



中国海洋大学

## 系统开发工具基础实验报告

上课时间:	周五 1-4 节
姓 名:	张仕达
学 号:	23010022094
指导老师:	周小伟

# 1 实验内容

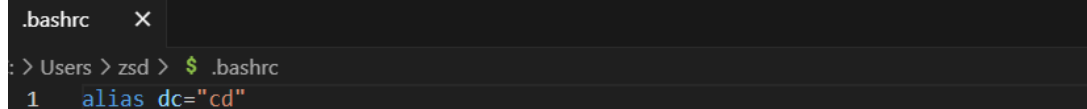
## 1.1 主题 1：命令行环境：别名

### 1.1.1 内容

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。

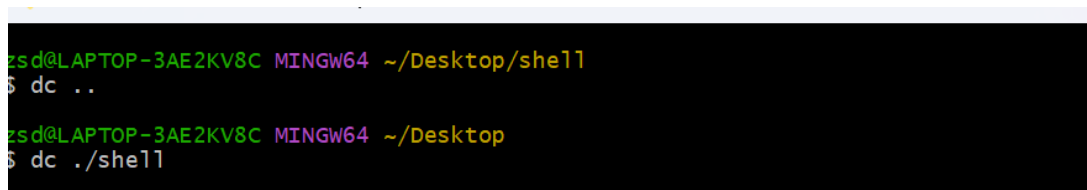
在用户目录里建一个 .bashrc 的文件

加入：alias dc="cd" 并保存即可



```
.bashrc X
: > Users > zsd > $ .bashrc
1  alias dc="cd"
```

### 1.1.2 结果



```
zsd@LAPTOP-3AE2KV8C MINGW64 ~/Desktop/shell
$ dc ..

zsd@LAPTOP-3AE2KV8C MINGW64 ~/Desktop
$ dc ./shell
```

## 1.2 主题 2：命令行环境：远端设备

### 1.2.1 内容

为虚拟机配置一个密钥

### 1.2.2 结果

```
[root@localhost ssh]# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519.
Your public key has been saved in /root/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:dXokG710w28q6B5QhFbWtgyM+AcR6L5MylkvkI58Y root@localhost.localdomain
The key's randomart image is:
+--[ED25519 256]--+
|      o+o      |
|      oo .o.   |
|      . =. = +  |
|      o +  o.O o o |
|    B X +S o.o  o |
| . = B * o..o   o |
| = +   = oo . .  |
| E .   o... .   |
| .      .o      |
+-----[SHA256]-----+
```

## 1.3 主题 3: python 基础练习

### 1.3.1 内容

斐波那契数列

```
1
2 nterms = int(input("输入需要的项数"))
3
4 n1 = 1
5 n2 = 1
6 count = 2
7
8 if nterms <= 0:
9     print("请输入一个正整数。")
10 elif nterms == 1:
11     print("斐波那契数列: ")
12     print(n1)
13 else:
14     print(n1,"",n2,end=" ",)
15     while count < nterms:
16         nth = n1 + n2
17         print(nth,end=" ",)
18         n1 = n2
19         n2 = nth
20         count += 1
```

### 1.3.2 结果

```
输入需要的项数5
1 , 1 , 2 , 3 , 5 ,
PS C:\Users\zsd\Desktop\python>
```

## 1.4 主题 4: python 基础练习

### 1.4.1 内容

判断质数

```
1 num = int(input("请输入一个数字: "))
2
3 if num > 1:
4     for i in range(2,num):
5         if (num % i) == 0:
6             print(num,"不是质数")
7             break
8     else:
9         print(num,"是质数")
10
11 else:
12     print(num,"不是质数")
```

### 1.4.2 结果

```
请输入一个数字: 9
9 不是质数
```

## 1.5 主题 5: python 基础练习

### 1.5.1 内容

返回一个数的阶乘

```
num = int(input("请输入一个数字: "))
factorial = 1
if num < 0:
    print("error")
elif num == 0:
    print("0 的阶乘为 1")
else:
    for i in range(1,num + 1):
        factorial = factorial*i
    print("%d 的阶乘为 %d" %(num,factorial))
```

