

报告二

王瑞洁 24020007127

2025.09.15

1 实验内容

1.1 python 入门基础

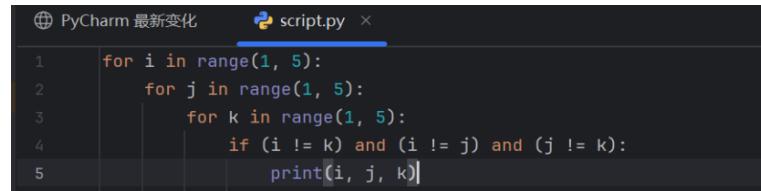
1.2 python 视觉运用

1.3 命令行环境

2 实验实例（20）

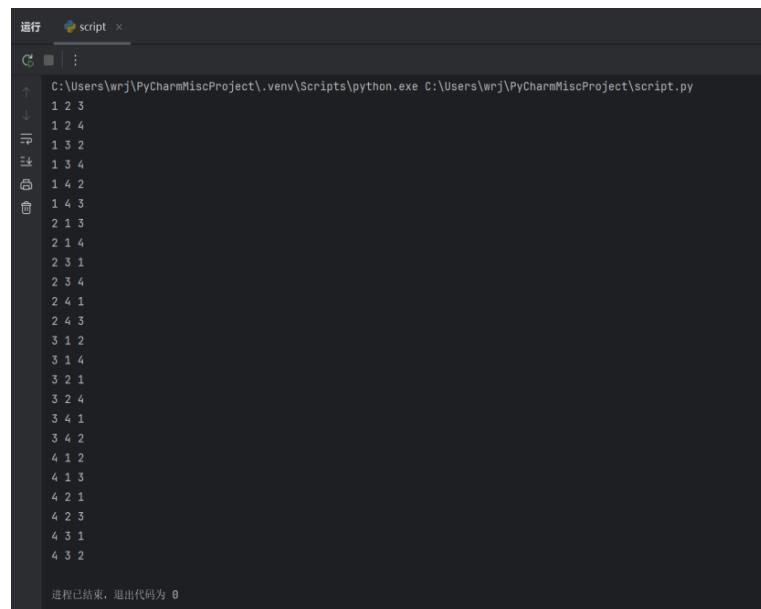
2.1 实例一

题目：有四个数字 1、2、3、4，能组成多少个互不相同且无重复数字的三位数？各是多少？



```
PyCharm 最新变化 script.py ×
1   for i in range(1, 5):
2       for j in range(1, 5):
3           for k in range(1, 5):
4               if (i != k) and (i != j) and (j != k):
5                   print(i, j, k)
```

将个十百位数放入 1、2、3、4 组成的三位数字所有情况排出来，再根据无重复数字的要求筛选满足条件的三位数字。运行结果展示如下：



```
运行 script ×
C:\Users\wrj\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\wrj\PyCharmMiscProject\script.py
1 2 3
1 2 4
1 3 2
1 3 4
1 4 2
1 4 3
2 1 3
2 1 4
2 3 1
2 3 4
2 4 1
2 4 3
3 1 2
3 1 4
3 2 1
3 2 4
3 4 1
3 4 2
4 1 2
4 1 3
4 2 1
4 2 3
4 3 1
4 3 2
```

2.2 实例二

题目：计算奖金利润提成。利润低于 10 万，提成 10%；高于 10 万，低于 20 万，该部分提成 7.5%；高于 20 万，低于 30 万，该部分提成 5%；高于 30 万，该部分提成 1%。根据输入的利润求发放的奖金额。

The screenshot shows the PyCharm IDE interface. The project navigation bar at the top indicates 'PyCharm 最新变化' and shows files 01.py and 02.py. The code editor displays a script named 02.py:

```
i=int(input('净利润为: '))
arr=[300000,200000,100000,0]
rat=[0.010,0.050,0.075,0.100]
r=0
for money in range(0,4):
    if i>arr[money]:
        r+=(i-arr[money])*rat[money]
    print((i-arr[money])*rat[money])
    i=arr[money]
print('最终奖金总额: ')
print(r)
```

The run output window below shows the execution results:

```
C:\Users\wrj\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\wrj\PyCharmMiscProject\02.py
净利润为: 180000
6000.0
10000.0
10000.0
最终奖金总额:
16000.0
进程已结束, 退出代码为 0
```

arr 存放奖金计算临界点，对应奖金区间的利润比例存放在 rat 里，有四个临界点，循环四次，比较输入的利润值和计算临界点，累加各部分的奖金分成，并打印出来，循环结束后显示最终的奖金总额。

2.3 实例三

题目：计算一个整数，加上 100 后是一个完全平方数，再加上 168 后也是一个完全平方数。

The screenshot shows the PyCharm IDE interface. The project navigation bar at the top indicates 'PyCharm 最新变化' and shows files 01.py, 02.py, and 03.py. The code editor displays a script named 03.py:

```
def find_number():
    for i in range(2,85):
        if 168*i==0:
            j=168//i
            if i>j and (i+j)%2==0:
                m=(i+j)//2
                n=(i-j)//2
                x=n*n-100
                print(x)
find_number()
```

The run output window below shows the execution results:

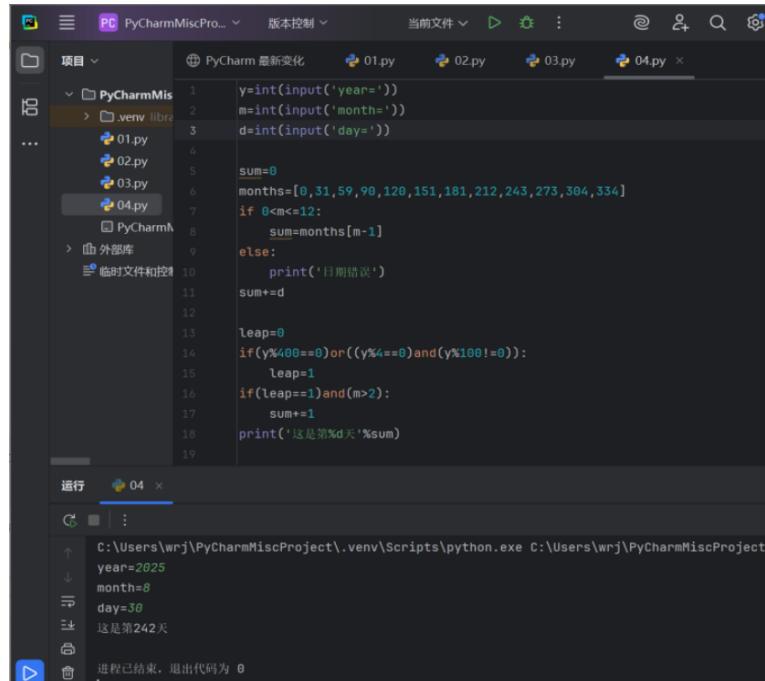
```
C:\Users\wrj\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\wrj\PyCharmMiscProject\03.py
-99
21
261
1581
进程已结束, 退出代码为 0
```

思路如下：

求 x , $x+100=m^2$, $x+100+168=n^2$
 $m^2-n^2=(m+n)(m-n)=168$
 $m+n=i$, $m-n=j$
 $i \cdot j=168 \Rightarrow i$ 和 j 至少有一个偶数
 令 $i \in [2, 84]$ 循环
 $m=(i+j)/2$, $n=(i-j)/2 \Rightarrow i$ 和 j 同奇或同偶
 $\Rightarrow i$ 和 j 同偶 且 $i > j$

2.4 实例四

题目：输入某年某月某日，判断这一天是这一年的第几天



```

1 y=int(input('year='))
2 m=int(input('month='))
3 d=int(input('day='))

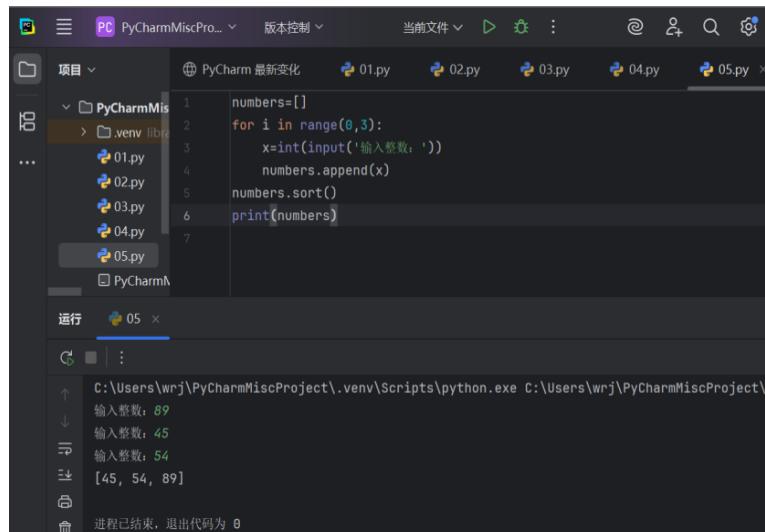
4 sum=0
months=[0,31,59,90,120,151,181,212,243,273,304,334]
if 0<m<=12:
    sum=months[m-1]
else:
    print('日期错误')
sum+=d

13 leap=0
if(y%400==0)or((y%4==0)and(y%100!=0)):
    leap=1
if(leap==1)and(m>2):
    sum+=1
print('这是第%d天'%sum)

```

2.5 实例五

题目：排序。输入三个整数，按由小到大的顺序排列。



```

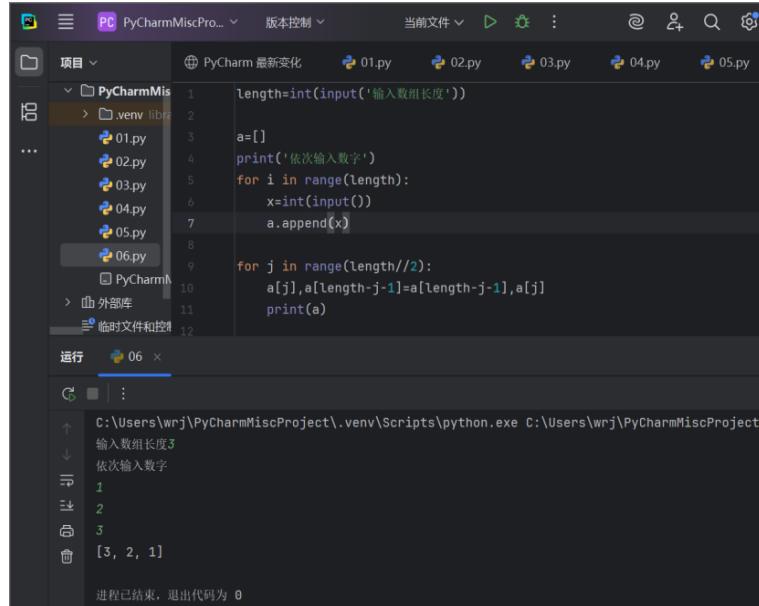
1 numbers=[]
2 for i in range(0,3):
3     x=int(input('输入整数: '))
4     numbers.append(x)
5 numbers.sort()
6 print(numbers)

