

隐藏攻略

一、巩固练习答案

第0课 print()与变量

1、(操作题) 给自己定个小目标

题目解析：

- 使用多个print()函数，分别把4个语句进行打印；
- 使用一个print()函数和换行符，在每个语句末尾加上一个换行符放到同一行进行打印；
- 使用一个print()函数和三引号，直接把四个语句粘贴到三引号内打印即可。

方法一：使用多个print()函数依次打印每一行内容。

```
1 print('目标：一个月入门python，并写出一个简单的项目！')
2 print('计划：每天花一个小时学习Python小课；每周复盘！')
3 print('惩罚：做不到，接下来一周不能喝奶茶！')
4 print('奖励：完成目标，奖励自己一款索尼蓝牙耳机！')
```

方法二：使用一个print()函数和换行符：在每一行后面加上换行符，然后打印出来。

```
1 print('目标：一个月入门python，并写出一个简单的项目！\n计划：每天花一个小时学习Python小课；每周复\n盘！\n惩罚：做不到，接下来一周不能喝奶茶！\n奖励：完成目标，奖励自己一款索尼蓝牙耳机！')
```

方法三：使用一个print()函数和三引号。直接把内容复制放到三引号内即可。

```
1 print('''目标：一个月入门python，并写出一个简单的项目！
2 计划：每天花一个小时学习Python小课；每周复盘！
3 惩罚：做不到，接下来一周不能喝奶茶！
4 奖励：完成目标，奖励自己一款索尼蓝牙耳机！'''')
```

2、参考答案：D

答案解析：每一次赋值都会改变变量的值，`cat_name = '三公主'`，这句才决定了`cat_name`的值；D错误，在终端显示的时候，是没有引号的。因此正确答案为C。

3、参考答案：ACE

答案解析：变量的命名不能以数字开头，故A错；变量的命名不能使用横杆连接，需要连接多个字符的时候，使用下划线，故B对，C错。命名的时候可以字母加数字，即D的形式。变量的命名只能使用一个词，E选项中用空格隔开，变成了2个词，故错误。

第1课 字符串、整数、浮点数

1、参考答案：C

答案解析： $3^{**}2$ 是3的二次方，结果是9； $7//2$ 是取7除以2结果的整数部分，即3.5的整数部分，为3；故答案为12。

2、参考答案：A

答案解析：整数和浮点数相加之后返回的是浮点数，所以`type(1+1.0)`的数据类型是`<class 'float'>`，选A。

3、参考答案：D

答案解析：`str(1.8)`的结果是字符串类型的1.8，即'1.8'。`int()`函数不能直接转换'1.8'，所以报错。

第2课 条件判断

1、参考答案：C

答案解析：首先明确，这是一个循环嵌套，最外层是一个多向判断，`if...elif...else`语句，只会执行其中一个语句；`if`语句中有一个`if...else...`语句；`elif`语句中也有一个`if...else...`语句。

第一句中有`price = 7000`,

因此，`price >= 10000`条件判断为假；

`price >= 5000`条件判断为真，执行语句`print('再讲讲价，抢救抢救')`；

`price >= 8000`条件判断为假，执行语句`print('每日三次劝说女友回心转意')`。

因此正确答案为C。

2、参考答案：BD

答案解析：年龄是6周岁的时候，进入判断语句，符合`elif 6 <= age < 14`，打印出：“车票打五折”；年龄是60周岁的时候，进入判断语句，不符合`if age<6`的条件，不符合`if 6 <= age < 14`的条件，也不符合`elif 14<=age < 60`的条件，所以进入`else`语句，打印出：“长者可以免费乘坐公交”。

因此正确答案为BD。

3、参考代码：

```
1 score = int(input('请输入你的成绩:'))
2 if score >=60:
3     print("恭喜你，你的成绩及格啦")
4 else:
5     print("你的成绩没有及格，要多加复习！")
```

