

隐藏攻略

一、巩固练习答案

第 0 课 `print()`与变量

第 1 课 字符串、整数、浮点数

第 2 课 条件判断

第 3 课 `input()`函数

第 4 课 列表、字典、元组

第 5~6 课 `for` 循环与 `while` 循环

第 9 课 函数

第 10 课 工作计量器

第 12~13 课 类与对象

第 15 课 编码和文件的读写

第 16 课 模块的定义与使用

二、项目与案例篇代码合集

基础类

项目 1: Excel 表格的合并

项目 2: 关键字快速查找 txt 文档

项目 3: 批量修改本地文件名

项目 4: 用 Python 定时爬取微博热搜

项目 5: 邮箱的附件发送

模板类

项目 6: 使用 Python 识别文字图片

项目 7: 统计词频制作词云图

项目 8: 使用 Python 绘制疫情地图

项目 9: 使用 opencv 绘制素描图

项目 10: 批量提取 pdf 文档图片

趣味类

项目 12: Python 将图片转换为字符画

项目 13: 图片的灰度处理及二值化

项目 14: 使用 **Turtle** 库绘制时钟

项目 15: 整人代码

项目 17: 制作一个计算器

三、原 Python 小课课后练习及解析

基础语法课程

第 0 关 `print()`与变量

课后练习 1: 打印皮卡丘

课后练习 2: 听见无脸男的声音

第 1 关 数据类型与转换

课后练习 1: 程序员的一人饮酒醉

课后练习 2: 非酋的吐槽

第 2 关 条件判断和条件嵌套

课后练习 1: 寻找宝石

课后练习 2: 美国队长的工资

第 3 关 `input()`函数

课后练习 1: 哈利波特的宠物

课后练习 2: 罗恩的减肥计划

课后练习 3: 古灵阁金币兑换

第 4 关 列表和字典

课后练习 1: 君子爱'数'取之有道

课后练习 2: 找到那只狼

第 5 关 for 循环和 while 循环

课后练习 1：数数字

课后练习 2：轮流坐前排

第 6 关 布尔值和四种语句

课后练习 1：囚徒困境

课后练习 2：记录困境中的选择

课后练习 3：演员的作品

第 7 关 小游戏大学问

课后练习 1：再来一盘

课后练习 2：再来一种

第 8 关 编程思维：如何解决问题

课后练习 1：老师我帮你

课后练习 2：老师我又来了

第 9 关 函数

课后练习 1：年终奖有多少

课后练习 2：hellokikty 抽奖器

第 10 关 工作计量器

课后练习 1：剪刀石头布

课后练习 2：让代码更简洁

第 11 关 编程思维：如何 debug

课后练习 1：一起来抓虫

课后练习 2：贴心的除法计算器

第 12 关 类与对象 1

课后练习 1：一次性说完

课后练习 2：重要的事情说三遍

第 13 关 类与对象 2

课后练习 1：老师和父亲

课后练习 2：学习时间记录

第 14 关 流浪图书馆

课后练习 1：书的分类

课后练习 2：我想看 TA 的书

第 15 关 编码和文件读写课后

课后练习 1：数据转移中的变化

课后练习 2：古诗默写

第 16 关 模块

课后练习 1：时间管理器

课后练习 2：再出古诗默写题

第 17 关 邮件发送

课后练习 1：自制动态二维码

课后练习 2：绝对值的多种求法

第 18 关 编程思维：产品思维

课后练习 1：拯救选择困难症

课后练习 2：模拟广告牌

爬虫精进课程

第 0 关 认识爬虫

课后练习 1：文章下载

课后练习 2：图像下载

课后练习 3：音频下载

第 1 关 HTML 基础

课后练习：我的书苑我做主

第 2 关 BeautifulSoup

课后练习 1：博客爬虫

课后练习 2：书店寻宝

课后练习 3：博客文章

第 3 关 BeautifulSoup 实践

课后练习：豆瓣爬虫

第 5 关 带参数请求数据

课后练习 1：歌词爬取

课后练习 2：头号粉丝

第 6 关 CSV&Excel

课后练习：存储电影信息

第 7 关 爬取特朗普新闻

课后练习：做个测单词的小工具

第 8 关 cookies

课后练习 1：自制翻译器

课后练习 2：图灵机器人

课后练习 3：nlpirc 人工智能

第 9 关 selenium

课后练习 1：博客达人

课后练习 2：Python 之禅

第 10 关 定时与邮件

课后练习：周末吃什么

第 11 关 协程

课后练习：煲剧达人

第 13 关 Scrapy 框架

课后练习：当当图书馆爬虫

第 14 关 Scrapy 实操

课后练习：豆瓣图书短评

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)
```

```

14     end_poly()
15
16     handForm = get_poly()
17     register_shape(name, handForm)
18
19     def init():
20         global secHand, minHand, hurHand, show
21         mode("logo")
22         #初始化三个表针元素
23         mkHand("secHand", 135)
24         mkHand("minHand", 110)
25         mkHand("hurHand", 90)
26         secHand = Turtle()
27         secHand.shape("secHand")
28         minHand = Turtle()
29         minHand.shape("minHand")
30         hurHand = Turtle()
31         hurHand.shape("hurHand")
32         for hand in secHand, minHand, hurHand:
33             hand.shapesize(1, 1, 3)
34             hand.speed(0)
35         #实例化Turtle对象
36         show = Turtle()
37         show.hideturtle()
38         show.penup()
39
40     def setup(radius):
41         #设置表框
42         reset()
43         pensize(7)
44         for i in range(60):
45             skip(radius)
46             if i % 5 == 0:
47                 forward(20)
48                 skip(-radius-20)
49             else:
50                 dot(5)
51                 skip(-radius)
52                 right(6)
53
54     def Week(t):
55         week = ["星期一", "星期二", "星期三",
56                 "星期四", "星期五", "星期六", "星期日"]

```

```

56     return week[t.weekday()]
57
58 def Date(t):
59     year = t.year
60     month = t.month
61     day = t.day
62     return "%s %d %d" % (year, month, day)
63
64 def tick():
65     #表针动态显示
66     t = datetime.today()
67     second = t.second + t.microsecond*0.000001
68     minute = t.minute + second/60.0
69     hour = t.hour + minute/60.0
70     secHand.setheading(6*second) #每秒秒针转动6度
71     minHand.setheading(6*minute) #每分钟分针转动6度
72     hurHand.setheading(30*hour) #每小时时针转动30度
73     tracer(False) #不显示绘制过程，直接显示结果
74     show.forward(65)
75     show.write(Week(t), align="center",
76                 font=("Courier", 14, "bold"))
77     show.back(130)
78     show.write(Date(t), align="center",
79                 font=("Courier", 14, "bold"))
80     show.back(50)
81     show.write("时钟", align="center",
82                 font=("Courier", 14, "bold"))
83     show.home()
84     tracer(True)
85     ontimer(tick, 1000) #刷新时间为1s，即为1s后继续调用tick
86
87 def main():
88     tracer(False)
89     init()
90     setup(160)
91     tracer(True)
92     tick()
93     mainloop()
94 main()

```

```

1 import os
2 import time
3
4 print('*****')
5 print('欢迎来到喊猪游戏')
6 print('*****\n')
7 content = input('说我是猪,不然就关你的机, 赶紧的! , "说不说"')
8
9 if content == "我是猪":
10     time.sleep(1)
11     print('明白就好啦, 真乖~\n')
12
13 else:
14     print("既然如此...")
15     time.sleep(3)
16     print("10秒后制裁你的电脑, 啊哈哈~~~~~")
17     time.sleep(1)
18     os.system('shutdown -r -t 10')

```

项目17：制作一个计算器

```

1 from tkinter import Tk, Button, Text, END
2 import math
3 # 定义一个栈
4 class Calculator():
5     def __init__(self):
6         self.items = []
7     def isEmpty(self):
8         return self.items == []
9     def push(self, item):
10        self.items.append(item)
11    def pop(self):
12        return self.items.pop()
13    def peek(self):
14        return self.items[-1]
15    def size(self):
16        return len(self.items)
17

```

```

18 # 将计算式转化为后序表达式
19 def postfixExpr(formulaList):
20     # 定义等级
21     prior = {}
22     prior["*"] = 3
23     prior["/] = 3
24     prior["+"] = 2
25     prior["-"] = 2
26     prior["("] = 1
27
28     opstack = Calculator()
29     postfixList = []
30     for token in formulaList:
31         if token == '(':
32             opstack.push(token)
33         elif token == ")":
34             topToken = opstack.pop()
35             while topToken != "(":
36                 postfixList.append(topToken)
37                 topToken = opstack.pop()
38             elif token in "*/+":
39                 while (not opstack.isEmpty()) and (prior[opstack.peek()]>= prior[token]) :
40                     postfixList.append(opstack.pop())
41                 opstack.push(token)
42             else :
43                 postfixList.append(token)
44
45         while not opstack.isEmpty():
46             postfixList.append(opstack.pop())
47
48     return " ".join(postfixList)      # 返回后序表达式
49
50 # 后序表达式计算
51 def postfixEval(postfixExpr):
52     operandStack = Calculator()
53     tokenLIst = postfixExpr.split()
54     for token in tokenLIst:
55         if token in "*/+":
56             number1 = operandStack.pop()
57             number2 = operandStack.pop()
58             result = doMath(token, number1, number2)
59             operandStack.push(result)

```

```

60         else:
61             operandStack.push(float(token))
62     return operandStack.pop()  # 返回计算结果
63
64 # 分步计算函数
65 def doMath(op, num1, num2):
66     if op == "*":
67         return num2 * num1
68     elif op == "/":
69         return num2 / num1
70     elif op == "+":
71         return num2 + num1
72     elif op == "-":
73         return num2 - num1
74
75 # 定义Value类，用于把值传到GUI界面的Text框中
76 class Value():
77
78     def __init__(self, value):
79         self.value = value
80     # 将输入按钮的值传到文本框中
81     def insert_value(self):
82         self.result_text=Application.result_text
83         self.result_text.insert(END, self.value)
84
85 # 定义Application类，用于画GUI和计算结果。
86 class Application(object):
87     window = Tk()    # 创建一个GUI界面
88     # 创建一个文本框
89     result_text = Text(window, background='azure')
90     # 更多背景色: http://www.science.smith.edu/dftwiki/index.php/Color_Charts_for_TKinter
91     result_text.place(x=12.5, y=8, width=325, height=60)
92
93     def __init__(self):
94         self.window.title(u'我的计算器') # 定义界面名称
95
96         # 设置窗口大小和位置
97         self.window.geometry('350x310+500+300')
98         self.window.minsize(350, 315)
99         self.window.maxsize(350, 315)
100
101        # 第1行按钮

```

```

102         self.submit_btn0 =
103             Button(self.window, text=u'(', command=Value('(').insert_value)
104             self.submit_btn0.place(x=12.5, y=76, width=60, height=40)
105             self.submit_btn1 = Button(self.window, text=u'C', command = self.clean)
106             self.submit_btn1.place(x=78.75, y=76, width=60, height=40)
107             self.submit_btn2 = Button(self.window, text=u'÷',
108                 command=Value('÷').insert_value)
109             self.submit_btn2.place(x=145, y=76, width=60, height=40)    #
110                 12.5+(60+6.25)*3
110             self.submit_btn4 = Button(self.window, text=u'←', command=self.backspace)
111             self.submit_btn4.place(x=277.5, y=76, width=60, height=40)
112             # 第2行按钮
113             self.submit_btn5 =
114                 Button(self.window, text=u') ', command=Value(')').insert_value)
114             self.submit_btn5.place(x=12.5, y=76+8+40, width=60, height=40)
115             self.submit_btn6 = Button(self.window, text=u'7', command =
116                 Value(7).insert_value)
116             self.submit_btn6.place(x=78.75, y=124, width=60, height=40)
117             self.submit_btn7 = Button(self.window, text=u'8',
118                 command=Value(8).insert_value)
118             self.submit_btn7.place(x=145, y=124, width=60, height=40)
119             self.submit_btn8 = Button(self.window, text=u'9',
120                 command=Value(9).insert_value)
120             self.submit_btn8.place(x=211.25, y=124, width=60, height=40)
121             self.submit_btn9 = Button(self.window, text=u'−',
122                 command=Value('−').insert_value)
122             self.submit_btn9.place(x=277.5, y=124, width=60, height=40)
123             # 第3行按钮
124             self.submit_btn10 =
125                 Button(self.window, text=u'π', command=Value(math.pi).insert_value)
125             self.submit_btn10.place(x=12.5, y=172, width=60, height=40)
126             self.submit_btn11 = Button(self.window, text=u'4', command =
127                 Value(4).insert_value)
127             self.submit_btn11.place(x=78.75, y=172, width=60, height=40)
128             self.submit_btn12 = Button(self.window, text=u'5',
129                 command=Value(5).insert_value)
129             self.submit_btn12.place(x=145, y=172, width=60, height=40)
130             self.submit_btn13 = Button(self.window, text=u'6',
130                 command=Value(6).insert_value)

```

```

131         self.submit_btn13.place(x=211.25, y=172, width=60, height=40)
132         self.submit_btn14 = Button(self.window, text=u'+',
133             command=Value('+').insert_value)
134         self.submit_btn14.place(x=277.5, y=172, width=60, height=40)
135         # 第4行按钮
136         self.submit_btn15 = Button(self.window, text=u'1/x', command=
137             Value('1/').insert_value)
138         self.submit_btn15.place(x=12.5, y=220, width=60, height=40)
139         self.submit_btn16 = Button(self.window, text=u'1',
140             command=Value(1).insert_value)
141         self.submit_btn16.place(x=78.75, y=220, width=60, height=40)
142         self.submit_btn17 = Button(self.window, text=u'2',
143             command=Value(2).insert_value)
144         self.submit_btn17.place(x=145, y=220, width=60, height=40)
145         self.submit_btn18 = Button(self.window, text=u'3',
146             command=Value(3).insert_value)
147         self.submit_btn18.place(x=211.25, y=220, width=60, height=40)
148         self.submit_btn19 = Button(self.window, text=u'=',
149             command=self.equal)
150         self.submit_btn19.place(x=277.5, y=220, width=60, height=88)
151         # 第5行按钮
152         self.submit_btn20 =
153             Button(self.window, text=u'x2', command=Value('^2').insert_value)
154         self.submit_btn20.place(x=12.5, y=268, width=60, height=40)
155         self.submit_btn21 = Button(self.window, text=u'%',
156             command=Value('%').insert_value)
157         self.submit_btn21.place(x=78.75, y=268, width=60, height=40)
158         self.submit_btn22 = Button(self.window, text=u'0',
159             command=Value(0).insert_value)
160         self.submit_btn22.place(x=145, y=268, width=60, height=40)
161         self.submit_btn23 = Button(self.window, text=u'.',
162             command=Value('.').insert_value)
163         self.submit_btn23.place(x=211.25, y=268, width=60, height=40)
164
165     def clean(self):      # 清空键
166         self.result_text.delete(0.0, END)
167         return
168     def backspace(self):   # 退格键
169         content = self.result_text.get(0.0, END).strip().split("\n")
170         self.result_text.delete(0.0, END)
171         for i in content[:-1]:
172             self.result_text.insert(END, i)

```

```

163         self.result_text.insert(END, '\n')
164
165         self.result_text.insert(END, content[-1][-1])
166         self.result_text.see(END) # 定位光标到最后
167
168     def equal(self):          # 等于键
169         formula = self.result_text.get(0.0, END).strip().split("\n")[-1] # 获取界面的计
算式子
170
171         print(formula, end=" ")
172         formula = formula.replace("^2", "") # 将平方保留一个符号^方便后面计算
173         try:
174             # 将数字和运算符号分离
175             formulaList1 = []
176             i = 0
177             while i < len(formula):
178                 if formula[i] in "0123456789.":
179                     cc = ''
180                     while i < len(formula):
181                         if formula[i] in "0123456789.":
182                             cc += formula[i]
183                         i += 1
184                     else:
185                         break
186                     i -= 1
187                     formulaList1.append(cc)
188
189             i += 1
190             # 将平方和百分号做处理
191             formulaList2 = []
192             for i in formulaList1:
193                 formulaList2.append(i)
194                 if i == "^":
195                     index = formulaList1.index("^")
196                     formulaList1[index] = "-" # 将^符号替换掉，方便查找第2个^
197                     number = formulaList1[index - 1]
198                     formulaList2.pop()
199                     formulaList2.append('*')
200                     formulaList2.append(number) # 转化为两个数相乘
201                 elif i == "%":
202                     index = formulaList1.index("%")
203                     formulaList1[index] = "-" # 将%符号替换掉，方便查找第2个%

```

