

实验报告 2

命令行环境，Python 与视觉应用

洪子翔

2024 年 9 月 12 日

目录

1 实验内容	3
1.1 命令行环境	3
1.1.1 PID	3
1.1.2 查看所有进程	3
1.1.3 终止进程	3
1.1.4 别名	4
1.1.5 配置文件	4
1.2 Python 基础	4
1.2.1 Hello World!	4
1.2.2 计算器	5
1.2.3 猜数字	5
1.2.4 反转字符串	6
1.2.5 列表处理	7
1.2.6 斐波那契数列	8
1.2.7 冒泡排序	9
1.2.8 质数判断	10
1.2.9 日历	11
1.2.10 文件处理	12

1.3	Python 视觉应用	13
1.3.1	图像读取与显示	13
1.3.2	转换灰度图像	13
1.3.3	图像缩放	14
1.3.4	图像旋转	14
1.3.5	裁剪与模糊	15
2	心得	15

1 实验内容

链接: <https://github.com/SoraRosa0514/system-dev-tools/tree/main/3>

1.1 命令行环境

1.1.1 PID

pid 是操作系统分配给正在运行的进程的唯一标识符

要获取 pid, 可以使用:

```
pgrep -f <name>
```

也可用 `echo $$` 获取当前进程的 pid

```
sora@sora-virtual-machine:~/Desktop$ echo $$
2288
```

1.1.2 查看所有进程

使用 `ps aux`

```
sora@sora-virtual-machine:~/Desktop$ ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.3	167976	12996	?	Ss	21:04	0:04	/sbin/init au
root	2	0.0	0.0	0	0	?	S	21:04	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	21:04	0:00	[pool_workque
root	4	0.0	0.0	0	0	?	I<	21:04	0:00	[kworker/R-rc
root	5	0.0	0.0	0	0	?	I<	21:04	0:00	[kworker/R-rc
root	6	0.0	0.0	0	0	?	I<	21:04	0:00	[kworker/R-sl
root	7	0.0	0.0	0	0	?	I<	21:04	0:00	[kworker/R-ne
root	9	0.0	0.0	0	0	?	I<	21:04	0:01	[kworker/0:0H
root	11	0.0	0.0	0	0	?	I	21:04	0:00	[kworker/u256
root	12	0.0	0.0	0	0	?	I<	21:04	0:00	[kworker/R-mm
root	13	0.0	0.0	0	0	?	I	21:04	0:00	[rcu_tasks_kt
root	14	0.0	0.0	0	0	?	I	21:04	0:00	[rcu_tasks_ru
root	15	0.0	0.0	0	0	?	I	21:04	0:00	[rcu_tasks_tr
root	16	0.0	0.0	0	0	?	S	21:04	0:00	[ksoftirqd/0]
root	17	0.1	0.0	0	0	?	I	21:04	0:07	[rcu_preempt]

1.1.3 终止进程

可以使用 `kill <pid>` 或 `pkill <name>` 后者会终止所有同名进程

1.1.4 别名

语法为 `alias <aliasname>="fullcommand"`

例如：

```
sora@sora-virtual-machine:~/Desktop$ alias fulls="ls -a -l -h -t --color=auto"
sora@sora-virtual-machine:~/Desktop$ fulls
total 40K
drwxr-xr-x  3 sora sora 4.0K  9月 11 22:19 .
drwxr-xr-x 14 sora sora 4.0K  9月 11 22:19 ..
-rwxrwxrwx  1 sora sora  18  9月 11 22:19 semester
-rw-r--r--  1 sora sora 12K  9月 11 21:23 .semester.swo
-rw-r--r--  1 sora sora 12K  9月  6 11:59 .semester.swp
drwxrwxr-x  2 sora sora 4.0K  9月  6 11:47 missing
```

这样就把原本的指令简化为 `fulls` 了

1.1.5 配置文件

配置文件是. 开头的文件，例如实验一的.gitconfig

可以在配置文件里使用 `if` 语句等，适配不同情况

`if [["$(uname)" == "Linux"]]; then {do_something}; fi` 上述配置只在 linux 系统生效

1.2 Python 基础

1.2.1 Hello World!

学习每个语言的开始都是这样的。Python 的 Helloworld 非常简单：

```
print("Hello World!")
```

结果如下：

```
D:\PycharmProjects\pythonProject\.venv\Scripts\python.exe D:\PycharmProjects\pythonProject\.venv\ex1.py
Hello World!