第3课 input()函数

1、(实操题) 薪酬计算器

答案解析：根据题干要求，整体的薪酬按照业绩分，可以分成四个区间，超过10万，8万至10万，5万至8万，小于5万，这四个区间可以作为if添加句的判断依据，然后根据绩效提成比例计算出提成工资，再加上基础工资即可。

参考代码：

```
1 salary_1 = 6000
2 salary_2 = int(input('你本月的销售业绩是多少元呢? '))
3 if salary_2 >= 100000:
4     salary_all = salary_1 + salary_2 * 10%
5 elif salary_2 >= 80000:
6     salary_all = salary_1 + salary_2 * 8%
7 elif salary_2 >= 50000:
8     salary_all = salary_1 + salary_2 * 6%
9 elif salary_2 < 50000:
10    salary_all = salary_1
11 print(salary_all)
```

第4课 列表、字典、元组

1、参考代码：

```
1 # 要求一:
2 score_chinese = scores['第二组']['小强']['语文'][1]
3 score_math = scores['第二组']['小强']['数学'][1]
4 print('小强期末两个学科的成绩如下：语文'+str(score_chinese)+ '分，数学'+str(score_math)+ '分。')
5 # 小强期末两个学科的成绩如下：语文95分，数学91分。
```

```
6
7 # 要求二:
8 scores['第一组']['小红']['数学'][1] = 92
9 print(scores['第一组']['小红']['数学'])
10 # [91, 92]
11
12 # 要求三:
13 score_xiaolan = scores['第二组']['小兰']
14 scores['第二组']['小蓝'] = score_xiaolan # 获取小兰的成绩赋值给小蓝
15 del scores['第二组']['小兰'] # 删除小兰
16 print(scores['第二组'])
17 # {'小强': {'语文': [92, 95], '数学': [90, 91]}, '小蓝': {'语文': [91, 95], '数学': [86, 98]}}
```

2、参考答案：B

答案解析：列表增加新元素使用append()函数即可，不需要赋值；列表删除元素使用del，注意后面需要加上空格，此题主要考查增删的格式。

3、参考答案：C

答案解析：字典中，有两个键为“小明”，出现重复时，只算一次，默认以后面的键值对为准，可以打印该字典查看该字典为{'小明': 90, '小红': 92, '小微': 99}。

第5~6课 for循环与while循环

1、参考答案：C

答案解析：i是字典的键，d[i]是字典键对应的值，第三次循环键为“十二”，键对应的值为“高铁二等座”。

2、参考答案：A

答案解析：当终端中打印出7之后紧接着count变量加到8。然后再次进入循环，满足if条件，执行countinue语句进入下一次循环，之后不断重复这个过程，变成了无限循环，所以终端中最后打印出来的结果是7。

3、参考答案：AD

答案解析："%"表示格化式一个对象为字符，所以本题的打印格式为“i VS j”；

在循环中，先遍历a，取1，然后遍历b，取4, 5, 6，得到：

1 VS 4

1 VS 5

1 VS 6

继续遍历a, 取2, 然后遍历b, 取4, 5, 6, 得到:

2 VS 4

2 VS 5

2 VS 6

继续遍历a, 取3, 然后遍历b, 取4, 5, 6, 得到:

3 VS 4

3 VS 5

3 VS 6

因此正确答案为AD。

4、参考代码:

```
1  i = 1000
2  j=0
3  while True:
4      i = i * 1.05
5      if i >=5000:
6          break
7      j+=1
8  print(j)
```

第9课 函数

1、参考答案: A

答案解析: global name是修改全局变量的意思, 所以在执行完name='蒙毅'后, name变量的值由'易小川'变为了'蒙毅'。

2、参考答案: A

答案解析: 此函数的功能是计算数字a平方加a, 所以当调用math(10)函数时, 10传递给了math(x)中的x, 然后进行运算, 并将运算结果存放到变量y。