```

204         formulaList2.pop()
205
206
207     global result      # 方便在except中调用
208     b = postfixExpr(formulaList2)          # 获取后序表达式
209     result = postfixEval(b)    # 获取计算结果值
210     print(result)
211
212     self.result_text.insert(END, '=')
213
214     self.result_text.insert(END, result)
215     self.result_text.insert(END, '\n')  # 另起一行
216     self.result_text.see(END)      # 定位光标到最后
217
218     except ValueError:
219
220         self.result_text.insert(END, result)
221         self.result_text.insert(END, '\n')
222         self.result_text.see(END)      # 定位光标到最后
223
224         print(result)
225
226     except:
227
228         self.result_text.insert(END, 'Error!')
229         self.result_text.insert(END, '\n')
230         self.result_text.see(END)      # 定位光标到最后
231
232     print("Error!")
233
234     def run(self):  # 运行tkinter, 展示界面
235
236         self.window.mainloop()
237
238
239 if __name__=="__main__":
240
241     app = Application()
242
243     app.run()

```

三、原Python小课课后练习及解析

基础语法课程

第0关 print()与变量

课后练习1：打印皮卡丘

- 题目解析:

- 直接复制题目要求打印的“皮卡丘”(在【书写代码】步骤里直接提供)，用print()打印。
- 皮卡丘涉及到换行，所以这里要使用三引号来换行。
- 注意：标点符号都要用英文。

- 参考代码:

```
1 print('''
2     ^    /|
3     /\_   <_/
4     / |   / /
5     | z _ , < /  /`\
6     |     \ /  \
7     Y     ^ / /
8     仁 、 •  c o < / 
9     () ^   + \ <
10    >- ._  イ | // 
11    / ^   / /<+ \ \
12    \_ /  (\_ /  | // 
13    7     + /
14    >-r____`--_
15    ''')
```

课后练习2：听见无脸男的声音

- 题目解析:

方法一：使用单引号+\n打印

- 使用 \' 表示打印一个单引号：使用单引号进行打印的时候，文本中有单引号的地方，使用转义符号把单引号直接打印出来。
- 需要换行，使用 \n。
- 打印函数print()。

- 参考代码:

```
1 print('千寻你好，人们叫我'无脸男'\n这个世界的人都选择无视我\n只有你看到了我并和我打招呼\n我感到很孤单，很孤单\n你愿意和我成为朋友吗？')
```

方法二：使用三引号打印

- 直接使用三引号，打印出内容里的单引号并且实现换行。

- 参考代码:

```
1 print(''千寻你好，人们叫我'无脸男'
2 这个世界的人都选择无视我
3 只有你看到了我并和我打招呼
4 我感到很孤单，很孤单
5 你愿意和我成为朋友吗？'')
```

方法三：使用双引号+\n

- 使用双引号的时候，不会和单引号产生歧义；
- 需要换行的地方用 \n。

```
1 print("千寻你好，人们叫我'无脸男'\n这个世界的人都选择无视我\n只有你看到了我并和我打招呼\n我感到很孤
单，很孤单\n你愿意和我成为朋友吗？")
```

第1关 数据类型与转换

课后练习1：程序员的一人饮酒醉

- 题目解析：
 - 非字符串要与字符串进行拼接的时候，需要使用str()先转换为字符串；
 - 使用 + 将字符串进行拼接；
 - 使用print()函数打印拼接结果。
- 参考代码：

```
1 number1 = 1
2 number2 = 2
3 unit1 = '人'
4 unit2 = '眼'
5 line1 = '我编程累'
6 line2 = '是bug相随'
7 sentence1 = '碎掉的节操满地堆'
8 sentence2 = '我只求今日能早归'
9
10 print(str(number1) + unit1 + line1 + sentence1)
11 print(str(number2) + unit2 + line2 + sentence2)
```

课后练习2：非酋的吐槽

- 题目解析:

- 小数形式的字符串'7.8'，不是整数形式的字符串，不能直接转化为整数，需要先使用float()转化为浮点数，再使用int()转化为整数，转化成整数之后，因为要用于字符串拼接，所以还要使用str()转化为字符串；
- 使用 + 将字符串进行拼接；
- 使用print()函数打印拼接结果。

- 参考代码:

```
1 slogan = '脸黑怪我咯'
2 number = '7.8'
3 unit = '张'
4 sentence = '蓝票一个SSR都没有'
5 word = slogan + str(int(float(number))) + sentence
6 print(word)
```

第2关 条件判断和条件嵌套

课后练习1：寻找宝石

- 题目解析:

- 我们需要设定一个变量来表示宝石的数量，可以把这个变量命名为stonenumber，并赋值为0。
- 按照题目需求我们需要选择多项选择的结构，条件是拥有宝石的数量，三种情况分别是大于三颗宝石，一到三颗宝石和零颗宝石。

- 参考代码:

```
1 stonenumber = 0
2 if stonenumber >= 4:
3     print('获得了打败灭霸的力量，反杀稳了')
4 elif 1<=stonenumber <= 3:
5     print('可以全员出动，殊死一搏')
6 else:
7     print('没办法了，只能尝试呼叫惊奇队长')
```

课后练习2：美国队长的工资

- 题目解析:

- 根据题干的要求，我们需要使用if...elif...else结构来处理小于500，500-1000和大于1000这三种情况。

- 在小于500的情况下，有小于100和100到500之间的两种情况，可以在这个if语句里面嵌套if..else语句来实现判断。
- 在大于1000的情况下条件也可以被细分为1000到20000和大于20000这两种情况，在这个条件里也去嵌套if...else结构实现判断。

- 参考代码：

```

1 money=80
2 if money<=500:
3     print('欢迎进入史塔克穷人榜前三名！')
4     if money<=100:
5         print("恭喜您荣获'美元队长'称号！")
6     else:
7         print('请您找弗瑞局长加薪。')
8 elif 500<money<=1000:
9     print('恭喜您至少可以温饱了。')
10 else:
11     print('经济危机都难不倒您！')
12     if money<=20000:
13         print('您快比托尼·史塔克有钱了！')
14     else:
15         print('您是不是来自于瓦坎达国？')
16
17 print('程序结束')

```

第3关 input()函数

课后练习1：哈利波特的宠物

- 题目解析：

- 运行代码之后，在终端输入一个名字，传给input()函数，并赋值给name变量；
- 然后和字符串【'哈利·波特的猫头鹰叫做'】拼接，并使用print()函数进行打印。

- 参考代码：

```

1 name = input('为哈利·波特的猫头鹰起个名字：')
2 print('哈利·波特的猫头鹰叫做' + name)

```

课后练习2：罗恩的减肥计划

- 题目解析：

- 运行代码之后，在终端输入一个数字，传给input()函数，并使用int()函数转化为整数，再赋值给number变量。
- 使用双向判断语句if...else...根据游戏规则对number的值进行判断。

- 参考代码：

```
1 number = int(input('请输入罗恩吃的巧克力数量：'))
2 #int将其转换为整数，之后好进行大小的比较
3 if number > 10:
4     print('罗恩要给哈利100块')
5 else:
6     print('哈利就给罗恩100块')
```

课后练习3：古灵阁金币兑换

- 题目解析：

- 根据题目的要求，我们判断可以通过 if条件嵌套的形式来解决问题：
 - 最外层的 if条件根据用户的需求判断是否需要帮助，只有两种情况可以使用双向判断：
 - 需要：嵌套第二层 if条件根据用户的需求判断需要什么类型的帮助，有三种情况使用多向判断：
 - 1 存取款->去存取款窗口；
 - 2 货币兑换：获取用户需要兑换的金加隆数量，并通过计算，打印出来需要支付的人民币；
 - 3 咨询->去存取款窗口。
 - 不需要->再见
 - 需要获取用户的意見的时候，都使用input()进行交互，获取用户的需求。
 - 为了使得用户输错的时候提示输入错误，需要再加一级else进行判断。

- 参考代码：

```
1 choice = input('您好，欢迎古灵阁，请问需要帮助吗？需要or不需要？')
2 if choice == '需要':
3     number = input('请问您需要什么帮助呢？1 存取款；2 货币兑换；3 咨询')
4     if number == '2':
5         print('金加隆和人民币的兑换率为1:51.3，即一金加隆=51.3人民币')
6         print('请问您需要兑换多少金加隆呢？')
7         money = input('请输入你需要兑换的金加隆')
8         print('好的，我知道了，您需要兑换' + money + '金加隆。')
9         print('那么，您需要付给我' + str(int(float(money) * 51.3)) + '人民币。')
10    elif number == '1':
```

```

11     print('请到存取款窗口办理')
12 elif number == '3':
13     print('请到咨询窗口咨询')
14 # 加上else语句当输入不是1、2、3时，提示输入错误
15 else:
16     print('输入错误，没有你需要的服务')
17
18 elif choice == '不需要':
19     print('好的，再见')
20 # 加上else语句当输入不是1、2、3时，提示输入错误
21 else:
22     print('输入错误')

```

第4关 列表和字典

课后练习1：君子爱‘数’取之有道

- 题目解析：
 - 第一题：把列表list1中的'love'取出来，并打印出来。
 - 观察list1 可以发现，list1 是一个嵌套字典的列表，外层是列表，共有3个元素，这3个元素都是字典，而且每个字典都只有一个键值对；
 - 要取出“love”，观察list1，可以找到它在列表最后一个元素中的字典里，作为键“爱”的值。可以使用偏移量-1或2，即list[-1] 或list[2]，取到列表，然后使用键“爱”取得。
 - 第二题：把字典dict1中的'love'取出来，并打印出来。
 - 观察dict1 可以发现，dict1 是一个嵌套列表的字典，外层是字典，共有3个键值对，这3个键值对中的值都是列表；
 - 要取出“love”，观察list1，可以找到它在dict1 中键为 3 对应的值（列表）中，是列表的第一个元素。首先要使用键 3 取出“love”所在的列表，然后再使用偏移量0 或-3 取出。
 - 第三题：将tuple1中的A和list2中的D打印出来。
 - 观察tuple1 可以发现，tuple1 是一个元组，“A”就是元组中的第一个元素，可以使用偏移量0 或-2 取得；
 - 观察list2 可以发现，是一个嵌套元组的列表，列表中共有3个元素，这3个元素都是元组。练习要求提取的字母“D”所在的元组在列表的第二个元素中，可以使用偏移量1 或-2 取得；取到元组之后，可以使用偏移量1 或-1 取到元素“D”。（1 或-2 和1 或-1 两两组合共有4种取法）
- 参考代码：

```

1 # 第一题代码
2 list1 = [{"嫉妒": "envy"}, {"恨": "hatred"}, {"爱": "love"}]
3 print(list1[2]['爱'])
4 print(list1[-1]['爱'])

5

6 # 第二题代码
7 dict1 = {1: ['cake', 'scone', 'puff'], 2: ['London', 'Bristol', 'Bath'], 3:
8     ['love', 'hatred', 'envy']}
9 print(dict1[3][0])
10 print(dict1[3][-3])

11 # 第三题代码
12 tuple1 = ('A', 'B')
13 list2 = [('A', 'B'), ('C', 'D'), ('E', 'F')]
14 print(tuple1[0])
15 print(list2[1][1])

```

课后练习2：找到那只狼

- 题目解析：
 - 观察townee 可以发现，它的最外层是一个列表，列表中有6个元素（看逗号，每个元素使用功能逗号分隔），分别是：
 - i. 第一个是字典 {"海底王国":['小美人鱼"海之王"小美人鱼的祖母"五位姐姐'],'上层世界':["王子", '邻国公主"]};
 - ii. 第二个是字符串 '丑小鸭'；
 - iii. 第三个是字符串 '坚定的锡兵'；
 - iv. 第四个是字符串 '睡美人'；
 - v. 第五个是字符串 '青蛙王子'；
 - vi. 第六个是列表 [{"主角": "小红帽", "配角1": "外婆", "配角2": "猎人"}, {"反面角色": "狼"}]， “狼”就藏该元素中。
 - 分析第六个元素，该元素也是一个列表，拥有2个元素，都是字典，分别是{"主角": "小红帽", "配角1": "外婆", "配角2": "猎人"}和{"反面角色": "狼"}。“狼”就藏在第二个元素中，以值的形式存在。
 - 综上得知，“狼”藏在townee 列表最后一个元素的列表中的字典里，作为键“反面角色”的值，即print(townee[5][1]['反面角色']), 也可以使用负向取值。
- 参考代码：

```

1 townee = [
2     {'海底王国': ['小美人鱼', '海之王', '小美人鱼的祖母', '五位姐姐'], '上层世界': ['王子', '邻国公主']},
3     '丑小鸭', '坚定的锡兵', '睡美人', '青蛙王子',
4     [{"主角": "小红帽", "配角1": "外婆", "配角2": "猎人"}, {"反面角色": "狼"}]

```

```
5      ]
6  print(townee[5][1]['反面角色'])
```

第5关 for循环和while循环

课后练习1：数数字

- 题目解析：
 - 使用for循环打印数字时可使用range()函数，循环内部可以加一个条件判断语句，当结果不等于4的时候打印数字。
 - 使用while循环打印数字时需要提前设置一个计数变量，每循环一次这个变量就加一，表示循环一次。因此循环条件可以根据这个计数变量的大小来设置。
- 参考代码：

```
1  # for循环的方法
2  for i in range(1,8):
3      if i !=4:
4          print(i)
5
6  # while循环的方法
7  a=0
8  while a<7:
9      a=a+1
10     if a !=4:
11         print(a)
```

课后练习2：轮流坐前排

- 题目解析：
 - 列表有append()和pop()两个方法，表示给列表增添或者删除元素。
 - append()方法接收一个参数，并且会把这个参数加到列表的末尾处。
 - pop()方法接收一个整数参数，参数表示删除列表中的第几个元素。
 - 可以使用pop()方法删除列表中的第一个元素，并把它赋值给一个变量以记录，然后再使用append()方法把这个变量加到列表的末尾位置，这样就实现了“换座位”的需求。
- 参考代码：

```
1 students = ['小明', '小红', '小刚']
2 for i in range(3):
3     student1 = students.pop(0) # 运用pop()函数，同时完成提取和删除。
4     students.append(student1) # 将移除的student1安排到最后一个座位。
5 print(students)
```

第6关 布尔值和四种语句

课后练习1：囚徒困境

- 题目解析：
 - 可以使用无限循环让双方选择，当满足一定条件时候跳出循环，可以使用input()语句接收双方的选择。
 - 针对双方都不认罪、一方认罪另一方不认罪、双方都认罪这三种情况，可以使用if...elif...else语句来进行条件判断。
- 参考代码：

```
1 while True:
2     a = input('A, 你认罪吗？请回答认罪或者不认：')
3     b = input('B, 你认罪吗？请回答认罪或者不认：')
4     if a == '认罪' and b == '认罪':
5         print('两人都得判10年，唉')
6     elif a == '不认' and b == '认罪':
7         print('A判20年, B判1年, 唉')
8     elif a == '认罪' and b == '不认':
9         print('A判1年, B判20年')
10    else:
11        print('都判3年, 太棒了')
12    break # 当满足开头提到的条件时，跳出循环。
```

课后练习2：记录困境中的选择

- 题目解析：
 - 我们可以使用一个列表来记录每一组实验者的选择，这个列表中的每一个元素就是一组实验者的选择。
 - 当使用input()函数获取到一组实验者的选择后，我们就可以用两个实验者的选择添加到列表中。使用列表的append()方法可以实现给列表中添加新元素。
 - 当满足条件跳出循环后，记录的最后一组就是选择了最优解的一组，也就是列表中的最后一个元素。

- 最后可以使用遍历列表的方式来打印实验者的组数和他们的选择。

- 参考代码:

```

1 n = 0
2 list_answer = []
3 while True:
4     n += 1
5     a = input('A, 你认罪吗? 请回答认罪或者不认: ')
6     b = input('B, 你认罪吗? 请回答认罪或者不认: ')
7     list_answer.append([a,b])
8     if a == '认罪' and b == '认罪':
9         print('两人都得判10年, 唉')
10    elif a == '不认' and b == '认罪':
11        print('A判20年, B判1年, 哎')
12    elif a == '认罪' and b == '不认':
13        print('A判1年, B判20年')
14    else:
15        print('都判3年, 太棒了')
16    break
17 print('第' + str(n) + '对实验者选了最优解。')
18 for i in range(n):
19     print('第' + str(i+1) + '对实验者的选择是: ' + str(list_answer[i]))

```

课后练习3：演员的作品

- 题目解析:

- 我们可以用一个字典来存放电影信息，字典的键是电影名，字典的值是对应电影的演员所组成的列表。
- 使用input()函数可以获取我们要查的演员名。
- 使用for循环遍历字典可以获取到字典的键，使用键值对对应的方式可取到字典的值，即是演员组成的列表，遍历演员列表就可找到具体的演员。

- 参考代码:

```

1 movies = {
2     '妖猫传': ['黄轩', '染谷将太'],
3     '无问西东': ['章子怡', '王力宏', '祖峰'],
4     '超时空同居': ['雷佳音', '佟丽娅'],
5 }
6
7 actor = input('你想查询哪个演员? ')
8 for movie in movies: # 用 for 遍历字典

```