### 1.5.2 结果

```
请输入一个数字: 5
5 的阶乘为 120
```

## 1.6 主题 6: python 基础练习

### 1.6.1 内容

输出字符转长度

```
demo1.py > ...
1  def findLen(str):
2      counter = 0
3      while str[counter:]:
4          counter += 1
5      return counter
6
7  str = "ddddddd"
8  print(findLen(str))
```

### 1.6.2 结果

```
8
8
```

## 1.7 主题 7: python scipy

### 1.7.1 内容

打印角度单位，返回弧度

```
from scipy import constants

print(constants.degree)
print(constants.arcmin)
print(constants.arcminute)
print(constants.arcsec)
print(constants.arcsecond)
```

### 1.7.2 结果

```
0.017453292519943295
0.0002908882086657216
0.0002908882086657216
4.84813681109536e-06
4.84813681109536e-06
```

## 1.8 主题 8: python scipy

### 1.8.1 内容

打印圆周率、黄金分割率

```
from scipy import constants

print(constants.pi)

print(constants.golden)
```

### 1.8.2 结果

```
3.141592653589793
1.618033988749895
```

## 1.9 主题 9: python numpy

### 1.9.1 内容

np.arange 返回一个向量组

```
demo1.py > ...
1  import numpy as np
2
3  A = np.arange(12).reshape(3, 4)
4  print(A)
```

### 1.9.2 结果

```
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
```

## 1.10 主题 10: python numpy

### 1.10.1 内容

array: 向量; array.ndim: 向量的维度;

array.shape: 向量有几行几列

array.size: 向量的尺寸  $2*3=6$ 。

```
1
2 import numpy as np
3
4 array = np.array([[1, 2, 3],
5                  [2, 3, 4]])
6
7
8 print(array)
9
10
11 print(f'number of dim: {array.ndim}')
12 print(f'shape: {array.shape}')
13 print(f"shape:{array.shape}")
14
15
16 print(f"size:{array.size}")
17
```

### 1.10.2 结果

```
[[1 2 3]
 [2 3 4]]
number of dim:2
shape:(2, 3)
size:6
```

## 1.11 主题 11: python numpy

### 1.11.1 内容

向量的分割

```
1 import numpy as np
2
3 A = np.arange(12).reshape(3, 4)
4
5 # 分割
6 print(np.split(A, 2, axis=1))
7 print(np.array_split(A, 3, axis=1))
8
9 # 纵向分割
10 print(np.vsplit(A, 3))
11
12 # 纵向分割
13 print(np.hsplit(A, 2))
14
```

### 1.11.2 结果

```
[array([[0, 1],
       [4, 5],
       [8, 9]]), array([[ 2,  3],
       [ 6,  7],
       [10, 11]])]
[array([[0, 1],
       [4, 5],
       [8, 9]]), array([[ 2],
       [ 6],
       [10]]), array([[ 3],
       [ 7],
       [11]])]
[array([[0, 1, 2, 3]]), array([[4, 5, 6, 7]]), array([[ 8,  9, 10, 11]])]
[array([[0, 1],
       [4, 5],
       [8, 9]]), array([[ 2,  3],
       [ 6,  7],
       [10, 11]])]
```

## 1.12 主题 12: python numpy

### 1.12.1 内容

向量的合并

```
1
2 import numpy as np
3
4 A = np.array([1, 1, 1])
5 B = np.array([2, 2, 2])
6
7 # 上下合并
8 C = np.vstack((A, B))
9
10 # 左右合并
11 D = np.hstack((A, B))
12
13 print(C)
14 print(D)
15 print(A.shape, C.shape)
16 print(A.shape, D.shape)
17
```



