

Week3 命令行环境、Python 入门基础、 python 视觉应用

黄琬晴

September 2025

本次实验的代码与实验报告上传在 github 仓库中：

<https://github.com/uuukyoo/systools>

目录

1 命令行环境-任务控制	1
1.1 任务控制	1
1.2 wait 命令使用	2
1.3 实现在不同的 bash 会话中等待	3
2 命令行环境-别名	5
2.1 为命令”cd” 创建别名	5
3 命令行环境-终端多路复用	5
3.1 建立新会话	5
3.2 切换会话	6
3.3 列出当前所有窗口	7
3.4 分割窗口	8
4 命令行环境-配置文件	9
4.1 为配置文件新建文件夹，并设置版本控制	9
4.2 添加配置文件	10

5 命令行环境-远端设备	11
5.1 创建 ssh 密钥对	11
5.2 配置 ssh	12
6 Python 入门基础	17
6.1 for 循环打印九九乘法表	17
6.2 while 循环计算阶乘	17
7 python 图像处理	18
7.1 改变图像的 RGB 值	18
7.2 增强图像对比度	19
7.3 高斯模糊	20
7.4 缩放图片	22
7.5 裁切图片	24
7.6 旋转翻转	25

1 命令行环境—任务控制

1.1 任务控制

我们可以使用类似 ps aux | grep 这样的命令来获取任务的 pid，然后您可以基于 pid 来结束这些进程。但我们其实有更好的方法来做这件事。在终端中执行 sleep 10000 这个任务。然后用 Ctrl-Z 将其切换到后台并使用 bg 来继续允许它。现在，使用 pgrep 来查找 pid 并使用 pkill 结束进程而不需要手动输入 pid。

先执行 sleep 10000 这个任务，按下 Ctrl-Z 可以看见出现已停止字样，此时该任务的进程切换到后台；再使用 bg 命令时期继续

```
hwq@hwq:~$ sleep 10000
^Z
[1]+  已停止                  sleep 10000
hwq@hwq:~$ bg
[1]+ sleep 10000 &
```

然后使用 pgrep 命令查找 pid，使用 pkill 和-f 标记即可

```
hwq@hwq:~$ pgrep sleep
15285
hwq@hwq:~$ pkill -f sleep
[1]+  已终止                  sleep 10000
```

不适用题目给的-af 是因为我的虚拟机的 pkill 命令不支持-a 选项，而使用-f 已经可以完成要求

```
hwq@hwq:~$ pkill -af sleep  
pkill: 无效的选项 -- a
```

用法：

```
pkill [options] <pattern>
```

选项：

1.2 wait 命令使用

如果您希望某个进程结束后再开始另外一个进程，应该如何实现呢？在这个练习中，我们使用 sleep 60 & 作为先执行的程序。一种方法是使用 wait 命令。尝试启动这个休眠命令，然后待其结束后再执行 ls 命令

使 wait \$(pgrep sleep);ls 命令， pgrep 的输入作为参数传给 wait，命令刚输入时不会执行 ls

```
hwq@hwq:~$ sleep 60 &  
[1] 17424  
hwq@hwq:~$ wait $(pgrep sleep);ls
```

等到 sleep 60 这一任务完成后，执行 ls 命令

```
hwq@hwq:~$ wait $(pgrep sleep);ls
[1]+  已完成                  sleep 60
公共      ca.crt          last3.txt        server2.py    unlock.c
模板      client.crt       last-modified.txt server.py     vir.crt
视频      client.key       lock             snap           vir.key
图片      debug_for.sh    lock.c          sorts.py      virtual.py
文档      file_to_send.txt marco_history.log sorts.py.lprof wk3_1_vir.py
下载      getlog.sh        marco.sh        starttime.txt wk3_2_vir.py
音乐      getlot.sh        out.log         test          wk4vir.py
桌面      httpserver.py   plug.vim        tocheck.sh
buggy.sh  lab5_vir.py    received_file.txt unlock
hwq@hwq:~$
```

1.3 实现在不同的 bash 会话中等待

但是，如果我们在不同的 bash 会话中进行操作，则上述方法就不起作用了。因为 `wait` 只能对子进程起作用。之前我们没有提过的一个特性是，`kill` 命令成功退出时其状态码为 0，其他状态则是非 0。`kill -0` 则不会发送信号，但是会在进程不存在时返回一个不为 0 的状态码。请编写一个 bash 函数 `pidwait`，它接受一个 pid 作为输入参数，然后一直等待直到该进程结束。您需要使用 `sleep` 来避免浪费 CPU 性能

