

1+X Python程序开发线上师资培训讲义

一、1+X Python程序开发标准解读及考核标准

1.标准解读

内容请参见标准解读PDF文档。

2.考核说明

2.1 平时考核 (权重为60%)

1) 不定期点名 (考勤)

2) 每天需要提交作业 (作业在当天课程结束后安排, 作业提交QQ群里的张老师 (846213108)), 提交截止时间为第二天9点前, 过期视为没交, 分数为零。

2.2 结业考试 (权重为40%)

结业考试需按项目需求完成对应的项目, 提交的最终时间为下周二 (6月2日中午12点前), 过期成绩将视该项目成绩为零, 成绩将在一周后出。

3.注意事项

1) 各位老师可以QQ群 (883411516) 中进行Python技术交流, 群里老师可以协助解决问题, 我公司的技术工程师也会不定期帮助老师们解决相关技术问题。(也即是我们为全国中高职及本科院校的老师提供了一个Python技术交流群)。

2) QQ群中必须实名: 院校名-姓名, 钉钉群中请实名。

3) 作业必须是本人在QQ群中亲自提交给张老师, 任何人不得转交。

4) 没有参加试点院校的院校, 可以参与试点院校申报, 我们会提供申报手册。申报过程中如有问题, 可以直接联系我们的区域负责人。

5) 老师们请不要随意发起签到, 签到是由上课老师发起的。

6) 大家不要在群里非技术或教学无关的信息, 否则会被踢出群的。

4.预告

今年分别举办至少两场暑期线下师资培训, 主要讲中高级的内容, 初级也会涉及, 小白也没问题。第一场暂定在云南昆明, 时间为7月25日-7月31日 (以具体的通知为准), 欢迎各位老师参加。

二、课程内容

2.1 Python应用编程基础

课前准备

- Python的由来

1989年 由Guido van Rossum开发, 1991年2月正式公开发布了Python的第一个版本



- Python的应用领域

中慧集团1+X Python程序开发



- Python的版本

最新版本3.9.5, 当前使用版本Python3.7.9

- Python的特点



VB.NET之父ALAN COOPER
(阿兰·库珀)
头顶茂密指数: ☆☆☆☆☆



C++之父BJARNE STROUSTRUP
(本贾尼·斯特劳斯普)
头顶茂密指数: ★☆☆☆☆



JAVA之父JAMES GOSLING
(詹姆斯·高林斯)
头顶茂密指数: ★★☆☆☆



PHP之父RASMUS LERDORF
(拉斯穆斯·勒多夫)
头顶茂密指数: ★★☆☆☆



Brendan Eich
1961 -
JavaScript

JAVASCRIPT之父BRENDAN EICH
(布伦丹·艾奇)
头顶茂密指数: ★★★★★



PYTHON之父GUIDO VAN ROSSUM
(吉多·范罗苏姆)
头顶茂密指数: ★★★★★

- Python的安装和运行

Python 下载地址 <https://www.python.org/>

Pycharm 下载地址 <https://www.jetbrains.com/pycharm/download/#section=linux>

2.1.1 学生信息管理系统

2.1.1.1 项目需求

学生信息管理系统

功能菜单

- 1 添加学生信息
- 2 查看所有学生信息
- 3 按姓名修改学生的成绩
- 4 按姓名删除学生信息
- 0 退出系统

说明：通过数字键选择菜单

1)写一个程序，任意输入n个学生的信息,形成字典后存于列表中。

学生的信息包括: 姓名(字符串), 年龄(整数), 成绩(整数); 循环输入学生信息, 直到输入学生姓名为空时结束输入, 最后形成字典列表: `L = [{'name':'xiaozhang', 'age':20, 'score': 100}, {'name':'xiaoli', 'age':21, 'score': 98}, {'name':'xiaowang', 'age':19, 'score': 89}, ...]`

2) 将得到的学生信息进行格式化打印:

name	age	score
xiaozhang	20	100
xiaoli	21	98
xiaowang	19	89

3)可以修改和删除学生成绩

4)实现打印菜单，并根据提示选择对应的操作

2.1.1.2 项目步骤

- 1.通过死循环进行学生信息（姓名、年龄、成绩）的录入，当输入学生姓名为空时，停止输入。
- 2.用字典封装一个学生的信息，并将字典逐个个追加到列表里
- 3.分析表格表头的规律，实现打印表头

4. 循环获取存储的学生信息，然后进行格式化打印学生信息
5. 实现修改和删除学生成绩的功能
6. 输出菜单，根据提示通过条件语句控制要执行的功能

2.1.1.3 项目代码

```
def show_menu():
    print("""
┌-----学生信息管理系统-----┐
│
│ ===== 功能菜单 ===== │
│
│ 1 添加学生信息              │
│ 2 查看所有学生信息          │
│ 3 按姓名修改学生的成绩      │
│ 4 按姓名删除学生信息        │
│ 0 退出系统                  │
│ =====                    │
│ 说明：通过数字键选择菜单    │
└-----┘
""")
)

def input_student():
    stufos = []
    while True:
        name = input("请输入学生姓名: ")
        if not name:
            break
        age = int(input("请输入学生年龄: "))
        score = int(input("请输入学生成绩: "))
        stu_dic = {'name': name, 'age': age, 'score': score}
        stufos.append(stu_dic)
    return stufos

def output_student(stufos):
    title1 = '+' + '-' * 15 + '+' + '-' * 5 + '+' + '-' * 7 + '+'
    title2 = '|%s|%s|%s|' % ('name'.center(15), 'age'.center(5), 'score'.center(7))
    print(title1)
    print(title2)
    print(title1)
    for stu in stufos:
        print('|%s|%s|%s|' % (
            stu['name'].center(15),
            str(stu['age']).center(5),
            str(stu['score']).center(7)))
    print(title1)

def modify_score(stufos):
```

```
name = input("请输入你要修改的学生名字:")
for stu in stuinfos:
    if name == stu['name']:
        score = int(input("请输入你要修改的学生成绩:"))
        stu['score'] = score
        print("成功修改%s的成绩为%s" % (name, score))
        return
print("没有姓名为%s的学生" % name)

def delete_student(stuinfos):
    name = input("请输入你要删除的学生姓名:")
    for stu in stuinfos:
        if name == stu['name']:
            stuinfos.remove(stu)
            print("删除%s成功" % name)
            return
    print("删除失败, 没有姓名为%s的学生" % name)

def main():
    students = []
    while True:
        show_menu()
        op = input("请选择你要做的操作:")
        if op == '1':
            students.extend(input_student())
        elif op == '2':
            output_student(students)
        elif op == '3':
            modify_score(students)
        elif op == '4':
            delete_student(students)
        elif op == 'q':
            return
        else:
            print("输入错误")
            input('请输入回车回主菜单.....')

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

2.1.1.4 运行结果

name	age	score
zhangsan	20	98
lisi	21	97
xiaowang	22	90

2.1.1.5 项目分析

此项目是对Python基础知识的综合运用，主要包括常用数据类型、循环语句、if语句以及函数，重点是要灵活运用字符串、列表、字典等数据类型、理清if、while和for语句的执行逻辑以及函数的用法。

2.1.2 房源信息分析

2.1.2.1 项目需求

中慧集团1+X Python程序设计(网内用)

房源信息分析系统

功能菜单

- 1 区域分析
- 2 商圈分析
- 3 楼盘分析
- 4 朝向分析
- 5 电梯分析
- 6 楼层分析
- 7 装修分析
- 8 年代分析
- 0 退出系统

说明：通过数字键选择菜单

从文件中读取房源数据，对数据进行分组和聚合计算，然后从区域、商圈、楼盘、朝向、电梯、楼层、装修、年代、价格等维度对房屋进行分析，要求打印菜单，并根据菜单对提示选择同的分析指标

2.1.2.2 项目步骤

- 1.从文件中读取房源数据，并对数据按特征整理存储
- 2.实现分组和聚合计算功能，并得到相应的结果数据
- 3.将上一步得到的结果数据进行格式化打印输出
- 4.制作打印菜单，并根据菜单执行不同的分析功能

2.1.2.3 项目代码

```
class HouseAnalysis:

    def __init__(self, file_name):
        self.file_name = file_name
```

```
self.district, self.region, self.village, self.direction, self.elevator, self.floor, self.layout, self.decoration, self.year, self.size, self.price, self.perprice = [], [], [], [], [], [], [], [], [], [], [], []
```

```
def read_data(self):  
    with open(self.file_name, encoding='utf-8') as f:  
        for line in f:  
            row = line.strip()  
            if not row.startswith('district'):  
                row_split = row.split(',')  
                self.district.append(row_split[0])  
                self.region.append(row_split[1])  
                self.village.append(row_split[2])  
                self.direction.append(row_split[3])  
                self.elevator.append(row_split[4])  
                self.floor.append(row_split[5])  
                self.layout.append(row_split[6])  
                self.decoration.append(row_split[7])  
                self.year.append(row_split[8])  
                self.size.append(row_split[9])  
                self.price.append(row_split[10])  
                self.perprice.append(row_split[11])
```

```
def group_and_agg(self, group_field, agg_field, aggfunc):  
    dic = {}  
    for a, b in zip(group_field, agg_field):  
        if a not in dic:  
            dic[a] = []  
        dic[a].append(float(b))  
    dic_agg = {}  
    for key, value in dic.items():  
        dic_agg[key] = aggfunc(value)  
    return dic_agg
```

```
def average(self, data):  
    return sum(data) / len(data)
```

```
def get_chinese_num(self, ch):  
    cont = 0  
    for c in ch:  
        if '\u4e00' <= c <= '\u9fa5' or c == chr(9632):  
            cont += 1  
    return cont
```

```
def print_pretty(self, line, length):  
    chinese_num = self.get_chinese_num(line)  
    space_num = length - chinese_num - len(line) - 2  
    print("|%s|" % (line, space_num * ' '))
```

```
def print_label(self, label, length):  
    chinese_num = self.get_chinese_num(label)  
    print_length = length - chinese_num - 2  
    print("|%s|" % label.center(print_length))
```