### 1.12.2 结果

```
[[1 1 1]
 [2 2 2]]
[1 1 1 2 2 2]
(3,) (2, 3)
(3,) (6,)
```

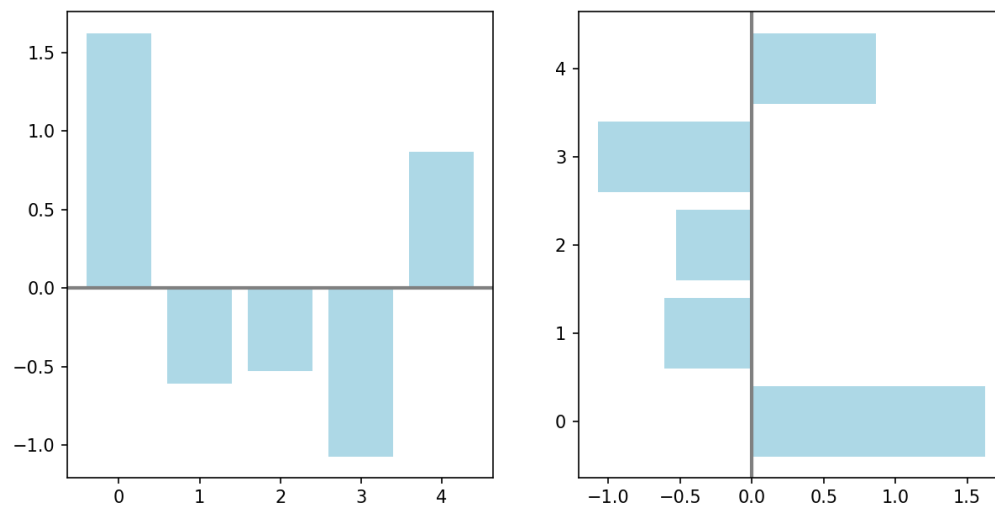
## 1.13 主题 13: python matplotlib

### 1.13.1 内容

绘制简单树状图

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 np.random.seed(1)
4 x = np.arange(5)
5 y = np.random.randn(5)
6
7 fig, axes = plt.subplots(ncols=2, figsize=plt.figaspect(1./2))
8
9 vert_bars = axes[0].bar(x, y, color='lightblue', align='center')
10 horiz_bars = axes[1].barh(x, y, color='lightblue', align='center')
11 #在水平或者垂直方向上画线
12 axes[0].axhline(0, color='gray', linewidth=2)
13 axes[1].axvline(0, color='gray', linewidth=2)
14 plt.show()
```

### 1.13.2 结果



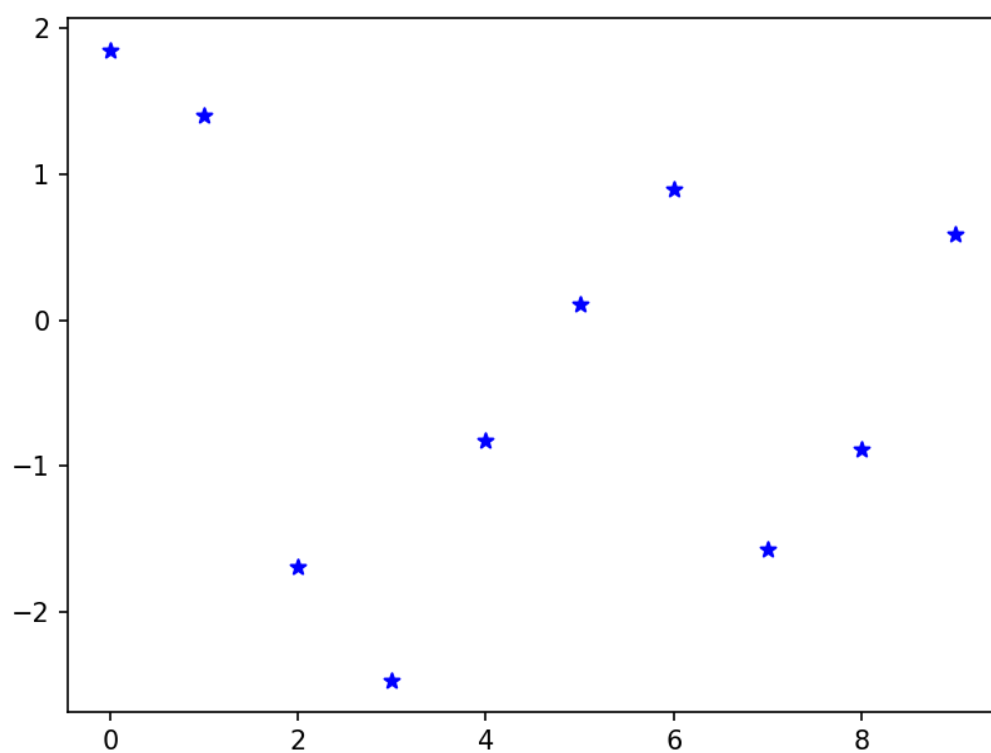
## 1.14 主题 14: python matplotlib

### 1.14.1 内容

绘制简单散点图