编写 `pypidwait()` 函数

```
mypidwait()
{
    while kill -0 $1 # 循环直到进程结束
    do
        sleep 1
    done
    ls
```

```
hwq@hwq:~$ source pidwait.sh
```

打开另一个终端窗口

```
hwq@hwq:~$ sleep 60 &  
[1] 17645
```

回到原来的窗口，执行以下命令

```
hwq@hwq:~$ pid=$(pgrep -n sleep)  
hwq@hwq:~$ mypidwait $pid;ls
```

可以看见先等待

```
hwq@hwq:~$ pid=$(pgrep -n sleep)  
hwq@hwq:~$ mypidwait $pid;ls  
bash: kill: (17645) - 没有那个进程  
公共 ca.crt last3.txt received_file.txt un  
模板 client.crt last-modified.txt server2.py un  
组织 client.key lock services_pk un
```

结束后再执行 ls

```
hwq@hwq:~$ pid=$(pgrep -n sleep)  
hwq@hwq:~$ mypidwait $pid;ls  
bash: kill: (17645) - 没有那个进程  
公共 ca.crt last3.txt received_file.txt un  
模板 client.crt last-modified.txt server2.py un  
组织 client.key lock services_pk un
```

2 命令行环境—别名

2.1 为命令”cd” 创建别名

创建一个 dc 别名，它的功能是当我们错误的将 cd 输入为 dc 时也能正确执行

使用 alias 命令，参数是 dc=”cd”，要注意等号两边不能有空格，不然会被识别为多个参数

```
hwq@hwq:~$ alias dc="cd"
```

创建别名后，使用”dc”的时候也能像使用”cd”的时候一样切换目录

```
hwq@hwq:~$ dc snap
hwq@hwq:~/snap$ dc ~
hwq@hwq:~$ cd snap
hwq@hwq:~/snap$ cd ~
```

3 命令行环境—终端多路复用

3.1 建立新会话

tmux new -s NAME 以指定名称开始一个新的会话

A screenshot of a terminal window titled "hwq@hwq: ~". The window title bar also shows the command "tmux ls". The terminal prompt is "hwq@hwq:~\$". At the bottom of the window, there is a green status bar with the text "[practice]0:bash*" on the left and the date/time "hwq" 22:37 21-9月 -25" on the right. The main body of the terminal is dark and empty.

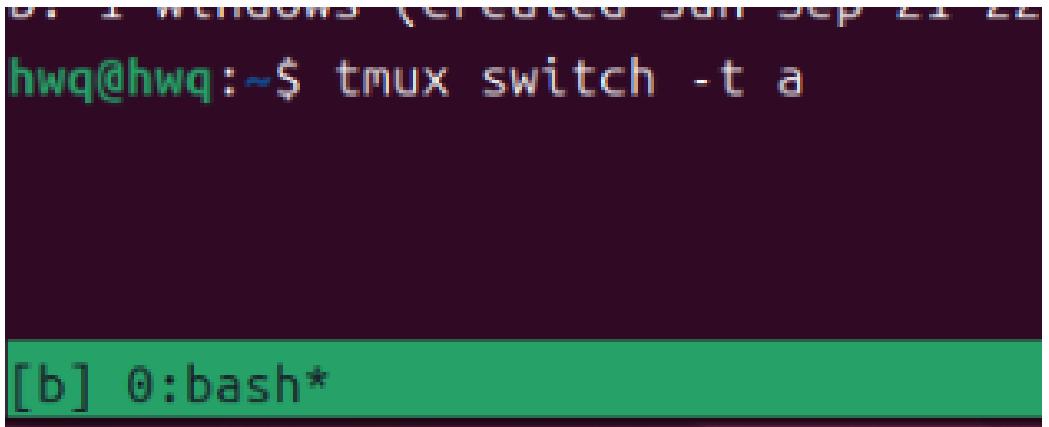
3.2 切换会话

如下有 ab 两对话

A screenshot of a terminal window titled "hwq@hwq: ~". The window title bar also shows the command "tmux ls". The terminal prompt is "hwq@hwq:~\$". The output of the "tmux ls" command is displayed, showing two windows: "a" and "b". Window "a" is described as "1 windows (created Sun Sep 21 22:46:27 2025)". Window "b" is described as "1 windows (created Sun Sep 21 22:48:03 2025) (attached)". At the bottom of the window, there is a green status bar with the text "[practice]0:bash*" on the left and the date/time "hwq" 22:37 21-9月 -25" on the right. The main body of the terminal is dark and empty.