3、参考代码:

```
1  product_name = input('请输入商品名称: ') # 商品名称
2  product_price= int(input('请输入商品单价: ')) # 商品单价
```

```
3 product_number= int(input('请输入商品数量: ')) # 商品数量
4
5 # 计算总价并打印相关信息
6 def product(x,y,z):
7     sum_price = y*z # 总价 = 单价*数量
8     print('入货名单如下: \n商品: %s, 单价: %d元, 数量: %d件, 总价: %d元' % (x,y,z,sum_price)) #
9     打印相关商品信息
10 product(product_name,product_price,product_number) # 调用函数
```

第10课 工作计量器

1、参考答案：C

答案解析：%nf是保留小数的意思，保留多少位小数，取决于n的值，故答案是%.2f。即保留两位小数的意思。

2、参考答案：ABD

答案解析：答案C值之所以是错误的，是因为math.ceil(2)，因为2是整数，不是浮点数，所以不能进行向上取整操作，选项C的答案结果应为2。

3、参考代码：

```
1 # 方法一：if...else语句
2 page_number = 3856%400 # '%'号是求余数的意思
3 if page_number!=0: # 当余数不等于0时，证明page_number的值是小数
4     page_number = int(3856/400)+1 # 将小数进行取整，并加1
5     print('需要的原稿纸张数为: %d张' % (page_number))
6 else:
7     print('需要的原稿纸张数为: %d张' % (3856/400)) #当余数等于0时，直接打印结果
8
9 # 方法二：math.ceil()函数
10 import math
11 page_number = math.ceil(3856/400) # 直接对运行结果进行向上取整
12 print('需要的原稿纸张数为: %d张' % (page_number))
```

第12~13课 类与对象

1、参考答案：D

答案解析：只调用Son_dog()类时，不会调用到character()方法，此时运行结果为空。

2、参考答案：C

答案解析：action是character()方法中定义的局部变量，不是实例属性，不能在方法外调用。

3、参考答案：B

答案解析：最后一行调用describe() 方法，分别给color、price、size传3个参数 '米色'、 '10'、 '中号'，然后执行第8行调用父类的describe() 方法，给size 传参数： '中号'，而color 使用调用时的赋值： '白色'。所以最终打印的结果是我花了10元买了一个白色的中号玩具，选B。

第15课 编码和文件的读写

1、参考代码：

```
1 a=input('输入名字：')# 请在这里写入你的名字  
2 print(a.encode('utf-8'))
```

2、参考答案：C

答案解析：A选项中open函数里的参数是'w'，会覆盖掉之前写入的内容。B选项中绝对路径前缺少了'r'，报错。D选项中'r'代表读取权限，并非写入。

第16课 模块的定义与使用

1、参考答案：C

答案解析：A选项报错e没有定义，B选项需要import math才可以，D选项报错e没有定义。

2、参考答案：不可以

答案解析：writerow()方法接收一个参数，参数需要是一个列表。

二、项目与案例篇代码合集

基础类

项目1：Excel表格的合并

```
1 import os, openpyxl      # openpyxl是第三方模块, 需安装
2
3 # 定义walkFile()函数获取文件夹的所有文件
4 def walkFile(path):
5     for root_path, dirs_list, file_list in os.walk(path):
6         # root_path是当前目录, dirs_list是子目录, file_list是子文件
7         Read_Write_File(root_path, file_list)
8
9 # 定义Read_Write_File()函数用于读写表格
10 def Read_Write_File(root_path, file_list):
11     # 新建表格
12     wb_w = openpyxl.Workbook()
13     sheet_w = wb_w.active
14     sheet_w.title = '汇总数据'  # 自定义一个工作表
15     # 遍历文件夹中的文件
16     for file in file_list:
17         file_path = os.path.join(root_path, file)
18         print(file_path)      # 打印文件中所有文件名
19
20         if '.xlsx' in file:  # 判断文件的扩展名是不是xlsx
21             wb_r = openpyxl.load_workbook(file_path)
22             sheet_name = wb_r.sheetnames[0] # 获取第0个工作表的名称
23             sheet_r = wb_r[sheet_name]      # 获取工作表
24
25             # 写入汇总表格并保存
26             for row in sheet_r.values:
27                 sheet_w.append(row)        # 写入新表格
28             wb_w.save('New_Excel.xlsx')  # 保存文件
29
30 # 定义main()函数, 获取路径并执行合并
31 def main():
32     path_pwd = os.getcwd()    # 获取当前工作路径
33     os.chdir(path_pwd)       # 切换到当前目录
34     path = path_pwd          # 使用相对路径获取 测试表2文件夹
35     walkFile(path)
36     print('合并完毕! ')
```

```
38 if __name__ == "__main__":
39     main()
```