```
print('|' + '+' * (length - 2) + '|')
```

```
def plot_bar(self, group_data, label, length):
    print('=' * length)
    self.print_label(label, length)
    sorted_data = sorted(group_data.items(), key=lambda x: x[1], reverse=True)[:15]
    divid = float(sorted_data[0][1]) / 10
    for d in sorted_data:
        bar_length = int(d[1] / divid)
        line = '%s %s %d' % (bar_length * chr(9632), d[0], d[1])
        self.print_pretty(line, length)
    print('=' * length)
```

```
def group_agg_bar(self, group_field, agg_field, agg_func, label, length=50):
    group_data = self.group_and_agg(group_field, agg_field, agg_func)
    self.plot_bar(group_data, label, length)
```

```
def show_menu(self):
    print("""
```

```
┌-----房源信息分析系统-----┐
│
│ ===== 功能菜单 ===== │
│
│ 1 区域分析                    │
│ 2 商圈分析                    │
│ 3 楼盘分析                    │
│ 4 朝向分析                    │
│ 5 电梯分析                    │
│ 6 楼层分析                    │
│ 7 装修分析                    │
│ 8 年代分析                    │
│ 0 退出系统                    │
│
│ =====                    │
│ 说明: 通过数字键选择菜单    │
│
└-----┘

""")
```

```
def analysis(self, item, label):
    self.group_agg_bar(item, self.price, self.average, '%s总价条形图(万元)' % label)
    self.group_agg_bar(item, self.perprice, self.average, '%s均价条形图(元)' % label)
    self.group_agg_bar(item, self.perprice, len, '%s住房数量条形图(套)' % label)
```

```
def run(self):
    self.read_data()
    while True:
        self.show_menu()
        op = input("请输入你要分析的项目:")
        if op == '1':
            self.analysis(self.district, '各区域')
        elif op == '2':
            self.analysis(self.region, '各商圈')
        elif op == '3':
            self.analysis(self.village, '各楼盘')
```

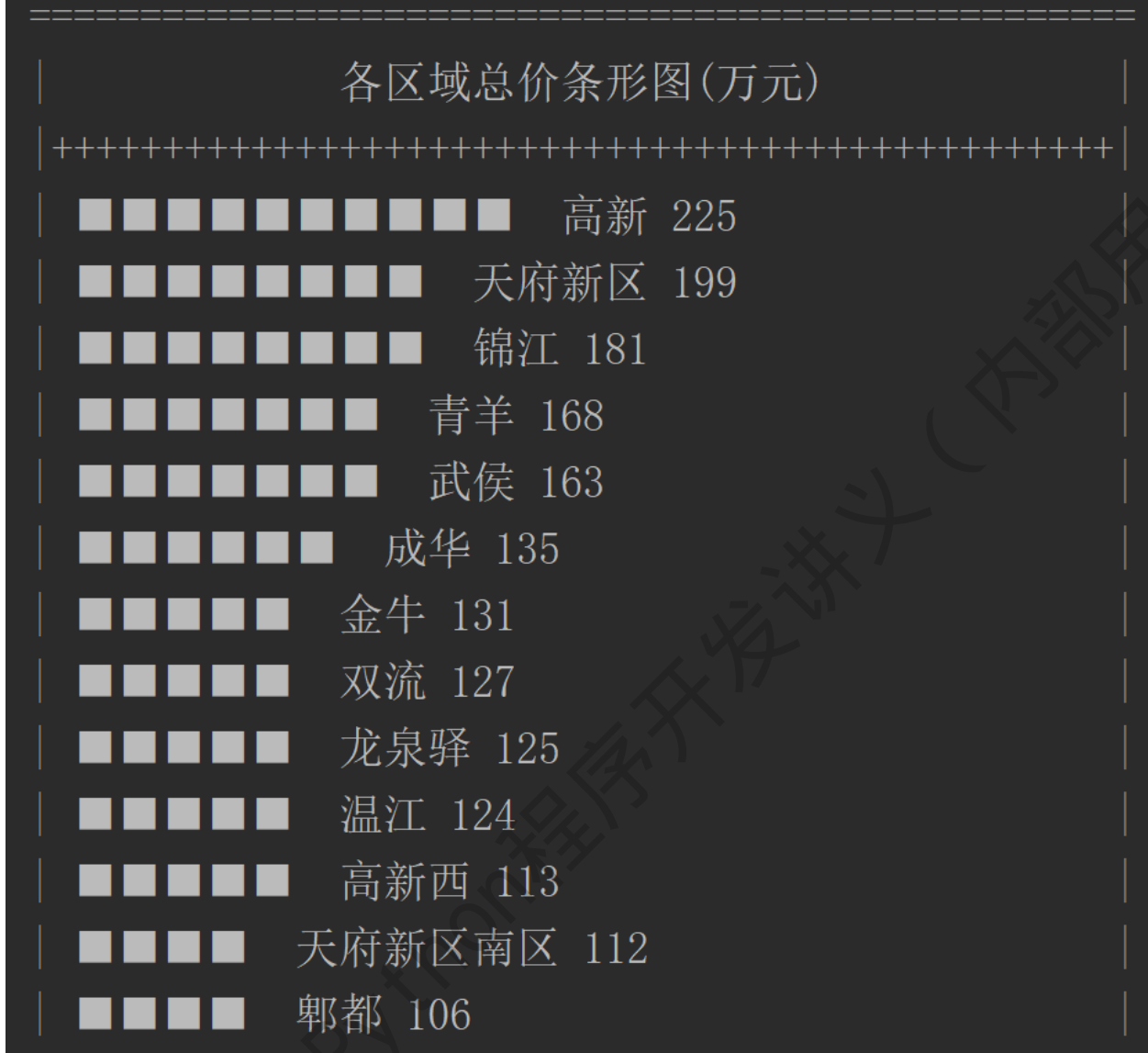
```
elif op == '4':
    self.analysis(self.direction, '各朝向')
elif op == '5':
    self.analysis(self.elevator, '有无电梯')
elif op == '6':
    self.analysis(self.floor, '楼层')
elif op == '7':
    self.analysis(self.decoration, '装修情况')
elif op == '8':
    self.analysis(self.year, '年代')
elif op == '0':
    return
else:
    print("你输入有误, 请重新输入")
q = input("是否回主菜单 (y代表是, 其它代表退出) ")
if q != 'y':
    return

if __name__ == '__main__':
    ha = HouseAnalysis('house.csv')
    ha.run()
```

2.1.2.4 运行结果

中慧集团1+X Python程序开发讲义(内部用)

请输入你要分析的项目: /



2.1.2.5 项目分析

该项目主要是基于面向对象编程实现的房源数据各维度的分析，重点需要掌握文件操作，通过字典、列表等实现数据的分组聚合，灵活使用字符串方法进行格式化打印；此外还需要慢慢体会函数参数的灵活运用，要树立面向对象的思维方式。

2.1.3 Python主要知识点讲解

2.1.3.1 Python基本语法

1. Python快速入门

1) Python中的数字

- 整型

用int表示，是不带小数部分的数字

例如 1 3 -5

- **浮点型**

用float表示,是带有小数部分的数字

例如 3.14 3.1 3.0 14 .14

- **布尔**

用bool表示,用来表示真和假两个状态的类型

True 表示真(条件满足或成立)

Flase表示假(条件不满足或不成立)

- **数字的算术运算**

加(+)、减(-)、乘(*)、除(/)

- **类型转换函数**

int(obj) 用于将一个对象转化为整型数

foat(obj) 用于将一个对象转换为浮点数

bool(obj) 用于将一个对象转换为布尔类型数

2) Python中的变量

- **什么是变量**

用来绑定数据对象的标识符

- **标识符的命名规则**

1.变量名必须为字母或下划线开头,后跟字母或下划线或数字

2.不能使用python的关键字命名, if、else、for、return、None、False等

- **赋值语句**

作用

用于创建变量并将变量绑定(或关联)在一个对象上

语法

变量名 = 表达式

赋值语句执行过程

首先把等号右侧表达式的值计算出来,然后在内存中寻找一个位置把值存储进去,然后创建变量并指向这个内存地址

```
a = 5  
b = 10
```

3) 基本输入输出函数

- **基本输入函数 input**

作用

从键盘上读取一个字符串(末尾的换行符(回车)会被删除)

格式

```
input('提示字符串')
```

说明

返回输入的字符串

- **基本输出函数print**

作用

将一系列的值以字符串形式输出到标准输出设备上，默认为终端。

格式

```
print(value, ..., sep=' ', end='\n')
```

4) 字符串

- **字符串的作用**

用来记录文本（文字）信息，比如姓名，电话，住址等。

- **字符串的表示方法**

在非注释中，凡是用引号括起来的部分都是字符串，可以是单引号(') 双引号(") 三单引号(''') 三双引号('\"'')。

- **创建字符串**

1) 直接用引号括起来

例如 'hello' "hello"

2) 通过str函数

例如 str(5)

- **字符串的算术运算**

+ += * *=

+ 运算符用于字符串的拼接

* 运算符用于生成重复的字符串

- **Python中常用的序列函数**

len(seq) 返回序列的长度

- **字符串的索引**

索引语法

```
字符串[下标]
```

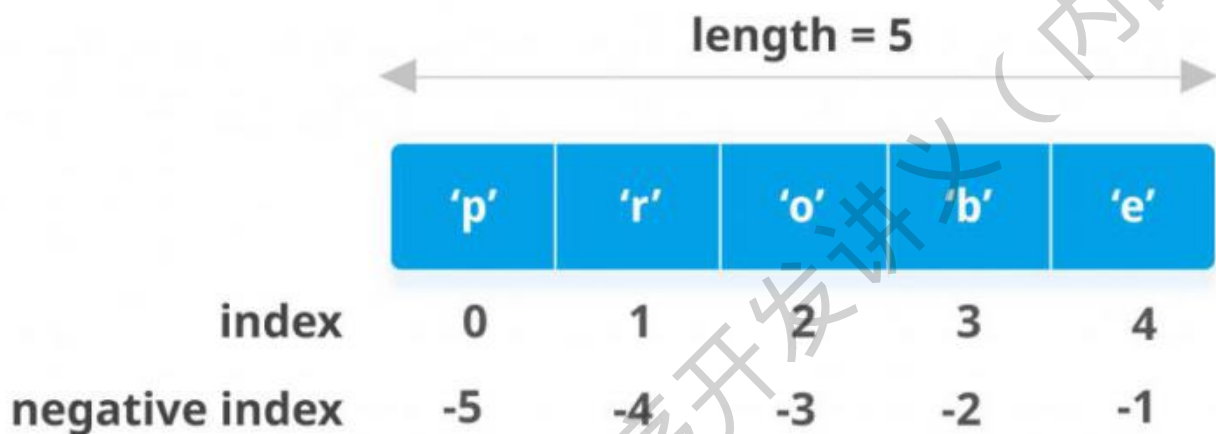
说明

python 序列都可以用索引(index) 来访问序列中的对象(元素)。

字符串是不可变的序列。

示例