```

使用 append 函数向列表 numbers 末尾添加元素，使用 sort 函数对添加完元素的列表排序。

2.6 实例六

题目：将一个数组逆序输出



The screenshot shows the PyCharm IDE interface. The project tree on the left shows a folder named 'PyCharmMisc' containing several Python files (01.py, 02.py, 03.py, 04.py, 05.py, 06.py). The code editor window displays the following Python script:

```
length=int(input('输入数组长度'))
a=[]
print('依次输入数字')
for i in range(length):
    x=int(input())
    a.append(x)

for j in range(length//2):
    a[j],a[length-j-1]=a[length-j-1],a[j]
    print(a)
```

The run tool window at the bottom shows the output of running file 06:

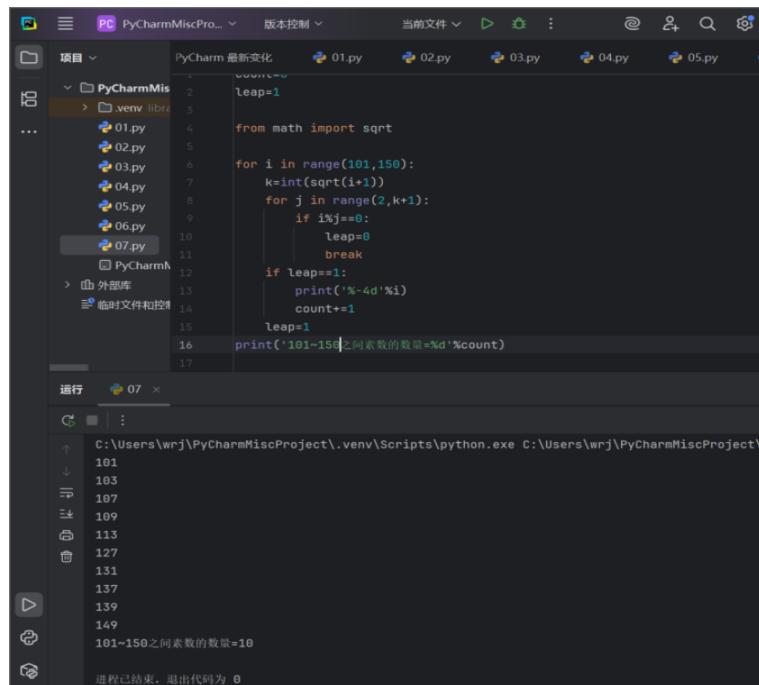
```
C:\Users\wrj\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\wrj\PyCharmMiscProject\06.py
输入数组长度3
依次输入数字
1
2
3
[3, 2, 1]
```

进程已结束，退出代码为 0

将首位的数字依次对换位置， $a[j],a[length-j-1]=a[length-j-1],a[j]$ ，只用循环数组长度的“一半（数组长度除以 2 取整）”次数即可。

2.7 实例七

题目：判断 101 到 150 之间有多少个素数，并输出全部素数



The screenshot shows the PyCharm IDE interface. The project tree on the left shows a folder named 'PyCharmMisc' containing several Python files (01.py, 02.py, 03.py, 04.py, 05.py, 06.py, 07.py). The code editor window displays the following Python script:

```
leap=1
from math import sqrt

for i in range(101,150):
    k=int(sqrt(i+1))
    for j in range(2,k+1):
        if i%j==0:
            leap=0
            break
        if leap==1:
            print('%-4d'%i)
            count+=1
    leap=1
print('101~150之间素数的数量=%d'%count)
```

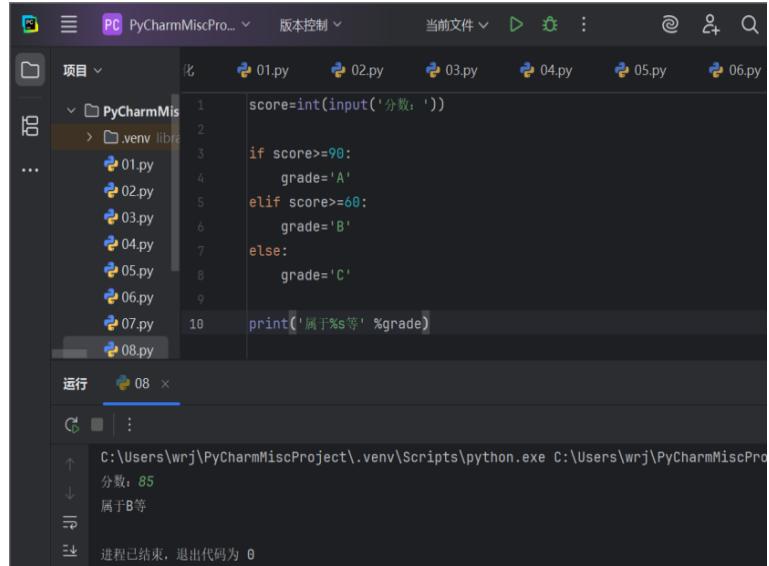
The run tool window at the bottom shows the output of running file 07:

```
C:\Users\wrj\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\wrj\PyCharmMiscProject\07.py
101
103
107
109
113
127
131
137
139
149
101~150之间素数的数量=10
```