项目2：关键字快速查找txt文档

```
1 import os, time
2
3 # 获取文件夹下的所有文档
4 def path():
5     print('———欢迎使用关键字搜索txt文档程序———')
6     time.sleep(1)
7     while True:
8         try:
9             folder_path = input('请输入您要查询的文件夹绝对路径: ')
10            file_list = os.listdir(folder_path)
11            break
12        except FileNotFoundError:
13            print('文件夹路径格式有误, 请重新输入\n')
14    return folder_path, file_list
15
16 # 关键字内容匹配
17 def txt_search(folder_path, file_list):
18     while True:
19         txt_count = 0
20         keywords = input('请输入您要查询的内容关键字: ')
21
22         for file_name in file_list:
23             if '.txt' in file_name:
24                 txt_path = folder_path + '\\\\' + file_name
25                 with open(txt_path, 'r', encoding='utf-8') as content:
26                     content = content.read()
27                     if keywords in content:
28                         print('包含关键字的txt文件名为: %s' %(file_name))
29                         txt_count += 1
30
31             if txt_count == 0:
32                 print('很抱歉, 您搜索的关键字不存在任何的txt文档中! ')
33
34             again = input('您是否继续查询, 是请输入:Y , 否请输入任意键退出' )
35             if again != 'Y':
36                 break
```

```
37
38 # 重复操作
39 def main():
40     while True:
41         folder_path,file_list = path()
42         txt_search(folder_path,file_list)
43         answer = input('您是否想重新查询，是请输入：Y，否请输入任意键退出')
44         if answer != 'Y':
45             break
46         print('————欢迎再次使用————')
47
48 if __name__ == "__main__":
49     main()
```

项目3：批量修改本地文件名

```
1 import os
2
3 path = input('输入文件所在的路径')
4 new_name = input('输入文件名（会自动给目录下的所有文件编序号）')
5 os.chdir(path) # 切换到当前路径
6 all_names = os.listdir(path) # 获取所有文件名（含拓展名）
7 file_names = [name for name in all_names if os.path.isfile(name)] # 筛选出文件
8
9 for i in range(len(file_names)):
10     extended_name = os.path.splitext(file_names[i])[1] # 获取文件后缀名
11     file_code = '_%03d'% (i+1) # 给文件序号，格式：文件名_001，随文件个数递增
12     newname = new_name+file_code+extended_name # 拼接新文件名
13     oldname = file_names[i] # 拼接原文件名
14     os.rename(oldname,newname) # 重命名
15     print(oldname,'>>>',newname)
16
17 print('已完成%s个文件的重命名。' % (i+1))
```