进程已结束，退出代码为 0
```

1.2.2 计算器

输入两个数字和符号，执行对应的运算。

```
1         num1 = float(input("请输入第一个数字\n"))
2         op = str(input("请输入运算符\n"))
3         num2 = float(input("请输入第二个数字\n"))
4
5         if op == '+':
6             print(num1 + num2)
7         elif op == '-':
8             print(num1 - num2)
9         elif op == '*':
10            print(num1 * num2)
11        elif op == '/':
12            if num2 == 0:
13                print("除数不能为0! ")
14            else:
15                print(num1 / num2)
16        else:
17            print("非法操作符! \n")
```

```
D:\PycharmProjects\pythonProject\.venv\Scripts\python.exe D:\PycharmProjects\pythonProject\.venv\ex2.py
请输入第一个数字
3
请输入运算符
+
请输入第二个数字
6
9.0
```

1.2.3 猜数字

设法猜出一个 1 100 内的随机整数

```
1      import random
2      answer = int(random.randint(1,100))
3      guess = int(0)
4      while guess != answer:
5          guess = int(input("Guess a number between 1 and 100: "))
6          if guess == answer:
7              print("Correct!\n")
8          elif guess < answer:
9              print("Too low!\n")
10         elif guess > answer:
11             print("Too high!\n")
```

```
Guess a number between 1 and 100: 70
Too low!

Guess a number between 1 and 100: 80
Too low!

Guess a number between 1 and 100: 90
Too low!

Guess a number between 1 and 100: 95
Too high!

Guess a number between 1 and 100: 92
Too low!

Guess a number between 1 and 100: 93
Too low!

Guess a number between 1 and 100: 94
Correct!
```

1.2.4 反转字符串

输入一个字符串，输出其反转

```

1      str = input("请输入字符串:\n")
2      reverse = ''
3      for i in range(0,len(str)):
4          reverse += str[len(str) - i - 1]
5      print( reverse )

```

```

D:\PycharmProjects\pythonProject\.venv\Scripts\python.exe D:\PycharmProjects\pythonProject\.venv\ex4.py
请输入字符串:
abcde
edcba
进程已结束，退出代码为 0

```

1.2.5 列表处理

实现列表初始化、添加、查找与删除元素，排序

```

1      my_list = [5, 3, 8, 1, 2]
2
3      print(f"初始列表: {my_list}")
4
5      new_element = int(input("请输入要添加的元素: "))
6      my_list.append(new_element)
7      print(f"添加元素后的列表: {my_list}")
8
9      element_to_remove = int(input("请输入要删除的元素: "))
10     if element_to_remove in my_list:
11         my_list.remove(element_to_remove)
12         print(f"删除元素后的列表: {my_list}")
13     else:
14         print(f"元素 {element_to_remove} 不在列表中。")
15
16     element_to_find = int(input("请输入要查找的元素: "))

```

```

17         if element_to_find in my_list:
18             print(f"元素 {element_to_find} 在列表中。")
19         else:
20             print(f"元素 {element_to_find} 不在列表中。")
21
22     my_list.sort()
23     print(f"排序后的列表: {my_list}")

```

```

初始列表: [5, 3, 8, 1, 2]
请输入要添加的元素: 4
添加元素后的列表: [5, 3, 8, 1, 2, 4]
请输入要删除的元素: 1
删除元素后的列表: [5, 3, 8, 2, 4]
请输入要查找的元素: 3
元素 3 在列表中。
排序后的列表: [2, 3, 4, 5, 8]

进程已结束，退出代码为 0

```

1.2.6 斐波那契数列

输入 n，输出斐波那契数列第 n 项

```

1         a = b = c = int(1)
2         num = int(input("Enter a number: "))
3         if num == 0 or num == 1:
4             print(1)
5         else:
6             for i in range(1, num):
7                 c = a + b
8                 a = b
9                 b = c
10            print(c)

```