进程已结束，退出代码为 0

2.8 实例八

题目：根据输入的分数，判定学生成绩的等级， ≥ 90 属于 A， $60 \sim 89$ 属于 B， ≤ 60 属于 C。要求使用条件运算符的嵌套。



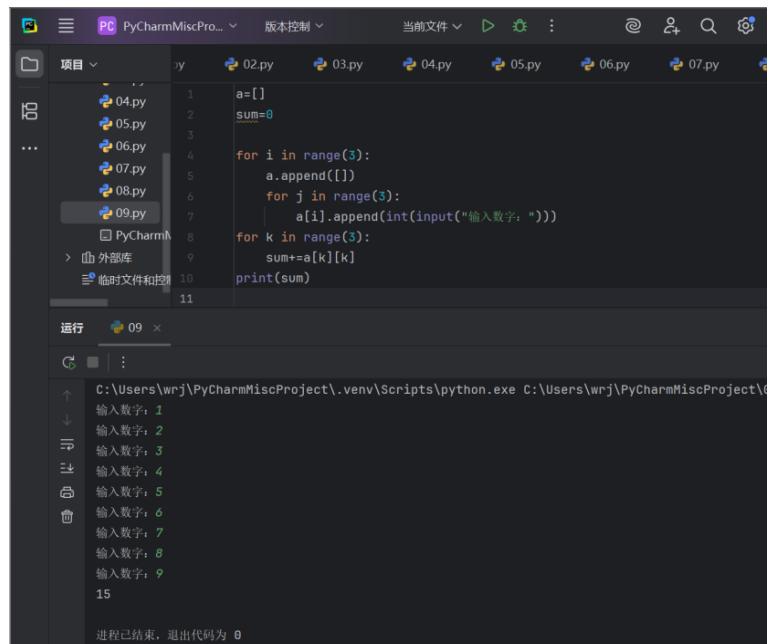
```
score=int(input('分数: '))
if score>=90:
    grade='A'
elif score>=60:
    grade='B'
else:
    grade='C'
print('属于%s等' %grade)
```

运行结果：

```
C:\Users\wrj\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\wrj\PyCharmMiscProject\08.py
分数: 85
属于B等
```

2.9 实例九

题目：求一个 3×3 矩阵主对角线元素之和



```
a=[]
sum=0
for i in range(3):
    a.append([])
    for j in range(3):
        a[i].append(int(input("输入数字: ")))
for k in range(3):
    sum+=a[k][k]
print(sum)
```

运行结果：

```
C:\Users\wrj\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\wrj\PyCharmMiscProject\09.py
输入数字: 1
输入数字: 2
输入数字: 3
输入数字: 4
输入数字: 5
输入数字: 6
输入数字: 7
输入数字: 8
输入数字: 9
15
```

2.10 实例十

题目：数组插入。给定一个已经排好有小到大的数组，输入一个数字，按由小到大的顺序插入数组，并输出

```

1 a=[1, 5, 10, 61, 78]
2
3 num=int(input("输入一个数字: "))
4 k=0
5 for i in range(len(a)):
6     if(num==a[i]):
7         k=i+1
8     else:
9         break
10 a.insert(k,num)
11 print("插入后的数组: ",a)

```

运行 10

```

C:\Users\wrj\PyCharmMiscProject\.venv\Scripts\python.exe C:/Users/wrj/PyCharmMiscProject/10.py
输入一个数字: 56
插入后的数组: [1, 5, 10, 56, 61, 78]

```

进程已结束，退出代码为 0

2.11 实例十一

在 cmd 里输入 pip install pillow 安装 pillow 库

我这里用的是 pycharm，在终端中输入 pip install pillow 安装 pillow 库

```

终端 本地 + ∨
Windows PowerShell
版权所有 (C) Microsoft Corporation. 保留所有权利。
安装最新的 PowerShell，了解新功能和改进：https://aka.ms/PSWindows

(.venv) PS C:\Users\wrj\PycharmProjects\PythonProject> pip insatll pillow
ERROR: unknown command "insatll" - maybe you meant "install"
(.venv) PS C:\Users\wrj\PycharmProjects\PythonProject> pip install pillow
Requirement already satisfied: pillow in c:/users/wrj/pycharmprojects/pythonproject/.venv/lib/site-packages (11.3.0)

[notice] A new release of pip is available: 25.1.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
(.venv) PS C:\Users\wrj\PycharmProjects\PythonProject> python.exe -m pip install --upgrade pip
Requirement already satisfied: pip in c:/users/wrj/pycharmprojects/pythonproject/.venv/lib/site-packages (25.1.1)
Collecting pip
  Downloading pip-25.2-py3-none-any.whl.metadata (4.7 kB)
  Downloading pip-25.2-py3-none-any.whl (1.8 MB)
    1.8/1.8 MB 2.7 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 25.1.1
    Uninstalling pip-25.1.1:
      Successfully uninstalled pip-25.1.1
      Successfully installed pip-25.2
(.venv) PS C:\Users\wrj\PycharmProjects\PythonProject>

```

利用 pillow 库更改图片的对比度

```

1 from PIL import Image, ImageEnhance
2
3 # 打开图像，使用原始字符串或转义反斜杠的路径
4 img_original = Image.open(r"C:\Users\wrj\Desktop\001.jpg")
5 # 显示原始图像
6 img_original.show("原图")
7
8 # 创建对比度增强对象
9 img_enhancer = ImageEnhance.Contrast(img_original)
10 # 增强对比度，参数就是倍数
11 img = img_enhancer.enhance(3.0)
12 # 显示增强对比度后的图像
13 img.show("对比图")

```



2.12 实例十二

同样的方法安装 numpy

```
(.venv) PS C:\Users\wrj\PycharmProjects\PythonProject> pip install numpy
Collecting numpy
  Downloading numpy-2.3.3-cp313-cp313-win_amd64.whl.metadata (60 kB)
  Downloading numpy-2.3.3-cp313-cp313-win_amd64.whl (12.8 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 6.1 MB/s  0:00:02
Installing collected packages: numpy
Successfully installed numpy-2.3.3
```