项目4：用Python定时爬取微博热搜

```
1 import requests,datetime,openpyxl, time,schedule
2 from bs4 import BeautifulSoup
3
```

```
4  # 爬取微博热榜新闻标题, 链接, 热度
5  def weiboSpider():
6      destination_url = 'https://s.weibo.com/top/summary'
7      destination = requests.get(destination_url)
8      soup = BeautifulSoup(destination.text, 'html.parser')
9      data_list = soup.find('div', class_='data').find_all('td', class_='td-02')
10     content_lists = []
11     for data in data_list:
12         title = data.find('a').text.strip()
13         title_href = data.find('a')['href']
14         href = 'https://s.weibo.com' + title_href
15         try:
16             heat = data.find('span').text
17
18         except AttributeError:
19             heat = '暂无'
20         content_lists.append([title, href, heat])
21     return content_lists
22
23 # 写入Excel文件
24 def writeExcel(content_lists):
25     date = datetime.datetime.now().strftime('%m-%d')      # 获取当天时间并转化为“月-日”格式
26     # 打开Excel文件, 如果不存在则新建Excel文件
27     try:
28         wb = openpyxl.load_workbook("微博热榜.xlsx")
29         sheets = wb.sheetnames
30         # 以当天日期为工作表名称, 判断工作表名称是否存在, 不存在则新建
31         if date in sheets:
32             sheet = wb[date]      # 工作表存在, 读取该工作表
33         else:
34             sheet = wb.create_sheet(date, 0) # 工作表不存在, 则新建, 0表示插入工作表在第0个位置
35             sheet.append(['标题', '链接', '热度'])
36
37     except:
38         wb = openpyxl.Workbook()
39         sheet = wb.active
40         sheet.title = date
41         sheet.append(['标题', '链接', '热度'])
42     # 遍历爬取数据, 并写入Excel文件
43     for content_list in content_lists:
44         sheet.append(content_list)
```

```
45     wb.save("微博热榜.xlsx")
46
47 # 调用爬虫和写入Excel文件
48 def main():
49     content_lists = weiboSpider()
50     writeExcel(content_lists)
51     print('成功爬取微博热榜，并写入Excel!')
52
53 # 执行定时爬取
54 schedule.every(2).hour.do(main) # 每隔2小时执行一次，修改数字可更改定时时间
55 while True:
56     schedule.run_pending()      # 检测并运行设定好的所有定时任务
57     time.sleep(1)
```

项目5：邮箱的附件发送

```
1  from openpyxl import load_workbook
2  import smtplib
3  from email.mime.text import MIMEText
4  from email.mime.multipart import MIMEMultipart
5
6  # 设置邮箱账号
7  account = input('请输入发送方邮箱账户: ')
8  # 设置邮箱授权码
9  token = input('请输入发送方邮箱授权码: ')
10
11 # 设置邮箱服务器，端口
12 # 这里是qq邮箱的服务器和端口，具体根据自己选择的邮箱来填写
13 smtp = smtplib.SMTP_SSL('smtp.qq.com', 465)
14
15 # 登录qq邮箱
16 smtp.login(account, token)
17
18 # 编写正文内容，content内容根据自己需求进行修改
19 content = '这是邮件的正文内容！'
20 email_content = MIMEText(content, 'plain', 'utf-8')
21
22 # 读取工作表文件数据，括号里根据本地文件的路径来填写
23 with open('info.xlsx', 'rb') as f:
24     file_data = f.read()
25
```

```
26 # 设置内容类型为附件
27 attachment = MIMEText(file_data, 'base64', 'utf-8')
28
29 # 设置附件标题以及文件类型, filename后面根据本地文件路径来填写
30 attachment.add_header('Content-Disposition', 'attachment', filename='info.xlsx')
31
32 # 创建复合邮件对象
33 msg = MIME Multipart()
34 # 添加正文到复合邮件对象中
35 msg.attach(email_content)
36 # 添加附件到复合邮件对象里
37 msg.attach(attachment)
38
39 # 设置发送者信息, 根据自己的发件信息来填写
40 msg['From'] = input('输入你的信息: ')
41 # 设置接受者信息
42 msg['To'] = input('输入对方的信息: ')
43 # 设置邮件标题
44 msg['Subject'] = '一封自由的邮件'
45 to_adder = input('请输入接收方的邮箱: ')
46
47 # 发送邮件, 收件邮箱地址根据情况修改填写
48 smtp.sendmail(account, to_adder, msg.as_string())
49 # 关闭邮箱服务
50 smtp.quit()
```

