**《Python高级应用》实验报告**

实验名称 ： 实验5网络爬虫

实验日期 ： 　 2024/11/11

姓 名 ： 　高心阳

学 号 ： 　 084623237

班 级 ： 　计算机232

成 绩 ：

**人工智能与信息技术学院**

**南京中医药大学**

|  |
| --- |
| **实验目的：** |
| 1. 熟悉request和beautifulsoup库的使用 2. 将网络爬虫结合到实际应用中 |
| **实验内容和要求** |
| 1. request库的练习：   尝试爬取百度网页、京东商品网页或亚马逊商品网页中的任意两个网页，并展示网页的前200个字符。  （\*在遭遇反爬虫，爬取失败时，尝试替换：  headers = {  'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 Safari/537.36' }  url = 'https://\*\*\*' r=requests.get(url, headers=headers) 等）   1. beautifulsoup库的练习：   （1） 阅读了解html标签对照表；  （2） 获取某天气网站(有相关天气信息等网站即可)上爬取包含七天南京天气的网页，使用bs库中的prettify()函数友好展示页面；  （3） 爬取未来七天南京的天气情况并保存到excel中。   1. 尝试爬取<https://xxgk.njucm.edu.cn/>网站中的“受理部门”、“联系电话”和“电子邮箱”的具体信息及相关的南京中医药大学的相关图片保存到本地。 2. 验证课件上的正则式相关代码。 |
| **运行结果（写清题号）** |
| |  | | --- | | **问题一** | | **代码** | | import requests url='https://www.baidu.com' response=requests.get(url) print(response.text[:200])  print('-------------------') url='https://jd.com' response=requests.get(url) print(response.text[:200]) | | **结果** | | <!DOCTYPE html>  <!--STATUS OK--><html> <head><meta http-equiv=content-type content=text/html;charset=utf-8><meta http-equiv=X-UA-Compatible content=IE=Edge><meta content=always name=referrer><link re  -------------------  <!DOCTYPE html>  <html>  <head>  <meta charset="utf8" version='1'/>  <title>京东(JD.COM)-正品低价、品质保障、配送及时、轻松购物！</title>  <meta name="viewport" content="width=device-width, initial-scale=1.0, maxim |  |  | | --- | | 问题二 | | 代码 | | import requests from bs4 import BeautifulSoup import pandas as pd  url = "https://www.weather.com.cn/weather/101190101.shtml" headers = {  "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.75 Safari/537.36" } response = requests.get(url, headers=headers) response.encoding = 'utf-8' soup = BeautifulSoup(response.text, "html.parser") forecast = [] ul = soup.find('ul', class\_='t clearfix') if ul:  lis = ul.find\_all('li')  for li in lis[:7]:  date = li.find('h1').get\_text()  weather = li.find('p', class\_='wea').get\_text()  temp\_high = li.find('p', class\_='tem').find('span')  temp\_high = temp\_high.get\_text().replace('℃', '')  if temp\_high  else ''  temp\_low=li.find('p',lass\_='tem').find('i').get\_text().replace('℃', '')  wind = li.find('p', class\_='win').find('span')['title']  wind\_speed = li.find('p', class\_='win').find('i').get\_text()   forecast.append({  "日期": date,  "天气": weather,  "最高温度(℃)": temp\_high,  "最低温度(℃)": temp\_low,  "风向": wind,  "风速": wind\_speed  })  df = pd.DataFrame(forecast) df.to\_excel("南京未来七天天气预报.xlsx", index=False) print("数据已保存到文件中") | | 结果 | | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **日期** | | **天气** | | **最高温度(℃)** | | **最低温度(℃)** | | **风向** | | **风速** | |  | 11日（今天） | | 晴 | |  | | 11 | | 东风 | <3级 | |  | 12日（明天） | | 多云 | | 22 | | 13 | | 东风 | 3-4级 | |  | 13日（后天） | | 多云 | | 22 | | 15 | | 东南风 | 4-5级转3-4级 | |  | 14日（周四） | | 多云转阵雨 | | 26 | | 17 | | 东北风 | <3级 | |  | 15日（周五） | | 阵雨 | | 24 | | 18 | | 东北风 | <3级 | |  | 16日（周六） | | 阴 | | 23 | | 16 | | 北风 | <3级 | |  | 17日（周日） | | 多云 | | 16 | | 11 | | 北风 | 4-5级 | |  |  | | --- | | **问题三** | | **代码** | | import requests from bs4 import BeautifulSoup import pandas as pd