```
demo1.py > ...  
1 import matplotlib.pyplot as plt  
2 import numpy as np  
3 x = np.arange(10)  
4 y = np.random.randn(10)  
5 plt.scatter(x, y, color='blue', marker='*')  
6 plt.show()  
7 |
```

### 1.14.2 结果



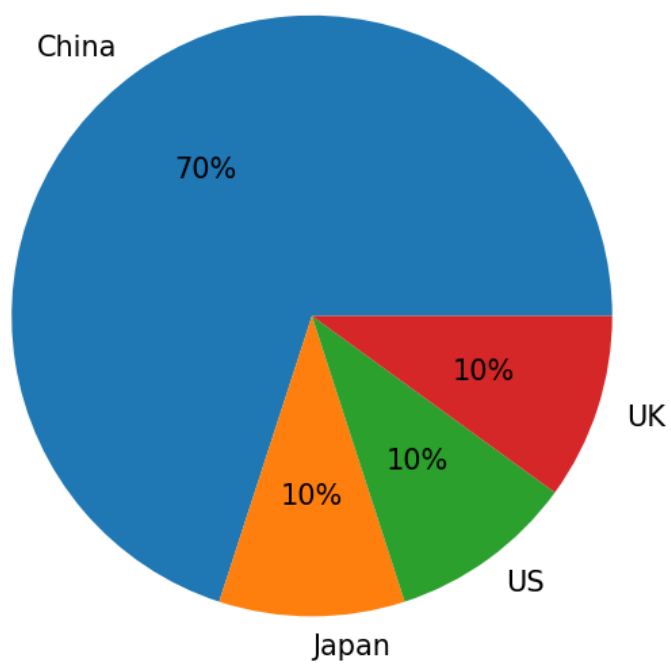
## 1.15 主题 15: python matplotlib

### 1.15.1 内容

绘制简单饼状图

```
demo1.py > ...  
1  import matplotlib.pyplot as plt  
2  labels = ['China', 'Japan', 'US', 'UK']  
3  fracs = [70, 10, 10, 10]  
4  
5  plt.pie(x=fracs, labels=labels, autopct='%0.0f%%')  
6  plt.show()
```

### 1.15.2 结果



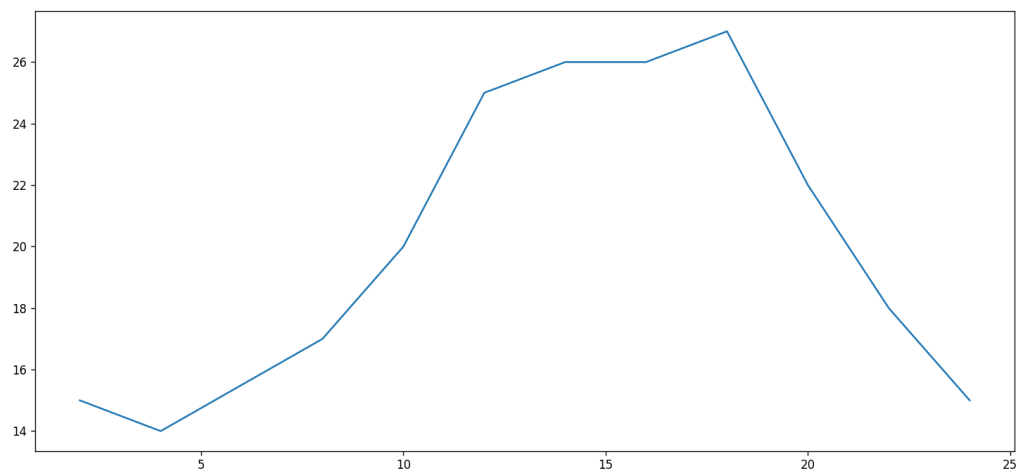
## 1.16 主题 16: python matplotlib

### 1.16.1 内容

绘制简单折线图