模板类

项目6：使用Python识别文字图片

```
1 # encoding:utf-8
2
3 # 权限认证文档代码示例修改, 获取access_token
4 import requests
5
6 # client_id 为官网获取的AK, client_secret 为官网获取的SK
7 AK = 【百度云应用的AK】 # 传入自己的AK
8 SK = 【百度云应用的SK】 # 传入自己的SK
```

```

9
10 host =
11     'https://aip.baidubce.com/oauth/2.0/token?grant_type=client_credentials&client_id=%s&client_secret=%s' % (AK, SK)
12 response = requests.get(host)
13 if response:
14     print(response.json()) # 返回信息中含有用户特定的信息 (access_token值)
15
16 # 文字识别文档代码示例修改, 获取图中文字
17 import requests
18 import base64
19
20 request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/accurate_basic"
21 # 二进制方式打开图片文件
22 f = open(r'[本地文件路径]', 'rb')      # 传入自己的文件路径
23 img = base64.b64encode(f.read())
24
25 params = {"image":img}
26 access_token = response.json()["access_token"] # 获取用户信息access_token
27 request_url = request_url + "?access_token=" + access_token # 加入用户信息
28 headers = {'content-type': 'application/x-www-form-urlencoded'}
29 contents = requests.post(request_url, data=params, headers=headers)
30 if contents:
31     print(contents.json())
32     # 遍历打印内容
33     for content in contents.json()['words_result']:
34         print('识别出来的内容为: \n', content['words'])

```

项目7：统计词频制作词云图

```

1 import re,collections,wordcloud,jieba
2 import matplotlib.pyplot as plt
3 from scipy.misc import imread
4
5 file = open('咖啡行业定义与分类.txt')
6 data = file.read()
7 file.close()
8
9 pattern = re.compile(u'\t|\n|\.|-|:|;|\\|\\(|\\?|"|"|:|;|")')
10 clear_data = re.sub(pattern, '', data)

```

```

11
12 clear_data = jieba.cut(clear_data, cut_all = False)
13 object_list = []
14 remove_words = [u'的', u', ',u'和', u'是', u'随着', u'对于', u'对', u'等', u'能', u'都',
15             u'。',u' ',u'、',u'中',u'在',u'了',,
16             u'通常',u'我们',u'需要',u'具有',u'不同',u'用',u'年',u'月',u'日',
17             u'日常',u'主要',u'同时',u'指出',u'我国',u'及',u'2019',u'达到',u'为',
18             u'占据',u'分别',u'2018',u'从',u'概括',u'较',u'将',u'为',u'受',
19             u'被',u'以',u'比',u'上',u'2',u'也',u'到',u'已']
20
21 for word in clear_data:
22     if word not in remove_words:
23         object_list.append(word)
24
25 word_counts = collections.Counter(object_list)
26 top_word_counts = word_counts.most_common(10)
27
28 mk=imread("咖啡.jpg")
29 word_cloud = wordcloud.WordCloud(
30     font_path='simhei.ttf',
31     mask=mk,
32     max_words=100,
33     max_font_size=100
34 )
35
36 word_cloud.generate_from_frequencies(word_counts)
37 image_colors = wordcloud.ImageColorGenerator(mk)
38 word_cloud.recolor(color_func=image_colors)
39
40 plt.imshow(word_cloud) # 显示词云
41 plt.axis('off') # 关闭坐标轴
42 plt.show() # 显示图像

```

项目8：使用Python绘制疫情地图

```

1 # 爬取数据
2 import requests, json
3 url = 'https://view.inews.qq.com/g2/getOnsInfo?name=disease_h5' # 数据接口
4 data = json.loads(requests.get(url).json()['data']) # 读取数据转换为 JSON 格式
5 update_time = data['lastUpdateTime'] # 更新时间