```
s = 'probe'
print(s[0]) # p
print(s[1]) # r
print(s[4]) # e
print(s[-1]) # e
```



• 字符串的方法

方法	作用
S.center(width[,fill])	将原字符串居中，左右默认填充空格
S.strip()	返回去掉左右空白字符的字符串
S.split(sep=None)	将字符串，使用sep作用为分隔符分割S字符串，返回分割后的字符串的列表，当不给定参数时，用空白字符作为分隔符进行分割
S.join(iterable)	用可迭代对象中的字符串，通过S拼接成一个新的字符串

• 字符串的格式化表达

运算符

%

作用

用来将对象填充到字符串中，方便字符串的拼接和显示，简化Python代码

例 '姓名: %s, 年龄: %s, 拥有%.2f元钱' % ('张三', 20, 9998.654)

2.Python程序控制结构

1) if语句

作用

让程序根据条件选择性的执行某条语句或某些语句

语法

```
if 真值表达式1:
```

```
    语句块1
```

```
elif 真值表达式2:
```

```
    语句块2
```

```
elif 真值表达式3:
```

```
    语句块3
```

```
...
```

```
else:
```

```
    语句块4
```

2) while语句

作用

根据一定条件，重复的执行一条语句或多条语句

语法

```
while 真值表达式:
```

```
    语句块
```

```
else:
```

```
    语句块
```

示例

用while语句打印 1 ~ 20 的整数(包含20)

break 语句

用于循环语句(while,for语句)中，用来终止当前循环语句的执行

死循环

死循环是指循环条件一直成立的循环;死循环通常用break语句来终止循环

3) for语句

作用

用来遍历可迭代对象的数据元素

语法

```
for 变量列表 in 可迭代对象:
```

```
    语句块1
```

```
else:
```

```
    语句块2
```

range()函数

语法

range(stop) 从零开始，每次生成一个整数后加1操作，直到stop为止（不包含stop）

range(start, stop[, step]) 从start开始，每次生成一个整数后移动step，直到stop为止(不包含stop,且step可以是负整数)

示例

输入一个整数用begin绑定，再输入一个整数用end绑定，打印出从begin~end(包含end)的所有偶数

3.Python中的列表

列表和字符串一样，也是Python中的一种序列，列表中可以装任何类型的对象。

- 列表的创建

通过[]创建

例:

```
[] # 空列表
```

```
[1, 2, 3, 4] # 4个整数
```

```
[1, 'a', 3.5, True, [1, 2, 3]] # 列表内可以是任何类型的对象
```

通过构造函数list()创建

例:

```
list() # 空列表
```

```
list('abcd') # 通过字符串创建一个列表
```

- 列表的索引

索引规则同字符串，不同之处在于可以索引赋值

- 列表的常用方法

方法	说明
L.append(x)	向列表中追加单个元素
L.extend(lst)	向列表中追加另一个列表
L.remove(x)	从列表中删除第一次出现在列表中的值

4.Python字典

字典和列表不同的地方在于，列表是有序的，而字典是无序的，并且字典是通过键来存取，字典中的数据以键(key)-值(value)对进行存储，且键不能重复。

- 创建字典

例：

```
d = {}  
  
d = {"a": 1, "b": 2}
```

- 字典取值

语法

dic[键]

- 修改添加元素

语法

dic[键] = 值

- 字典的方法

方法	作用
D.keys()	返回可迭代的dict_keys集合对象
D.values()	返回可迭代的dict_values值对象
D.items()	返回可迭代的dict_items对象

2.1.3.2 Python高级运用

1.函数

在Python中，有些语句块需要重复使用，如果我们通过复制粘贴的方式去使用这些代码，就会使得我们的程序太过冗余，这种情况下，我们可以通过函数将这些需要复用的语句块封装起来。

- 函数的定义和调用

函数定义语法

```
def 函数名(形参):  
    语句块  
  
    [return [表达式]]
```

说明

return语句用于函数中，结束当前函数的执行，返回到调用该函数的地方

return语句后跟的表达式可以省略，省略后相当于return None

return语句也可以省略，此时函数执行完最后一条语句后返回None(相当于在最后加了一

函数调用语法

函数名(实参)

示例

写一个函数，输入2个整数，返回2个数中的最大值

- 函数的参数

函数的形参

必选参数

必须参数，即函数调用时必须传的参数

例：

```
def helloworld(a):  
    print("你传入的参数是:", a)  
  
helloworld("nice") # 调用时必须传参
```

默认参数 (缺省参数)

可以为形参设置默认值，当调用带有默认值参数的函数时，如果没有为该参数传参，此时函数将会直接使用函数定义时设置的默认值，如果你有为该参数传参，则以你传入的参数为准。

例：

```
def fun(name, age=20, address='成都'):  
    print("姓名是:%s" % name)  
    print("年龄是:%s岁" % age)  
    print("地址是:%s" % address)  
  
fun("张三", 22) # age和address是默认参数，可以传参，可以不传参
```

函数的传参方式 (实参)

位置传参

实参与形参是按位置来依次对应的

例：

```
def myfun(a, b, c):  
    print("a=", a)  
    print("b=", b)  
    print("c=", c)  
  
myfun(1, 2, 3) # 1、2、3依次按照位置传递给a、b、c
```

关键字传参

传参时，根据形参的名称给形参赋值，实参和形参按名称进行匹配

例：

```
def myfun(a, b, c):  
    print("a=", a)  
    print("b=", b)  
    print("c=", c)  
  
myfun(a=1, b=2, c=3) # 1、2、3依次按照位置传递给a、b、c
```

- **lambda表达式**

创建一个匿名函数对象，和def的区别在于不提供函数名

语法

lambda [参数1, 参数2, ...]: 表达式 [] 里的内容可以省略

示例

通过lambda表达式计算两个数的和

- **sorted()函数**

sorted()函数可将可迭代对象的数据进行排序，生成排序后的列表

语法

sorted(iterable, key=None, reverse=False)

示例

L = [5, -2, -4, 0, 3, 1] 将L列表按照绝对值大小排序

2.文件操作

文件是用来存储数据的，并且可以实现持久化存储。在Python里面是通过文件对象来操作文件。

- **文件打开函数open()**

语法格式

f = open(file, mode='r', encoding=None)

file: 文件路径

mode: 读取模式，默认为r

encoding: 编码方式，windows下默认为gbk

模式	说明
r	读模式，默认模式
w	写模式
a	追加模式

- 文件操作方法

方法	说明
f.readline()	读取一行数据
f.readlines()	返回每行字符串的列表
f.write()	将字符串写入文件
f.close()	关闭文件

- with语句

可以通过with语句处理文件关闭

```
with open(file) as f:
    xxxx
```

- 文件的迭代读取

open函数得到的文件对象是一个可迭代对象，可以用for循环进行遍历

```
with open(file) as f:
    for line in f:
        print(line)
```

2.1.3.3 Python面向对象

1.面向对象的概念



什么是对象

对象是拥有具体属性值(名词)和行为(动词)的实体, 比如在现实生活中的实际存在的一辆车, 一个人, 一个苹果等。对象, 又称实例, 在Python中, 一切皆对象, 比如我们之前使用的整数、字符串、列表、字典等。

什么是面向对象

面向对象就是在程序中, 把一切都看做对象, 基于这些对象来编程。当你要做事情的时候, 你不会亲自去做, 而是调用一些对象去做。

什么是类

类可以理解为对象的模板, 其中定义了同类对象应该具有的属性和方法, 类可以用来创建一个或多个同类对象。类是一个抽象的概念, 而对象是类具象化的结果。类比于现实世界中的事务, 比如汽车就相当于是一个类, 提到它我们知道它应具有行驶的功能, 而车型、重量、最大行驶速度、车龄等等是汽车应该具有的属性。而一辆具体的汽车是一个对象, 在这个对象中, 车型、重量等属性有了具体的值。

类的创建语法

```
class 类名(继承列表):  
    语句块
```

2. 实例方法

语法


```
class 类名(继承列表):  
  
    def 实例方法名(self, 参数1, 参数2, ...):  
  
        '''实例方法的文档字符串'''  
  
        语句块
```

作用

用于描述一个对象的行为,让此类型的全部对象都拥有相同的行为

说明

实例方法实质是函数,是定义在类内的函数。实例方法至少有一个形参,第一个形参代表调用这个方法的实例,一般命名为'self'

调用

实例.实例方法名(调用传参)

示例

创建一个Car类, 定义实例方法run

3.实例属性

每个实例都可以有自己的变量,此变量称为实例变量(也叫实例属性)

实例属性的使用

实例.实例属性名

实例属性的赋值

可以通过"="为实例属性赋值

实例.实例属性名 = 值

4.初始化方法

作用

对新创建的对象添加实例变量(属性)

语法

```
class 类名(继承列表):  
  
    def __init__(self [,形参列表]):  
  
