

# 温州大学瓯江学院

WENZHOU UNIVERSITY OUJIANG COLLEGE

## 《Python 网络爬虫》大作业

题    目：    《Python 网络爬虫》大作业

分    院：    理工分院

班    级：    16 计算机科学与技术三

姓    名：    应苗苗

学    号：    16219111302

完成日期：    2019 年 06 月 16 日

温州大学瓯江学院教务部

二〇一二年十一月制

# 目 录

1. 项目需求.....	1
2. 相关技术简介.....	2
2.1 Selenium.....	2
2.2 Scrapy.....	2
2.3 Redis.....	2
2.4 Cookie.....	2
2.5 深度优先.....	2
2.6 广度优先.....	2
3. 软件设计及关键代码.....	3
3.1 12306 自动登录.....	3
3.2 Scrapy.....	5
3.3 Redis 分布式爬虫.....	8
3.4 解决中文乱码.....	9
3.5 百度百科.....	10

# 1. 项目需求

1. 12306 自动登陆
2. 使用 scrapy 框架爬取豆瓣电影 top250 数据存入 CSV 文件
3. Redis 分布式爬虫
4. 解决中文乱码
5. 百度百科深度优先的递归爬虫和广度优先的多线程爬虫

Github 账号:mlgmya

项目地址: <https://github.com/mlgmya/Python.git>

## 2. 相关技术简介

### 2.1 Selenium

Selenium 是一个用于测试网站的自动化测试工具，支持各种浏览器包括 Chrome、Firefox、Safari 等主流界面浏览器，同时也支持 phantomJS 无界面浏览器。

### 2.2 Scrapy

Scrapy, Python 开发的一个快速、高层次的屏幕抓取和 web 抓取框架，用于抓取 web 站点并从页面中提取结构化的数据。Scrapy 用途广泛，可以用于数据挖掘、监测和自动化测试。

### 2.3 Redis

Redis 是一个开源的使用 ANSI C 语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value 数据库，并提供多种语言的 API。从 2010 年 3 月 15 日起，Redis 的开发工作由 VMware 主持。从 2013 年 5 月开始，Redis 的开发由 Pivotal 赞助。

### 2.4 Cookie

Cookie，有时也用其复数形式 Cookies，指某些网站为了辨别用户身份、进行 session 跟踪而储存在用户本地终端上的数据（通常经过加密）。定义于 RFC2109 和 2965 中的都已废弃，最新取代的规范是 RFC6265。（可以叫做浏览器缓存）

### 2.5 深度优先

深度优先搜索，是图论中的经典算法。其利用深度优先搜索算法可以产生目标图的相应拓扑排序表，利用拓扑排序表可以方便的解决很多相关的图论问题，如最大路径问题等等。

### 2.6 广度优先

广度优先遍历是连通图的一种遍历策略。因为它的思想是从一个顶点 V0 开始，辐射状地优先遍历其周围较广的区域，故得名。

## 3. 软件设计及关键代码

### 3.1 12306 自动登录

基本步骤：输入用户名和密码

下载验证码

破解验证码

点击验证码

登录成功

首先使用 selenium 打开 12306 网站，根据用户名和密码的 id 自动输入，验证码通过其 id 中 src 值下载并保存，然后打开 <http://littlebigluo.qicp.net:47720/> 页面，通过窗口句柄切换所要操作的页面，输入保存图片的路径并点击验证，获取验证码的正确答案，然后返回 12306 页面，切换窗口，根据所获的验证码答案依次点击对应的图片所在位置，当验证出错时，可尝试五次。验证后点击登陆，登录成功。

扫码登录

账号登录

18758622377

●●●●●●●●●●

请点击下图中所有的鞭炮

刷新



立即登录

[注册12306账号](#) | [忘记密码?](#)

请点击下图中的所有鞭炮

刷新



扫码登录

账号登录

18758622377

密码输入框

请点击下图中的所有鞭炮

刷新



立即登录

[注册12306账号](#) | [忘记密码?](#)

扫码登录

账号登录

用户名 / 邮箱 / 手机号

密码

验证成功, 跳转中...

刷新



恭喜! 完成验证。

立即登录

[注册12306账号](#) | [忘记密码?](#)

## 3.2 Scrapy

基本步骤:

1. 创建一个 scrapy 项: `scrapy startproject DBmovie`

2. 定义 Item:

```
import scrapy

class DoubanMovieItem(scrapy.Item):
    ranking = scrapy.Field() #排名
    movie_name = scrapy.Field() #电影名称
    score = scrapy.Field() #评分
```

4. 编写爬虫 spider:

打开豆瓣电影网页, 循环分别获取电影排名、名称、评分。当存在下一页链接时, 跳转至下一页, 直至全部获取。

```
from DBmovie.items import DoubanMovieItem
import scrapy

class DoubanMovieTop250spider(scrapy.Spider):
    name = 'douban_movie_top250'
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36',
    }
```

```
def start_requests(self):
    url = 'https://movie.douban.com/top250'
    yield scrapy.Request(url, headers=self.headers)
```

```
def parse(self, response):
    item = DoubanMovieItem()
    movies = response.xpath('//ol[@class="grid_view"]/li')
    for movie in movies:
        item['ranking'] = movie.xpath('//div[@class="pic"]/em/text()').extract()[0]
        item['movie_name'] = movie.xpath('//div[@class="hd"]/a/span[1]/text()').extract()[0]
        item['score'] = movie.xpath('//div[@class="star"]/span[@class="rating_num"]/text()').extract()[0]
    yield item
```

```
next_url = response.xpath('//span[@class="next"]/a/@href').extract() #获取下一页链接
if next_url:
    next_url = 'https://movie.douban.com/top250' + next_url[0]
```

```
yield scrapy.Request(next_url, headers=self.headers)
```

#### 4. 设置配置文件

```
BOT_NAME = 'DBmovie'

SPIDER_MODULES = ['DBmovie.spiders']

NEWSPIDER_MODULE = 'DBmovie.spiders'

ITEM_PIPELINES = {
    'DBmovie.pipelines.DbmoviePipeline': 300,
}
```

#### 5. 爬取保存至 CSV 文件

Cd DBmovie

Scrapy crawl douban\_movie\_top250

```
# -*- coding: utf-8 -*-

# Define your item pipelines here
#
# Don't forget to add your pipeline to the ITEM_PIPELINES setting
# See: https://doc.scrapy.org/en/latest/topics/item-pipeline.html
```

```
import os
import csv
```

```
class DbmoviePipeline(object):
    def __init__(self):
        # csv 文件的位置,无需事先创建
        store_file = "D:\\xx\\爬虫\\实验\\dzy\\scrapy\\DBmovie\\movies.csv"
        print("*****")
        # 打开(创建)文件
        self.file=open(store_file,'a+',encoding='utf-8',newline='')
        # csv 写法
        self.writer = csv.writer(self.file, dialect="excel")
```

```
def process_item(self, item, spider):
    # 判断字段值不为空再写入文件
    print("正在写入.....")
    if item['ranking']:
        # 主要是解决存入 csv 文件时出现的每一个字以', '隔离
        self.writer.writerow([item['ranking'],item['movie_name'],item['score']])
    return item
```

```
def close_spider(self, spider):
    # 关闭爬虫时顺便将文件保存退出
    self.file.close()
```



```

PS D:\xx\爬虫\实验\dzy\scrapy\DBmovie> scrapy crawl douban_movie_top250
2019-06-14 14:08:46 [scrapy.utils.log] INFO: Scrapy 1.6.0 started (bot: DBmovie)
2019-06-14 14:08:46 [scrapy.utils.log] INFO: Versions: lxml 4.3.2.0, libxml2 2.9.5, cssselect 1.0.3, parsel 1.0.0, Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AMD64)], pyOpenSSL 1.0.2, cryptography 2.6.1, Platform Windows-10-10.0.10586-SP0
2019-06-14 14:08:46 [scrapy.crawler] INFO: Overridden settings: {'SPIDER_MODULES': ['DBmovie.spiders'], 'NEWSPIDER_MODULE': 'DBmovie.spiders'}
2019-06-14 14:08:46 [scrapy.extensions.telnet] INFO: Telnet Password: 6f402a702c7e14d7
2019-06-14 14:08:46 [scrapy.middleware] INFO: Enabled extensions:
['scrapy.extensions.logstats.LogStats',
'scrapy.extensions.corestats.CoreStats',
'scrapy.extensions.telnet.TelnetConsole']
2019-06-14 14:08:48 [scrapy.middleware] INFO: Enabled downloader middlewares:
['scrapy.downloadermiddlewares.robotstxt.RobotsTxtMiddleware',
'scrapy.downloadermiddlewares.httpauth.HttpAuthMiddleware',
'scrapy.downloadermiddlewares.downloadtimeout.DownloadTimeoutMiddleware',
'scrapy.downloadermiddlewares.defaultheaders.DefaultHeadersMiddleware',
'scrapy.downloadermiddlewares.useragent.UserAgentMiddleware',
'scrapy.downloadermiddlewares.retry.RetryMiddleware',
'scrapy.downloadermiddlewares.redirect.MetaRefreshMiddleware',
'scrapy.downloadermiddlewares.httpcompression.HttpCompressionMiddleware',
'scrapy.downloadermiddlewares.redirect.RedirectMiddleware',
'scrapy.downloadermiddlewares.cookies.CookiesMiddleware',
'scrapy.downloadermiddlewares.httpproxy.HttpProxyMiddleware']

```

WPS | movies.csv

文件 | 开始

剪切 | 复制 | 格式刷 | 宋体 | 11 | B | I | U | 表格 | 插入 | 数据 | 窗口 | 帮助

	A	B	C	D
1	1	肖申克的救	9.6	
2	2	霸王别姬	9.6	
3	3	这个杀手不	9.4	
4	4	阿甘正传	9.4	
5	5	美丽人生	9.5	
6	6	泰坦尼克号	9.3	
7	7	千与千寻	9.3	
8	8	辛德勒的名	9.5	
9	9	盗梦空间	9.3	
10	10	忠犬八公的	9.3	
11	11	机器人总动	9.3	
12	12	三傻大闹宝	9.2	
13	13	海上钢琴师	9.2	
14	14	放牛班的春	9.3	
15	15	楚门的世界	9.2	
16	16	大话西游之	9.2	
17	17	星际穿越	9.2	
18	18	龙猫	9.2	
19	19	教父	9.3	
20	20	熔炉	9.3	
21	21	无间道	9.1	
22	22	肖申克的救	9.6	