```
D:\PycharmProjects\pyth
Enter a number: 7
21

进程已结束，退出代码为 0
```

1.2.7 冒泡排序

输入列表长度 n 与列表内容，从小到大排序并输出

```
1      import random
2
3      def bubble_sort( arr ):
4          n = len(arr)
5          for i in range(n):
6              swapped = False
7              for j in range(0, n-i-1):
8                  if arr[j] > arr[j+1]:
9                      arr[j], arr[j+1] = arr[j+1], arr[j]
10                     swapped = True
11             if not swapped:
12                 break
13         return arr
14
15     num = int(input("请输入列表的长度: "))
16
17     random_list = [random.randint(1, 100) for _ in range(num)]
18     print(f"原始列表: {random_list}")
19
20     sorted_list = bubble_sort(random_list)
```

21

```
print(f"排序后的列表: {sorted_list}")
```

```
请输入列表的长度: 7
原始列表: [96, 58, 88, 91, 57, 76, 39]
排序后的列表: [39, 57, 58, 76, 88, 91, 96]

进程已结束, 退出代码为 0
```

1.2.8 质数判断

写一个判定质数的函数

```
1      def isPrime(num):
2          if num <= 1:
3              print("False\n")
4              return False
5          else:
6              i = 2;
7              while i ** 2 <= num:
8                  if num % i == 0:
9                      print("False\n")
10                     return False
11                     i += 1
12                     print("True\n")
13                     return True
14
15     num = int(input("Enter a number: "))
16     isPrime(num)
```

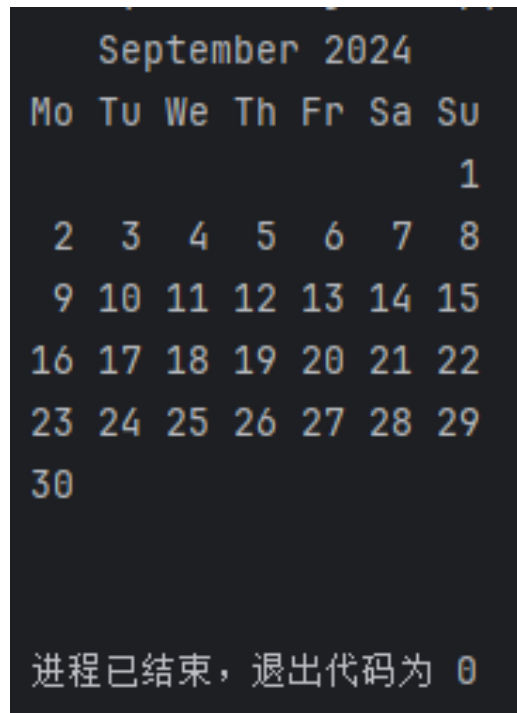
```
Enter a number: 7
True

进程已结束，退出代码为 0
```

1.2.9 日历

打印当前月份的日历

```
1      import calendar
2      from datetime import datetime
3
4      def display_current_month_calendar():
5          now = datetime.now()
6          year = now.year
7          month = now.month
8
9          cal = calendar.TextCalendar()
10         month_calendar = cal.formatmonth(year, month)
11
12         print(month_calendar)
13
14     if __name__ == "__main__":
15         display_current_month_calendar()
```



1.2.10 文件处理

从一个文本中读取内容，反转并写入另一个文件

```
1      def reverse_file_content ( input_file , output_file ):  
2  
3          with open(input_file , 'r') as file :  
4              content = file.read()  
5  
6              reversed_content = content[::-1]  
7  
8          with open(output_file , 'w') as file :  
9              file.write(reversed_content)  
10  
11          input_file = input("请输入要读取的文件名: ")  
12          output_file = input("请输入要写入的文件名: ")
```