        语句块
```

示例

为刚才创建的Car类添加初始化方法，添加品牌和颜色属性

2.1.3.4 作业

1.写代码实现如下功能：给出一个年份，判断是否为闰年并打印结果。闰年规则：每四年一闰，每百年不闰，四百年又是一个闰年

例：2016年是闰年；2100年 不是闰年；2400年 是闰年

2.用字符串s = "ABC" 和 s2 = "123" 生成如下列表：

['A1', 'A2', 'A3', 'B1', 'B2', 'B3', 'C1', 'C2', 'C3']

2.2 网络爬虫

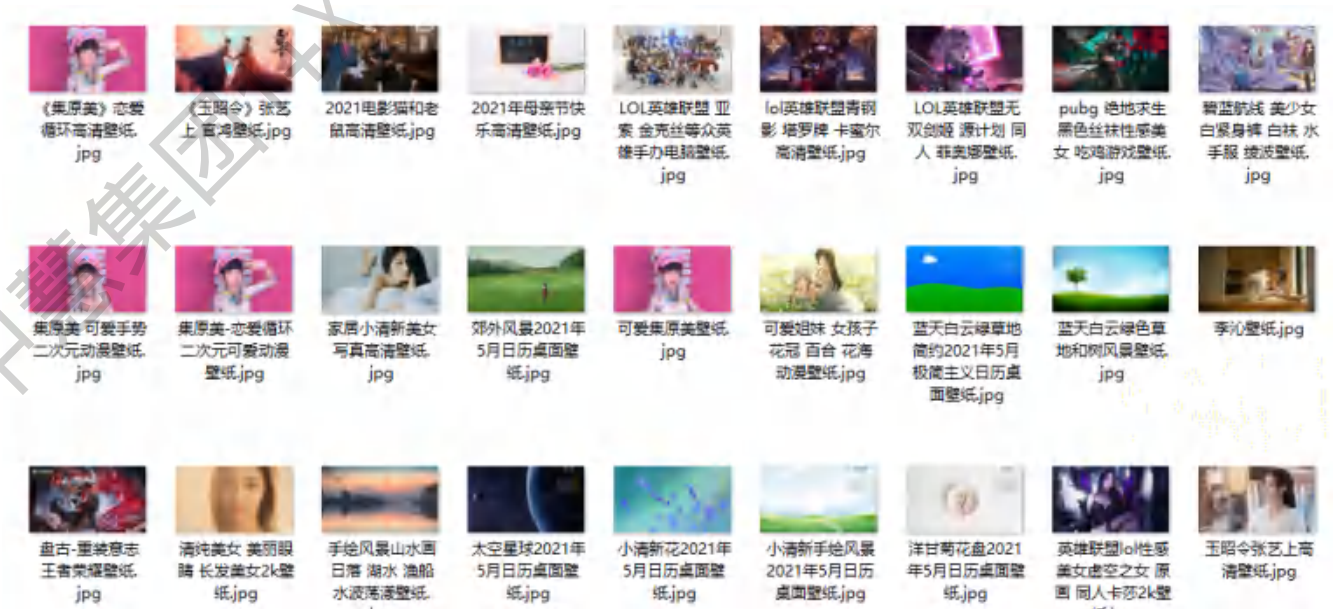


2.2.1 项目导读

1、图片批量下载

网站地址：<http://www.netbian.com/>

运行结果：



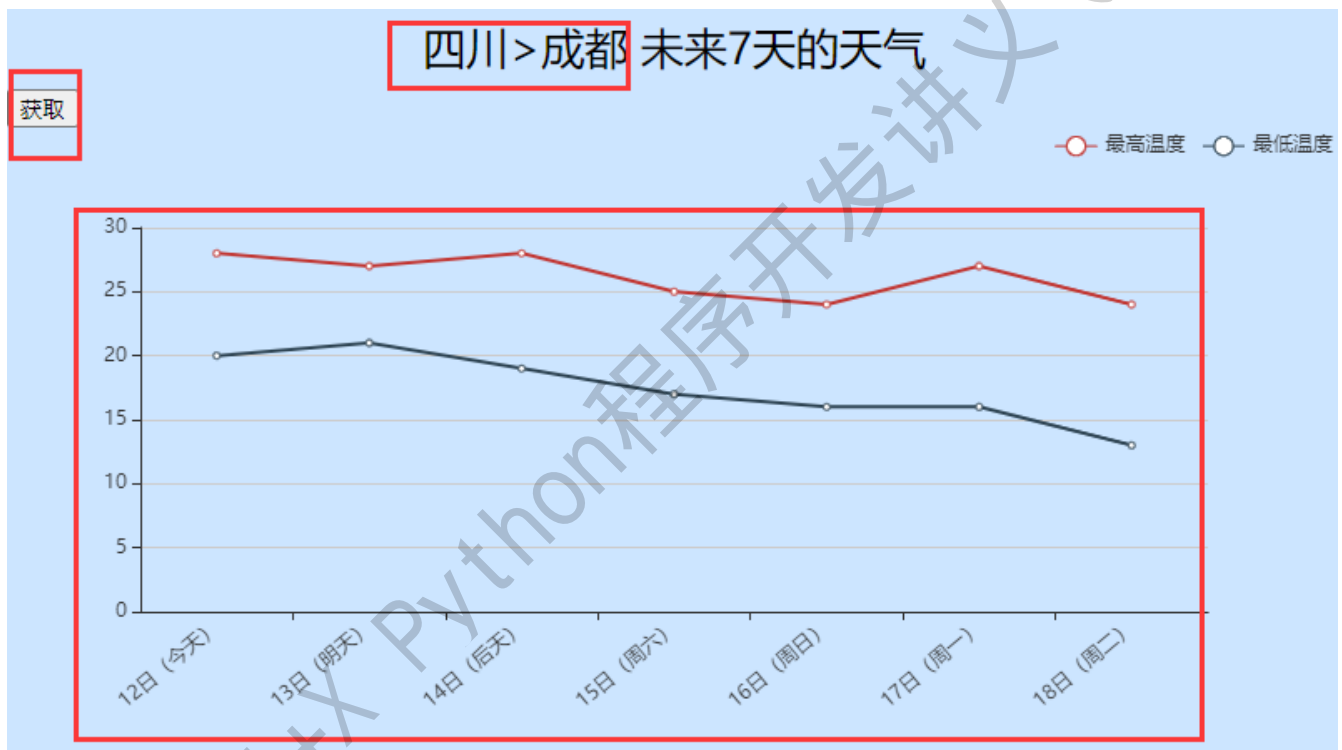
2、全国各城市未来七日温度可视化

网站地址: <http://www.weather.com.cn/weather/101270101.shtml>

运行结果:

```
C:\Users\ASUS\Desktop\html\python_best
C:/Users/ASUS/Desktop/html/python_bes
请输入你要查询的地区天气: 成都
正则搜索你要查询 成都 的天气....

Process finished with exit code 0
```



2.2.2 项目实施步骤

1.认识爬虫 2.模拟浏览器发送请求 3.提取网页内容技术 (xpath) 4.实现数据可视化技术

2.2.3 项目涉及知识

2.2.3.1.爬虫语法

1.认识爬虫

网络爬虫, 又被称为网页蜘蛛, 网络机器人, 在FOAF社区中间, 更经常的称为网页追逐者, 是一种按照一定的规则, 自动地抓取万维网信息的程序或者脚本, 另外一些不常用的名字还有蚂蚁、自动索引、模拟程序或者蠕虫。

简单来讲，爬虫就是一个探测机器，它的基本操作就是模拟人的行为去各个网站溜达，点看到的信 息背回来。就像一只虫子在一幢楼里不知疲倦地爬来爬去。每天放出无数爬虫到各个网站，把他们的信息抓回来,并保存

2.爬虫的使用场景

- 个人信息检索系统
- 特定信息收集系统
- 自动填写调查问卷
- 电话号码收集系统
- 爬虫分析热度排行
- **爬虫进行股票分析**
- 爬虫网站定向数据
- 视频网站视频批量下载
- 购物网站比价系统
- 文章批量下载
- 飞机票比价系统 (以一个地方到另外一个地方飞机票)
- 招聘公司爬虫招聘信息
- 爬虫房产网站做房产分析
- 财务报表下载助手
- 排行分析之指数分析
- 畅销书排行分析
- 验证码破解
- **用户拓展关系分析**
- 模拟登陆系统
- 文件下载助手开发
- 抢票软件
- 构建代理IP池
- 分析网页结构
- 房产数据分析
- 音乐网站批量下载
- 城市旅游数据分析
- **购物网站数据挖掘分析**
- V企业信息、分类信息、房地产信息、电商信息
- 论坛发帖、问答推广、效果回访、网上购物、宝贝收藏、投票点赞
- 爬取微信公众号，进行数据分析
- 头条自动发文章，赚点广告费
- 网站薅羊毛
- 淘宝关键词展位排名
- 新闻数据分析
- **为人工智能，科学分析提供大量的数据**

3.HTTP请求

客户端：浏览器（谷歌）

服务端：服务器（百度）

以百度网站为案例：

<http://www.baidu.com> 网页请求地址



百度一下

- 百度热搜
- 1 中国仍是世界第一人口大国
 - 2 成都坠亡学生家属不认同教育局通报
 - 3 中国男比女多3490万人
 - 4 云南直升机坠毁海 4名机组人员遇难
 - 5 中国平均每个家庭人口不足3人
 - 6 31省区市新增确诊14例均为境外输入

(内部用)

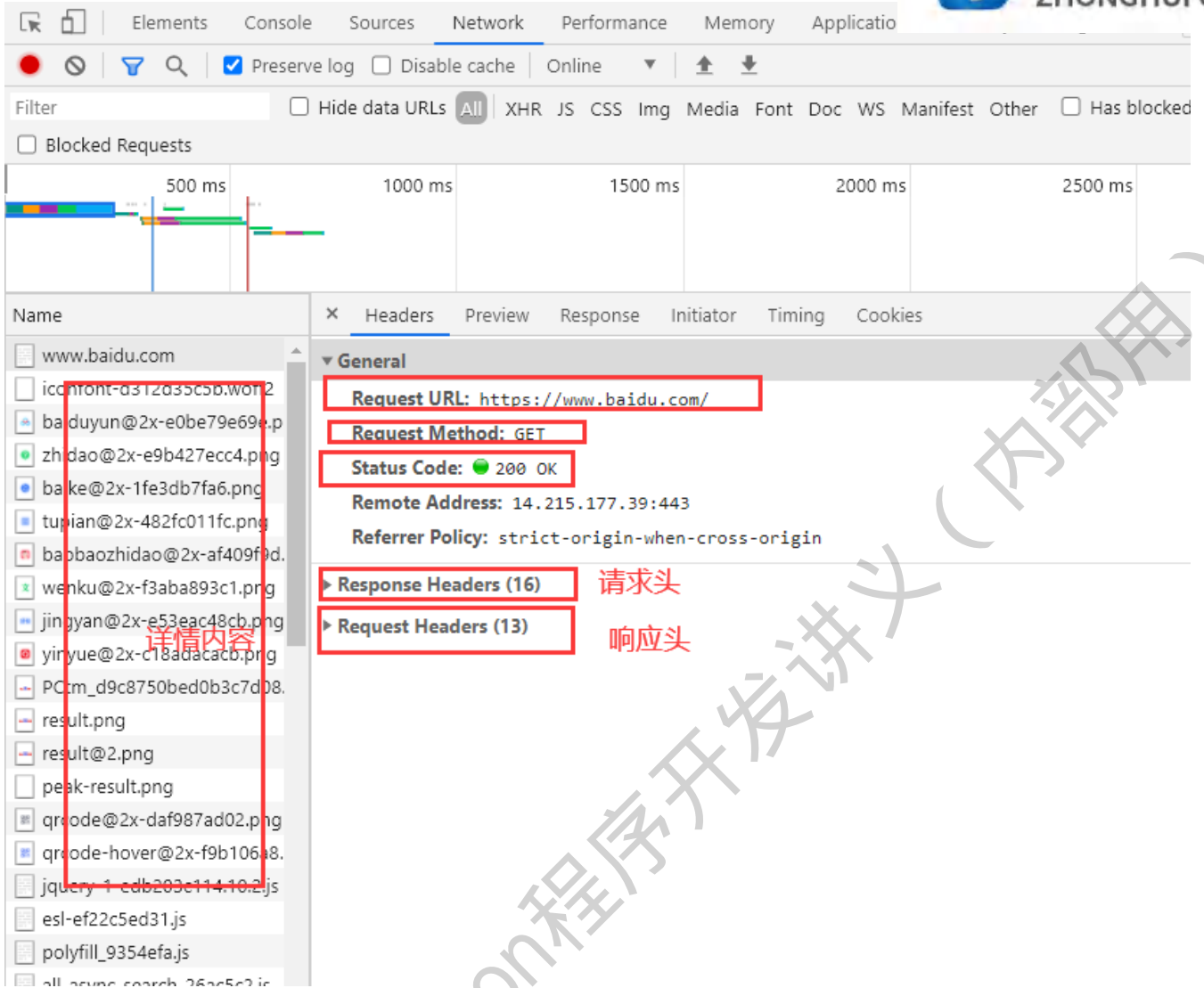
服务器返回给浏览器的响应体

百度热搜

- 1 中国仍是世界第一人口大国
- 2 成都坠亡学生家属不认同教育局通报
- 3 中国男比女多3490万人