```
9     actors = movies[movie] # 读取各个字典的主演表
10    if actor in actors:
11        print(actor + '出演了电影' + movie)
```

第7关 小游戏大学问

课后练习1：再来一盘

- 题目解析：
 - 方案一：开启一个无限循环，设定一个跳出循环的条件。当得到肯定回复时，继续进行循环过程；当得到否定回复时，运行break，停止循环，退出游戏。
 - 方案二：设置一个变量并赋值为布尔值True，然后进入while循环，循环的条件为变量的布尔值。接着通过input()函数接收输入的内容，接着使用if...else语句来根据输入的内容改变变量的布尔值决定是否跳出循环。
- 参考代码：

```
1 #方法一
2 import time
3 import random
4 while True:
5     player_victory = 0
6     enemy_victory = 0
7     for i in range(1,4):
8         time.sleep(1.5)
9         print(' \n——现在是第 %s 局——' % i)
10        player_life = random.randint(100,150)
11        player_attack = random.randint(30,50)
12        enemy_life = random.randint(100,150)
13        enemy_attack = random.randint(30,50)
14        print('【玩家】\n血量: %s\n攻击: %s' % (player_life,player_attack))
15        print('-----')
16        time.sleep(1)
17        print('【敌人】\n血量: %s\n攻击: %s' % (enemy_life,enemy_attack))
18        print('-----')
19        time.sleep(1)
20        while player_life > 0 and enemy_life > 0:
21            player_life = player_life - enemy_attack
```

```
22     enemy_life = enemy_life - player_attack
23     print('你发起了攻击, 【玩家】剩余血量%s' % player_life)
24     print('敌人向你发起了攻击, 【敌人】的血量剩余%s' % enemy_life)
25     print('-----')
26     time.sleep(1.2)
27 if player_life > 0 and enemy_life <= 0:
28     player_victory += 1
29     print('敌人死翘翘了, 你赢了! ')
--
```

```

30     elif player_life <= 0 and enemy_life > 0:
31         enemy_victory += 1
32         print('悲催，敌人把你干掉了！')
33     else:
34         print('哎呀，你和敌人同归于尽了！')
35     if player_victory > enemy_victory:
36         time.sleep(1)
37         print('\n【最终结果：你赢了！】')
38     elif enemy_victory > player_victory:
39         print('\n【最终结果：你输了！】')
40     else:
41         print('\n【最终结果：平局！】')
42     a1 = input('要继续游戏吗，请输入n退出，输入其他继续：')
43     if a1 == 'n':
44         break

```

```

1 #方法二
2 import time
3 import random
4
5 player_victory = 0
6 enemy_victory = 0
7 again = True
8 while again:
9     for i in range(1,4):
10         time.sleep(1.5)
11         print(' \n——现在是第 %s 局——' % i)
12         player_life = random.randint(100,150)
13         player_attack = random.randint(30,50)
14         enemy_life = random.randint(100,150)
15         enemy_attack = random.randint(30,50)
16
17         print('【玩家】\n血量：%s\n攻击：%s' % (player_life,player_attack))
18         print('-----')
19         time.sleep(1)
20         print('【敌人】\n血量：%s\n攻击：%s' % (enemy_life,enemy_attack))
21         print('-----')
22         time.sleep(1)
23
24         while player_life > 0 and enemy_life > 0:
25             player_life = player_life - enemy_attack
26             enemy_life = enemy_life - player_attack

```

```

27         print('你发起了攻击, 【玩家】剩余血量%s' % player_life)
28         print('敌人向你发起了攻击, 【敌人】的血量剩余%s' % enemy_life)
29         print('-----')
30         time.sleep(1.2)
31
32     if player_life > 0 and enemy_life <= 0:
33         player_victory += 1
34         print('敌人死翘翘了, 你赢了! ')
35     elif player_life <= 0 and enemy_life > 0:
36         enemy_victory += 1
37         print('悲催, 敌人把你干掉了! ')
38     else:
39         print('哎呀, 你和敌人同归于尽了! ')
40
41     if player_victory > enemy_victory :
42         time.sleep(1)
43         print('\n【最终结果: 你赢了! 】')
44     elif enemy_victory > player_victory:
45         print('\n【最终结果: 你输了! 】')
46     else:
47         print('\n【最终结果: 平局! 】')
48
49     a1 = input('要继续游戏吗, 请输入n退出, 输入其他继续: ') # 在 while True 循环中设置跳出条件。
50     if a1 == 'n':
51         again = False
52     else:
53         again = True

```

课后练习2：再来一种

- 题目解析：
 - 我们可以使用str.format()函数来取代之前学过的字符串格式化方法。
 - format()函数中的参数需要和之前字符串中的占位符{}匹配，当{}里没有数字，则按照默认的位置顺序匹配，如果{}中有数字，则按照数字去进行匹配。
- 参考代码：

```

1 import time
2 import random
3
4 player_victory = 0

```

```

5 enemy_victory = 0
6
7 while True:
8     for i in range(1, 4):
9         time.sleep(1.5)
10        print(' \n——现在是第 {} 局——'.format(i))
11        player_life = random.randint(100, 150)
12        player_attack = random.randint(30, 50)
13        enemy_life = random.randint(100, 150)
14        enemy_attack = random.randint(30, 50)
15
16        print(' 【玩家】 \n血量: {} \n攻击: {}'.format(player_life, player_attack))
17        print('-----')
18        time.sleep(1)
19        print(' 【敌人】 \n血量: {} \n攻击: {}'.format(enemy_life, enemy_attack))
20        print('-----')
21        time.sleep(1)
22
23        while player_life > 0 and enemy_life > 0:
24            player_life = player_life - enemy_attack
25            enemy_life = enemy_life - player_attack
26            print('敌人发起了攻击, 【玩家】剩余血量{}'.format(player_life))
27            print('你发起了攻击, 【敌人】的血量剩余{}'.format(enemy_life))
28            print('-----')
29            time.sleep(1.2)
30
31        if player_life > 0 and enemy_life <= 0:
32            player_victory += 1
33            print('敌人死翘翘了, 你赢了! ')
34        elif player_life <= 0 and enemy_life > 0:
35            enemy_victory += 1
36            print('悲催, 敌人把你干掉了! ')
37        else:
38            print('哎呀, 你和敌人同归于尽了! ')
39
40    if player_victory > enemy_victory :
41        time.sleep(1)
42        print('\n【最终结果: 你赢了! 】 ')
43    elif enemy_victory > player_victory:
44        print('\n【最终结果: 你输了! 】 ')
45    else:
46        print('\n【最终结果: 平局! 】 ')

```

```
47
48     a1 = input('要继续游戏吗, 请输入n退出, 输入其他继续: ')
49     if a1 == 'n':
50         break
```

第8关 编程思维：如何解决问题

课后练习1：老师我帮你

- 题目解析：
 - 合并列表，可用append()函数或者extend()函数将两个列表的合并。
 - 列表排序，可用sort()函数或者sorted()函数将列表的内容排序。
- 参考代码：

```
1 # 合并列表
2 # 使用append()函数
3 list1 = [91, 95, 97, 99]
4 list2 = [92, 93, 96, 98]
5 for i in list2:
6     list1.append(i)
7 print(list1)
8
9 # 使用extend()函数
10 list1 = [91, 95, 97, 99]
11 list2 = [92, 93, 96, 98]
12 list3 = list1
13 list3.extend(list2)
14 print(list3)
```

课后练习2：老师我又来了

- 题目解析1：
 - 使用for循环叠加成绩，然后除以总人数或直接使用numpy模块中的mean()函数，求出平均值。
 - 由于学生的成绩已经被集中到一个列表里，所以可以用for循环遍历这个列表来取出成绩。取出的成绩与平均值进行判断，将小于平均值的成绩存放到新的列表里。
- 参考代码：

```

1 # 方法一：通过for循环求出平均值并判断
2 scores1 = [91, 95, 97, 99, 92, 93, 96, 98]
3 sum = 0
4 scores2 = []
5
6 for score in scores1:
7     sum = sum + score # 叠加成绩
8     average = sum/len(scores1) # 用len()函数计算出总人数，即列表的元素数量
9 print('平均成绩是: {}'.format(average))
10
11 for score in scores1:
12     if score < average: # 判断是否小于平均值
13         scores2.append(score)
14     continue
15 print('低于平均成绩的有: {}'.format(scores2))

```

```

1 # 方法二，通过mean()函数，直接求出平均值
2 import numpy as np
3
4 scores1 = [91, 95, 97, 99, 92, 93, 96, 98]
5 scores2 = []
6
7 average = np.mean(scores1) # 一行解决。
8 print('平均成绩是: {}'.format(average))
9
10 for score in scores1:
11     if score < average:
12         scores2.append(score)
13     continue
14 print('低于平均成绩的有: {}'.format(scores2))

```

第9关 函数

课后练习1：年终奖有多少

- 题目解析：
 - 定义两个函数，一个函数的功能是计算奖金，另一个函数的功能是打印员工的姓名，入职时长和所获奖金的

相关信息。

- 使用input()函数，输入员工姓名与入职时长，调用计算奖金的函数，计算出对应的奖金金额。
 - 将奖金金额传递到打印功能的函数，将相关信息用print()函数打印出来。
- 参考代码：

```
1 # 计算奖金金额
2 def bonus(month):
3     if month < 6:
4         money = 500
5         return money
6     elif 6 <= month <= 12:
7         money = 120 * month
8         return money
9     else:
10        money = 180 * month
11        return money
12
13 # 打印相关信息
14 def info(name, month):
15     gain = bonus(month) # 调用bonus()函数，并将奖金金额存放到gain变量
16     print('%s来了%s个月，获得奖金%s元' % (name, month, gain))
17
18 info('大聪', 14)
```

课后练习2：hellokikty抽奖器

- 题目解析：
 - 通过随机抽取luckylist列表中的元素，来作为中奖的人员，所以luckylist列表中的元素，为可变变量，可以作为函数的参数名。
 - 将抽奖代码封装成函数，将luckylist列表中的元素作为函数的参数名。
 - 代码封装完函数后，调用函数，给参数名进行赋值。

- 参考代码：

```
1 import random
2 import time
3
4 # 将抽奖程序封装成函数
5 def choujiang(q,w,e): # 定义一个抽奖函数，带有3个参数，也就是3位候选人
6     luckylist = [q,w,e] # 定义一个中奖名单的列表
7     a = random.choice(luckylist) # 在中奖名单里面随机选择
```

```

8     print('开奖倒计时', 3)
9     time.sleep(1)
10    print('开奖倒计时', 2)
11    time.sleep(1)
12    print('开奖倒计时', 1)
13    time.sleep(1)
14    image = '''
15    /\_)o<
16    |       \ \
17    | o . o|
18    \____/
19    ...
20
21    print(image)
22    print('恭喜' + a + '中奖! ')
23 choujiang('虚竹', '萧峰', '段誉')  # 调用函数，给参数名q,w,e赋值

```

第10关 工作计量器

课后练习1：剪刀石头布

- 题目解析：
 - 使用random.choice()函数，实现电脑随机亮券；使用input()函数，实现用户亮券。
 - 使用while循环，判断用户输入的是否是石头、剪刀或布，若不是，则重新输入。
 - 通过穷举赢的情况，使用if...elif...else语句判断出哪方胜出。
- 参考代码：

```

1 punches = ['石头', '剪刀', '布']
2 computer_choice = random.choice(punches) # 电脑亮券
3 user_choice = input('请出拳：(石头、剪刀、布) ') # 请用户亮券
4 while user_choice not in punches: # 当用户输入错误，提示错误，重新输入
5     print('输入有误，请重新出拳')
6     user_choice = input() # 用户重新输入
7
8 # 双方亮拳信息
9 print('——战斗过程——')
10 print('电脑出了: %s' %(computer_choice))
11 print('你出了: %s' %(user_choice))

```

```

12
13 # 胜负结果
14 print('——结果——')
15 if user_choice == computer_choice:
16     print('平局! ')
17 elif (user_choice == '石头' and computer_choice == '剪刀') or (user_choice == '剪刀' and
18     computer_choice == '布') or (user_choice == '布' and computer_choice == '石头'):
19     print('你赢了! ')
20 else:
21     print('你输了! ')

```

课后练习2：让代码更简洁

- 题目解析：
 - 使用index()函数，优化用户赢的判断条件。
 - 可以先确定电脑随机选到的选项在列表punches中的索引位置，将其和玩家的选项联系起来。（如电脑出布，那么玩家的选项要出剪刀才能赢，通过元素的索引位置，观察规律）
 - 用户必赢规律：当用户亮券所对应的位置索引值是电脑所亮券所对应的位置索引值-1时，用户必赢。
- 参考代码：

```

1 import random
2 punches = ['石头', '剪刀', '布']
3 computer_choice = random.choice(punches)
4 user_choice = input('请出拳：(石头、剪刀、布) ')
5 while user_choice not in punches:
6     print('输入有误，请重新出拳')
7     user_choice = input()
8
9 print('——战斗过程——')
10 print('电脑出了: %s' % computer_choice)
11 print('你出了: %s' % user_choice)
12
13 print('——结果——')
14 if user_choice == computer_choice:
15     print('平局! ')
16 elif user_choice == punches[punches.index(computer_choice)-1]: # 判断电脑位置索引值-1的元素
17     print('你赢了! ')
18 else:
19     print('你输了! ')

```

第11关 编程思维：如何debug

课后练习1：一起来抓虫

- 题目解析：
 - 代码一：主要考察全局变量与局部变量的作用域，属于知识点debug。
 - 代码二：主要考察代码的运行逻辑，属于思维不清debug。
 - 代码三：主要考察代码的特殊性，属于被动掉坑debug。

● 参考代码：

```
1 # 代码一
2 scores = {'语文':89, '数学':95, '英语':80}
3 def get_average(scores):
4     sum_score = 0 # sum_score 作为函数内部的局部变量，从而可以为函数所用。
5     for subject, score in scores.items():
6         sum_score += score
7         print('现在的总分是%d'%sum_score)
8     ave_score = sum_score/len(scores)
9     print('平均分是%d'%ave_score)
10 get_average(scores)
```

```
1 # 代码二
2 not_bad_word = True
3 while not_bad_word:
4     x = input('请给旺财取个外号：')
5     if x == '小狗' or x == '汪汪': # 只要外号是两个中的一个，就会生气。
6         not_bad_word = False
7         print('我生气了，不想理你了！')
8
9 print('对不起，以后我不会这么叫你了')
```

```
1 # 代码三
2 deposit = [100,300,900,2000,5000,0,2000,4500]
3 for i in range(1, len(deposit)):
4     if deposit[i-1] == 0: # 判断除数等于0时，特殊处理。
5         print('你上次存款为 0 哦！')
```

```
6     else:  
7         times = deposit[i]/deposit[i-1]  
8         print('你的存款涨了%f倍' %times)
```

课后练习2：贴心的除法计算器

- 题目解析：
 - 使用input()函数，输入被除数，除数，并将数据类型转换为浮点型。
 - 通过try...except语句，排除掉输入了非数值（即不属于整数和浮点数）或被除数为零的情况。
- 参考代码：

```
1  print('\n欢迎使用除法计算器! \n')  
2  while True:  
3      try:  
4          x = input('请你输入被除数: ')  
5          y = input('请你输入除数: ')  
6          z = float(x)/float(y)  
7      except ZeroDivisionError: # 排除掉除数不为0的情况  
8          print('0是不能做除数的! ')  
9      except ValueError: # 排除掉非数值的情况  
10         print('除数和被除数都应该是整值或浮点数! ')  
11     else:  
12         print(x, '/', y, '=', z)  
13         break
```

第12关 类与对象1

课后练习1：一次性说完

- 题目解析：一次性“说完”居住地和出生地。
 - 新建一个方法，在这个方法中调用实例方法born和live，调用实例方法时需要加上self，即 self.方法名()。
 - 创建完之后，需要在类外创建一个实例，然后调用新建的方法。创建实例之后self就指向新建的实例，所以可以使用实例名.方法名()的方式进行调用。
- 参考代码：

```
1  class Chinese:
```

```

2     def __init__(self,hometown,region):
3         self.hometown = hometown
4         self.region = region
5         print('程序持续更新中.....')
6
7     def born(self):
8         print('我生在%s。'%self.hometown)
9
10    def live(self):
11        print('我在%s。'%self.region)
12
13    # 新建方法，调用上面的两个方法（注：方法名可自定义）。
14    def citys(self):
15        self.born()
16        self.live()
17
18 wufeng = Chinese('广东', '深圳')
19 wufeng.citys()  # 调用新建实例方法

```

课后练习2：重要的事情说三遍

- 题目解析：
 - 第一题：打招呼
 - 先创建一个Robot类，然后定义一个初始化方法。由于需要输入名字和称呼，所以需要使用input()语句来获取这两个值。
 - 然后要使得代码能够运行，需要进行实例化调用。
 - 第二题：打印三遍愿望
 - 定义一个名为say_wish()的实例方法之后，需要获取愿望，故要使用功能input()函数获取愿望。
 - 然后将愿望打印三遍，打印多遍用循环。
 - 最后还是实例化调用类，创建类对象后调用实例方法say_wish()。
- 参考代码：