实现 RGB 通道分离和展示效果

```
1  from PIL import Image
2  import numpy as np
3
4  # 打开图像并转换为 NumPy 数组
5  img = np.array(Image.open(r"C:\Users\wrj\Desktop\001.jpg"))
6
7  # 分离开处理 RGB 通道
8  # 处理红色通道：将绿、蓝通道置 0
9  img_red = img.copy()
10 img_red[:, :, (1, 2)] = 0
11 # 处理绿色通道：将红、蓝通道置 0
12 img_green = img.copy()
13 img_green[:, :, (0, 2)] = 0
14 # 处理蓝色通道：将红、绿通道置 0
15 img_blue = img.copy()
16 img_blue[:, :, (0, 1)] = 0
17
18 # 拼接原图和三个单通道图，便于对比展示
19 img_rgb = np.concatenate([img, img_red, img_green, img_blue], axis=1)
20
21 # 将 NumPy 数组转回图像并显示
22 Image.fromarray(img_rgb).show()
```

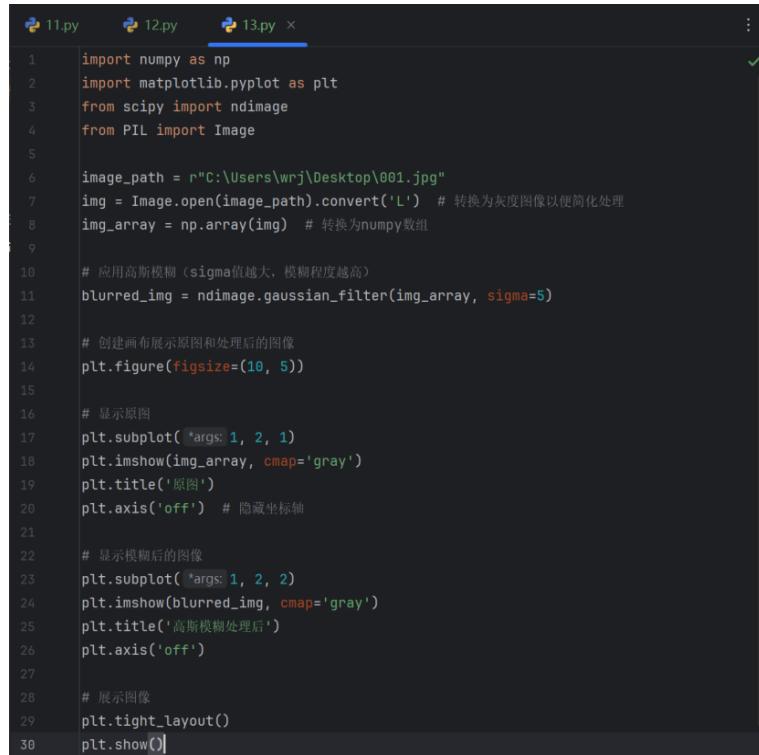


2.13 实例十三

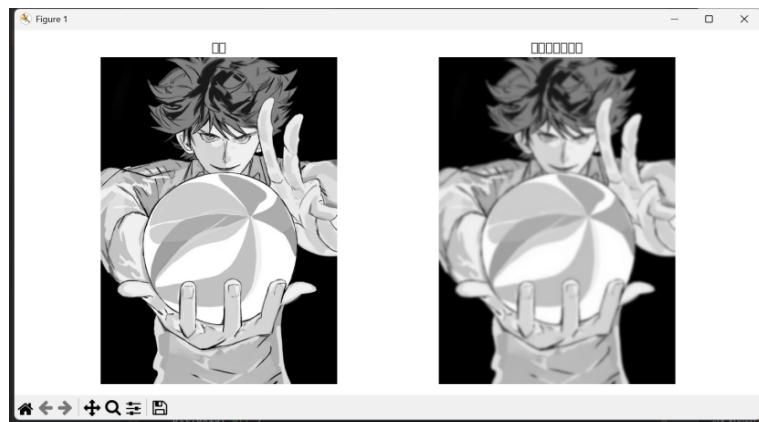
同样的方法安装 scipy 库和 matplotlib 库

```
(.venv) PS C:\Users\wrj\PycharmProjects\PythonProject> pip install scipy
Collecting scipy
  Downloading scipy-1.16.2-cp313-cp313-win_amd64.whl.metadata (60 kB)
Requirement already satisfied: numpy<2.6,>=1.25.2 in c:\users\wrj\pycharmprojects\pythonproject\.venv\lib\site-packages (from scipy) (2.3.3)
  Downloading scipy-1.16.2-cp313-cp313-win_amd64.whl (38.5 MB)
     38.5/38.5 MB 5.5 MB/s  0:00:07
Installing collected packages: scipy
Successfully installed scipy-1.16.2
(.venv) PS C:\Users\wrj\PycharmProjects\PythonProject> pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.1.0-cp313-cp313-win_amd64.whl.metadata (11 kB)
```

先灰度处理以便简化模糊



```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy import ndimage
4 from PIL import Image
5
6 image_path = r"C:\Users\wrj\Desktop\001.jpg"
7 img = Image.open(image_path).convert('L') # 转换为灰度图像以便简化处理
8 img_array = np.array(img) # 转换为numpy数组
9
10 # 应用高斯模糊 (sigma值越大，模糊程度越高)
11 blurred_img = ndimage.gaussian_filter(img_array, sigma=5)
12
13 # 创建画布展示原图和处理后的图像
14 plt.figure(figsize=(10, 5))
15
16 # 显示原图
17 plt.subplot(*args: 1, 2, 1)
18 plt.imshow(img_array, cmap='gray')
19 plt.title('原图')
20 plt.axis('off') # 隐藏坐标轴
21
22 # 显示模糊后的图像
23 plt.subplot(*args: 1, 2, 2)
24 plt.imshow(blurred_img, cmap='gray')
25 plt.title('高斯模糊处理后')
26 plt.axis('off')
27
28 # 展示图像
29 plt.tight_layout()
30 plt.show()
```