```

```
6 China = data['areaTree'][0]['children'] # 全国各地具体数据
7
8 map_data = []
9 for i in China :
10     province = i["name"]      # 获取省份名称
11     province_confirm = i["total"]["confirm"]
12     map_data.append((i["name"],province_confirm))    # 保存省份名称和该省确诊人数
13
14 # 绘制地图
15 from pyecharts.charts import Map
16 from pyecharts import options as opts
17 data_label = "确诊"
18 map_title = "全国疫情图"
19 c = (Map()           # 声明一个map对象
20       .add(data_label, map_data, "china")    # 添加数据
21       # 设置标题和颜色
22       .set_global_opts(title_opts=opts.TitleOpts(title=map_title),
23                           visualmap_opts=opts
24                           .VisualMapOpts(split_number=6,          # 分成6个区间
25                                         is_piecewise=True,    # 将颜色图例分开展示
26                                         pieces=[{"min":1,"max":9,"label":"1-9人","color":"#ffefd7"},
27                                                 {"min":10,"max":99,"label":"10-99人","color":"#ffd2a0"},
28                                                 {"min":100,"max":499,"label":"100-499人","color":"#fe8664"},
29                                                 {"min":500,"max":999,"label":"500-999人","color":"#e64b47"},
30                                                 {"min":1000,"max":9999,"label":"1000-9999
人","color":"#c91014"},,
31                                                 {"min":10000,"label":"10000人及以上","color":"#9c0a0d"}]
32       ))
33   )
34
35 c.render("全国实时疫情.html") # 生成html文
件.set_global_opts(title_opts=opts.TitleOpts(title=map_title),
36                     visualmap_opts=opts
37                     .VisualMapOpts(split_number=6,          # 分成6个区间
38                                   is_piecewise=True,    # 将颜色图例分开展示
39                                   pieces=[{"min":1,"max":9,"label":"1-9人","color":"#ffefd7"},
40                                           {"min":10,"max":99,"label":"10-99人","color":"#ffd2a0"},
41                                           {"min":100,"max":499,"label":"100-499人","color":"#fe8664"},
42                                           {"min":500,"max":999,"label":"500-999人","color":"#e64b47"},
43                                           {"min":1000,"max":9999,"label":"1000-9999
人","color":"#c91014"},,
44                                           {"min":10000,"label":"10000人及以上","color":"#9c0a0d"}]
```

```
45           ]))
46     )
47
48 c.render("全国实时疫情.html") # 生成html文件
```

项目9：使用opencv绘制素描图

```
1 import cv2
2 def dodging(image, mask):    # 亮化操作
3     return cv2.divide(image, 255 - mask, scale=256)
4
5 def control_to_sketch(image, to_image):
6     img_rgb = cv2.imread(image)
7     img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)
8     img_gray_inv = 255 - img_gray
9     img.blur = cv2.GaussianBlur(img_gray_inv, ksize=(21, 21),
10                                sigmaX=0, sigmaY=0)
11    img_blend = dodging(img_gray, img.blur)
12    cv2.imshow("sketch", img_blend)
13    cv2.waitKey(100)
14    cv2.destroyAllWindows()
15    cv2.imwrite(to_image, img_blend)
16
17 image = '小猫咪.jpg'      # 此处需要传递一个图片的路径
18 to_image = '小猫咪1.jpg'    # 此处传递生成图片的路径
19 control_to_sketch(image, to_image)
```