import os from urllib.parse import urljoin  url = "https://xxgk.njucm.edu.cn/" response = requests.get(url) response.encoding = 'utf-8'  soup = BeautifulSoup(response.text, "html.parser") department\_data = [] tables=soup.find\_all('table',{'background':'/\_upload/tpl/01/c0/448/template448/images/sliderbg.png'})  for table in tables:  dept\_info = {}  header = table.find('td', style="font-size:12px; font-weight:bold;")  dept\_info['部门名称'] = header.get\_text(strip=True) if header else ""  p = table.find('p')  if p:  lines = p.get\_text(separator='\n', strip=True).split('\n')  for line in lines:  if '：' in line:  key, value = line.split('：', 1)  key = key.strip()  value = value.strip()  dept\_info[key] = value  department\_data.append(dept\_info)  df = pd.DataFrame(department\_data) df.to\_excel('南京中医药大学信息公开部门信息.xlsx', index=False) print("信息已保存到 '南京中医药大学信息公开部门信息.xlsx' 文件中。")  image\_tags = soup.find\_all('img')  image\_urls = [img['src'] for img in image\_tags if 'src' in img.attrs] os.makedirs('images', exist\_ok=True) for i, img\_url in enumerate(image\_urls):  img\_url = urljoin(url, img\_url)  img\_response = requests.get(img\_url, stream=True)  if img\_response.status\_code == 200:  ext = os.path.splitext(img\_url)[1]  with open(f'images/image\_{i + 1}{ext}', 'wb') as f:  for chunk in img\_response.iter\_content(1024):  f.write(chunk)  print(f"已下载 image\_{i + 1}{ext}")  else:  print(f"下载 {img\_url} 失败") | | **结果** | | **信息已保存到 '南京中医药大学信息公开部门信息.xlsx' 文件中。**  **已下载 image\_1.jpg**  **（省略……）**  **已下载 image\_67.gif**  **下载 https://xxgk.njucm.edu.cn/（省略……）失败**  **（省略……）**  **下载https://xxgk.njucm.edu.cn/（省略……） 失败**   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **部门名称** | **受理部门** | **办公时间** | **联系电话** | **传真号码** | **电子邮箱** | **办公地址** | **邮政编码** | | 信息公开办公室 | 信息公开办公室 | 9:00—12:00、13:30—17:00 | 025-85811001 | 025-85811006 | xiaoban@njucm.edu.cn | 南京市仙林大道138号 | 210023 | | 信息公开监督检查办公室 | 信息公开监督检查办公室 | 9:00—12:00、13:30—17:00 | 025—85811025 |  | jiwei@njucm.edu.cn | 南京市仙林大道138号 | 210023 | |  |  | | --- | | **问题四** | | **代码** | | import re reg=r"\d" m=re.search(reg,"abc123cd") print(m)  reg=r"b\d+" m=re.search(reg,"a12b123c") print(m)  reg=r"ab+" m=re.search(reg,"acabc") print(m) reg=r"ab\*" m=re.search(reg,"acabc") print(m)  reg=r"ab?" m=re.search(reg,"abbcabc") print(m)  s="xaxby" m=re.search(r"a.b",s) print(m)  s="xaabababy" m=re.search(r"ab|ba",s) print(m)  reg=r"a\nb?" m=re.search(reg,"ca\nbcabc") print(m)  reg=r"car\b" m=re.search(reg,"The car is black") print(m)  reg=r"x[0-9]y" m=re.search(reg,"xyx2y") print(m)  reg=r"x[^ab0-9]y" m=re.search(reg,"xayx2yxcy") print(m) | | **结果** | | <re.Match object; span=(3, 4), match='1'>  <re.Match object; span=(3, 7), match='b123'>  <re.Match object; span=(2, 4), match='ab'>  <re.Match object; span=(0, 1), match='a'>  <re.Match object; span=(0, 2), match='ab'>  <re.Match object; span=(1, 4), match='axb'>  <re.Match object; span=(2, 4), match='ab'>  <re.Match object; span=(1, 4), match='a\nb'>  <re.Match object; span=(4, 7), match='car'>  <re.Match object; span=(2, 5), match='x2y'>  <re.Match object; span=(6, 9), match='xcy'> | |
| **实验的体会与建议** |
|  |