```

13
14         reverse_file_content ( input_file , output_file )
15
16     print(f"文件内容已反转并写入到 {output_file}")

```

1.3 Python 视觉应用

原图片已上传。

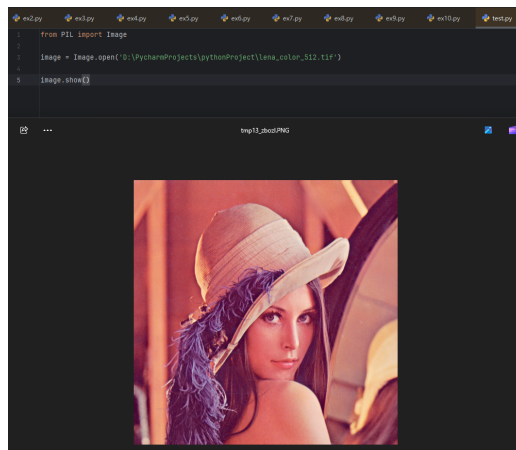
1.3.1 图像读取与显示

```

1     from PIL import Image
2     image = Image.open('D:\PycharmProjects\pythonProject\lena_color_512.tif')
3     image.show()
4     image.save(D:\PycharmProjects\pythonProject\img.png)

```

结果如下：



1.3.2 转换灰度图像

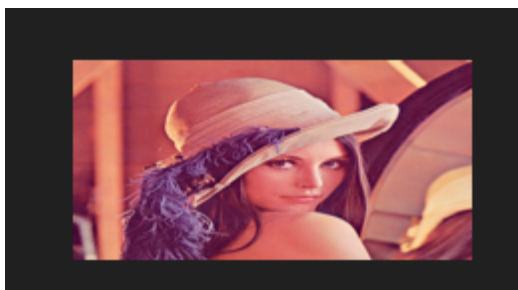
加上 `gray = image.convert('L')` 即可



1.3.3 图像缩放

```
img.resize((width,height))
```

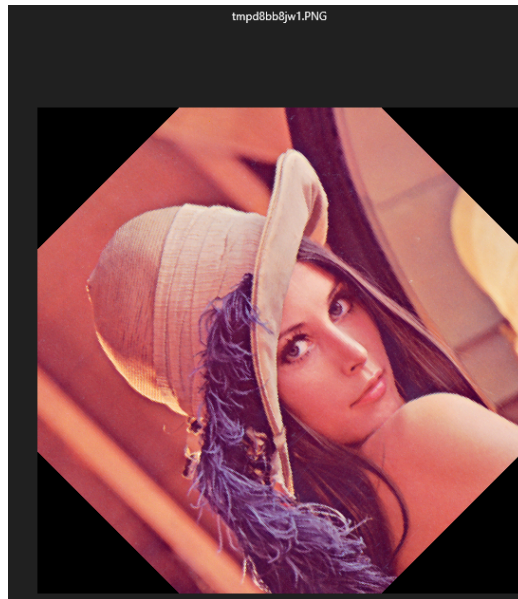
把图像修改为 200*100:



1.3.4 图像旋转

```
img.rotate(degree)
```

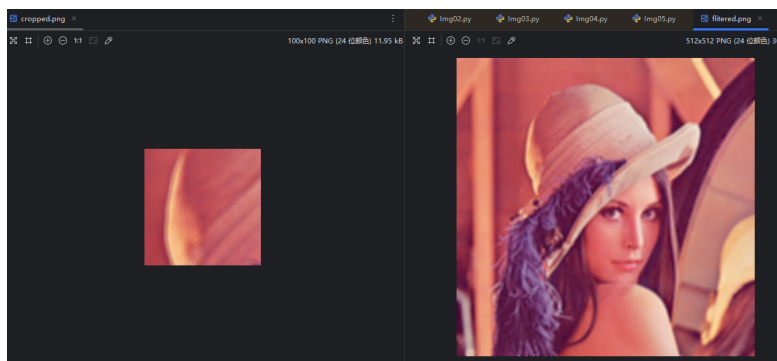
图像旋转 45°



1.3.5 裁剪与模糊

裁剪: `img.crop(left,upper,right,lower)`

滤镜: `img.fliter(ImageFliter.type)` 以下将图片裁剪并模糊:



2 心得

通过这节课的学习，我学到了许多新的东西，相信他们会在之后的学习和工作中发挥大用处。