现在在 b，使用如下命令

```
huawei@huawei:~$ tmux switch -t a
```



```
[b] 0:bash*
```

切换到 a



```
huawei@huawei:~$
```

```
[a] 0:bash*
```

3.3 列出当前所有窗口

<C-b> w 列出当前所有窗口

```
(0) - a: 1 windows (attached)
(1) └─> 0: [tmux]*
(2) - b: 1 windows
(3) └─> 0: bash*

^ sort: index) —————
CDROM .. 1 windows (created Sun Sep 21 22:46:27 2025) (attached)
hwq@hwq:~$ tmux switch -t b
can't find session: b
hwq@hwq:~$ █
```

3.4 分割窗口

- $\langle C-b \rangle$ " 水平分割
 - $\langle C-b \rangle$ % 垂直分割

```
hwq@hwq:~$ tmux ls
a: 1 windows (created Sun Sep 21 22:46:27 2025)
) 
b: 1 windows (created Sun Sep 21 22:48:03 2025
) (attached)
hwq@hwq:~$ tmux switch -t a
hwq@hwq:~$
```

Ubuntu 24.04.1 LTS amd64

```
[b] 0:bash* "hwq" 22:59 21-9月 -25
```

4 命令行环境—配置文件

4.1 为配置文件新建文件夹，并设置版本控制

为您的配置文件新建一个文件夹，并设置好版本控制

```
hwq@hwq:~$ mkdir ~/gits/dotfiles
hwq@hwq:~$ git init ~/gits/dotfiles
提示：使用 'master' 作为初始分支的名称。这个默认分支名称可能会更改。要在新仓库中
提示：配置使用初始分支名，并消除这条警告，请执行：
提示：
提示： git config --global init.defaultBranch <名称>
提示：
提示：除了 'master' 之外，通常选定的名字有 'main'、'trunk' 和 'development'。
提示：可以通过以下命令重命名刚创建的分支：
提示：
提示： git branch -m <name>
已初始化空的 Git 仓库于 /home/hwq/gits/dotfiles/.git/
hwq@hwq:~$
```

4.2 添加配置文件

在其中添加至少一个配置文件，比如说您的 shell，在其中包含一些自定义设置（可以从设置 \$PS1 开始）转到目录，将本机配置文件拷贝到目录中

```
hwq@hwq:~/gits/dotfiles$ cd ~/gits/dotfiles
hwq@hwq:~/gits/dotfiles$ cp ~/.vimrc ./
hwq@hwq:~/gits/dotfiles$ cp ~/.bashrc ./
hwq@hwq:~/gits/dotfiles$ cp ~/.profile ./
hwq@hwq:~/gits/dotfiles$ ls -a
.  ..  .bashrc  .git  .profile  .vimrc
hwq@hwq:~/gits/dotfiles$ 
```

无法提交因为没设置用户名

```
hwq@hwq:~/gits/dotfiles$ git add .
hwq@hwq:~/gits/dotfiles$ git commit -m "Add initial configuration files"
作者身份未知
```

*** 请告诉我您是谁。

运行

设置用户信息，将 git 的配置文件也添加在文件夹中（里面包含了用户信息

```
hwq@hwq:~/gits/dotfiles$ git config --global user.name "hwq"
hwq@hwq:~/gits/dotfiles$ git config --global user.email "520002888@qq.com"
hwq@hwq:~/gits/dotfiles$ cp ~/.gitconfig ./
```

提交

```
hwq@hwq:~/gits/dotfiles$ git add .
hwq@hwq:~/gits/dotfiles$ git commit -m "Add initial configuration files"
[master (根据提交) 1ce2a4e] Add initial configuration files
 4 files changed, 153 insertions(+)
  create mode 100644 .bashrc
  create mode 100644 .gitconfig
  create mode 100644 .profile
  create mode 100644 .vimrc
```