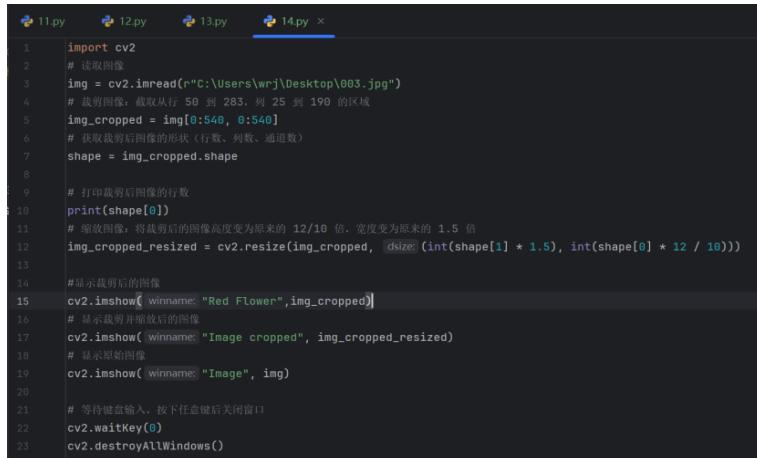


2.14 实例十四

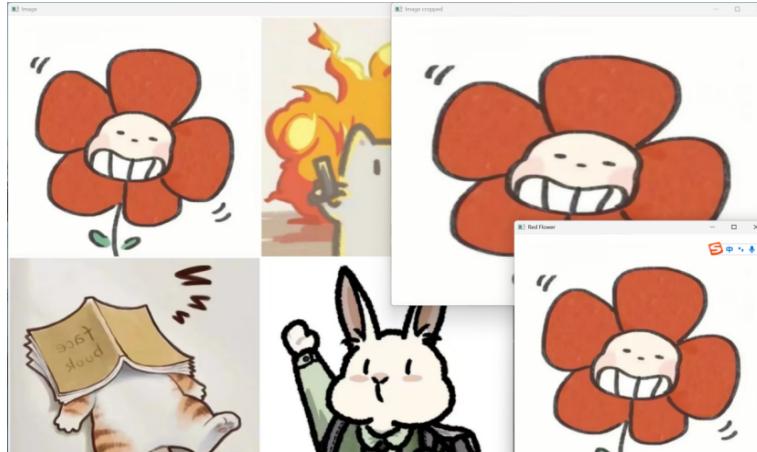
同样的方法安装 opencv-python 库

```
(.venv) PS C:\Users\wrj\PycharmProjects\PythonProject> pip install opencv-python
Collecting opencv-python
  Downloading opencv_python-4.12.0.88-cp37-abi3-win_amd64.whl.metadata (19 kB)
Collecting numpy<2.3.0,>=2 (from opencv-python)
  Downloading numpy-2.2.6-cp313-cp313-win_amd64.whl.metadata (60 kB)
  Downloading opencv_python-4.12.0.88-cp37-abi3-win_amd64.whl (39.0 MB)
    ━━━━━━━━━━━━━━━━ 39.0/39.0 MB 4.8 MB/s 0:00:08
  Downloading numpy-2.2.6-cp313-cp313-win_amd64.whl (12.6 MB)
    ━━━━━━━━━━━━━━ 12.6/12.6 MB 5.3 MB/s 0:00:02
Installing collected packages: numpy, opencv-python
  Attempting uninstall: numpy
    Found existing installation: numpy 2.3.3
    Uninstalling numpy-2.3.3:
      Successfully uninstalled numpy-2.3.3
Successfully installed numpy-2.2.6 opencv-python-4.12.0.88
```

利用 opencv-python 库为实现图片裁剪以及裁剪后的比例调整



```
1 import cv2
2 # 读取图像
3 img = cv2.imread(r"C:\Users\wrj\Desktop\003.jpg")
4 # 裁剪图像：裁取从行 50 到 283，列 25 到 190 的区域
5 img_cropped = img[50:283, 25:190]
6 # 获取裁剪后图像的形状（行数、列数、通道数）
7 shape = img_cropped.shape
8
9 # 打印裁剪后图像的行数
10 print(shape[0])
11 # 缩放图像：将裁剪后的图像高度变为原来的 12/10 倍，宽度变为原来的 1.5 倍
12 img_cropped_resized = cv2.resize(img_cropped, dsize=(int(shape[1] * 1.5), int(shape[0] * 12 / 10)))
13
14 # 显示裁剪后的图像
15 cv2.imshow("Red Flower", img_cropped)
16 # 显示裁剪并缩放后的图像
17 cv2.imshow("Image cropped", img_cropped_resized)
18 # 显示原始图像
19 cv2.imshow("Image", img)
20
21 # 等待键盘输入，按下任意键后关闭窗口
22 cv2.waitKey(0)
23 cv2.destroyAllWindows()
```



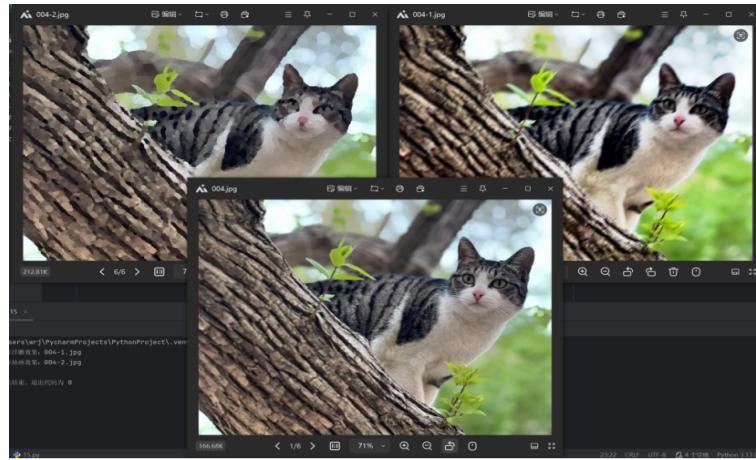
2.15 实例十五

先安装 ImageMagick，再用和前面实例同样的方法安装 wand 库

```
(.venv) PS C:\Users\wrj\PycharmProjects\PythonProject> pip install wand
Collecting wand
  Downloading Wand-0.6.13-py2.py3-none-any.whl.metadata (4.0 kB)
  Downloading Wand-0.6.13-py2.py3-none-any.whl (143 kB)
Installing collected packages: wand
Successfully installed wand-0.6.13
```