```
demo1.py > ...
1  from matplotlib import pyplot as plt
2
3  x=range(2,26,2)
4  #纵坐标
5  y=[15,14,15.5,17,20,25,26,26,27,22,18,15]
6  #设置图片大小
7  plt.figure(figsize=(20,8),dpi=80)
8  #绘图
9  plt.plot(x,y)
10 #展示
11 plt.show()
```

### 1.16.2 结果



## 1.17 主题 17: python PIL

### 1.17.1 内容

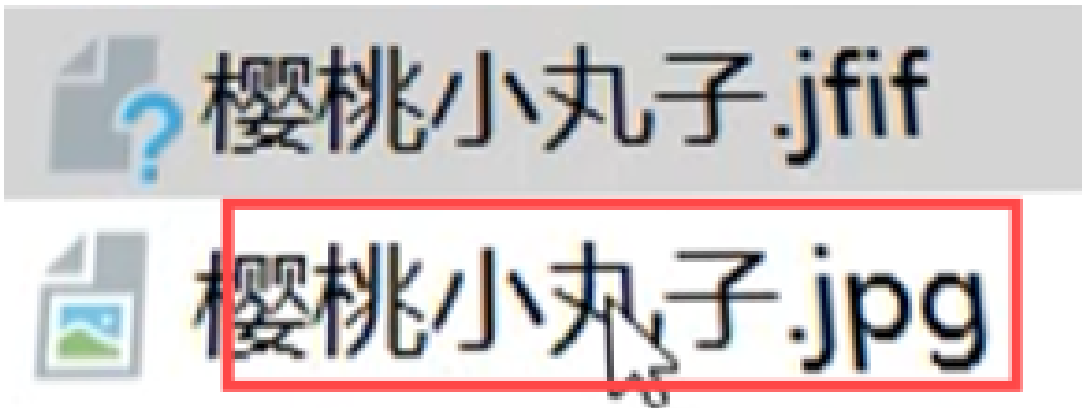
#### 1. 调用图片

```
# 使用PIL 打开、显示和保存图片  
#1. 调用模块  
from PIL import Image  
#2. 打开图片  
im1=Image.open('1.jfif')
```

#### 2. 可展示图片，也可保留图片，可保留不同格式，不同路径

```
#3. 显示图片  
im1.show()  
# 4. 保存文件  
im1.save('樱桃小丸子.jfif')  
#5. 保存到不同地址  
im1.save('C:/Users/86132/Desktop/樱桃小丸子.jfif')  
#6 保存不同格式的图片  
im1.save('樱桃小丸子.jpg')
```

#### 1.17.2 结果



## 1.18 主题 18: python PIL

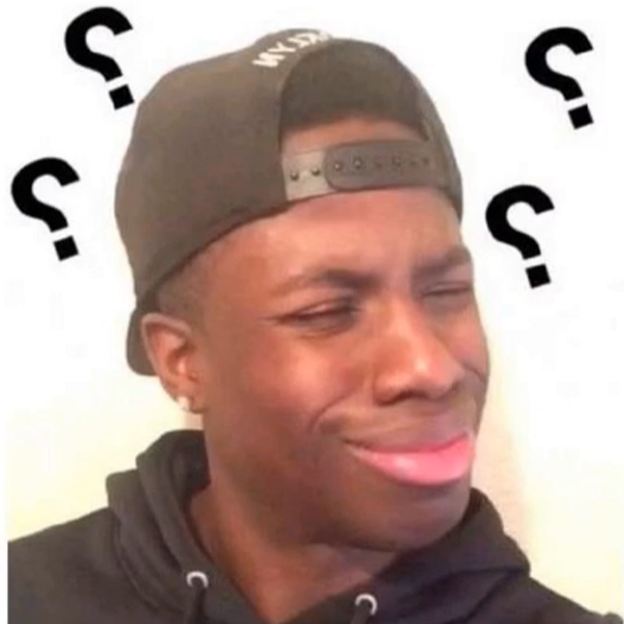
### 1.18.1 内容

实现镜像图片

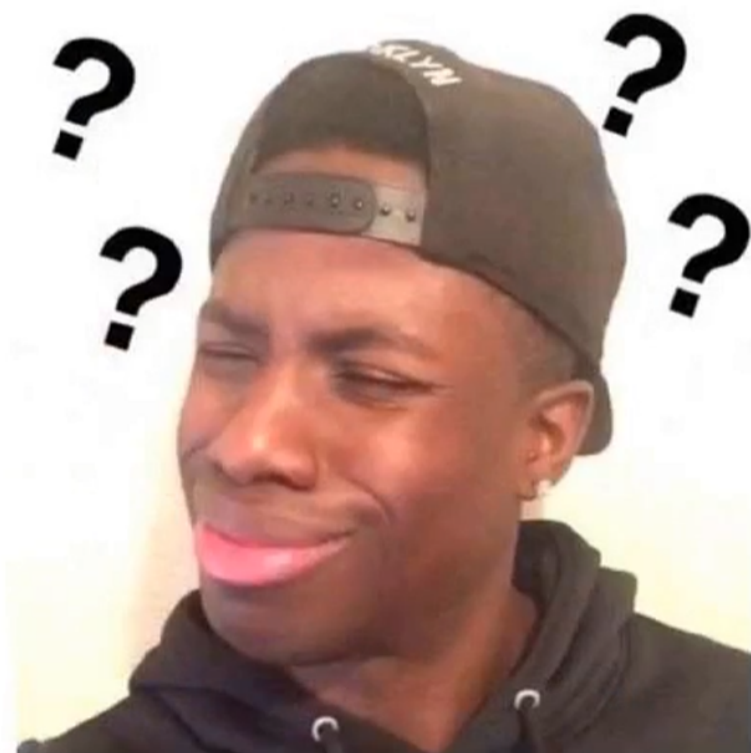
#PIL 镜像功能