5 命令行环境—远端设备

5.1 创建 ssh 密钥对

前往 `/.ssh/` 并查看是否已经存在 SSH 密钥对。如果不存在，请使用 `ssh-keygen -o -a 100 -t ed25519` 来创建一个

```
hwq@hwq:~$ ssh-keygen -o -a 100 -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/hwq/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hwq/.ssh/id_ed25519
Your public key has been saved in /home/hwq/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:sHXCpGx6C8JCaBTzfhYzT8u4mAgnqfeGs1IxBr89Nc hwq@hwq
The key's randomart image is:
+--[ED25519 256]--+
| +. . |
| o o . +
| o+ . +=.+ .
| +o*. oO+.o |
| .+=+++=o+S |
| ...o=+.o E |
| ....+ .o |
| oo=.. |
| .+==. |
+----[SHA256]-----+
```

使用 `ls ~/.ssh` 命令可以查看刚刚生成的公私钥，`id_ed25519` 和 `id_ed25519.pub`

```
hwq@hwq:~$ ls ~/.ssh  
authorized_keys  id_ed25519  id_ed25519.pub
```

5.2 配置 ssh

将虚拟机设置为 NAT 模式检查虚拟机是否配置 SSH 执行如下命令，显示没有那个文件或目录，说明没有配置

```
hwq@hwq:~$ /etc/init.d/iptables start  
bash: /etc/init.d/iptables: 没有那个文件或目录
```

执行以下命令配置 SSH、重启 SSH 服务器

```
hwq@hwq:~$ sudo apt install openssh-server  
[sudo] hwq 的密码：  
正在读取软件包列表... 完成  
正在分析软件包的依赖关系树... 完成  
正在读取状态信息... 完成  
openssh-server 已经是最新版 (1:9.6p1-3ubuntu13.14)。  
下列软件包是自动安装的并且现在不需要了：  
 libgl1-amber-dri libglapi-mesa libllvm17t64 python3-netifaces  
使用'sudo apt autoremove'来卸载它(它们)。  
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 7 个软件包未被升级。  
。  
hwq@hwq:~$ sudo service ssh restart
```

打开/etc/ssh/sshd_config 文件

```
hwq@hwq:~$ sudo vi /etc/ssh/sshd_config
```

去掉 Port 22 前的 # 号，开启 22 端口

```
# systemctl restart sshd
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
```

去掉 PermitRootLogin 和 StrictModes 前的 # 号，并将 prohibit-password 改为 yes，允许 root 用户通过 ssh 登录并且开启 StrictModes

```
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
```

查看虚拟机的 Ip 地址

```
hwq@hwq:~$ sudo service ssh restart
hwq@hwq:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.93.129 netmask 255.255.255.0 broadcast 192.168.93.255
              inet6 fe80::20c:29ff:fece:a093 prefixlen 64 scopeid 0x20<link>
                ether 00:0c:29:ce:a0:93 txqueuelen 1000 (以太网)
                  RX packets 127548 bytes 151352234 (151.3 MB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 13387 bytes 1118843 (1.1 MB)
                  TX errors 0 dropped 24 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
              inet6 ::1 prefixlen 128 scopeid 0x10<host>
                loop txqueuelen 1000 (本地环回)
                  RX packets 2444 bytes 304374 (304.3 KB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 2444 bytes 304374 (304.3 KB)
                  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

再 xshell 上新建会话



输入 root



密码为虚拟机 root 的密码



点击确定后可以看见连接成功

```
请参见 https://ubuntu.com/copyright 或者运行：sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted
applicable law.

/usr/bin/xauth:  file /root/.Xauthority does not exist
root@hwq:~#
```

6 Python 入门基础

6.1 for 循环打印九九乘法表

```
for i in range(1, 10):
    for j in range(1, i + 1):
        print (f"\{j} * {i} = {j * i}", end="\t")
    print()
```