```

1  # 第一题代码
2  class Robot:
3      def __init__(self):
4          self.name = input('我还没有名字，请给我起个名字吧：')
5          self.user = input('你叫什么名字：')
6          print('你好%s, 我叫%s, 很高兴认识你!' % (self.user, self.name))
7  robot1 = Robot()

```

```

8
9  # 第二题代码
10 class Robot:
11     def __init__(self):
12         self.name = input('我还没有名字, 请给我起个名字吧: ')
13         self.user = input('你叫什么名字: ')
14         print('你好%s, 我叫%s, 很高兴认识你!' %(self.user, self.name))
15     def say_wish(self):
16         wish = input('告诉我一个你的愿望吧: ')
17         print('%s的愿望是: %s' %(self.user, wish))
18         for i in range(3):
19             print(wish)
20
21 robot2 = Robot()
22 robot2.say_wish()

```

第13关 类与对象2

课后练习1：老师和父亲

- 题目解析：
 - 创建两个子类，同时继承已有的两个类，一个是(Teacher ,Father)，另一个是(Father ,Teacher)。
 - 在其中任选一个子类进行定制：将 face 属性的值改变为'gentle'。
 - 最后创建实例time3 和time4 分别调用两个子类，再用实例对象调用face 属性并打印。
- 参考代码：

```

1  # 第一题代码
2  class Teacher:
3      face = 'serious'
4      job = 'teacher'
5
6  class Father:
7      face = 'sweet'
8      parenthood = 'dad'
9
10 class TeacherMore(Teacher, Father):
11     pass
12
13 # 定义一个新的类或函数，如果其内部无须其他代码，也必须加上pass使得定义成立。

```

```

14 class FatherMore(Father, Teacher):
15     face = 'gentle'
16
17 time3 = TeacherMore()
18 time4 = FatherMore()
19 print(time3.face)
20 print(time4.face)

```

课后练习2：学习时间记录

- 题目解析：
 - 如果编程开发人员学 Python 的话，学习效率默认为1。则在继承之后，给rate 添加一个默认参数即可。
 - job 的属性为 programmer，即给job 添加要给默认参数。
 - 因为学习效率不同的人员可能不同，可以通过外部传值进行修改，所以默认参数添加到子类对应的方法中，而工作是固定的都是programmer，所以在调用父类的初始函数的时候，给job 传入一个参数，这样子就不会受到外部传入的参数的影响。

- 参考代码：

```

1  class Student:
2
3      def __init__(self, name, job=None, time=0.00, time_effective=0.00):
4          self.name = name
5          self.job = job
6          self.time = time
7          self.time_effective = time_effective
8
9      def count_time(self, hour, rate):
10         self.time += hour # 在类外进行多次调用时，会累计时间
11         self.time_effective += hour * rate
12
13 class Programmer(Student):
14     def __init__(self, name): # 因为job、time、time_effective都有了默认值，所以这里可以不用写
15         Student.__init__(self, name, job='programmer', time=0.00, time_effective=0.00)
16
17     def count_time(self, hour, rate=1):
18         Student.count_time(self, hour, rate)
19         #这里也是为了简写代码而直接调用类Student中的方法count_time。
20
21 student1 = Student('韩梅梅')
22 student2 = Programmer('李雷')

```

```
22 print(student1.job)
23 print(student2.job)
24 student1.count_time(10, 0.8)
25 student2.count_time(10)
26 print(student1.time_effective)
27 print(student2.time_effective)
```

第14关 流浪图书馆

课后练习1：书的分类

- 题目解析：
 - 创建一个子类FictionBook，继承父类Book。
 - 在子类改造父类的初始化方法，让程序能够打印出实例book的相关信息。由于初始函数不能使用return 返回字符串，而要打印相关的信息，所以需要加入本课的新知识点__str__()方法对信息以字符串形式进行返回。
- 参考代码：

```
1 class Book:
2     def __init__(self, name, author, comment, state = 0):
3         self.name = name
4         self.author = author
5         self.comment = comment
6         self.state = state
7
8 # 创建一个Book类的子类 FictionBook
9 class FictionBook(Book):
10    def __init__(self, name, author, comment, state = 0, type = '虚构类'):
11        Book.__init__(self, name, author, comment, state = 0)
12        self.type = type
13 # 这里是类的继承和改写，在初始化方法中增加了默认参数type = '虚构类'，而为了简写代码，直接调用Book类中的初始化方法，因为Book类中初始化方法本身不含参数type，故增加15行代码self.type = type。
14    def __str__(self):
15        status = '未借出'
16        if self.state == 1:
17            status = '已借出'
18        return '类型: %s 名称: 《%s》 作者: %s 推荐语: %s\n状态: %s' % (self.type,
19                           self.name, self.author, self.comment, status)
```

```
20 book = FictionBook('囚鸟', '冯内古特', '我们都是受困于时代的囚鸟')
21 print(book)
```

课后练习2：我想看TA的书

- 题目解析：
 - 要求补全show_author_book()方法，先用条件判断语句判断该作者在不在列表authors里，需要使用if条件句，直接判断输入的作者是否在作者列表中。
 - 如果不在就打印“很可惜，我们暂时没有收录这位作者的作品”。
 - 如果在，则遍历列表books的每个实例，当实例属性author与输入的作者名相等，就打印该实例。

- 参考代码：

```
1 class Book:
2     def __init__(self, name, author, comment, state = 0):
3         self.name = name
4         self.author = author
5         self.comment = comment
6         self.state = state
7     def __str__(self):
8         status = '未借出'
9         if self.state == 1:
10             status = '已借出'
11     #这里表示当state=1时，书籍状态为"已借出"
12     return '名称:《%s》作者:%s 推荐语: %s\n状态: %s' % (self.name, self.author,
13         self.comment, status)
14
15 class BookManager:
16     authors = []
17     def __init__(self):
18         book1 = Book('撒哈拉的故事', '三毛', '我每想你一次，天上便落下一粒沙，从此便有了撒哈拉。')
19         book2 = Book('梦里花落知多少', '三毛', '人人都曾拥有荷西，虽然他终会离去。')
20         book3 = Book('月亮与六便士', '毛姆', '满地都是六便士，他却抬头看见了月亮。')
21     #这里是实例化Book类并赋值给book1、book2、book3
22     self.books = [book1, book2, book3]
23     #将三个实例化类装进books列表中
24     self.authors.append(book1.author)
25     self.authors.append(book2.author)
26     self.authors.append(book3.author)
27     #将每个实例中的author装进authors列表
```

```

27
28     def menu(self):
29         while True:
30             print('1.查询书籍')
31             choice = int(input('请输入数字选择对应的功能: '))
32             if choice == 1:
33                 self.show_author_book()
34             else:
35                 print('感谢使用!')
36                 break
37
38     def show_author_book(self):
39         author = input('请输入想查询作家的名称: ')
40         if author in self.authors:
41             print(author + '的作品有: ')
42             for book in self.books:
43 #从books列表中遍历出每一个实例类
44                 if book.author == author:
45 #book.author表示调用实例的author属性
46                     print(book)
47             else:
48                 print('很可惜, 我们暂时没有收录这位作者的作品')
49
50 manager = BookManager()
51 manager.menu()

```

第15关 编码和文件读写课后

课后练习1：数据转移中的变化

- 题目解析：
 - 编码与解码：使用encode()函数和decode()函数可以实现编码与解码。
 - 通过文件读写复制图片：可以通过open()函数使用rb权限来打开一个图片文件，并把文件赋值给一个变量。再用open()函数使用wb权限打开一个新的图片文件，并把之前的变量写入，这样就完成了复制图片的操作。
- 参考代码：

```

1 # 1. 分别使用gbk和utf-8编码自己的名字，并打印。（以吴枫为例）
2 print('吴枫'.encode('gbk'))

```

```
3 print('吴枫'.encode('utf-8'))
4 # 结果为:
5 # b'\xe4\xce\xe2\xb7\xe3'
6 # b'\xe4\xce\x90\xb4\xe6\x9e\xab'
7
8 # 2. 复制上一步得到的结果, 进行解码, 分别打印。
9 print(b'\xe4\xce\xe2\xb7\xe3'.decode('gbk'))
10 print(b'\xe4\xce\x90\xb4\xe6\x9e\xab'.decode('utf-8'))
```

```
1 #练习2
2 with open('photo2.png', 'rb') as file: # 以"rb"模式打开图片
3     data = file.read()
4     with open('photo3.png', 'wb') as newfile: # 以"wb"模式写入
5         newfile.write(data)
```

```
1 #练习3-1
2 file1 = open('scores.txt', 'r', encoding='utf-8')
3 file_lines = file1.readlines()
4 file1.close()
5
6 final_scores = []
7 for i in file_lines:
8     data = i.split()
9     sum = 0 # 先把总成绩设为0
10    for score in data[1:]: # 遍历列表中第1个数据和之后的数据
11        sum = sum + int(score) # 然后依次加起来, 但分数是字符串, 所以要转换
12    result = data[0]+str(sum)+'\n' # 结果就是学生姓名和总分
13    print(result)
14    final_scores.append(result)
15 print(final_scores)
16
17 sum1 = open('winner.txt', 'w', encoding='utf-8')
18 sum1.writelines(final_scores)
19 sum1.close()
```

```
1 #练习3-2
2 file1 = open('winner.txt', 'r', encoding='utf-8')
3 file_lines = file1.readlines()
4 file1.close()
5
```

```

6  dict_scores = {}
7  list_scores = []
8  final_scores = []
9
10 for i in file_lines: # i是字符串。
11     print(i)
12     name = i[:-4] # 取出名字 (注：字符串和列表一样，是通过偏移量来获取内部数据。)
13     score = int(i[-4:-1]) # 取出成绩
14     dict_scores[score] = name # 将名字和成绩对应存为字典的键值对 (注意：这里的成绩是键)
15     list_scores.append(score)
16
17 list_scores.sort(reverse=True) # reverse，逆行，所以这时列表降序排列，分数从高到低。
18
19 for i in list_scores:
20     result = dict_scores[i] + str(i) + '\n'
21     final_scores.append(result)
22
23 print(final_scores) # 最终结果
24
25 winner_new = open('winner_new.txt', 'w', encoding='utf-8')
26 winner_new.writelines(final_scores)
27 winner_new.close()

```

课后练习2：古诗默写

- 题目解析：
 - 创建一个列表来存放需要默写的内容，列表中的每一个元素即是需要默写的诗句。
 - 使用open()函数打开诗词文件后，使用readlines()函数读取内容，得到一个列表。列表中的每一个元素是诗句的每一行内容。
 - 使用open()函数打开诗词文件，并用循环来比对原始诗词和我们需要默写的内容，如果一致就把原始诗词中相应的诗句替换成下划线，如果不一致则原封不动。

- 参考代码：

```

1 list_test = ['一弦一柱思华年。\\n', '只是当时已惘然。\\n'] # 将要默写的诗句放在列表里。
2 with open ('poem2.txt', 'r') as f:
3     lines = f.readlines()
4     print(lines)
5 with open('poem2.txt', 'w') as new:
6     for line in lines:
7         if line in list_test: # 属于默写列表中的句子，将其替换成横线。

```

```
8         new.write('_____。 \n')
9     else:
10        new.write(line)
```

第16关 模块

课后练习1：时间管理器

- 题目解析：
 - 设定一个无限循环，方便任务管理器的程序可以被反复执行。
 - 设定一个目标任务时间并赋值给变量，使用time.time()方法记录任务开始的时间。
 - 使用for ... in range():的程序结构来倒计时任务，range()函数中设置步长为-1，即可起到倒计时的效果。
 - 任务结束后继续使用time.time()函数记录任务结束的时间，并用任务结束的时间减去任务开始的时间，将结果记为实际任务时间，将实际任务时间记入txt文档中。
- 参考代码：

```
1 import time
2
3 input("欢迎使用\"时间管理器\"！请按回车继续。")
4 while True:
5     task_name = input('请输入任务名：')
6     task_time = int(input('觉得自己至少可以专注这个任务多少分钟？输入 N 分钟'))
7     input('此次任务信息：\n我要完成的任务：%s\n我至少要专注：%d分钟\n按回车开始计时：' %
8           (task_name, task_time))
9     start = time.time() # 开始计时
10    start_time = time.strftime('%Y-%m-%d %H:%M:%S', time.localtime(time.time())) # 格式化日期
11    # for t in range(task_time*60, 0, -1): # 实际代码：分钟转成秒要乘60，用-1来倒计时。
12    for t in range(task_time, 0, -1):
13        info = '请专注任务，还要保持专注 ' + str(t) + ' 秒哦！'
14        print(info, end="")
15        print("\b" * (len(info) * 2), end="", flush=True)
16    #\b是退格的意思，将原有的的内容从终端退出去。
17    #len()是计算字符串长度，在这里是计算info字符串的长度，因为汉字占2个字节，所以乘以2。
18    #"\b"*(len(info)*2)的意思是将终端里的内容退出。
19    #end=""是不换行的意思，flush=True是刷新的意思，即刷新终端内容后不换行。
```

```

20         time.sleep(1)
21     print('你已经专注了 %d 分钟，很棒~再加把劲，完成任务!' % task_time) # 倒计时后，才继续运行之后
22     的代码。
23
24     # 询问任务是否完成
25     task_status = input('请在任务完成后按输入y:')
26     if task_status == 'y':
27         end = time.time() # 一定用户按了 y，就记下结束时间。
28         end_time = time.strftime('%Y-%m-%d %H:%M:%S', time.localtime(time.time())) # 日
29         期格式化
30         actual_time = int((end - start) / 60) # 始末时间相减，从秒换算到分，除以60。
31         start_end = start_time + '—' + end_time + '\n'
32         with open('timelog3.txt', 'a', encoding='utf-8') as f:
33             f.write(task_name + ' 的预计时长为: ' + str(task_time) + '分钟\n')
34             f.write(task_name + ' 的实际时长为: ' + str(actual_time) + '分钟,具体时间为: ' +
35             start_end)
36         again = input('建立一个新任务请按 y, 退出时间日志记录器请按 q: ')
37         if again == 'q':
38             break
39     else:
40         print('抱歉，你的输入有误。请重启时间记录器。')
41
42     print('愿被你善待的时光，予你美好的回赠。')

```

课后练习2：再出古诗默写题

- 题目解析：
 - 替换诗句的逻辑同第15关的古诗默写练习题
 - 使用os.replace(src,dst)函数来修改文件名，src是原文件名，dst是目标文件名。
- 参考代码：

```

1 import os
2 list_test = ['一弦一柱思华年。\\n', '只是当时已惘然。\\n']
3
4 with open ('poem3.txt', 'r') as f:
5     lines = f.readlines()
6
7 with open('poem_new.txt', 'w') as new:
8     for line in lines:
9         if line in list_test:
10             new.write('_____。\\n')

```

```
11     else:  
12         new.write(line)  
13  
14 os.replace('poem_new.txt', 'poem3.txt')
```

第17关 邮件发送

课后练习1：自制动态二维码

- 题目解析：
 - 注意代码中参数的名称不能改变，picture参数和save_name两个参数可以写为绝对路径。
- 参考代码：

```
1 # 先导入模块
2 from MyQR import myqr
3 myqr.run(
4     words='http://weixin.qq.com/r/kzlje9TEE4lsrzAY92yB',
5     # 扫描二维码后，显示的内容，或是跳转的链接
6     version=5,    # 设置容错率
7     level='H',    # 控制纠错水平，范围是L、M、Q、H，从左到右依次升高
8     picture='she-3.gif',  # 图片所在目录，可以是动图
9     colorized=True,   # 黑白(False)还是彩色(True)
10    contrast=1.0,    # 用以调节图片的对比度，1.0 表示原始图片。默认为1.0。
11    brightness=1.0,   # 用来调节图片的亮度，用法同上。
12    save_name='Python.gif',  # 控制输出文件名，格式可以是 .jpg, .png , .bmp , .gif
13 )
14 # save_dir= r'C:\Users\user\Desktop\test'# '指定二维码生成的地址，填写路径就行'
```

课后练习2：绝对值的多种求法

- 题目解析：
 - 方法一：使用条件判断语句，负数时候将变量改为它的相反数，正数时候不变。
 - 方法二：使用内置的abs()函数计算。
 - 方法三：使用math库的fabs()函数计算。
- 参考代码：

```
1 import math
2 # 方法1：条件判断
3 def abs_value1():
4     a = float(input('1.请输入一个数字: '))
5     if a >= 0:
6         a = a
7     else:
8         a = -a
9     print('绝对值为: %f' % a)
10
11 # 方法2：内置函数 abs()
```