```
<!--STATUS K-->
<html>
<head>
<body class="style">
<script>
<script type="text/css" id="s_is_result_css" style="display:none">
<script type="text/css" id="s_index_off_css" style="display:none">
<script>
<div id="head">
<div id="s_top_wrap" class="s-top-wrap s-isindex-wrap" style="left: 0px">
<div id="s_top_left" class="s-top-left s-isindex-wrap">
<div id="s_top_right" class="s-top-right s-isindex-wrap">
<div id="head_wrapper" class="head_wrapper s-isindex-wrap nologin">
<div id="s_side_wrapper" class="s-isindex-wrap">
<div id="bottom_layer" class="s-bottom-layer s-isindex-wrap">
<div class="s_tab" id="s_tab">
<div id="s_side_wrapper">
<div id="wrapper_wrapper">
<div class="c-tips-container" id="c-tips-container">
<script>
<script type="text/javascript">
<script type="text/javascript" src="https://dss0.bdstatic.com/9ajibjoh_023ndCf/static/awermap/js/...>
```

Python 爬虫 (内部用)



Network tab details for `www.baidu.com`:

- Request URL:** `https://www.baidu.com/`
- Request Method:** `GET`
- Status Code:** `200 OK`
- Remote Address:** `14.215.177.39:443`
- Referrer Policy:** `strict-origin-when-cross-origin`
- Response Headers (16):** 请求头
- Request Headers (13):** 响应头

4. Requests 安装

作用: 模拟浏览器, 发送网络请求, 并接收服务器返回的数据

```
pip install requests
```

5. 发起响应, 获取网页

```
# 网络请求的第三方库
import requests
# 百度地址
url = "http://www.baidu.com"
# 模拟浏览器发起请求
response = requests.get(url)
# 获取浏览器状态码: 200成功
print(response) # <Response [200]>
# 解析服务器返回内容
print(response.content.decode()) #<!--STATUS OK--><html> <head><meta http-equiv=content-
```

6. 反爬

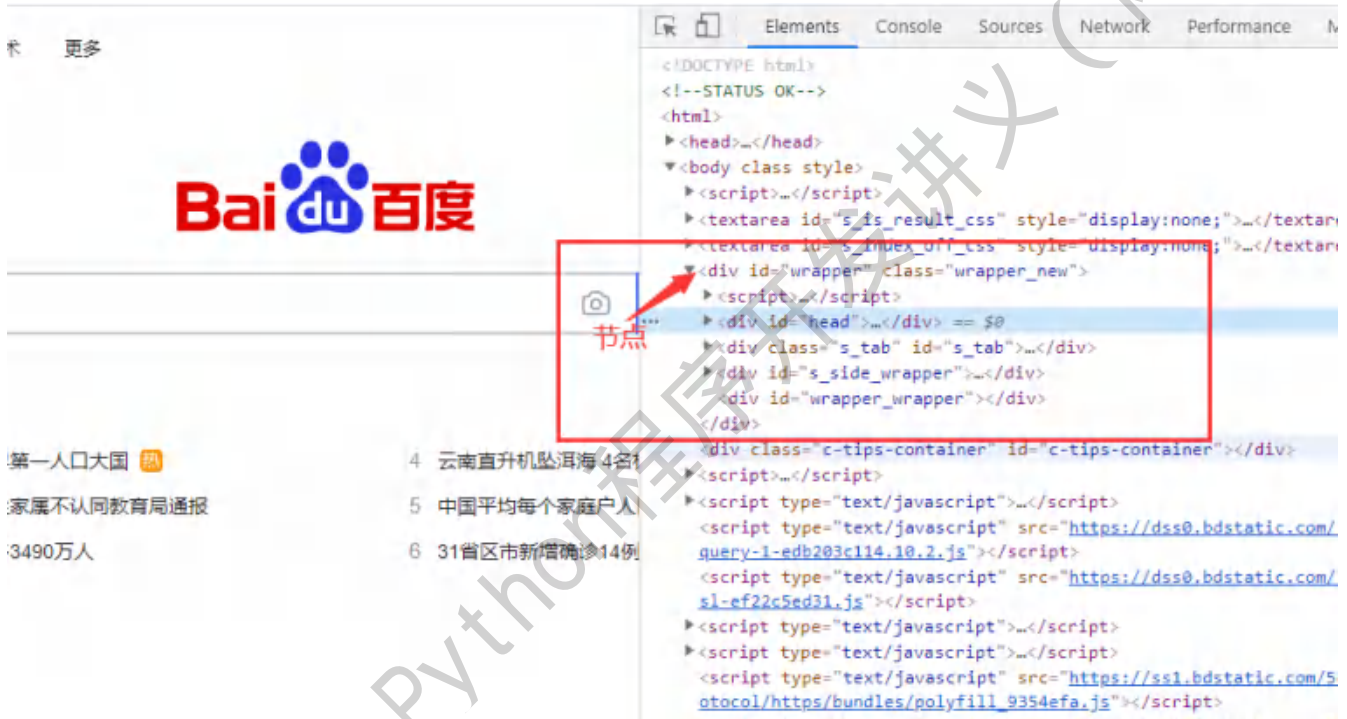
```
import requests
url = "http://www.baidu.com"
header = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36'}
response = requests.get(url, headers=header)
print(response.content.decode())
```

7.XPath语法, 提取数据

安装Python第三方库: lxml, 提取数据

```
pip install lxml
```

1) XPath语法--节点



2) XPath语法

以百度贴吧为例: <https://tieba.baidu.com/f?kw=%CC%F9%B0%C9&fr=ala0&tpl=5>

选取节点

nodename	选取此节点的所有子节点。
/	从根节点选取。
//	从匹配选择的当前节点选择文档中的节点，而不考虑它们的位置。
.	选取当前节点。
..	选取当前节点的父节点。
@	选取属性。

```

import requests
from lxml import etree
url = "https://tieba.baidu.com/f?kw=%CC%F9%B0%C9&fr=ala0&tp1=5"
response = requests.get(url)
# print(response)
# print(response.content.decode())
# 获取网页内容
content = response.content.decode()
# 将网页内容通过通过lxml梳理
html_str = etree.HTML(content)
# 使用xpath提取网页内容
p_obj = html_str.xpath("//p") # 全文搜索p标签
print(p_obj) # [<Element p at 0x22eb3e0e888>],p所在的对象
# 找到class为card_slogan的p标签
p_class = html_str.xpath("//p[@class='card_slogan']")
print(p_class) # [<Element p at 0x1f325c4af88>],p标签的位置
# 找到class为card_slogan的p标签下的文字
p_text = html_str.xpath("//p[@class='card_slogan']/text()")
print(p_text) # ['百度贴吧吧, 吧友交流的家園']
    
```

案例作业:



```

import requests
from lxml import etree
url = "https://tieba.baidu.com/f?kw=%CC%F9%B0%C9&fr=ala0&tp1=5"
response = requests.get(url)
# print(response)
# print(response.content.decode())
# 获取网页内容
content = response.content.decode()
# 将网页内容通过通过lxml梳理
html_str = etree.HTML(content)
#获取"看贴"文字
a_text= html_str.xpath("//a[@id='tab_forumname']/text()")
print(a_text) # ['看贴']
    
```



```
# 获取"看帖","图片","精品","视频" 文字
li = html_str.xpath("//li[@class=' j_tbtnav_tab ']") # 属性相等
print(li) # [<Element li at 0x1d4c5e5cd88>, <Element li at 0x1d4c615adc8>, <Element li at 0x1d4c615ad08>]
for i in li:
    a_txt = i.xpath("./a/text()") # 当前li标签下的a标签下的文字
    print(a_txt) # ['图片'] ['精品'] ['视频']
```

选取节点谓词

<code>/bookstore/book[1]</code>	选取属于 bookstore 子元素的第一个 book 元素。
<code>/bookstore/book[last()]</code>	选取属于 bookstore 子元素的最后一个 book 元素。
<code>/bookstore/book[last()-1]</code>	选取属于 bookstore 子元素的倒数第二个 book 元素。
<code>//title[@lang='eng']</code>	选取所有 title 元素, 且这些元素拥有值为 eng 的 lang 属性。