```
PS D:\Pythoncode> & D:/Users/HuQ/AppData/Local/Programs/Python/Python38/python.exe "D:/Pythoncode/practice/fors.py"
1 * 1 = 1
1 * 2 = 2      2 * 2 = 4
1 * 3 = 3      2 * 3 = 6      3 * 3 = 9
1 * 4 = 4      2 * 4 = 8      3 * 4 = 12     4 * 4 = 16
1 * 5 = 5      2 * 5 = 10     3 * 5 = 15     4 * 5 = 20     5 * 5 = 25
1 * 6 = 6      2 * 6 = 12     3 * 6 = 18     4 * 6 = 24     5 * 6 = 30     6 * 6 = 36
1 * 7 = 7      2 * 7 = 14     3 * 7 = 21     4 * 7 = 28     5 * 7 = 35     6 * 7 = 42     7 * 7 = 49
1 * 8 = 8      2 * 8 = 16     3 * 8 = 24     4 * 8 = 32     5 * 8 = 40     6 * 8 = 48     7 * 8 = 56     8 * 8 = 64
1 * 9 = 9      2 * 9 = 18     3 * 9 = 27     4 * 9 = 36     5 * 9 = 45     6 * 9 = 54     7 * 9 = 63     8 * 9 = 72     9 * 9 = 81
```

6.2 while 循环计算阶乘

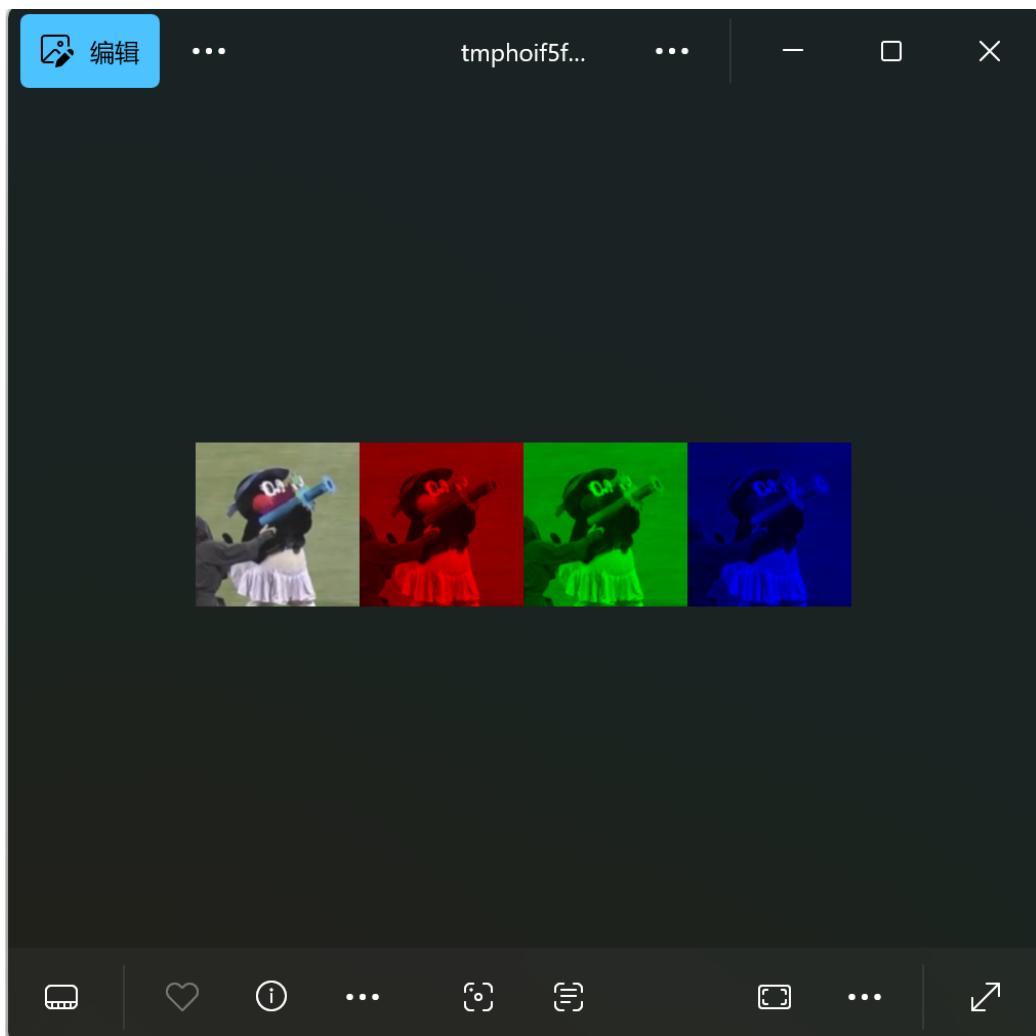
```
n = int(input("输入一个正整数"))
i = 1
result = 1
while i <= n:
    result *= i
    i += 1
print(f"\{n}的阶乘等于{result}")
```