```

12 def abs_value2():
13     a = float(input('2.请输入一个数字: '))
14     a = abs(a)
15     print('绝对值为: %f' % a)
16
17 # 方法3: 内置模块 math
18 def abs_value3():
19     a = float(input('3.请输入一个数字: '))
20     a = math.fabs(a)
21     print('绝对值为: %f' % a)
22
23 # 运行函数, 查验一下。
24 abs_value1()
25 abs_value2()
26 abs_value3()

```

第18关 编程思维：产品思维

课后练习1：拯救选择困难症

- 题目解析：
 - ‘不知道吃什么有两种原因’，一是完全不知道要吃什么，二是在几家店之间犹豫。
 - 可先使用if...elif...else语句，判断出是属于上述的那种情况，并根据用户选择的情况，进行相关的食物推送。
 - 定义菜名选择函数，无论用户选择那种情况，都需要对推送的菜名进行选择，所以此函数可以多次复用。
- 参考代码：

```

1 import random
2
3 # 将需要用到的表格和变量放在开头
4 list_food = ['KFC', '蒸菜馆', '楼下快餐店', '桂林米粉', '东北饺子', '金牌猪脚饭', '三及第汤饭']
# 备选菜单，可自定义。
5 list_choice = []
6
7 # 菜名选择
8 def choose(list):
9     while True:
10         food = random.choice(list)
11         judgement = input('去吃【%s】好不好啊？同意的话输入y，不想吃直接回车即可。' % (food))

```

```

12         if judgement == 'y':
13             print('去吃【%s】！就这么愉快地决定啦!' % (food))
14             break
15
16 # 判断环节
17 reason = int(input('你不知道吃的原因是：1.完全不知道吃什么；2.在几家店之间徘徊（请输入1或
18 2）：'))
19 if reason == 1:
20     choose(list_food)
21 elif reason == 2:
22     add = True
23     while add:
24         choice = input('请输入让你犹豫的店名（注：一家一家输，完成后输入y）：')
25         if choice != 'y': # 这个判断语句，是为了不将 y 也添加到菜单里。
26             list_choice.append(choice)
27         if choice == 'y':
28             add = False
29     choose(list_choice)
30 else:
31     print('抱歉，目前还不支持第三种情况—不过，你可以加代码哦。')

```

课后练习2：模拟广告牌

- 题目解析：

- 定义一个变量，用于存放关于广告牌的显示内容。
- os.system()函数，对屏幕内容清空，再使用字符串拼接，将广告词的第一个元素与剩下的所有元素进行拼接，成为新的广告词。（例如广告词：‘你好啊， python! ’，拼接后尾： ‘!你好啊， python’）。
- 根据规律，每一次的清屏，广告词都会进行变化，只要设计清屏的时间够快，躲过人类的反应时间，那么就可以形成滚动的效果。
- 最后添加while循环来不停的滚动广告词。

- 参考代码：

```

1 import os, time
2
3 def main():
4     content = ' 风变编程，陪你一起学Python ' # 广告词可自定义。
5     while True:
6         os.system('clear') # windows系统将'clear'改为'cls'。
7         print(content)

```

```
8         content = content[1:] + content[0]
9         time.sleep(2)
10
11 if __name__ == '__main__': # 类里面学到的检测方法，在函数中其实也可以用。
12     main()
```

爬虫精进课程

第0关 认识爬虫

课后练习1：文章下载

- 题目解析：
 - 调用requests库，使用requests.get()获取文件，返回Response对象。
 - 由于该链接是一个文本，需要把Response对象转化为字符串输出，所以使用text属性返回数据。
- 参考代码：

```
1 import requests
2
3 destnation_url = 'https://localprod.pandateacher.com/python-manuscript/crawler-
4           html/exercise/HTTP%93%8D%E5%BA%94%E7%8A%B6%E6%80%81%E7%A0%81.md'
5
6 res = requests.get (destnation_url)
7 print(res.status_code) # 查看响应码
8 article=res.text      # 把Response对象的内容以字符串的形式返回
9 print(article)
```

课后练习2：图像下载

- 题目解析：
 - 调用requests库，使用requests.get('URL')获取文件，返回Response对象。
 - 由于该链接是一个图片，需要把Response对象转化为二进制输出，所以使用content属性返回数据。
 - 返回的二进制数据通过打开文件>写入文件>关闭文件，将图片文件储存到本地。
- 参考代码：

```
1 import requests
2 res = requests.get('https://res.pandateacher.com/2019-01-12-15-29-33.png')
```

```
3 # 发出请求，并把返回的结果放在变量res中
4 pic=res.content      # 把Reponse对象的内容以二进制数据的形式返回
5
6 photo = open('spider.jpg', 'wb') # 新建了一个文件ppt.jpg，可以添加文件路径存放到指定位置
7 photo.write(pic)    # 写入pic的二进制内容
8 photo.close()       # 关闭文件
```

课后练习3：音频下载

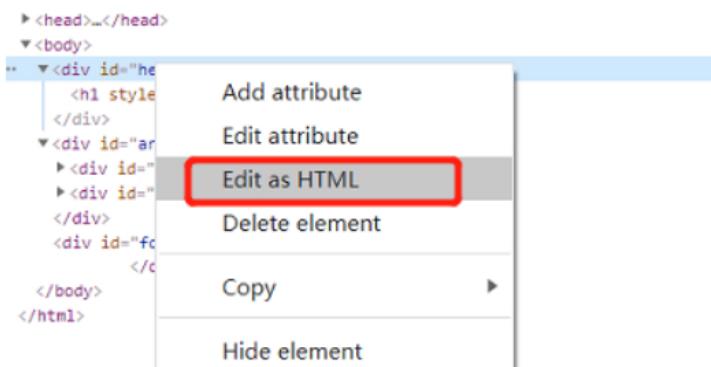
- 题目解析：
 - 音频的下载原理，同图片。
- 参考代码：

```
1 import requests
2
3 res=requests.get('https://static.pandateacher.com/Over%20The%20Rainbow.mp3')
4 # 发出请求，并把返回的结果放在变量res中
5 mp3=res.content # 把Reponse对象的内容以二进制数据的形式返回
6
7 music = open('rainbow.mp3', 'wb')# 新建了一个文件ppt.jpg，可以添加文件路径存放到指定位置
8 music.write(mp3) # 获取pic的二进制内容
9 music.close()    # 关闭文件
```

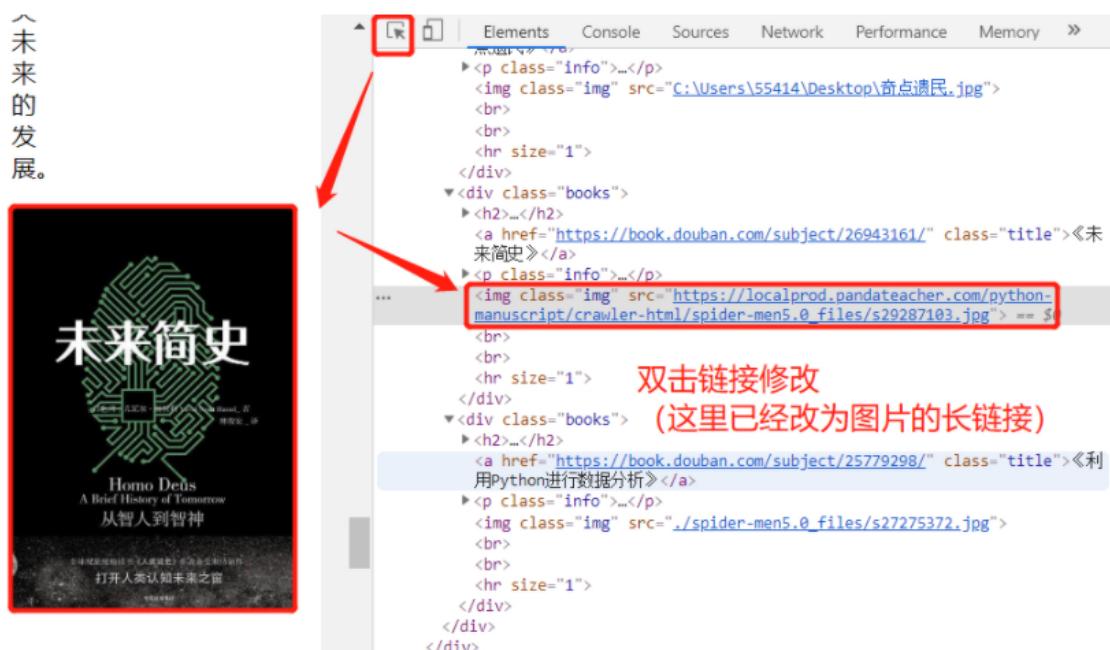
第1关 HTML基础

课后练习：我的书苑我做主

- 题目解析：
 - 右键element代码>Edit as HTML，会出现一个编辑框，可以尽情修改HTML代码。



- 注意：直接下载网页代码，保存到.html文件中，直接用浏览器打开时，会出现图片显示不了。此时修改一下图片的链接即可，可以改为图片的网页链接，也可以执行下载，使用本地路径的绝对路径。



- 参考代码：

- 参考修改链接：<https://localprod.pandateacher.com/python-manuscript/crawler-html/exercise/01-01-test.html>

⚠ 注意，修改图片是直接将自己要替换的图片的链接在检查里面双击替换。

第2关 BeautifulSoup

课后练习1：博客爬虫

- 题目解析：
 - 第1步：获取数据，使用requests.get()
 - 第2步：解析数据，使用BeautifulSoup(网页源代码的字符串格式,'html.parser')
 - 第3步：提取数据，使用find_all()，提取之后返回一个Tag集合，需要遍历提取每一个值，然后使text属性提取字符串。
 - 寻找数据所在标签，可以借用“小箭头”定位；
 - 定位到数据之后，查看数据标签能否被精准定位到，可以调出搜索框搜索，一般有2种判断方法，一种是标签唯一，一种是数据的数量和标签的数量刚好对应；
 - 这里的评论数量可能只有一条也可以有多条，可以根据数据数量和标签数量判断：像以下这一条评论，最小的一个标签是<p>但是搜索的时候发现，源代码中有很多<p>标签，不能准确定位，然后就往上一级去找，找到"comment-content"属性的时候，发现一共有7个，刚好和左边的评论个数（7个）一致！（评



- 找到标签数和数据数量一致时，可以尝试爬虫，该作业可以准确爬取到。

- 参考代码：

```
1 import requests
2 from bs4 import BeautifulSoup
3 # 获取数据
4 destination_url = 'https://wordpress-edu-3autumn.localprod.oc.forchange.cn/all-about-
the-future_04/'
5 destination = requests.get(destination_url)           # 返回一个response对象，赋值给
destination
6 # 解析数据
7 soup = BeautifulSoup(destination.text, 'html.parser')      # 把网页解析为BeautifulSoup
对象
8 # 提取数据
9 comments = soup.find_all('div', class_= 'comment-content') # 通过匹配属性提取出我们想要的元
素
10 for comment in comments: # 遍历列表，取出列表中的每一个值
11     print(comment.text) # 打印评论的文本
```

课后练习2：书店寻宝

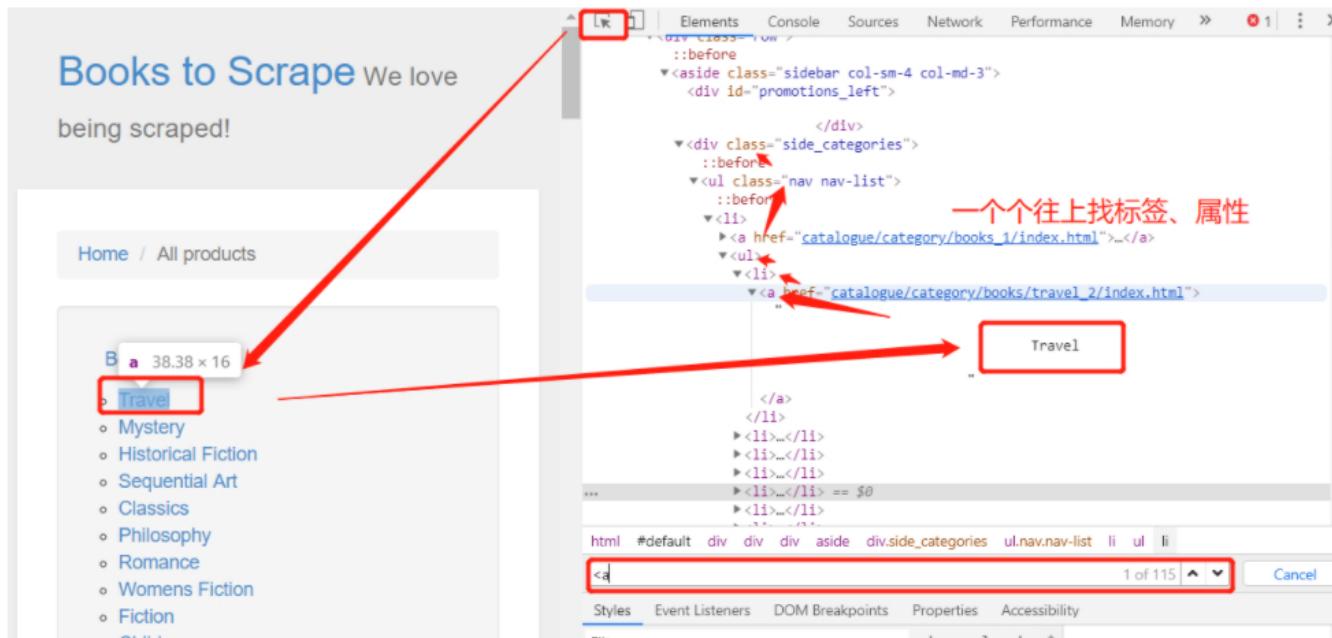
第一个小练习

- 题目解析：

- 第1步：获取数据，使用`requests.get()`。
- 第2步：解析数据，使用`BeautifulSoup(网页源代码的字符串格式, 'html.parser')`。
- 第3步：提取数据，使用`find_all()`和`find()`交替使用，提取数据Tag对象，最终根据数据需要呈现的内容（字符

串或属性值) 选择text属性或["属性名"]提取。

- 定位数据所在标签 (方法同爬虫第2关课后练习1定位数据方法) 。
- 该练习比上个练习加了一个难度, 就是数据所在的标签并不能精准定位到所要爬取的数据, 因为标签的数量远大于所要爬取数据的数量, 此时需要一级一级往上找能够精准定位到的父级标签。
- 从依次通过- >
-> - 到
才能准确定位到。



- 注意一点: 如果属性值有多个, 会使用空格隔开, 取其中一个即可 (如:
中有两个属性值nav和nav-list)

- 参考代码:

```
1 import requests
2 from bs4 import BeautifulSoup
3
4 res_bookstore = requests.get('http://books.toscrape.com/')
5 bs_bookstore = BeautifulSoup(res_bookstore.text, 'html.parser')
6 list_kind = bs_bookstore.find('ul', class_='nav').find('ul').find_all('li') # 需要提取好几层
7
8 for tag_kind in list_kind:
9     tag_name = tag_kind.find('a').text      # 提取字符
10    print(tag_name.strip())               # 去除字符串首尾空格、空行等
```

第二个小练习

- 题目解析:

- 该练习在上一个练习上也递增了难度——爬取多条数据。
- 先定位数据所在标签。
- 爬取多条数据, 需要找到每一条数据的共同父级标签; 需要一级一级往上寻找数据的最小父级标签 (一开始

不熟悉可以把各级的父级标签写下来，然后对照每一个数据的父级，找到对应的共同父级，一般10级以内可以找到）。

- 找到共同父级标签之后，可以调出element的搜索框，先在网页中查找一遍判断是否可以准确定位到，也可以在编译器敲代码进行爬取测试；如果不行，再往上找父级标签，然后再网页查找或爬取测试，直到能够准确定位到。
- 确定可以定位到的共同父级标签之后，需要再查看小标签能否精准定位，就是查看在该共同标签内能不能精准定位到我们需要的数据的位置，也可以通过网页查找或爬取测试。
- 注意一点：取星级是取class属性的属性值，需要使用['class']进行取值。

- 参考代码：

```
1 import requests
2 from bs4 import BeautifulSoup
3 res_bookstore =
4     requests.get('http://books.toscrape.com/catalogue/category/books/travel_2/index.html')
5
6 bs_bookstore = BeautifulSoup(res_bookstore.text, 'html.parser')
7 list_books = bs_bookstore.find_all(class_='product_pod')
8 for tag_books in list_books:
9     tag_name = tag_books.find('h3').find('a') # 定位到该a标签需要先提取h3
10    list_star = tag_books.find('p', class_="star-rating")
11    tag_price = tag_books.find('p', class_="price_color")
12
13    print('书名:', tag_name['title']) # 这里用到了tag['属性名']提取属性值
14    print('评分:', list_star['class'][1]) # 提取class属性的值，返回列表['star-rating',
15    'Two'], 然后取第1个索引值
16    print('价格:', tag_price.text + '\n' + '-----' + '\n') # 打印，此处换行符可以去掉，是为了
17    让数据更加清晰地分隔开
```

课后练习3：博客文章

- 题目解析：
 - 做该练习时，同学们可以参考课后练习2中的搜索过程进行探索；
 - 该练习和上一个练习也有不同之处：爬取链接。
 - 爬取链接的取法同属性值的取法，不过取出链接之后，由于链接是短链接（只有后面一段，没有域名等，相当于相对路径），所以需要在element上查看完整链接，把鼠标放在链接上即可出现完整链接，也可以点击链接调整到新页面直接复制地址栏的地址。

- 参考代码：