利用 wand 库实现图像的优化效果和浮雕效果

```
1  from wand.image import Image
2  import os
3  import subprocess
4
5  def apply_image_effects():
6      input_path = r"C:\Users\wrj\Desktop\004.jpg"
7
8      if not os.path.exists(input_path):
9          print("错误：找不到文件 {input_path}")
10         return
11
12     # 打开原始图像
13     with Image(filename=input_path) as img:
14
15         # 浮雕效果
16         with img.clone() as embossed:
17             embossed.emboss(radius=4, sigma=6)
18             embossed.save(filename='004-1.jpg')
19             print("已保存浮雕效果：004-1.jpg")
20
21         # 油画效果
22         with img.clone() as painted:
23             painted.oil_paint(radius=8)
24             painted.save(filename='004-2.jpg')
25             print("已保存油画效果：004-2.jpg")
26
27     apply_image_effects()
```



2.16 实例十六

- 题目要求：(1) 执行 sleep 10000，用 Ctrl+z 暂停并切换到后台
(2) 用 pgrep 找进程 PID，用 pkill 结束进程
(3) 编写 pidwait 函数：接受 PID 为参数，等待进程结束

```
wrj@wrjubuntu:~$ sleep 10000
20060224wrj
^Z
[1] + 7557 suspended sleep 10000
wrj@wrjubuntu:~$ bg
[1] + 7557 continued sleep 10000
wrj@wrjubuntu:~$ pgrep -af "sleep 10000"
7557 sleep 10000
wrj@wrjubuntu:~$ pkill -f "sleep 10000"
[1] + 7557 terminated sleep 10000
wrj@wrjubuntu:~$ jobs
wrj@wrjubuntu:~$
```

在 zshrc 脚本末尾添加这段代码：

```

pidwait(){
    local pid=$1
    if [[ -z $pid || ! $pid =~ ^[0-9]+$ ]]; then
        echo "use: pidwait <PID>"
        return 1
    fi
    while kill -0 $pid 2>/dev/null; do
        sleep 1
    done
    echo "$pid"
}
-- INSERT --

```

先启动 sleep 60 & 这个后台进程，根据输出的样例，调用 pidwait 等待这个 PID。

```

wrj@wrjubuntu ~ $ vim ~/.zshrc
wrj@wrjubuntu ~ $ source ~/.zshrc
wrj@wrjubuntu ~ $ sleep 60 &
[1] 8007
wrj@wrjubuntu ~ $ pidwait 8007
[1] + 8007 done      sleep 60
8007

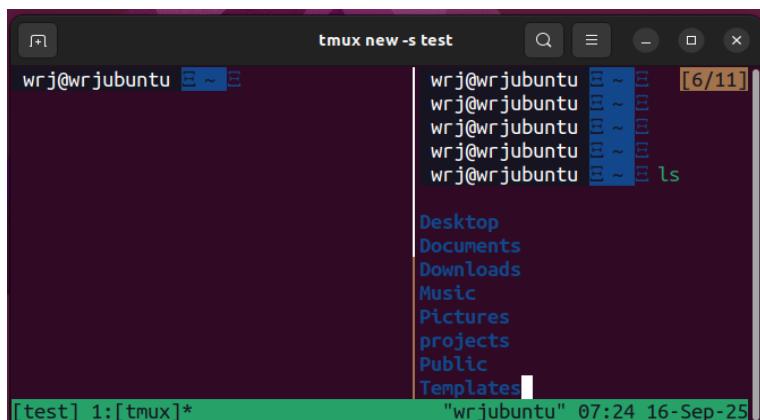
```

2.17 实例十七

题目要求：

- (1) 完成 tmux 教程，自定义 tmux 配置
- (2) 熟练使用 tmux 分屏、切换窗口、分离/链接会话

先新建一个会话并分屏。按 **ctrl+b** 组合键，在立即按 **v**，会垂直分屏，按 **ctrl+b+** 右方向键会切换到右侧面板，在右侧面板输入 **ls**，再按 **ctrl+b+** 左方向键会回到左侧面板。



2.18 实例十八

题目要求：

- (1) 创建 **dc** 别名，实现输入 **dc** 等同于 **cd**
- (2) 查看最常用的 8 条命令，为他们创建别名
在 zshrc 脚本里面添加如下部分，最后生效配置即可实现

```

# alias ll="ls -lh"
# alias gs="git status"
# alias gc="git commit -m"
# alias dc="cd"
# alias mkdir="mkdir -p"
# alias df="df -h"
# alias gss="git ststus -s"
# alias sshvm="ssh vm"

