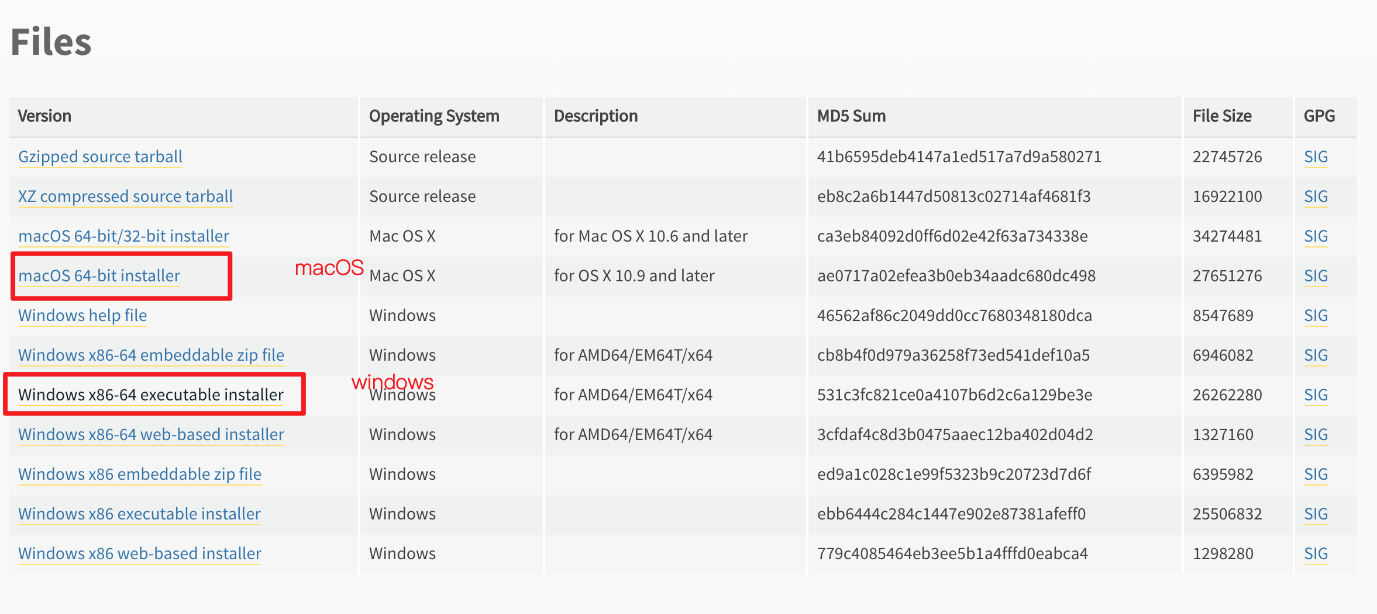
【扩展练习】结合给出的ch01-4.py。编程实现加载并显示三幅图像，其中一幅通过指定文件名的方式读取，一幅图片通过用户键盘输入指定文件，最后一幅通过用户输入资源文件夹下任一一个图片名。**（本题要求写实验报告，格式要求见目录页），并解答以下问题：\*args这个参数如何解释，\*号在Python中的作用，两个\*\*号呢？**