```
1 import requests
2 from bs4 import BeautifulSoup
3
4 url_destination = 'https://spidermen.cn/'
5 res_destination = requests.get(url_destination)
6 print(res_destination.status_code) # 打印响应码
7
8 bs_articles = BeautifulSoup(res_destination.text, 'html.parser')
9 list_articles = bs_articles.find_all('header', class_ = "entry-header") # 首先找到每篇文章所在的相同的元素
10 for tag_article in list_articles: # 遍历列表
11     tag_title = tag_article.find('h2', class_ = "entry-title")          # 找文章标题
12     tag_url = tag_article.find('a', rel = "bookmark")      # 找文章链接
13     tag_date = tag_article.find('time', class_ = "entry-date published") # 找文章发布时间
14     print(tag_title.text, '发布于：', tag_date.text)        # 打印文章标题与发布时间
15     print(tag_url['href'])           # 换行打印文章链接，需要使用属性名提取属性值
```

第3关 BeautifulSoup实践

课后练习：豆瓣爬虫

- 题目解析：
 - 经典三步骤：获取数据、解析数据、提取数据，前面两步同学们应该较为熟悉了，难点就是在第三步。
 - 该练习有一定的难度，不过经历过下厨房之后，应该不会觉得太难。还是使用定位数据标签>查找共同父级标签>判断标签是否可行的思路进行验证数据标签和父级标签的可行性。具体操作也可以参考爬虫第2关的课

后练习讲解，这里大概讲一下操作路径：定位到数据标签之后>往上找父级标签，最终聚焦到最小父级标签<div class="item">。

1 分别定位数据的位置

2 接下来往上找最小父级标签

- 相比下厨房，加了一个要求：循环获取多个页面的数据。这里就需要对多个页面的网址进行查看，可以把网址放到一起，进行对比，寻找规律。经过一番对比，可以很明了的发现，除了第一页，其他的页面只有start=后面的参数不同，这些参数都是25的倍数，除以25之后，便是2,3,4,5,6.....所以可以确定第2页之后，第n页的网址中start=后面的参数是 $25*n$ ；再根据该规律，应用到第1页进行测试，结果发现可行，故十页中的网址都可以通过n的值来确定，而n的值可以使用for循环赋值，即for n in range(10)，再对网址做拼接即可。
- 注意3点：
 - 最小父级标签比较方便往下定位数据，如果在最小父级标签的上级标签中也可以定位到的话，也可以没有唯一答案，比如：使用`bs.find('div', class_="article").find_all("li")`和`bs = bs.find('ol', class_="grid_view").find_all("li")`也可以定位到所需要的数据，感兴趣可以尝试一下。
 - 如果不加headers（请求头）爬取不到数据，可以加上试试，详见以下参考代码第2行，注意在第8行处需要传入headers；
 - 由于推荐语部分数据缺失，需要加一个判断条件把没有推荐语的数据进行处理，想见以下参考代码第17~21行，当然也可以使用功能try...except...进行判断（参考以下代码第23~27行）。

- 参考代码：

```
1 import requests, bs4
2 headers = {'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36'}
3 # 增加请求头，放在get()前面即可
```

```

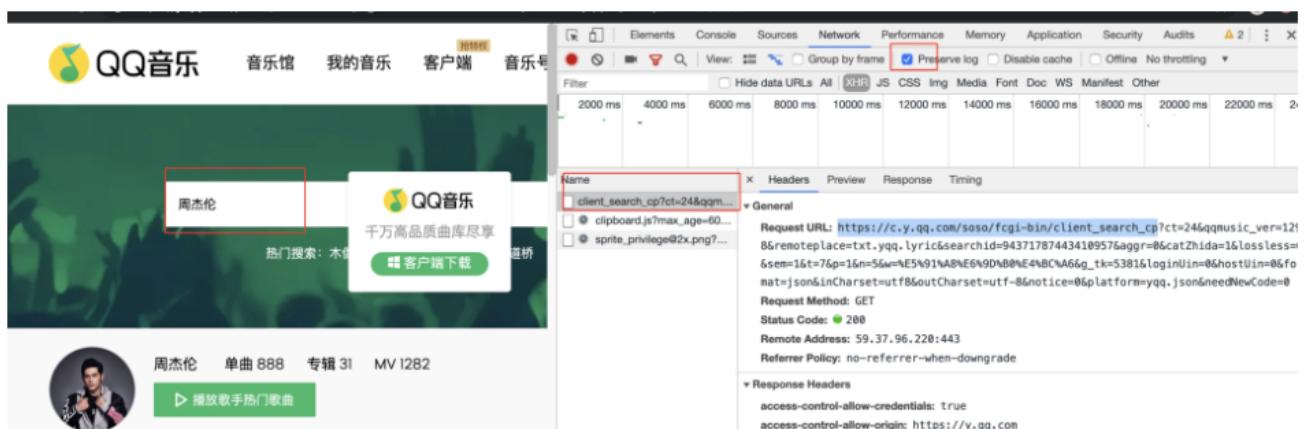
4
5     for x in range(10):
6         url = 'https://movie.douban.com/top250?start=' + str(x * 25) + '&filter='
7         # res = requests.get(url)
8         res = requests.get(url, headers=headers)  # 向get()函数添加参数headers。
9         bs = bs4.BeautifulSoup(res.text, 'html.parser')
10        bs = bs.find_all('div', class_="item")
11        for titles in bs:
12            num = titles.find('em', class_="").text
13            title = titles.find('span', class_="title").text
14            comment = titles.find('span', class_="rating_num").text
15            url_movie = titles.find('a')['href']
16
17            if titles.find('span', class_="inq") != None:
18                tes = titles.find('span', class_="inq").text
19            else:
20                tes = "没有评论语"
21            print(num + '.' + title + '—' + comment + '\n' + '推荐语: ' + tes + '\n' +
url_movie)

```

第5关 带参数请求数据

课后练习1：歌词爬取

- 题目解析：
 - 第0步-打开QQ音乐网站-输入关键字‘周杰伦’-点击歌词。 (图一)
 - 第1步-寻找在请求中的歌词信息。 (图二)
 - 第2步-改变网址中page变量的值，打开Network - XHR-client_search - Headers - Query String Parameters，观察里面参数的变化。



access-control-expose-headers: Area
area: sz
content-encoding: gzip

3 / 9 requests | 6.2 KB / 6.5 KB tr
Console What's New X
Highlights from the Chrome 75 update
Meaningful preset values when autocompleteing CSS functions
Properties like "filter" that takes functions for values now autocomplete to previewable values in the Styles pane.
Clear site data from the Command Menu
Press Command + Shift + P and run the "Clear Site Data" command to clear cookies, storage, and more.

图一

QQ音乐 音乐馆 我的音乐 客户端

龙卷风 - 周杰伦

龙卷风 - 周杰伦 (Jay Chou)

词: 徐若瑄

曲: 周杰伦

单曲 专辑 MV 歌单 用户 歌词

龙卷风 - 周杰伦

龙卷风 - 周杰伦 (Jay Chou)

词: 徐若瑄

曲: 周杰伦

编曲: 周杰伦

客户端

Elements Console Sources Network Memory Application Security Audits

Search Filter Hide data URLs All (client_search_cp) 1000 ms 1500 ms 2000 ms 2500 ms 3000 ms 3500 ms 4000 ms 4500 ms

Name Headers Preview Response Cookies Timing

client_search_cp: {code: 0, ...} code: 0
data: {keyword: "周杰伦", lyric: {curnum: 5, curpage: 1, list: [...], totalnum: 376}, keyword: "周杰伦"}
lyric: {curnum: 5, curpage: 1, list: [...], totalnum: 376}
curnum: 5
curpage: 1
list: [...]
0: {albumid: 8218, albummid: "000f01724fd7TH", albumname: "Jay", albumname_hilite: "Jay", albumid: 8218, albummid: "000f01724fd7TH", albumname: "Jay", albumname_hilite: "Jay", chinesesinger: 0, content: "龙卷风 - 周杰伦 (Jay Chou)\n词: 徐若瑄\n曲: 周杰伦\n", docid: "14303915167107409415", download_url: "http://soso.music.qq.com/cgi-bin/fcg_download_lrc.q?song=龙卷风&ct=24&qqmusic_ver=1.2.8.20130315", interval: 250, isonly: 1, lyric: "曲: 周杰伦\n编曲: 周杰伦\n制作人: 周杰伦\n", media_mid: "0039Q1KR3RmT3I", msgid: 13, nt: 1248851160, pay: {payalbum: 0, payalbumprice: 0, paydownload: 1, payinfo: 1, payplay: 1, payurl: 0}, payalbum: 0, payalbumprice: 0, paydownload: 1, payinfo: 1, payplay: 1, payurl: 0, preview: {trybegin: 66348, tryend: 150127, triesize: 960887}, pubtime: 973526400, pure: 0, singer: [{id: 4558, mid: "0025Nh1N2yWrP4", name: "周杰伦", name_hilite: "周杰伦"}, size128: 4016134, size320: 10037805, sizeape: 26976920, sizeflac: 27794719}

怎么找数据？

1、根据每条线的位置，一般靠右的是下一层，要一层一层往下，而不能直接提取

2、遇到0,1这种键名的，一般都是列表，所以需要用for循环后，再往下找

13 / 52 requests | 8.5 KB / 24.0 KB transferred | Finish...

图二

- 参考代码：

```
1 import requests
2 import json
3 # 引用requests,json模块
4
5 url = 'https://c.y.qq.com/soso/fcgi-bin/client_search_cp'
6 headers = {
7     'referer':'https://y.qq.com/portal/search.html',
8     'user-agent':'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36'
9     # 标记了请求从什么设备, 什么浏览器上发出
10 }
11
12 for x in range(5):
13     params = {
14         'ct':'24',
15         'qqmusic_ver': '1298',
16         'new_json':'1',
17         'remoteplace':'sizer.yqq.lyric_next',
18         'searchid':'94267071827046963',
19         'aggr':'1',
20         'cr':'1',
21         'catzhida':'1',
22         'lossless':'0',
23         'sem':'1',
24         't':'7',
25         'p':str(x+1),
26         'n':'10',
27         'w':'周杰伦',
28         'g_tk':'1714057807',
29         'loginUin':'0',
30         'hostUin':'0',
31         'format':'json',
32         'inCharset':'utf8',
33         'outCharset':'utf-8',
34         'notice':'0',
35         'platform':'yqq.json',
36         'needNewCode':'0'
37     }
38
39 res = requests.get(url, params = params)
40 #下载该网页, 赋值给res
41 jsonres = json.loads(res.text)
```

```

42     #使用json来解析res.text
43     list_lyric = jsonres['data']['lyric']['list']
44     #一层一层地取字典，获取歌词的列表
45
46     for lyric in list_lyric:
47         #lyric是一个列表，x是它里面的元素
48         print(lyric['content'])
49     #以content为键，查找歌词

```

课后练习2：头号粉丝

- 题目解析：
 - 观察xhr并爬取其他歌手的音乐信息（包括歌曲名，专辑名，播放时长，播放链接）。
 - 打开Network - XHR-client_search - Headers - Query String Parameters，通过观察，参数'w'表示的是歌手的名字，因此，我们可以改变参数'w'，来达到爬取不同歌手的音乐信息。
- 参考代码：

```

1 import requests, json
2
3 # 引用requests模块
4 url = 'https://c.y.qq.com/soso/fcgi-bin/client_search_cp'
5 singer = input('你喜欢的歌手是谁呢？')
6 for x in range(6):
7     params = {
8         'ct':'24',
9         'qqmusic_ver': '1298',
10        'new_json':'1',
11        'remoteplace':'txt.yqq.song',
12        'searchid':'70717568573156220',
13        't':'0',
14        'aggr':'1',
15        'cr':'1',
16        'catzhida':'1',
17        'lossless':'0',
18        'flag_qc':'0',
19        'p':str(x+1),
20        'n':'20',
21        'w':singer,
22        'g_tk':'714057807',
23        'loginUin':'0',
24        'hostUin':'0',

```

```

25     'format':'json',
26     'inCharset':'utf8',
27     'outCharset':'utf-8',
28     'notice':'0',
29     'platform':'yqq.json',
30     'needNewCode':'0'
31 }
32 # 将参数封装为字典
33 res_music = requests.get(url,params=params)
34 # 调用get方法，下载这个列表
35 json_music = res_music.json()
36 # 使用json()方法，将response对象，转为列表/字典
37 list_music = json_music['data']['song']['list']
38 # 一层一层地取字典，获取歌单列表
39 for music in list_music:
40     # list_music是一个列表，music是它里面的元素
41     print(music['name'])
42     # 以name为键，查找歌曲名
43     print('所属专辑: '+music['album']['name'])
44     # 查找专辑名
45     print('播放时长: '+str(music['interval'])+'秒')
46     # 查找播放时长
47     print('播放链接: https://y.qq.com/n/yqq/song/'+music['mid']+'.html\n\n')
48     # 查找播放链接

```

第6关 CSV&Excel

课后练习：存储电影信息

- 题目解析：
 - 使用excel表格存储：
 - 爬取逻辑同第三关豆瓣练习。
 - 使用openpyxl库创建工作簿和工作表之后，在第一行A1到E1位置写上表头内容。
 - 使用工作表对象的append()方法可以一行一行地向表格中添加内容，append()函数的参数是一个列表。
 - 使用csv表格存储：
 - 使用open函数可以创建一个用于读取文件的对象。再用csv.writer()函数创建一个适用于csv表格的写入对象。
 - 使用csv表格写入对象的writerow()方法可以按行写入内容，函数的参数是列表形式。

- 参考代码:

```
1 # Excel存储
2 import requests, random, bs4, openpyxl
3 wb=openpyxl.Workbook()
4 #创建工作薄
5 sheet=wb.active
6 #获取工作薄的活动表
7 sheet.title='movies'
8 #工作表重命名
9 sheet['A1'] ='序号'          #加表头, 给A1单元格赋值
10 sheet['B1'] ='电影名'         #加表头, 给B1单元格赋值
11 sheet['C1'] ='评分'          #加表头, 给C1单元格赋值
12 sheet['D1'] ='推荐语'         #加表头, 给D1单元格赋值
13 sheet['E1'] ='链接'          #加表头, 给E1单元格赋值
14
15 headers = {'User-Agent':
16 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5) AppleWebKit/537.36 (KHTML, like Gecko)
17 Chrome/85.0.4183.121 Safari/537.36'}
18
19 for x in range(10):
20     url = 'https://movie.douban.com/top250?start=' + str(x*25) + '&filter='
21     res = requests.get(url,headers = headers)
22     bs = bs4.BeautifulSoup(res.text, 'html.parser')
23     bs = bs.find('ol', class_="grid_view")
24     for titles in bs.find_all('li'):
25         num = titles.find('em',class_="").text
26         title = titles.find('span', class_="title").text
27         comment = titles.find('span',class_="rating_num").text
28         url_movie = titles.find('a')['href']
29
30         if titles.find('span',class_="inq") != None:
31             tes = titles.find('span',class_="inq").text
32             sheet.append([num, title, comment, tes, url_movie])
33             # 把num, title, comment, tes和url_movie写成列表, 用append函数多行写入Excel
34             print(num + '.' + title + '—' + comment + '\n' + '推荐语: ' + tes + '\n' +
35             url_movie)
36         else:
37             sheet.append([num, title, comment, None,url_movie])
38             print(num + '.' + title + '—' + comment + '\n' + '\n' + url_movie)
39
40 wb.save('movieTop250.xlsx')
```