```
PS D:\Pythoncode> & D:/Users
输入一个正整数6
6的阶乘等于720
PS D:\Pythoncode>
```

7 python 图像处理

7.1 改变图像的 RGB 值

```
from PIL import Image
import numpy as np
img = np.array(Image.open('tsubami.jpg'))
img_red = img.copy()
img_red[:, :, (1, 2)] = 0
img_green = img.copy()
img_green[:, :, (0, 2)] = 0
img_blue = img.copy()
img_blue[:, :, (0, 1)] = 0
img_ORGB = np.concatenate((img, img_red, img_green, img_blue), axis=1)
img_converted = Image.fromarray(img_ORGB)
img_converted.show()
```



7.2 增强图像对比度

```
PS D:\Pythoncode\figureprocess> pip install pillow
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Requirement already satisfied: pillow in d:\users\hwq\appdata\local\programs\python\python38\lib\site-packages (10.4.0)
PS D:\Pythoncode\figureprocess> 
```

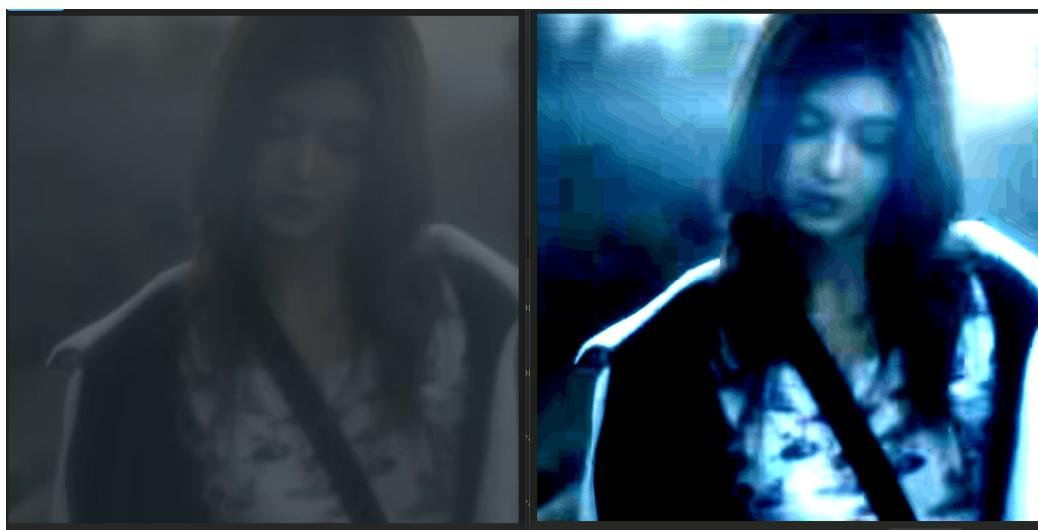
```
from PIL import Image, ImageEnhance
img_original = Image.open("dark.png")
img_original.show("Original Image")
img = ImageEnhance.Contrast(img_original)
```

```
img增强(3.8).show("Image With More Contrast")
```

会分别弹出原图与增强对比度的图片（左原图，右增强对比度）



尝试修改参数，看效果有何不同（如将 enhance() 的参数从 3.8 改为 10）



7.3 高斯模糊

使用 scipy 库

```
from PIL import Image
import numpy as np
from scipy.ndimage import gaussian_filter
import matplotlib.pyplot as plt

img = Image.open("dive.jpg").convert("L")

img_array = np.array(img)

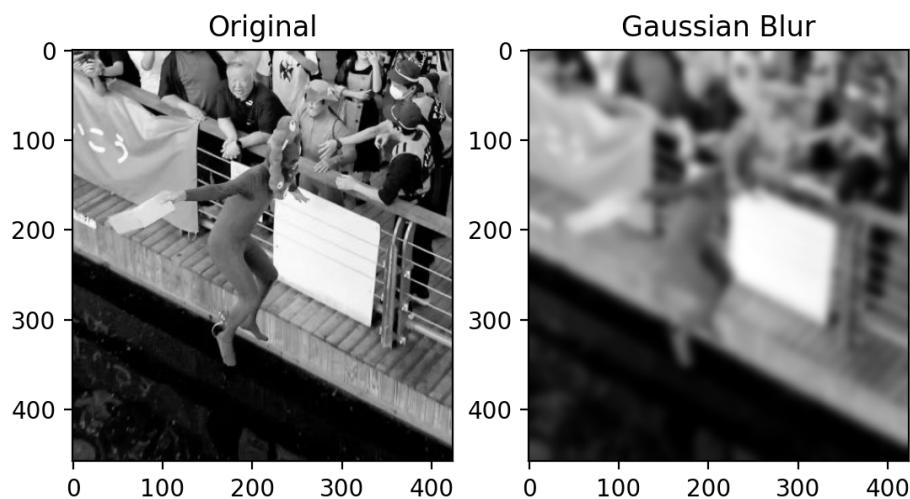
result = gaussian_filter(img_array, sigma=5)