例如：用户键盘输入指定文件：/use/desktop/football.jpg，回车后显示相应的图片

用户输入资源文件夹下任一图片名称：football.jpg，回车后显示相应的图片

1. **Python安装及环境配置**
   1. **Python安装**

**网址：**<https://www.python.org/downloads/release/python-370/>

****

* 1. **pycharm下载安装**

**网址：**<https://www.jianshu.com/p/9797be7874d6>

链接是mac系统的，但是Windows一样

**Pycharm官网网址：**<https://www.jetbrains.com/pycharm/>

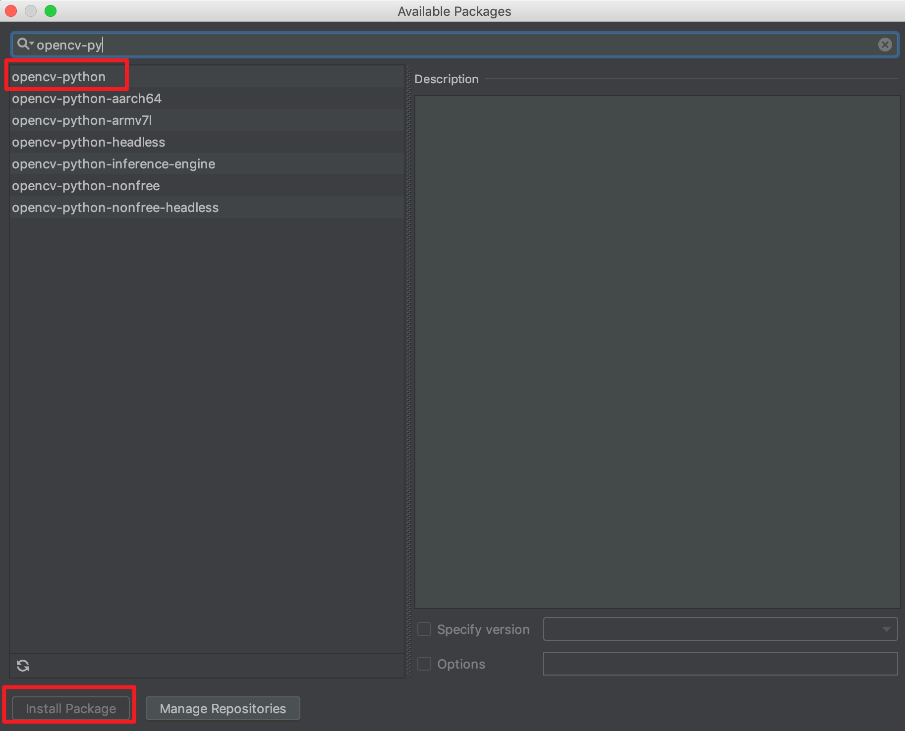
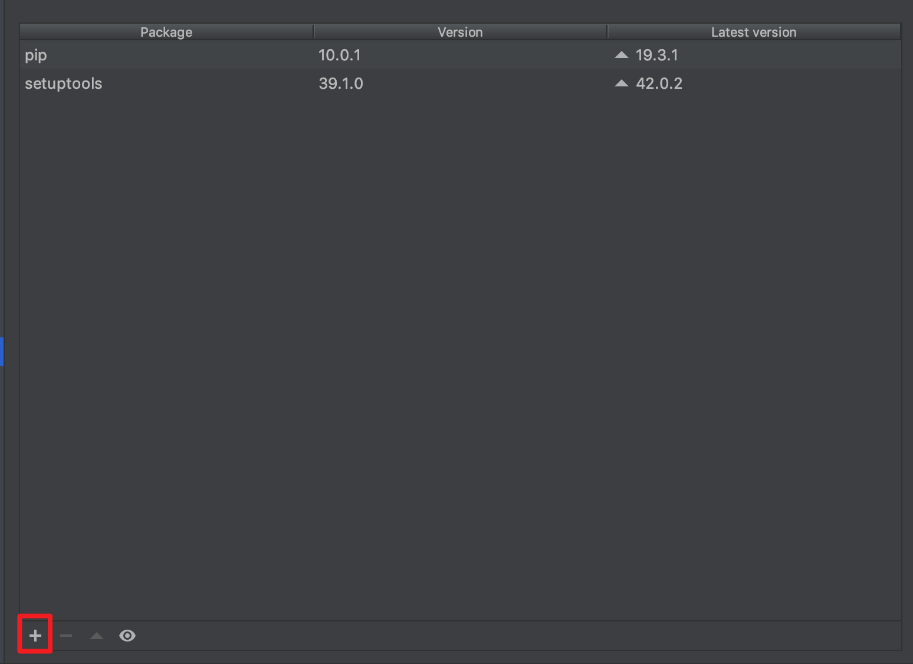
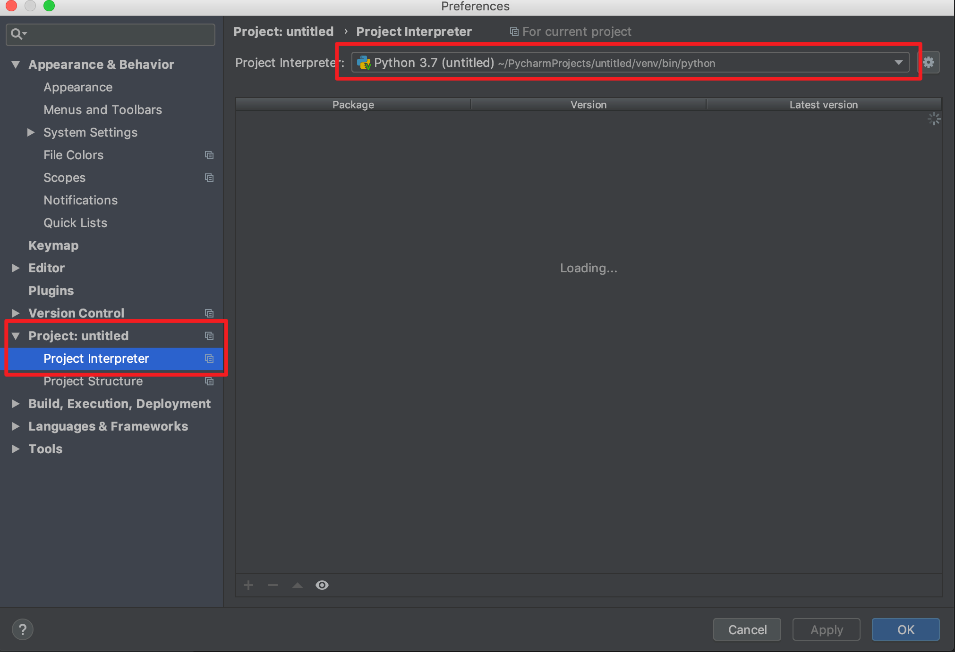
****

* 1. **opencv-python库安装**

1. **Windows系统**

**方法一：打开命令行cmd，输入pip install opencv-python，等待安装即可**

**方法二：打开pycharm，创建好工程后，打开设置**

****

**选择安装即可。**

**方法三：采用anaconda来控制环境，这个适用于Python用的特别多的，挺好用，网上也有教程。**

1. **macOS系统**

**方法一：在终端输入pip3 install opencv-python**

**方法二（推荐）：安装homebrew后，终端输入，brew opencv-python，配置问题可自行百度。**

**方法三：采用anaconda。**