```

1  # csv方法
2  import requests, random, bs4, csv
3  #引用csv模块。
4  csv_file=open('movieTop250.csv', 'w', newline='')
5  #调用open()函数打开csv文件，传入参数：文件名“movieTop250.csv”、写入模式“w”、newline=''.
6  writer = csv.writer(csv_file)
7  # 用csv.writer()函数创建一个writer对象。
8  writer.writerow(['序号', '电影名', '评分', '推荐语', '链接'])
9  #调用writer对象的writerow()方法，可以在csv文件里写入title:'序号', '电影名', '评分', '推荐语',
10 '链接'
11
12 headers = {'User-Agent':
13     'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5) AppleWebKit/537.36 (KHTML, like Gecko)
14     Chrome/85.0.4183.121 Safari/537.36'}
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32

```

csv方法

import requests, random, bs4, csv

#引用csv模块。

csv_file=open('movieTop250.csv', 'w', newline='')

#调用open()函数打开csv文件，传入参数：文件名“movieTop250.csv”、写入模式“w”、newline=''。

writer = csv.writer(csv_file)

用csv.writer()函数创建一个writer对象。

writer.writerow(['序号', '电影名', '评分', '推荐语', '链接'])

#调用writer对象的writerow()方法，可以在csv文件里写入title:'序号', '电影名', '评分', '推荐语', '链接'

headers = {'User-Agent':

'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5) AppleWebKit/537.36 (KHTML, like Gecko)

Chrome/85.0.4183.121 Safari/537.36'}

for x in range(10):

url = 'https://movie.douban.com/top250?start=' + str(x*25) + '&filter='

res = requests.get(url, headers = headers)

bs = bs4.BeautifulSoup(res.text, 'html.parser')

bs = bs.find('ol', class_="grid_view")

for titles in bs.find_all('li'):

num = titles.find('em', class_="").text

title = titles.find('span', class_="title").text

comment = titles.find('span', class_="rating_num").text

url_movie = titles.find('a')['href']

if titles.find('span', class_="inq") != None:

tes = titles.find('span', class_="inq").text

把num, title, comment, tes和url_movie写成列表，用append函数多行写入Excel

writer.writerow([num + '.' + title + '—' + comment + '\n' + '推荐语: ' +

tes + '\n' + url_movie])

else:

writer.writerow([num + '.' + title + '—' + comment + '\n' + '\n' +

url_movie])

csv_file.close()

第7关 爬取特朗普新闻

课后练习：做个测单词的小工具

- 题目解析：
 - 第0步-选择题库。
 - 第1步-根据选择的题库，获取50个单词。
 - 第2步-让用户选择认识的单词：此处，要分别记录下用户认识哪些，不认识哪些。
 - 第3步-对于用户认识的单词，给选择题让用户做：此处要记录用户做对了哪些，做错了哪些。这一步是第0步和第2步的组合——涉及到第0步中的选择，也涉及到第2步的数据记录。
 - 第4步-生成报告：50个单词，不认识多少，认识多少，掌握多少，错了多少。生成报告主要有三部分：第输出统计数据；打印错题集；把错题集保存到本地。
- 参考代码：

```
1 import requests
2 link = requests.get('https://www.shanbay.com/api/v1/vocabtest/category/')
3 #先用requests下载链接。
4
5 js_link = link.json()
6 #解析下载得到的内容。
7
8 # ----第0步-选择题库-----
9 bianhao = int(input('''请输入你选择的词库编号，按Enter确认
10 1, GMAT 2, 考研 3, 高考 4, 四级 5, 六级
11 6, 英专 7, 托福 8, GRE 9, 雅思 10, 任意
12 >'''))
13 #让用户选择自己想测的词库，输入数字编号。int()来转换数据类型
14 ciku = js_link['data'][bianhao-1][0]
15
16 # ----第1步-根据选择的题库，获取50个单词-----
17 test=requests.get('https://www.shanbay.com/api/v1/vocabtest/vocabularies/?category='+ci
ku)
18 #利用用户输入的数字编号，获取题库的代码。如果以输入“高考”的编号“3”为例，那么ciku的值就是，在字典
js_link中查找data的值，data是一个list，查找它的第bianhao-1，也就是第2个元素，得到的依然是一个
list，再查找该list的第0个元素。最后得到的就是我们想要的NCEE。
19 words = test.json()
20 #下载并解析用于测试的50个单词。
21
22 danci = []
23 #新增一个list，用于统计用户认识的单词
24 words_knows = []
```

```

25 #创建一个空的列表，用于记录用户认识的单词。
26 not_knows = []
27 #创建一个空的列表，用于记录用户不认识的单词。
28 print ('测试现在开始。如果你认识这个单词，请输入Y，否则直接敲Enter: ')
29
30 # ----第2步-让用户选择认识的单词----
31 n=0
32 for x in words['data']:
33 #启动一个循环，循环的次数等于单词的数量。
34     n=n+1
35     print ("\n第"+str(n)+"个: "+x['content'])
36     #加一个\n，用于换行。
37     answer = input('认识请敲Y，否则敲Enter: ')
38     #让用户输入自己是否认识。
39     if answer == 'Y':
40         #如果用户认识：
41         danci.append(x['content'])
42         words_knows.append(x)
43         #把这个单词，追加进列表words_knows。
44     else:
45         #否则
46         not_knows.append(x)
47         #把这个单词，追加进列表not_knows。
48 print ('\n在上述'+str(len(words['data']))+'个单词当中，有'+str(len(danci))+'个是你觉得自己认识的，它们是：')
49 print (danci)
50
51 # ----第3步-对于用户认识的单词，给选择题让用户做----
52 print ('现在我们来检测一下，你有没有真正掌握它们：')
53 wrong_words = []
54 right_num = 0
55 for y in words_knows:
56     print ('\n\n'+A:+y['definition_choices'][0]['definition'])
57     #我们改用A、B、C、D，不再用rank值，下同
58     print ('B:'+y['definition_choices'][1]['definition'])
59     print ('C:'+y['definition_choices'][2]['definition'])
60     print ('D:'+y['definition_choices'][3]['definition'])
61     xuanze = input('请选择单词\"'+y['content']+\"的正确翻译（填写数字即可）：')
62     dic = {'A':y['definition_choices'][0]['rank'], 'B':y['definition_choices'][1]
['rank'], 'C':y['definition_choices'][2]['rank'], 'D':y['definition_choices'][3]['rank']}

```

```

63     #我们创建一个字典，搭建起A、B、C、D和四个rank值的映射关系。
64     if dic[xuanze] == y['rank']:
65         #此时dic[xuanze]的内容，其实就是rank值，此时的代码含义已经和之前的版本相同了。
66         right_num += 1
67     else:
68         wrong_words.append(y)
69
70 # ----第4步-生成报告----
71 print ('现在，到了公布成绩的时刻：')
72 print ('在'+str(len(words['data']))+'个'+js_link['data'][bianhao-1][1]+'词汇当中，你认识其中'+str(len(danci))+'个，实际掌握'+str(right_num)+'个，错误'+str(len(wrong_words))+'个。')
73 #这是句蛮复杂的话，对照前面的代码和json文件你才能理解它。一个运行示例是：在50个高考词汇当中，你认识其中30个，实际掌握25个，错误5个。
74
75 save = input ('是否打印并保存你的错词集？填入Y或N： ')
76 #询问用户，是否要打印并保存错题集。
77 if save == 'Y':
78     #如果用户说是：
79     f = open('错题集.txt', 'a+')
80     #在当前目录下，创建一个错题集.txt的文档。
81     print ('你记错的单词有：')
82     f.write('你记错的单词有：\n')
83     #写入"你记错的单词有：\n"
84     m=0
85     for z in wrong_words:
86         #启动一个循环，循环的次数等于，用户的错词数：
87         m=m+1
88         print (z['content'])
89         #打印每一个错词。
90         f.write(str(m+1) +'. '+ z['content']+'\n')
91         #写入序号，写入错词。
92     print ('你不认识的单词有：')
93     f.write('你没记住的单词有：\n')
94     #写入"你没记住的单词有：\n"
95     s=0
96     for x in not_knows:
97         #启动一个循环，循环的次数等于，用户不认识的单词数。
98         print (x['content'])
99         #打印每一个不认识的单词。
100        f.write(str(s+1) +'. '+ x['content']+'\n')
101        #写入序号，写入用户不认识的词汇。

```

```

102     print ('错词和没记住的词已保存至当前文件目录下，下次见！')
103     #告诉用户，文件已经保存好。
104     #在网页版终端运行时，文件会被写在课程的服务器上，你看不到，但它的确已经存在。
105 else:
106     #如果用户不想保存：
107     print('下次见！')
108     #输出“下次见！”

```

第8关 cookies

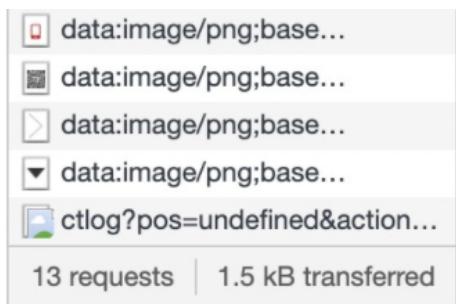
课后练习1：自制翻译器

- 题目解析：
 - 受制于篇幅，我们只介绍爬虫部分的逻辑。首先打开有道网页，同时打开检查模式，选择network。
 - 输入一个单词，然后从network里找到图中圈出的部分，点击Headers并下拉，找到From Data，其中的内容是我们需要的参数。

The screenshot shows a browser window with the Youdao Translate website open. The input field contains 'bye' and the output is '再见'. On the right, the developer tools Network tab is active, showing a list of network requests. One request, 'translate_o?smartresult=dict...', is highlighted with a red box. In the Headers section of the Network tab, there is a 'From Data' section containing the parameter 'translate_o?smartresult=dict...'. The Headers section also displays other details like Request URL, Method, Status Code, and Headers.

- 点击preview后发现需要找的翻译路径是'translateResult-0-0-tgt'，如下图。

Name	Headers	Preview	Response	Initiator	»
translate_o?smartresult=dict...	<pre> {translateResult: [[{tgt: "再见", src: "bye"}]] errorCode: 0 smartResult: {entries: ["", "n. 轮空 (参赛者无对"], "tgt": "再见", "src": "bye"}] type: "en2zh-CHS" </pre>				
rlog.php?_npid=fanyiweb&_n...					
ctllog?pos=undefined&action...					
ctllog?pos=undefined&action...					
data:image/png;base...					
data:image/png;base...					
data:image/png;base...					



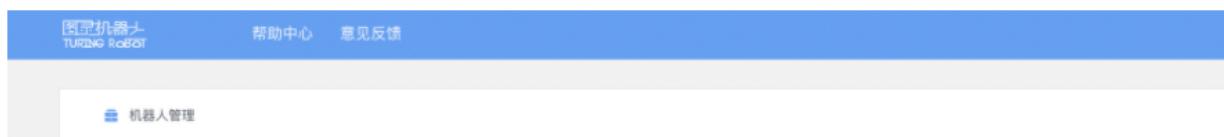
- 参考代码：

```
1 import requests, json
2 #调用了两个模块。requests负责上传和下载数据，json负责解析。
3
4
5 word = input('你想翻译什么呀？')
6 url='http://fanyi.youdao.com/translate?smartresult=dict&smartresult=rule'
7 #使用post需要一个链接。
8 data={'i': word,
9       'from': 'AUTO',
10      'to': 'AUTO',
11      'smartresult': 'dict',
12      'client': 'fanyideskweb',
13      'doctype': 'json',
14      'version': '2.1',
15      'keyfrom': 'fanyi.web',
16      'action': 'FY_BY_REALTIME',
17      'typoResult': 'false'}
18 #将需要post的内容，以字典的形式记录在data内。
19 r = requests.post(url,data)
20 #post需要输入两个参数，一个是刚才的链接，一个是data，返回的是一个Response对象。
21 answer=json.loads(r.text)
22 #你可以自己尝试print一下r.text的内容，然后再阅读下面的代码。
23 print ('翻译的结果是：'+answer['translateResult'][0][0]['tgt'])
```

课后练习2：图灵机器人

- 题目解析：

- 首先我们进入图灵机器人官网<http://www.tuling123.com/>，然后登陆注册一个账号，并创建机器人。创建好机器人之后我们会得到apikey。



The screenshot shows a list of five robots managed on the platform. Each robot has a small icon, a name, a daily chat count (all 0), an API key, and a version status (Free Edition). A red box highlights the API keys for all five robots.

机器人	今日聊天数量: 0	apikey: [redacted]	机器人版本: 免费版	设置	升级/续费
图灵机器人0	今日聊天数量: 0	apikey: 4b3a... [redacted]	机器人版本: 免费版	设置	升级/续费
图灵机器人2	今日聊天数量: 0	apikey: 49afe... [redacted]	机器人版本: 免费版	设置	升级/续费
图灵机器人3	今日聊天数量: 0	apikey: cd90... [redacted]	机器人版本: 免费版	设置	升级/续费
图灵机器人4	今日聊天数量: 0	apikey: 2c92f... [redacted]	机器人版本: 免费版	设置	升级/续费

创建机器人

- 接下来我们点开帮助中心的api文档，来查看图中的信息。

The screenshot shows the API V2.0 documentation page. The left sidebar has a '帮助中心' section with a search bar and a 'API V2.0接入文档' link highlighted with a red box. The right panel displays the API details:

- 接口地址:** `http://openapi.tuling123.com/openapi/api/v2`
- 请求方式:** HTTP POST
- 请求参数:** 请求参数格式为 json
- 请求示例:**

```
{
  "reqType": 0,
  "perception": {
    "inputText": {
      "text": "附近的酒店"
    },
    "inputImage": {
      "url": "imageUrl"
    },
    "selfInfo": {
      "location": {
        "city": "北京",
        "province": "北京",
        "street": "信息路"
      }
    },
    "userInfo": {
      "apiKey": "",
      "userId": ""
    }
}
```

- 从参数说明中可以看到，只有参数 perception 和 userinfo 才是必须的。
- 继续看下面的输出示例，就是我们请求之后的输出形式，text数据就是我们需要的数据。

输出参数

输出示例:

```
{
  "intent": [
    {
      "code": 10005,
      "intentName": "",
      "actionName": "",
      "parameters": {
        "nearby_place": "酒店"
      }
    }
  ],
  "results": [
    {
      "groupType": 1,
      "resultType": "url",
      "values": [
        {
          "url": "http://m.elong.com/hotel/0101/nlist/#inDate=2016-12-10&outDate=2016-12-11&keyw"
        }
      ]
    }
  ]
}
```

```
        "groupType": 1,
        "resultType": "text",
        "values": [
            {
                "text": "亲，已帮你找到相关酒店信息"
            }
        ]
    ]
```

- 参考代码：

```
1 import requests
2 import json
3
4 userid = str(1)
5 # 1 可以替换成任何长度小于32的字符串哦
6 apikey = str('A')
7 # 这里的A, 记得替换成你自己的apikey哦~
8
9 # 创建post函数
10 def robot(content):
11     # 图灵api
12     api = r'http://openapi.tuling123.com/openapi/api/v2'
13     # 创建post提交的数据
14     data = {
15         "perception": {
16             "inputText": {
17                 "text": content
18             }
19         },
20         "userInfo": {
21             "apiKey": apikey,
22             "userId": userid,
23         }
24     }
25     # 转化为json格式
26     jsondata = json.dumps(data)
27     # 发起post请求
28     response = requests.post(api, data = jsondata)
29     # 将返回的json数据解码
30     robot_res = json.loads(response.content)
31     # 提取对话数据
32     print(robot_res["results"][0]['values']['text'])
33
34 for x in range(10):
35     content = input("talk:")
36     # 输入对话内容
```

```

37     robot(content)
38     if x == 10:
39         break
40     # 十次之后就结束对话，数字可以改哦，你想几次就几次
41
42 #当然咯，你也可以加一些stopwords，只要说了这些词就可以终止聊天
43 while True:
44     content = input("talk:")
45     # 输入对话内容
46     robot(content)
47     if content == 'bye':
48         # 设置stopwords
49         break
50
51 #但是，我觉得吧，喜欢和聊天机器人玩的都是话痨，所以，可以最后加个死循环，如下：
52
53 # 创建对话死循环
54 while True:
55     # 输入对话内容
56     content = input("talk:")
57     robot(content)

```

课后练习3：nlpirc人工智能

- 题目解析：

- 第0步，打开网页与检查，输入关键字‘音乐剧’，点击网页中的“Word2vec”按钮（联想词汇）。
- 第1步，观察XHR中新增加的请求“getWord2Vec.do”，点击Headers，发现关键字‘音乐剧’是请求参数data里面参数值。
- 第2步，点击Perview，观察网页的返回数据，发现是json数据，需提取数据在请求中的‘v2wlist’列表中。
- 总体思路：使用post请求获取json数据（带data参数）-将json数据转换为字典-使用for循环遍历‘v2wlist’列表-分别提取近义词和相关度

- 参考代码：

```

1 import requests, json
2
3 # 请求并数据（带data参数）
4 url = 'http://ictclas.nlpir.org/nlpir/index6/getWord2Vec.do'
5 headers = {'user-agent':'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36'}
6

```

```

7 words = input('请输入你想查询的词汇：')
8 data = {'content':words}
9 res = requests.post(url,data=data,headers=headers)
10 data=res.text
11
12 # 把json数据转换为字典
13 data1=json.loads(data)
14
15 用for循环遍历'w2vlist'列表
16 f=0 # 变量f的作用是统计共有多少个近义词
17 for i in data1['w2vlist']: # i提取出来是字符串，如i='近义词.相关度'
18     f=f+1 # 近义词数量加1
19     word = i.split(',') # 以','号切割字符串，并转为列表，如word=['近义词','相关度']
20     print ('('+str(f)+')+word[0]+', 其相关度为'+word[1]) # 如word[0]是近义词，word[1]是相关
度

```

第9关 selenium

课后练习1：博客达人

- 题目解析：
 - 使用selenium实例化一个浏览器对象，再用其get()方法获取登录页。
 - 接下来我们定位到登录使用的用户名输入框和密码输入框输入账号信息。本习题我们可以不用注册账号，直接使用课程中提供的账号密码进行操作。
 - 点击《未来已来（三）——同九义何汝秀》文章标题，进入文章页面。注意从博客首页进入文章页面时，需要用到 find_element_by_partial_link_text 通过链接的部分文本获取超链接。
 - 定位到评论区的标签，模拟输入评论，点击发表评论。
- 参考代码：