fig = plt.figure()
plt.gray()

ax1 = fig.add_subplot(121)
ax1.set_title("Original")
ax1.imshow(img_array)

ax2 = fig.add_subplot(122)
ax2.set_title("Gaussian Blur")
ax2.imshow(result)

plt.show()
```



7.4 缩放图片

```
from PIL import Image, ImageEnhance  
img_original = Image.open("dive.jpg")  
img_original.show("Original Image")  
img = img_original.resize((50, int(img_original.size[1] * 50 / img_original.size[0])))  
img.show("Image changed")
```

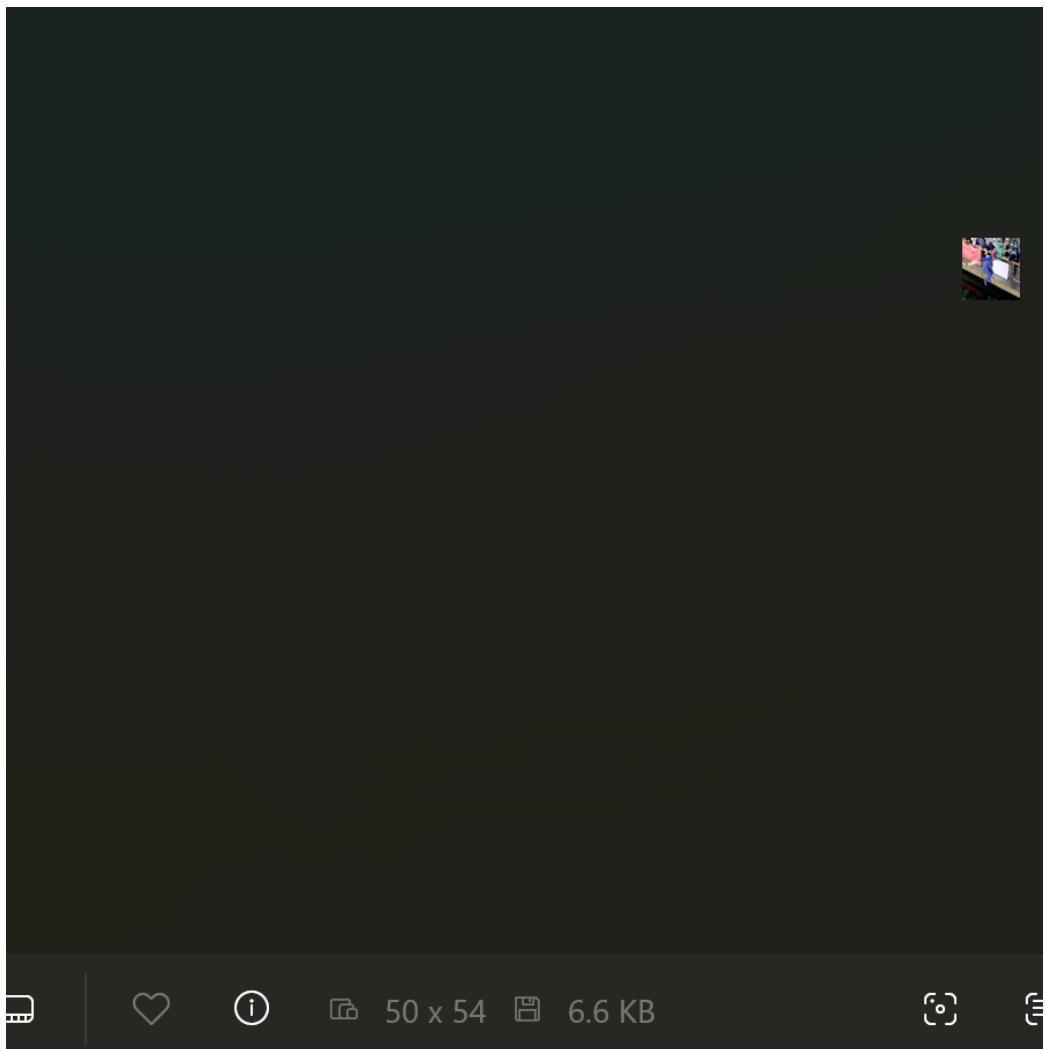
图片浏览器左下角的信息可以看出大小的变化



424 x 458

283.9 KB



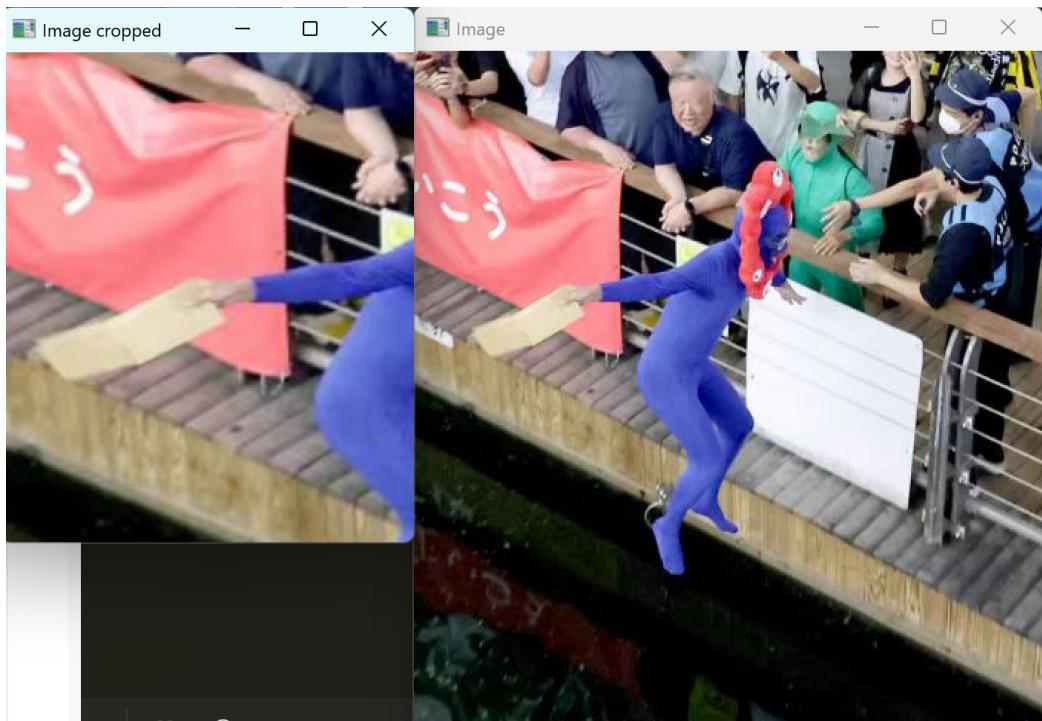


7.5 裁切图片

使用 cv 库进行对图像的裁切

```
import cv2
img = cv2.imread("dive.jpg")
imgCropped = img[50:283,25:190]
shape = imgCropped.shape
print(shape[0])
imgCropped = cv2.resize(imgCropped,(shape[0]*12//10,shape[1]*2))
```

```
cv2.imshow("Image cropped",imgCropped)
cv2.imshow("Image",img)
cv2.waitKey(0)
```



7.6 旋转翻转

使用 PIL 库对图像进行旋转与翻转

```
from PIL import Image

img = Image.open("tsubami.jpg")
img.show("original")
# 旋转图片
rotated_img = img.rotate(90) # 顺时针旋转90度
rotated_img.show("Rotated 90°")

# 翻转图片
```

```
flipped_horizontal = img.transpose(Image.FLIP_LEFT_RIGHT) # 水平翻转  
flipped_vertical = img.transpose(Image.FLIP_TOP_BOTTOM) # 垂直翻转  
  
flipped_horizontal.show("Flipped Horizontal")  
flipped_vertical.show("Flipped Vertical")
```