```
from PIL import Image
im1=Image.open('表情包.jpg')
im1.show()
im1_lr=im1.transpose(Image.FLIP_LEFT_RIGHT)
im1_lr.show()
```

镜像前:



镜像后:





### 1.18.2 结果

## 1.19 主题 19: python PIL

### 1.19.1 内容

#### 1. 设置两张图片

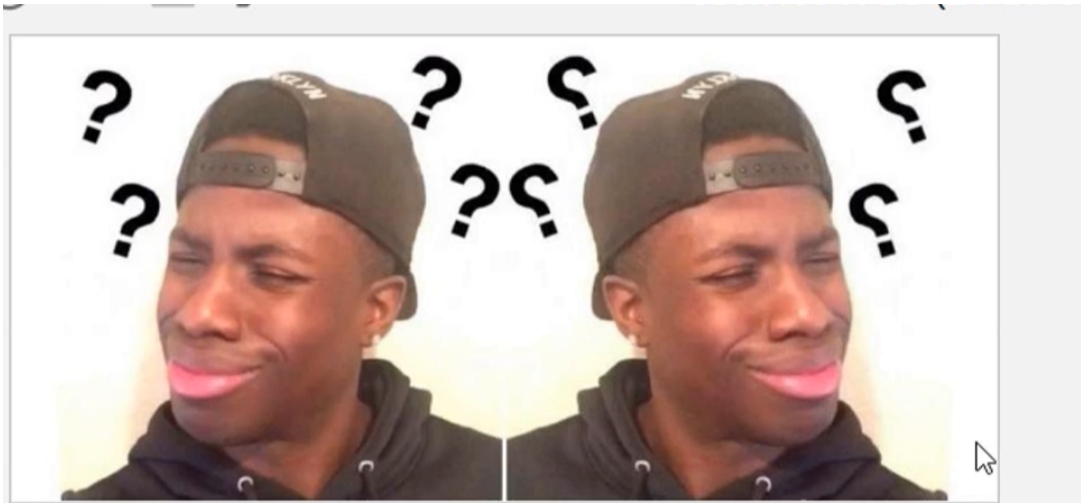
```
from PIL import Image, ImageDraw
im1=Image.open('表情包.jpg')
im2=Image.open('表情包_1r.jpg')

#新建空白图片
#Image.new('RGB/RGBA', 图像尺寸(x,y), (r0-255, g0-255, b0-255))
newim=Image.new('RGB', (850,400), (255,255,255))