```
import requests
from lxml import etree
url = "https://tieba.baidu.com/f?kw=%CC%F9%B0%C9&fr=ala0&tp1=5"
response = requests.get(url)
# print(response)
# print(response.content.decode())
# 获取网页内容
content = response.content.decode()
# 将网页内容通过通过lxml梳理
html_str = etree.HTML(content)

# 获取"看帖"href属性
a_href = html_str.xpath("//a[@id='tab_forumname']/@href")
print(a_href) # ['/f?kw=%E8%B4%B4%E5%90%A7&ie=utf-8&tab=main']

# 找到精品属性
# 方法1:
a_list = html_str.xpath("//a[@class='j_nav_local_tab_link j_tbtnav_tab_a']")
print(a_list) # [<Element a at 0x181fe4fca08>, <Element a at 0x181fe6cad8>, <Element a at 0x181fe507f08>, <Element a at 0x181fe6ce808>]
a_list2_text = a_list[2].xpath("./@href")
print(a_list2_text) # ['/f?kw=%E8%B4%B4%E5%90%A7&ie=utf-8&tab=good']
# 方法2:
a_list3 = html_str.xpath("//ul[@class='nav_list j_nav_list']/li[3]/a/@href")
print(a_list3) # ['/f?kw=%E8%B4%B4%E5%90%A7&ie=utf-8&tab=good']
```

选取若干路径

径表达式	结果
//book/title //book/price	选取 book 元素的所有 title 和 price 元素。
//title //price	选取文档中的所有 title 和 price 元素。
/bookstore/book/title //price	选取属于 bookstore 元素的 book 元素的所有 title 元素，以及文档中所有的 price 元素。



```

import requests
from lxml import etree
url = "https://tieba.baidu.com/f?kw=%CC%F9%B0%C9&fr=ala0&tpl=5"
response = requests.get(url)
# print(response)
# print(response.content.decode())
# 获取网页内容
content = response.content.decode()
# 将网页内容通过通过lxml梳理
html_str = etree.HTML(content)

# 提取多项内容
a_text = html_str.xpath("//ul[@class='forum_dir_info bottom_list clearfix']//a[@rel='nofollow']/text()")
print(a_text) # ['贴吧活动与交流']
a_list_text = html_str.xpath("//a[@class='j_nav_local_tab_link j_tbnav_tab_a']/text()")
print(a_list_text) # ['看贴', '图片', '精品', '视频']
# 合并提取
a_list = html_str.xpath("//ul[@class='forum_dir_info bottom_list clearfix']//a[@rel='nofollow']/text() | //a[@class='j_nav_local_tab_link j_tbnav_tab_a']/text()")
print(a_list) # ['贴吧活动与交流', '看贴', '图片', '精品', '视频']
    
```

2.2.3.2 项目实训--彼岸桌面壁纸下载

项目需求：

根据彼岸网页地址，下载桌面图片，并保存到本地img文件夹中

实施步骤：

1. 安装requests库: pip install requests
2. 安装xpath库: pip install lxml
3. 获取彼岸网站url地址
4. 发起requests响应，获取服务器内容
5. 使用XPath提取图片地址和名字
6. 根据图片地址提取图片内容
7. 下载图片，并保存

项目代码

```
import requests
from lxml import etree
import time
import os
# 彼岸桌面
# 1.获取url链接地址
# def get_url():
#     url = "http://www.netbian.com/"
#     return url
def get_url():
    url_list = ["http://www.netbian.com/"]
    for i in range(2,100):
        url_list.append("http://www.netbian.com/index_{}.htm".format(i))
    return url_list
# 2.发起响应
def get_response(url):
    time.sleep(0.3)
    response = requests.get(url)
    content = response.content
    return content
# 3.提取数据列表
def get_content(content):
    html_str = etree.HTML(content.decode("gbk"))
    content_list = []
    li_list = html_str.xpath("//div[@class='list']/ul/li")
    # print(li_list)
    # 4.提取图片的url地址和图片名
    for li in li_list:
        item = {}
        item["name"] = li.xpath("./a/b/text()")
        item["img_href"] = li.xpath("./a/img/@src")
        if item["name"]:
            content_list.append(item)
    return content_list

#4下载并保存图片
def download_img(content_list):
    for content in content_list:
        img_name = content["name"][0]
```

```

path_img = "./img"
if os.path.exists(path_img):
    pass
else:
    os.mkdir(path_img)
with open(path_img+"/{}.jpg".format(img_name),"wb") as f:
    f.write(get_response(content["img_href"][0]))
    print("{}.....下载成功".format(img_name))

# 运行函数
def run():
    # url = get_url()
    url_list = get_url()
    for url in url_list:
        content = get_response(url)

        content_list = get_content(content)

        download_img(content_list)

if __name__ == '__main__':
    run()
    
```

运行结果:



项目分析:

此项目是对Xpath语法的应用和总结，需要有扎实的Python语言基础.重点在爬虫实施步骤和Xpath语法规则

2.2.3.3 Echarts图形化显示

1.Echarts认识

ECharts 是一个使用 JavaScript 实现的开源可视化库，涵盖各行业图表，满足各种需求。

ECharts 遵循 Apache-2.0 开源协议，免费商用。

ECharts 兼容当前绝大部分浏览器 (IE8/9/10/11, Chrome, Firefox, Safari等) 及兼容展示。

2.Echarts 使用

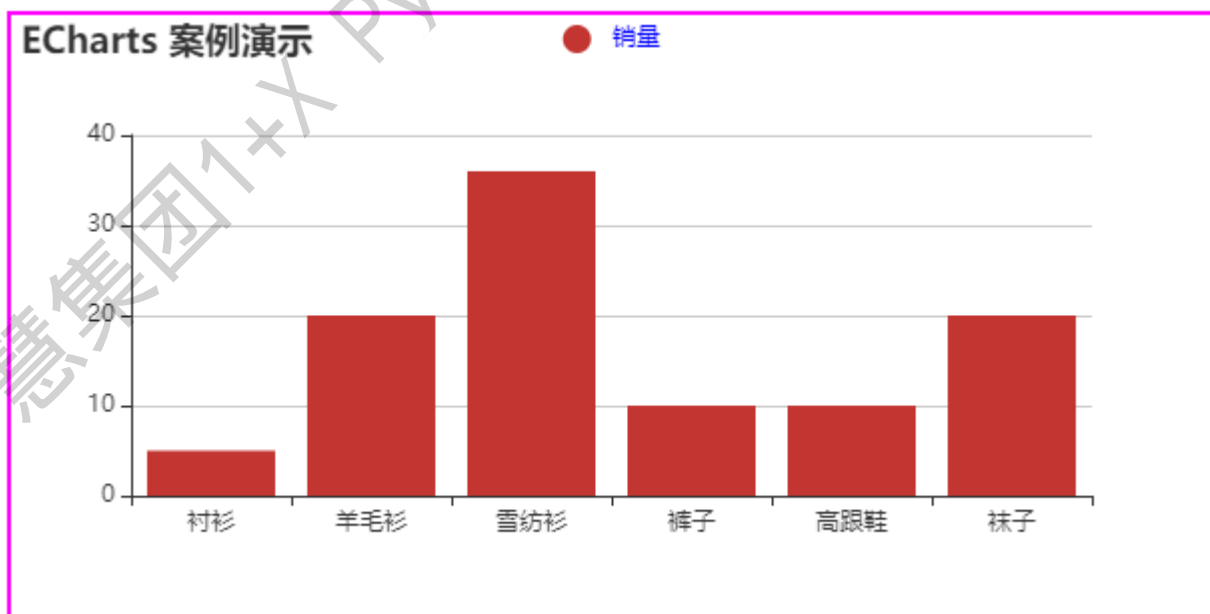
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6     <script src="echarts.min.js"></script>
7 </head>
8 <body>
9
10 </body>
11 </html>
```

3.Echarts语法--柱状图

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <script src="echarts.min.js"></script>
  <style>
    #echar{
      /*盒子宽度*/
      width: 600px;
      /*盒子高度*/
      height: 300px;
      /*盒子边框*/
      border: 2px solid magenta;
    }
  </style>
</head>
<body>
  <div id="echar"></div>
</body>
<script>
// 基于准备好的dom, 初始化echarts实例
var myChart = echarts.init(document.getElementById('echar'));
// 绘制图表
myChart.setOption({
  //为图表配置标题:
  title: {
```

```
text: 'Echarts 案例演示'  
},  
// 图例组件, 图例组件展现了不同系列的标记(symbol), 颜色和名字。  
legend: {  
  data: [{  
    name: '销量',  
    // 强制设置图形为圆。  
    icon: 'circle',  
    // 设置文本为; 蓝色  
    textStyle: {  
      color: 'blue'  
    },  
    left: "right"  
  }]  
},  
//X 轴 配置要在 x 轴显示的项:  
xAxis: {  
  data: ['衬衫', '羊毛衫', '雪纺衫', '裤子', '高跟鞋', '袜子']  
},  
//Y 轴 配置要在 y 轴显示的项。  
yAxis: {  
},  
//系列列表:每个系列通过 type 决定自己的图表类型:  
series: [{  
  name: '销量',  
  type: 'bar',  
  data: [5, 20, 36, 10, 10, 20]  
}]  
})  
</script>  
</html>
```

运行结果:



修改柱状图颜色:

//系列列表:每个系列通过 type 决定自己的图表类型:

```
series: [{
  name: '销量',
  type: 'bar',
  // 修改颜色
  itemStyle: {
    normal: {
      color: "green"
    }
  },
  data: [5, 20, 36, 10, 10, 20]
}]
```

修改图形:

- type: 'bar': 柱状/条形图
- type: 'line': 折线/面积图
- type: 'pie': 饼图
- type: 'scatter': 散点 (气泡) 图
- type: 'effectScatter': 带有涟漪特效动画的散点 (气泡)
- type: 'radar': 雷达图
- type: 'tree': 树型图
- type: 'treemap': 树型图
- type: 'sunburst': 旭日图
- type: 'boxplot': 箱形图
- type: 'candlestick': K线图
- type: 'heatmap': 热力图
- type: 'map': 地图
- type: 'parallel': 平行坐标系的系列
- type: 'lines': 线图
- type: 'graph': 关系图
- type: 'sankey': 桑基图
- type: 'funnel': 漏斗图
- type: 'gauge': 仪表盘
- type: 'pictorialBar': 象形柱图
- type: 'themeRiver': 主题河流

多图绘制

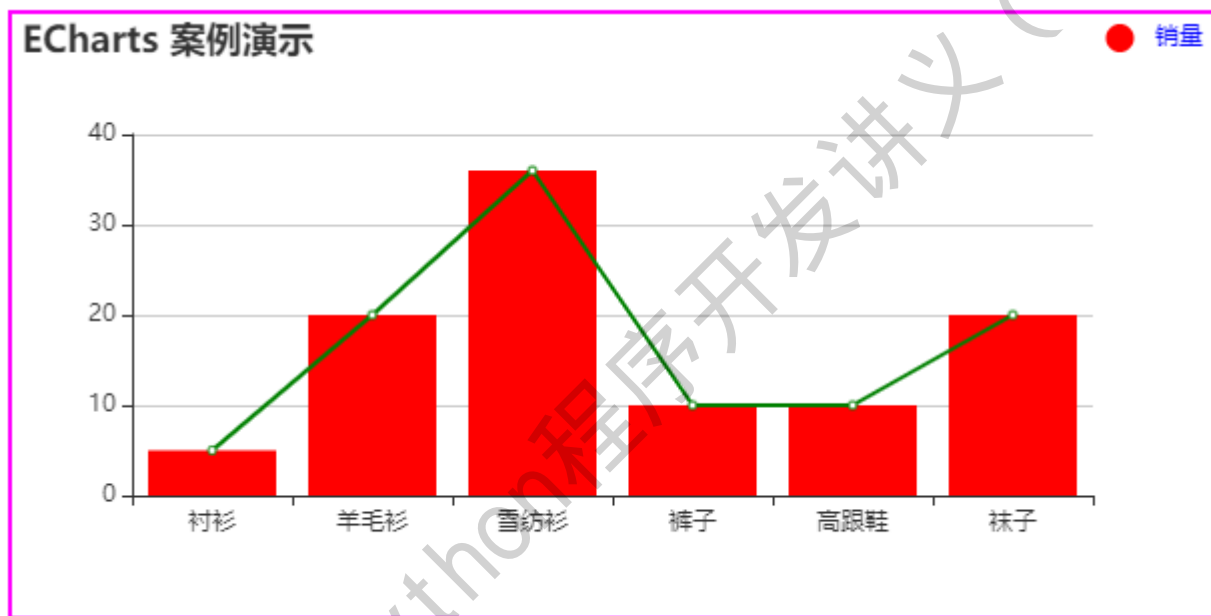
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <script src="echarts.min.js"></script>
  <style>
    #echar{
      /*盒子宽度*/
      width: 600px;
      /*盒子高度*/
```

```
        height: 300px;
        /*盒子边框*/
        border: 2px solid magenta;
    }
</style>
</head>
<body>
<div id="echar"></div>
</body>
<script>
// 基于准备好的dom, 初始化echarts实例
var myChart = echarts.init(document.getElementById('echar'));
// 绘制图表
myChart.setOption({
    //为图表配置标题:
    title: {
        text: 'ECharts 案例演示'
    },
    // 图例组件, 图例组件展现了不同系列的标记(symbol), 颜色和名字。
    legend: {
        data: [{
            name: '销量',
            // 强制设置图形为圆。
            icon: 'circle',
            // 设置文本为; 蓝色
            textStyle: {
                color: 'blue'
            }
        }],
        // 图例位置
        left:"right"
    },
    //X 轴 配置要在 X 轴显示的项:
    xAxis: {
        data: ['衬衫', '羊毛衫', '雪纺衫', '裤子', '高跟鞋', '袜子']
    },
    //Y 轴 配置要在 Y 轴显示的项。
    yAxis: {
    },
    //系列列表:每个系列通过 type 决定自己的图表类型:
    series: [{
        name: '销量',
        type: 'bar',
        // 修改颜色
        itemStyle:{
            normal:{
                color:"red"
            }
        },
        data: [5, 20, 36, 10, 10, 20]
    },{
        name: '销量',
        type: 'line',
```



```
// 修改颜色
itemStyle:{
  normal:{
    color:"green"
  }
},
data: [5, 20, 36, 10, 10, 20]
}
]
})
</script>
</html>
```

运行结果



2.2.3.4 Python自带的简单服务器

开启服务器语法：


```
python -m http.server 8090
```

```
Microsoft Windows [版本 10.0.19042.928]
(c) Microsoft Corporation。保留所有权利。
```

```
(python_best) C:\Users\ASUS\Desktop\html\python_best>python -m http.server 8090
Serving HTTP on 0.0.0.0 port 8090 (http://0.0.0.0:8090/) ...
```

浏览器访问语法：

http://127.0.0.1:8090



← → ↻ ⓘ 127.0.0.1:8090

应用 百度一下, 你就知道 QQ邮箱 译 百度翻译-200种语... 散

Directory listing for /

- [.idea/](#)
- [.ipynb_checkpoints/](#)
- [2021年饼图.png](#)
- [Include/](#)
- [Lib/](#)
- [python_best/](#)
- [pyvenv.cfg](#)
- [Scripts/](#)
- [spider/](#)
- [绘图.ipynb](#)
- [饼图.png](#)

2.2.3.5 提取JSON文件并显示

demo.json文件

```
[  
  {"value":235, "name":"视频广告"},  
  {"value":274, "name":"联盟广告"},  
  {"value":310, "name":"邮件营销"},  
  {"value":335, "name":"直接访问"},  
  {"value":400, "name":"搜索引擎"}  
]
```

`$.get`, 提取并显示:

注意: 使用`$.get`, 需要加载jQuery, 并且需要开启Python服务器

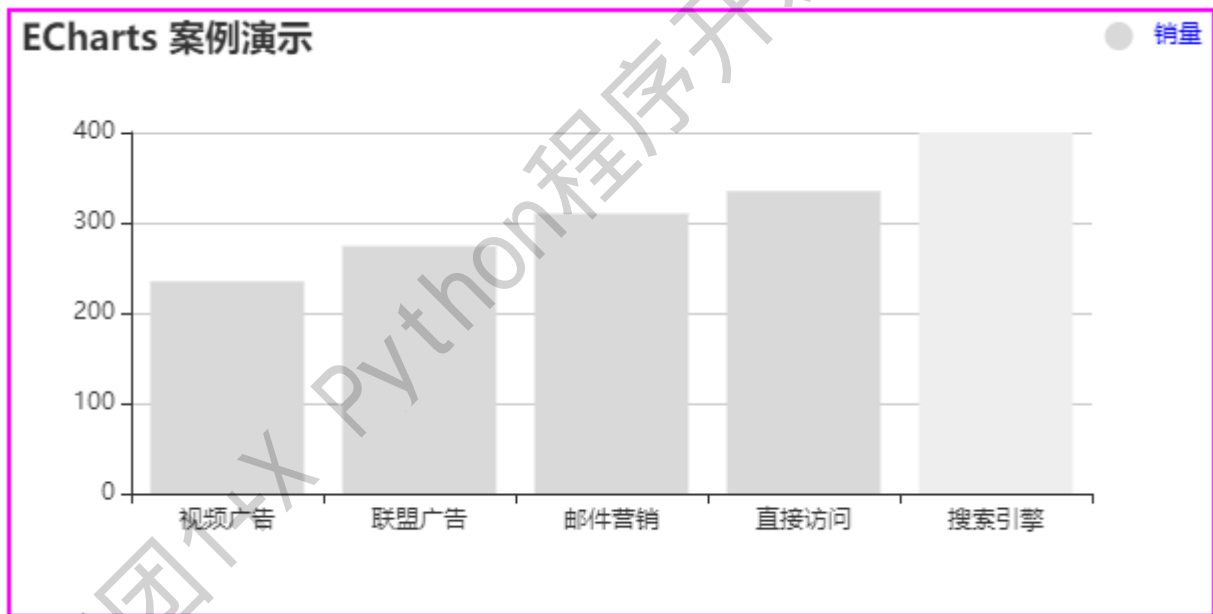
```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>Title</title>  
  <script src="jquery.min.js"></script>
```

```
<script src="echarts.min.js"></script>
<style>
  #echar{
    /*盒子宽度*/
    width: 600px;
    /*盒子高度*/
    height: 300px;
    /*盒子边框*/
    border: 2px solid magenta;
  }
</style>
</head>
<body>
<div id="echar"></div>
</body>
<script>
// 基于准备好的dom, 初始化echarts实例
var myChart = echarts.init(document.getElementById('echar'));

$.get("./demo.json",function (data) {
  // console.log(data); # 获取的data数据
  var x_list = [];
  var y_list = [];
  for (var i=0;i<data.length;i++){
    x_list.push(data[i].name);
    y_list.push(data[i].value);
  }
  // 绘制图表
  myChart.setOption({
    //为图表配置标题:
    title: {
      text: 'Echarts 案例演示'
    },
    // 图例组件, 图例组件展现了不同系列的标记(symbol), 颜色和名字。
    legend: {
      data: [{
        name: '销量',
        // 强制设置图形为圆。
        icon: 'circle',
        // 设置文本为; 蓝色
        textStyle: {
          color: 'blue'
        }
      }],
      // 图里位置
      left:"right"
    },
    //x 轴 配置要在 x 轴显示的项:
    xAxis: {
      data: x_list
    },
    //y 轴 配置要在 y 轴显示的项。
    yAxis: {
```

```
},  
//系列列表:每个系列通过 type 决定自己的图表类型:  
series: [{  
  name: '销量',  
  type: 'bar',  
  // 修改颜色  
  itemStyle: {  
    normal: {  
      // 十六进制颜色表示方式  
      color: "#d9d9d9"  
    }  
  },  
  data: y_list  
}  
]  
})  
});  
  
</script>  
</html>
```