```

2.19 实例十九

题目要求：配置文件 dotfiles 管理

- (1) 新建 Dotfiles 文件夹，用 git 版本控制
- (2) 添加至少一个配置文件，包含自定义设置
- (3) 编写安装脚本，实现新设备快速部署

先初始化 dotfiles 仓库，新建 projects 文件夹并初始化 git，移动配置文件到仓库，再创建软连接

```

wrj@wrjubuntu ~ % mkdir -p ~/projects/dotfiles
wrj@wrjubuntu ~ % cd ~/projects/dotfiles
wrj@wrjubuntu ~/projects/dotfiles % git init
hint: Using 'master' as the name for the initial branch. This default branch name
      is subject to change. To configure the initial branch name to use in all
      of your new repositories, which will suppress this warning, call:
      hint:   git config --global init.defaultBranch <name>
      hint:
      hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
      hint: 'development'. The just-created branch can be renamed via this command:
      hint:
      hint:   git branch -m <name>
Initialized empty Git repository in /home/wrj/projects/dotfiles/.git/
wrj@wrjubuntu ~/projects/dotfiles % master % git remote add origin https://
github.com/WRJ123wrj/dotfiles.git
wrj@wrjubuntu ~/projects/dotfiles % master % mv ~/.zshrc ~/projects/dotfile
s/
wrj@wrjubuntu ~/projects/dotfiles % master % mv ~/.tmux.conf ~/projects/dot
files/
wrj@wrjubuntu ~/projects/dotfiles % master % mkdir -p ~/projects/dotfiles/s
sh_config
wrj@wrjubuntu ~/projects/dotfiles % master % mv ~/.ssh/config ~/projects/do
tfiles/ssh_config/
mv: cannot stat '/home/wrj/.ssh/config': No such file or directory
x wrj@wrjubuntu ~/projects/dotfiles % master % mv ~/.ssh/config ~/projects/
dotfiles/ssh_config/
mv: cannot stat '/home/wrj/.ssh/config': No such file or directory
x wrj@wrjubuntu ~/projects/dotfiles % master % ln -s ~/projects/dotfiles/.z
shrc ~/.zshrc
wrj@wrjubuntu ~/projects/dotfiles % master % ln -s ~/projects/dotfiles/.tmu
x.conf ~/.tmux.conf
wrj@wrjubuntu ~/projects/dotfiles % master % ln -s ~/projects/dotfiles/ssh_
config/config ~/.ssh/config
wrj@wrjubuntu ~/projects/dotfiles % master % git add .
wrj@wrjubuntu ~/projects/dotfiles % master % git commit -m "Initial commi
t: add zsh/tmux/ssh config"

```

编写 install.sh 脚本，保存退出， chmod +x 添加可执行权限

```

vim ~/dotfiles/install.sh
#!/bin/bash
set -e

declare -A configs=()
[".zshrc"]="~/.zshrc"
[".tmux.conf"]="~/.tmux.conf"
["ssh_config/config"]="~/.ssh/config"
)

for src in "${!configs[@]}"; do
    src_path="~/dotfiles/$src"
    dest_path="${configs[$src]}"

    if [ ! -f "$dest_path" ] || [ ! -L "$dest_path" ]; then
        echo "备份 $dest_path 到 $dest_path.bak"
        mv "$dest_path" "$dest_path.bak"
    fi

    echo "创建软链接：$src_path -> $dest_path"
    ln -s "$src_path" "$dest_path"
done

echo "安装完成！"

```

2.20 实例二十

配置 ssh 环境，ssh 免密登录配置（主机到虚拟机）

(1) 在 windows 主机生成 SSH 密钥对，生成后主机 /.ssh 目录下会有一个私钥文件和一个公钥文件，私钥文件不可泄露，公钥文件要传到虚拟机上

```

wrj@wrjubuntu:~/projects/dotfiles

wrj@LAPTOP-1D16IK15 MINGW64 /
$ ssh-keygen -o -a 100 -t ed25519 -f ~/.ssh/vm_key
Generating public/private ed25519 key pair.
Created directory '/c/Users/wrj/.ssh'.
Enter passphrase for "/c/Users/wrj/.ssh/vm_key" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/wrj/.ssh/vm_key.
Your public key has been saved in /c/Users/wrj/.ssh/vm_key.pub
The key fingerprint is:
SHA256:Krr740sgfwazCnS62EGN2iJgxVwTZ3U33jkZ95sQUIw wrj@LAPTOP-1D16IK15
The key's randomart image is:
+--[ED25519 256]--+
| +.0.. 0*=.* |
| o . + .E.oo=o |
| + . . . |
| .o . . o |
| oo*.. S o |
| +*o= . |
|= = = . |
|++.*.. |
|o==+o |
+---[SHA256]----+

```

(2) 将公钥传到虚拟机上，使用 ssh-copy-id 工具，将公钥添加到虚拟机的授权列表

```

wrj@LAPTOP-1D16IK15 MINGW64 /
$ ssh-copy-id -i ~/.ssh/vm_key.pub wrj@192.168.43.130
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/c/Users/wrj/.ssh/vm_key.pub"
The authenticity of host '192.168.43.130 (192.168.43.130)' can't be established.
ED25519 key fingerprint is SHA256:F2+wI08w0//15IF4KQsbGUzk6PpgUlasePD13R/lCk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
wrj@192.168.43.130's password:

Number of key(s) added: 1

```

(3) 配置主机的别名，编辑主机的 /.ssh/config 文件，添加到虚拟机的链接配置，之后只用输入 ssh vm 就可以访问

3 总结和体会

本次实验围绕 python 基础编程、python 视觉运用以及命令行环境三大核心主题，在 python 基础部分，我熟练掌握了 python 语言环境下循环嵌套、条件判断、列表等操作，学习使用了算法优化来降低程序的复杂度；在 python 视觉部分，我学习使用了 pillow 库、numpy 库、opencv-python 库和 wand 库，分别完成了图像对比度调整、RGB 通道分离、图像裁剪、浮雕和油画效果显示，体会到了多库协同在视觉任务中的优势；在命令行环境部分，我遇到了很多问题，例如在配置环境时，遇到了 git 终端中 github 身份验证的问题，查询后发现，git 操作不再支持密码验证，于是我通过生成 github 个人访问令牌，并在 push 时用令牌代替密码解决了这个问题。这次实验不仅使我掌握了具体的技术和工具使用，也培养了我的系统思维和解决问题的能力。