#拼接图片
newim.paste(im1,(0,0))
newim.paste(im2,(425,0))
newim.show()
```

#### 2. 设置新图片作拼接

### 1.19.2 结果



## 1.20 主题 20: python PIL

### 1.20.1 内容

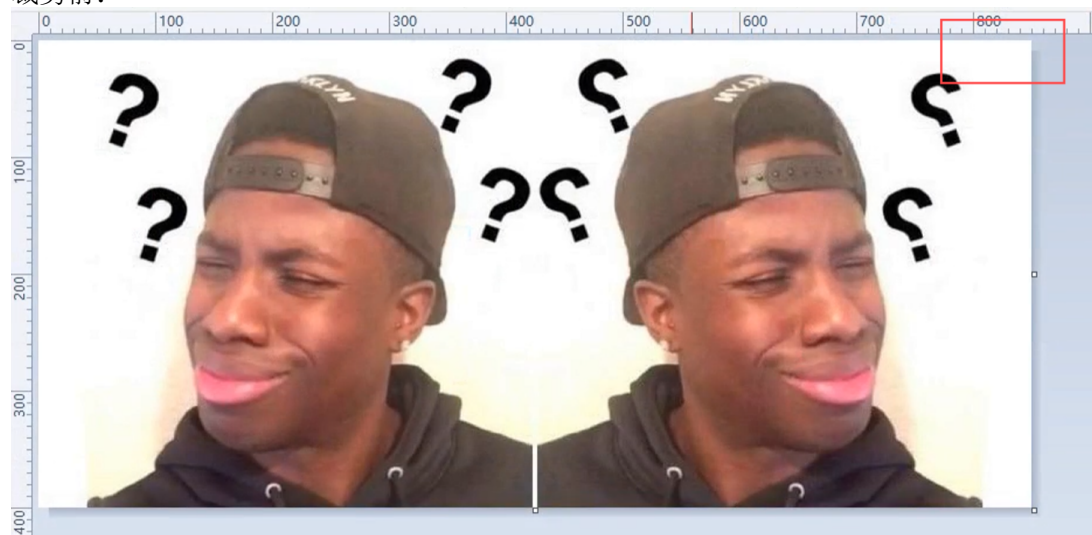
图片裁剪，去除多余白边

*#图片裁剪*

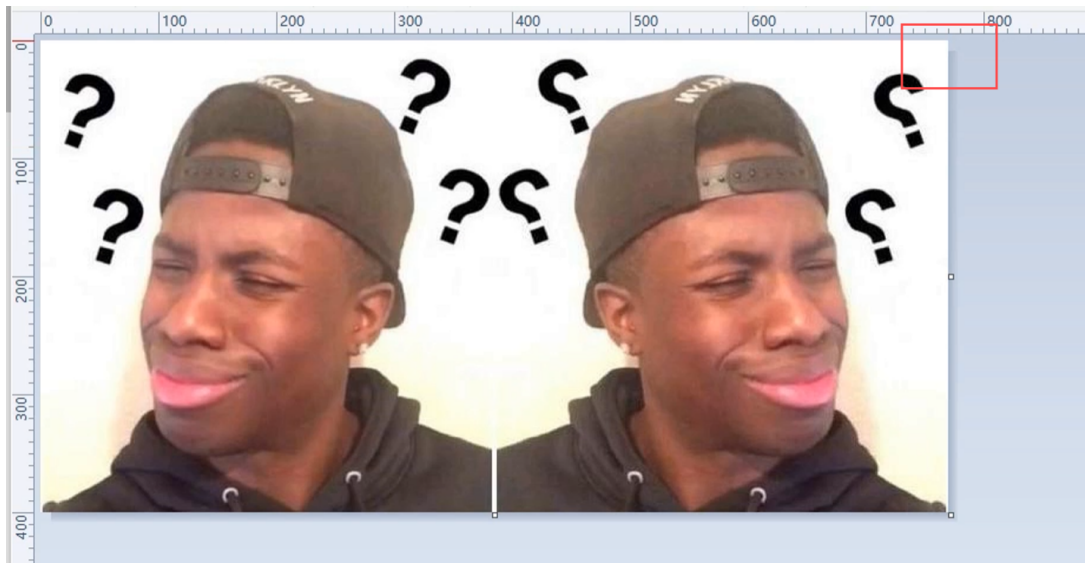
```
from PIL import Image
im1=Image.open('表情包左右拼接.jpg')
# im1.crop((a,b,c,d))。a,b为左上角的位置，c,d为右下角的像素位
im1_cut=im1.crop((40,0,810,400))
im1_cut.show()
im1_cut.save('表情包裁剪.jpg')
```

### 1.20.2 结果

裁剪前:



裁剪后:



## 2 解题感悟

这次的学习内容非常多，学的很累，第一次接触 python 后立马就跟着学了 PIL,matplotlib 等库，但总体来说收获也比前两次课要多的多。也是体会到了 python 这门语言的独特之处，希望可以借此机会好好深入了解一下 python。

[github 地址](#) [\[点击\]](#)