显示效果:



2.2.3.6 项目实训--全国各个城市未来七日温度变化图

项目需求:

需要使用到Python中爬虫技术, 根据输入的城市名, 获取到全国各个城市的天气数据, 将数据保存成JSON文件, 并使用Echarts进行可视化显示

实施步骤:

1. 获取全国各个成绩的JS表
2. 根据输入城市名, 判断是否, 在JS表中
3. 在JS提取不同城市的ID

4. 根据ID获取七日气温变化图url地址
5. 发起响应, 获取数据
6. 提取未来七日温度数据
7. 保存成JSON文件
8. 开始Python服务器
9. 创建HTML文件
10. 使用\$.get获取天器的JSON数据
11. 使用Echarts绘制最高温度和最低温度, 未来七日温度折线图

项目代码

weather_spider.py文件:

```
import requests
import json
from lxml import etree
import os
# 输入天气绘制成表
# 城市地址
def get_city_list():
    city_url = "https://j.i8tq.com/weather2020/search/city.js"
    response = requests.get(city_url)
    city_js = response.content.decode()
    citys = city_js.split("=")[-1]
    # print(citys)
    city_dic = json.loads(citys)
    # print(city_dic.items())
    city_list = []
    for k,v in city_dic.items():
        # print(k)
        # print(v)
        for k1,v1 in v.items():
            # print(v1)
            for k2, v2 in v1.items():
                # print(v2)
                city_list.append(v2)
    return city_list

# city_list 所有天气城市
def get_city_url(city_list):
    input_city = input("请输入你要查询的地区天气: ")
    status = False
    for city in city_list:
        if input_city == city["NAMECN"]:
            status = True
            city_urls = "http://www.weather.com.cn/weather/{}.shtml".format(city["AREAID"])
            print("正则搜索你要查询 {} 的天气....".format(input_city))
            return city_urls

# 发起响应
def get_response(url):
    response = requests.get(url)
    content = response.content.decode()
```

```
return content

# 提取数据列表
def get_content_list(content):
    html_str = etree.HTML(content)
    # 地址
    city = html_str.xpath("//div[@class='crumbs fl']/a[2]/text()")[0]
    town = html_str.xpath("//div[@class='crumbs fl']/a[3]/text()")[0]
    # print(city,town)

    li_list = html_str.xpath("//ul[@class='t clearfix']/li")
    weather_list = []
    for li in li_list:
        item = {}
        item["address"] = city+">" + town
        item["week_date"] = li.xpath("./h1/text()")[0]
        item["max_temp"] = li.xpath("./p[@class='tem']/span/text()")[0]
        item["min_temp"] = li.xpath("./p[@class='tem']/i/text()")[0].split("°C")[0]
        item["wea"] = li.xpath("./p[@class='wea']/text()")[0]
        # print(item)
        weather_list.append(item)
    return weather_list

#保存成json文件
def save_json(weather_list):
    file_path = "./source"
    if os.path.exists(file_path):
        with open(file_path + "/weather.json", "w", encoding="utf-8") as f:
            json.dump(weather_list, f, ensure_ascii=False, indent=2)
    else:
        os.mkdir(file_path)
        with open(file_path + "/weather.json", "w", encoding="utf-8") as f:
            json.dump(weather_list, f, ensure_ascii=False, indent=2)

def run():
    city_list = get_city_list()
    url = get_city_url(city_list)
    # url = "http://www.weather.com.cn/weather/101010200.shtml"
    if url:
        #发起响应
        content = get_response(url)
        #提取数据
        weather_list = get_content_list(content)
        #保存
        save_json(weather_list)
    else:
        print("你输入的城市地区名有误, 未获取到需要的天气.....")

if __name__ == '__main__':
    run()
```

服务器运行:

```
(python_best) C:\Users\ASUS\Desktop\html\python_best>python -m http.server 8090
Serving HTTP on 0.0.0.0 port 8090 (http://0.0.0.0:8090/) ...
127.0.0.1 - - [12/May/2021 10:51:31] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [12/May/2021 10:51:31] code 404, message File not found
127.0.0.1 - - [12/May/2021 10:51:31] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [12/May/2021 11:02:07] "GET /spider HTTP/1.1" 301 -
127.0.0.1 - - [12/May/2021 11:02:07] "GET /spider/ HTTP/1.1" 200 -
```

weather_echarts.html

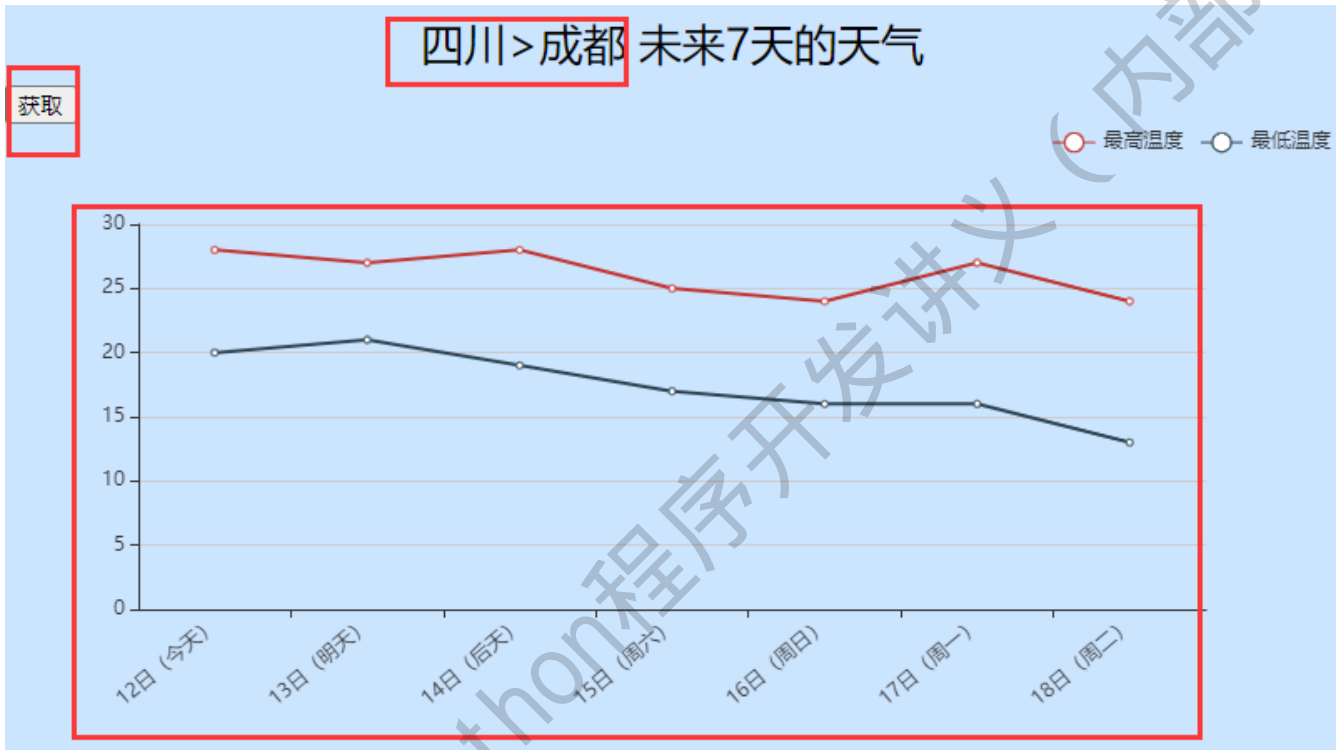
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>weather</title>
  <style>
    .wea-box{
      width: 800px;
      height: 450px;
      margin: 100px auto;
      background-color: rgba(153,204,255,0.5);
    }
    .wea-tit{
      font-size: 26px;
      text-align: center;
      line-height: 50px;
    }
    .wea-map{
      height: 350px;
      /*border: 2px solid red;*/
    }
  </style>
</head>
<body>
<div class="wea-box">
  <!-- 地址-->
  <div class="wea-tit" >
    <span id="wea_tit"></span> 未来7天的天气
  </div>
  <input type="button" value="获取" id="btn">
  <!-- 可视化-->
  <div class="wea-map" id="wea_map"></div>
</div>
<script src="jquery.min.js"></script>
<script src="echarts.min.js"></script>
<script>
  $("#btn").click(function(){
    var wea_map = document.getElementById("wea_map");
    var my_map = echarts.init(wea_map);
    var wea_date = [];
```

```
var max_temp = [];  
var min_temp = [];  
$.get("./source/weather.json",function (weather_list) {  
    // console.log(weather_list);  
    $("#wea_tit").html(weather_list[0]["address"]);  
    for (var i=0;i<weather_list.length;i++){  
        wea_date.push(weather_list[i]["week_date"]);  
        max_temp.push(weather_list[i]["max_temp"]);  
        min_temp.push(weather_list[i]["min_temp"]);  
    }  
    console.log(wea_date);  
    console.log(max_temp);  
    var option1 = {  
    xAxis:{  
        data:wea_date,  
        // 斜体显示  
        axisLabel:{  
            rotate:40  
        }  
    },  
    // 图例  
    legend:{  
        left:"right"  
    },  
    yAxis:{  
  
    },  
    series:[{  
        name:"最高温度",  
        type:"line",  
        data:max_temp  
    },  
    {  
        name:"最低温度",  
        type:"line",  
        data: min_temp  
    }  
    ]  
};  
my_map.setoption(option1);  
});  
})  
</script>  
</body>  
</html>
```

运行结果:


```
C:\Users\ASUS\Desktop\html\python_bes
C:/Users/ASUS/Desktop/html/python_bes
请输入你要查询的地区天气: 成都
正则搜索你要查询 成都 的天气....

Process finished with exit code 0
```



项目分析:

重点是在: 爬虫--》服务器--》可视化, 实现流程提别重要, 需要能熟练使用XPath语句, 网页分析和Echars绘图语法。

2.2.3.7 习题

编程题:

使用Python爬虫技术, 获取网站: <https://www.runoob.com/try/xml/books.xml>, 中class=en的title节点文字和price节点中的文字, 并保存成JSON文件, 开启服务器, 使用Echarts绘制柱状图。