项目10：批量提取pdf文档图片

```
1 # 从pdf中提取图片
2 def pdf_pic(path, pic_path):
3     t0 = time.clock()
4
5     # 使用正则表达式来查找图片
6     checkXO = r"/Type(?=.*XObject)"
7     checkIM = r"/Subtype(?=.*Image)"
8
9     # 打开pdf
10    doc = fitz.open(path)
```

```
11
12     # 图片计数
13     imgcount = 0
14     lenXREF = doc._getXrefLength()
15
16     # 打印PDF的信息
17     print(" 页数: {}, 对象: {}".format(len(doc), lenXREF - 1))
18
19     # 遍历每一个对象
20     for i in range(1, lenXREF):
21         # 定义对象字符串
22         text = doc._getXrefString(i)
23         isXObject = re.search(checkXO, text)
24
25         # 使用正则表达式查看是否是图片
26         isImage = re.search(checkIM, text)
27
28         # 如果不是对象也不是图片，则continue
29         if not isXObject or not isImage:
30             continue
31         imgcount += 1
32
33         # 根据索引生成图像
34         pix = fitz.Pixmap(doc, i)
35         # 根据pdf的路径生成图片的名称
36         new_name = "img{}.png".format(imgcount)
37         new_name = new_name.replace(':', '')
38
39         # 如果pix.n<5,可以直接存为PNG, 否则先转换CMYK
40         if pix.n < 5:
41             pix.writePNG(os.path.join(pic_path, new_name))
42         else:
43             pix0 = fitz.Pixmap(fitz.csRGB, pix)
44             pix0.writePNG(os.path.join(pic_path, new_name))
45             pix0 = None
46
47         # 释放资源
48         pix = None
49         t1 = time.clock()
50         print("运行时间:{}s".format(t1 - t0))
51         print("提取了{}张图片".format(imgcount))
52
```

```
53 if __name__=='__main__':
54     path = input('请输入需要批量提取图片的pdf文档路径: ') # pdf文档路径
55     pic_path = input('请输入保存图片的文件夹路径: ') # 保存图片的文件夹路径
56     pdf_pic(path, pic_path)
```

趣味类

项目12：Python将图片转换为字符画

```
1 from PIL import Image
2
3 char_set = ['.', '!', '~', '@', '#', '$', '%', '^', '&', '*', '(', ')', '-', '+']
4 # 是我们的字符画所使用的字符集
5
6 # 设置字符画的宽和高
7 pic_width = 55
8 pic_height = 35
9
10 # 将256灰度映射到70个字符上，RGB值转字符的函数，alpha为透明度
11 def get_char(R, G, B, alpha=256):
12     if alpha == 0:
13         return ' '
14
15     else:
16         pic_gray = int(0.2126 * R + 0.7152 * G + 0.0722 * B) # 计算灰度
17         pic_unit = (256.0 + 1) / len(char_set)
18         return char_set[int(pic_gray / pic_unit)] # 不同的灰度对应着不同的字符
19
20 # 输入图片路径，打印出字符画
21 def main():
22     pic = input('请输入图片所在位置 (图片文件路径) : ')
23     img = Image.open(pic)
24     im = img.resize((pic_width, pic_height), Image.NEAREST)
25     content = ""
26     for i in range(pic_height):
27         for j in range(pic_width):
28             content += get_char(*im.getpixel((j, i)))
```

```
29         content += '\n'
30     print(content)
31
32 if __name__ == '__main__':
33     main()
```

项目13：图片的灰度处理及二值化

```
1 # 图片二值化
2 from PIL import Image
3 img = Image.open('咖啡.jpg') # 这里的图片替换成本地图片路径
4 # 模式L为灰色图像，每个像素占有8个比特位，0表示纯黑，255表示纯白
5 Img = img.convert('L')
6 Img.save("咖啡2.jpg") # 生成的灰度图片路径
7 # 自定义灰度界限
8 reference = 128
9 table = []
10 for i in range(256):
11     if i < reference:
12         table.append(0)
13     else:
14         table.append(1)
15 # 图片二值化
16 photo = Img.point(table, '1')
17 photo.save("咖啡3.jpg") # 生成的二值图片路径
```

项目14：使用Turtle库绘制时钟

```
1 from turtle import *
2 from datetime import *
3
4 def skip(step):
5     penup()
6     forward(step)
7     pendown()
8
9 def mkHand(name, length):
10    reset() #清空窗口并重设为默认值
11    skip(-length*0.1)
```

```
12 begin_poly()  
13 forward(length*1.1)
```