```

1 import time
2 from selenium import webdriver
3
4 # 获取用户输入的评论内容
5 while True:
6     comment_content = input('请输入你想要的评论的内容，按回车提交：')
7     if comment_content == '':
8         print('&' * 5, '评论内容不允许为空', '&' * 5)
9     else:

```

```

10     break
11
12 driver = webdriver.Chrome() # 实例化浏览器对象
13 driver.get('https://wordpress-edu-3autumn.localprod.oc.forchange.cn/wp-login.php') # 访问页面
14 time.sleep(2)
15
16 # 定位到用户名输入框，输入用户名
17 login_name = driver.find_element_by_id('user_login')
18 login_name.send_keys('forchangeman')
19 time.sleep(1)
20 # 定位到密码输入框，输入密码
21 password = driver.find_element_by_id('user_pass')
22 password.send_keys('forchange123')
23 # 定位到登录按钮，并点击按钮
24 submit_btn = driver.find_element_by_id('wp-submit')
25 submit_btn.click()
26 time.sleep(2)
27
28 # 通过链接的部分文本定位到‘《未来已来（三）——同九义何汝秀》’这篇文章
29 # 获取到该文章对应的a标签（超链接），并点击链接进入文章详情页
30 article_link = driver.find_element_by_partial_link_text('同九义何汝秀')
31 article_link.click()
32
33 # 进入文章详情页，定位到该页面下编写评论的文本框，输入内容
34 comment_area = driver.find_element_by_id('comment')
35 comment_area.send_keys(comment_content)
36 time.sleep(2)
37 # 定位到提交按钮，点击该按钮提交评论
38 comment_submit = driver.find_element_by_id('submit')
39 comment_submit.click()
40
41 # 评论成功10秒后关闭浏览器
42 time.sleep(10)
43 driver.close()
44 print('#' * 6, '评论成功，浏览器已关闭', '#' * 6)

```

课后练习2：Python之禅

- 题目解析：
 - 使用selenium实例化一个浏览器对象，再用get()方法获取“你好，蜘蛛侠”的网页。

- 找到输入框的标签，输入你喜欢的老师和助教，点击提交。
- 网页跳转到Python之禅后，提取Elements的源代码里Python之禅文本的标签。
 - 方法一：使用selenium库自身提供的标签定位方法定位标签。
 - 方法二：使用selenium库中浏览器对象的page_source属性获取经过渲染的网页源代码，然后使用BeautifulSoup库的标签定位方法去定位。

- 参考代码：

```

1  # 方法一: selenium
2  from selenium import webdriver # 从selenium库中调用webdriver模块
3  import time
4
5  driver = webdriver.Chrome() # 声明浏览器对象
6  driver.get('https://localprod.pandateacher.com/python-manuscript/hello-spiderman/') #
访问页面
7  time.sleep(2) # 暂停两秒，等待浏览器缓冲
8
9  teacher = driver.find_element_by_id('teacher') # 找到【请输入你喜欢的老师】下面的输入框位置
10 teacher.send_keys('必须是吴枫呀') # 输入文字
11 assistant = driver.find_element_by_name('assistant') # 找到【请输入你喜欢的助教】下面的输入框
位置
12 assistant.send_keys('都喜欢') # 输入文字
13 button = driver.find_element_by_class_name('sub') # 找到【提交】按钮
14 button.click() # 点击【提交】按钮
15 time.sleep(1)
16
17 contents = driver.find_elements_by_class_name('content') # 定位到Python之禅所在的标签
18 for content in contents:
19     title = content.find_element_by_tag_name('h1').text # 提取标题
20     chan = content.find_element_by_tag_name('p').text # 提取正文
21     print(title + '\n' + chan + '\n') # 打印标题与正文
22 driver.close()

```

```

1  # 方法二: selenium+BeautifulSoup
2  from selenium import webdriver # 从selenium库总调用webdriver模块
3  import time
4  from bs4 import BeautifulSoup
5
6  driver = webdriver.Chrome() # 声明浏览器对象
7  driver.get('https://localprod.pandateacher.com/python-manuscript/hello-spiderman/') #
访问页面
8  time.sleep(2) # 暂停两秒，等待浏览器缓冲

```

```

9
10 teacher = driver.find_element_by_id('teacher') # 定位到【请输入你喜欢的老师】下面的输入框位置
11 teacher.send_keys('必须是吴枫呀') # 输入文字
12 assistant = driver.find_element_by_name('assistant') # 定位到【请输入你喜欢的助教】下面的输入
    框位置
13 assistant.send_keys('都喜欢') # 输入文字
14 button = driver.find_element_by_class_name('sub') # 定位到【提交】按钮
15 button.click() # 点击【提交】按钮
16 time.sleep(1) # 等待一秒
17
18 pageSource = driver.page_source # 获取页面信息
19 soup = BeautifulSoup(pageSource, 'html.parser') # 使用bs解析网页
20 contents = soup.find_all(class_="content") # 找到源代码Python之禅中文版和英文版所在的元素
21 for content in contents: # 遍历列表
22     title = content.find('h1').text # 提取标题
23     chan = content.find('p').text.replace(' ', '') # 提取Python之禅的正文，并且去掉文字前面
        的所有空格
24     print(title + chan + '\n') # 打印Python之禅的标题与正文
25 driver.close()

```

第10关 定时与邮件

课后练习：周末吃什么

- 题目解析：
 - 爬虫：按照爬虫第3关爬取下厨房的方法进行序号、菜名、链接和原料的爬取。
 - 发送邮件：使用email和smtplib两个库按照基础语法第17关的方法进行邮件发送。
 - 设置定时任务：使用schedule库设置定时任务，注意：可以不把发送邮件的时间设置为固定时间。
- 参考代码：

```

1 import requests
2 import smtplib
3 import schedule
4 import time
5 from bs4 import BeautifulSoup
6 from email.mime.text import MIMEText
7 from email.header import Header
8

```

```

9 account = input('请输入你的邮箱: ')
10 password = input('请输入你的密码: ')
11 receiver = input('请输入收件人的邮箱: ')
12
13 def recipe_spider():
14     headers={'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 Safari/537.36'}
15     res_foods = requests.get('http://www.xiachufang.com/explore/',headers=headers)
16     bs_foods = BeautifulSoup(res_foods.text,'html.parser')
17     list_foods = bs_foods.find_all('div',class_='info pure-u')
18     list_all = ''
19     num=0
20     for food in list_foods:
21         num=num+1
22         tag_a = food.find('a')
23         name = tag_a.text.strip()
24         url = 'http://www.xiachufang.com'+tag_a['href']
25         tag_p = food.find('p',class_='ing ellipsis')
26         ingredients = tag_p.text.strip()
27         food_info = '''
28             序号: %s
29             菜名: %s
30             链接: %s
31             原料: %s
32             '''%(num,name,url,ingredients)
33         list_all=list_all+food_info
34     return(list_all)
35
36 def send_email(list_all):
37     global account,password,receiver
38     mailhost='smtp.qq.com'
39     qqmail = smtplib.SMTP()
40     qqmail.connect(mailhost,25)
41     qqmail.login(account,password)
42     content= '亲爱的，本周的热门菜谱如下'+list_all
43     message = MIMEText(content, 'plain', 'utf-8')
44     subject = '周末吃个啥'
45     message['Subject'] = Header(subject, 'utf-8')
46     try:
47         qqmail.sendmail(account, receiver, message.as_string())
48         print ('邮件发送成功')

```

```

49     except:
50         print ('邮件发送失败')
51     qqmail.quit()
52
53 def job():
54     print('开始一次任务')
55     list_all = recipe_spider()
56     send_email(list_all)
57     print('任务完成')
58
59 schedule.every().tuesday.at("11:27").do(job) #部署每周三的13: 15执行函数的任务
60 while True:
61     schedule.run_pending()
62     time.sleep(1)

```

第11关 协程

课后练习：煲剧达人

- 题目解析：
 - 该练习和第3关课后练习豆瓣爬虫比较相似，面对多个网址，分析思路也是一样，翻页查看网址规律，该练习第一页的网址不同于其他9页，需要单独写，其他九页的网址，只有一个数字不同，可以通过for循环来控制。
 - 找标签先使用小箭头定位，然后测试爬取。测试成功之后，再加上gevent库相关代码，进行多爬虫爬取数据。
 - 不同的是，该网址中的电影，缺失内容（导演、主演、推荐语）的数量比较多，所以在取的时候，直接把所有都取到即可。找到标签<div class_="mov_con">下的所有<p>标签，使用加号+进行拼接。
 - 注意：存储时可以使用utf-8-sig编码防止csv在Excel中打开出现乱码，uft-8-sig带有签名（Bom），Bom一般出现在文本文件头部，Unicode编码标准中用于标识文件是采用哪种格式的编码。Excel 在读取 csv 文件的时候是通过读取文件头上的 BOM 来识别编码的，如果文件头无 BOM 信息，则默认按照 Unicode 编码读取，如果使用 utf-8 编码生成 csv 文件（没有生成BOM信息），Excel 就会自动按照 Unicode 编码读取，就会出现乱码问题了。
- 参考代码：

```

1 from gevent import monkey
2 monkey.patch_all()
3 import gevent, requests, bs4, csv

```

```

4  from gevent.queue import Queue
5
6  work = Queue()
7  url_1 = 'http://www.mtime.com/top/tv/top100/'
8  work.put_nowait(url_1)
9
10 url_2 = 'http://www.mtime.com/top/tv/top100/index-{page}.html'
11 for x in range(2,11):
12     real_url = url_2.format(page=x)
13     work.put_nowait(real_url)
14
15 def crawler():
16     headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36'}
17     while not work.empty():
18         url = work.get_nowait()
19         res = requests.get(url,headers=headers)
20         bs_res = bs4.BeautifulSoup(res.text, 'html.parser')
21         datas = bs_res.find_all('div', class_="mov_con")
22         for data in datas:
23             TV_title = data.find('a').text
24             data = data.find_all('p')
25             TV_data =''
26             for i in data:
27                 TV_data =TV_data + i.text + ' '
28             writer.writerow([TV_title,TV_data])
29             print([TV_title,TV_data])
30
31 csv_file = open('timetop.csv', 'w', newline='', encoding='utf-8-sig')
32 writer = csv.writer(csv_file)
33
34 task_list = []
35 for x in range(3):
36     task = gevent.spawn(crawler)
37     task_list.append(task)
38 gevent.joinall(task_list)
39 csv_file.close()

```

第13关 Scrapy框架

课后练习：当当图书馆爬虫

- 题目解析：

- 分析网址：翻页查看网址规律，该网站的网址规律比较统一，只有一个数字不同，可以使用for循环控制变量来给网址赋值该数字；
- 定位标签：借助小箭头进行定位，查找标签所有数据都在标签`<ul class="bang_list clearfix bang_list_mode">`下，详细数据标签如下：
 - 书名在`<div class="name">`
 - 作者在`<div class="publisher_info">`
 - 价格在``
- 文件编写：
 - 在`items.py`文件里定义item；
 - 在`bestsellers.py`文件里编写spider；
 - 在`settings.py`文件里修改设置；
 - 在`main.py`文件里写入运行scrapy的代码，并点击运行。
- 注意：课堂系统里并没有显示Scrapy的项目里的所有文件，只显示了需要修改或编辑的文件。

- 参考代码：

```
1 # items.py文件参考代码
2 import scrapy
3 class DangdangItem(scrapy.Item):
4     name = scrapy.Field()
5     author = scrapy.Field()
6     price = scrapy.Field()
```

```
1 # bestsellers.py文件里spider参考代码
2 import scrapy
3 import bs4
4 from ..items import DangdangItem
5
6 class DangdangSpider(scrapy.Spider):
7     name = 'dangdang'
8     allowed_domains = ['http://bang.dangdang.com']
9     start_urls = []
10    for x in range(1, 4):
11        url = 'http://bang.dangdang.com/books/bestsellers/01.00.00.00.00.00-
12        year-2018-0-1-' + str(x)
13        start_urls.append(url)
```

```

13     def parse(self, response):
14         soup = bs4.BeautifulSoup(response.text, 'html.parser')
15         elements = soup.find('ul', class_="bang_list clearfix
16             bang_list_mode").find_all('li')
17         for element in elements:
18             item = DangdangItem()
19             item['name'] = element.find('div', class_="name").find('a')['title']
20             item['author'] = element.find('div', class_="publisher_info").text
21             item['price'] = element.find('div', class_="price").find('span',
22                 class_="price_n").text
23             yield item

```

```

1 # settings.py文件参考代码 (添加请求头, 修改ROBOTSTXT_OBEY的值)
2 USER_AGENT = 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/71.0.3578.98 Safari/537.36'
3 # Obey robots.txt rules
4 ROBOTSTXT_OBEY = True

```

```

1 # main.py文件参考代码
2 from scrapy import cmdline
3 cmdline.execute(['scrapy', 'crawl', 'dangdang'])

```

第14关 Scrapy实操

课后练习：豆瓣图书短评

- 题目解析：
 - 分析网址：翻页查看网址规律，前面已经分析过了，规律就是，只有一个数字不同，可以使用for循环控制变量来给网址赋值该数字；
 - 定位标签：借助小箭头进行定位，查找短评所有数据都在标签<div class="comment">下，详细数据标签如下：
 - 发表人在<div class="comment">下的第1个（索引值）<a>标签
 - 内容在
 - 文件编写：
 - 在items.py文件里定义item；
 - 在comments.py文件里编写spider；

- 在settings.py文件里修改设置；
 - 在pipelines.py文件里写入存储成Excel的代码；
 - 在main.py文件里写入运行scrapy的代码，并点击运行。
- 注意：
 - 如果你的代码成功运行，在【文件】里的root文件夹下的douban文件夹里，你就可以找到你创建的存储数据的Excel文件，将其下载到本地，你就能用Excel打开查看。
 - scrapy的项目和spider的文件，已经在练习系统里提前为你创建好了。
- 参考代码：

```

1 # items.py文件参考代码
2 import scrapy
3 class DoubantopItem(scrapy.Item):
4     book_name = scrapy.Field()
5     ID_name = scrapy.Field()
6     comment = scrapy.Field()

```

```

1 # comments.py文件里编写spider
2 import scrapy,bs4
3 from ..items import DoubantopItem
4
5 class DoubantopSpider(scrapy.Spider):
6     name = 'doubantop'
7     allowed_domains = ['https://book.douban.com']
8     start_urls = []
9     for x in range(2):
10         url = 'https://book.douban.com/top250?start=' + str(x * 25)
11         start_urls.append(url)
12
13     def parse(self, response):
14         soup = bs4.BeautifulSoup(response.text,'html.parser')
15         datas = soup.find_all('tr',class_='item')
16         for data in datas:
17             book_url = data.find_all('a')[1]['href']
18             comment_url = book_url+'comments/'
19             yield scrapy.Request(comment_url,callback=self.parse_comment)
20
21     def parse_comment(self,response):
22         soup = bs4.BeautifulSoup(response.text,'html.parser')
23         book_name = soup.find('div',id='content').text.split()[0]
24         datas = soup.find_all('div',class_='comment')
25         for data in datas:

```

```
26         item = DoubantopItem()
27
28         item['book_name'] = book_name
29
30         item['ID_name'] = data.find_all('a')[1].text
31         item['comment'] = data.find('span', class_='short').text
32
33         yield item
```

```
1 # settings.py文件参考代码 (添加请求头, 修改ROBOTSTXT_OBEY的值, 取消6~8行的注释)
2 USER_AGENT = 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/71.0.3578.98 Safari/537.36'
3 # Obey robots.txt rules
4 ROBOTSTXT_OBEY = True
5 ITEM_PIPELINES = {
6     'doubantop.pipelines.DoubantopPipeline': 300,
7 }
```

```
1 # pipelines.py文件参考代码
2 import openpyxl
3 class DoubantopPipeline(object):
4     def __init__(self):
5         self.wb = openpyxl.Workbook()
6         self.ws = self.wb.active
7         self.ws.append(['书名', 'ID名', '短评'])
8
9     def process_item(self, item, spider):
10        line = [item['book_name'], item['ID_name'], item['comment']]
11        self.ws.append(line)
12        return item
13
14    def close_spider(self, spider):
15        self.wb.save('book.xlsx')
16        self.wb.close()
```

```
1 # main.py文件参考代码
2 from scrapy import cmdline
3 cmdline.execute(['scrapy', 'crawl', 'doubantop'])
```

使用腾讯文档创作和发布 多人协作 / 远程办公 / 信息收集

我也要用 >