### 3.3 Redis 分布式爬虫

基本步骤:

1. 下载 redis, 启动 redis 服务, 安装 redis 库

2. 加入任务队列

首先创建 `r=Redis(host='10.218.107.18',port=6379,password='redisredis')` 并连接到 redis 服务器, 然后使用 `r.keys('*')` 将 redis 服务器中所有 keys 都打印出来。接着读取网站地址, 对于 `link_list` 的每一个连接, 通过 `requests` 和 `BeautifulSoup` 获取其中的图片链接, 然后使用 `r.lpush('img_url',img_url)` 将链接注入 redis 数据库中, 最后 `r.llen('img_url')` 输出当前图片 url 数量。

3. 读取任务队列并下载图片

首先连接 redis 服务器, 然后使用 `url=r.lpop('img_url')` 获取队列中的图片链接, 接着使用 `while` 循环对每一张图片链接使用 `requests` 获取图片并保存下来。

Master.py

```
thon.exe
' 'c:\Users\dre\.vscode\extensions\ms-python.python-2019.5.18875\pythonFiles\ptvsd_launcher.py' --port '61644' 'd:\xx\爬虫\实验\dzy\redis\master.py'
开始分布爬虫
[b'img_url']
加入的图片url: //www.baidu.com/img/bd_logo1.png
加入的图片url: //www.baidu.com/img/bd_logo1.png?qua=high
加入的图片url: //www.baidu.com/img/baidu_jgylogo3.gif
加入的图片url: //www.baidu.com/img/baidu_resultlogo@2.png
现在图片链接的个数为 1488
加入的图片url: //mat1.gtimg.com/pingjs/ext2020/qindex2018/dist/img/qc_logo_2x.png
加入的图片url: //mat1.gtimg.com/pingjs/ext2020/test2017/netwatch.png
加入的图片url: //img1.gtimg.com/ninja/2/2018/10/ninja153907290259802.png
加入的图片url: //img1.gtimg.com/ninja/2/2018/10/ninja153907291410277.png
加入的图片url: //inews.gtimg.com/newsapp_ls/0/9333223318_640330/0
加入的图片url: //inews.gtimg.com/newsapp_ls/0/9332469959_640330/0
加入的图片url: //inews.gtimg.com/newsapp_ls/0/9325256313_640330/0
加入的图片url: //inews.gtimg.com/newsapp_ls/0/9331734357_150120/0
```

Slave.py

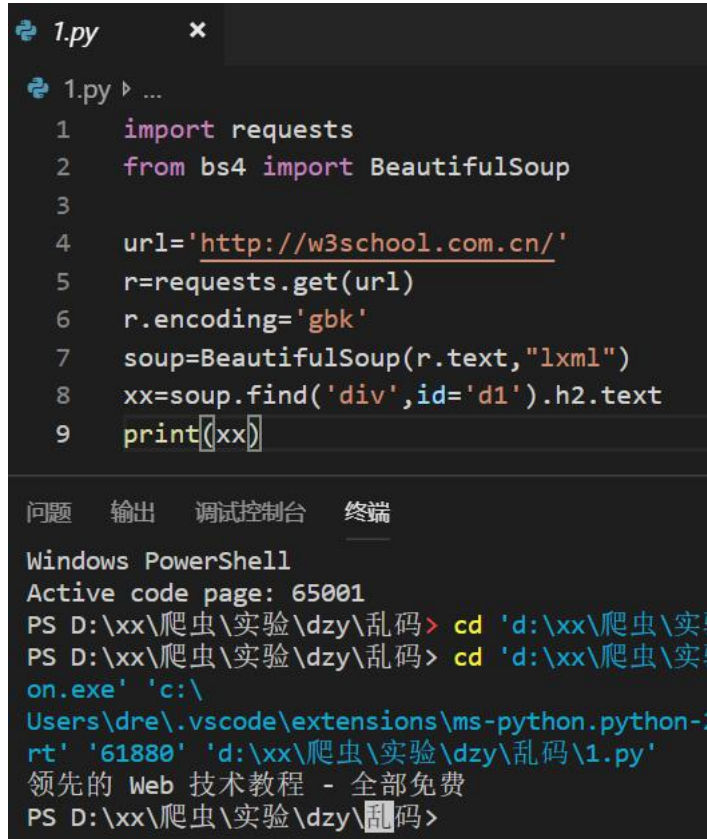
```
PS D:\xx\爬虫\实验\dzy\redis> cd 'd:\xx\爬虫\实验\dzy\redis'; $(env:PYTHON=python.exe)
thon.exe
' 'c:\Users\dre\.vscode\extensions\ms-python.python-2019.5.18875\pythonFiles\ptvsd_launcher.py' --port '61670' 'd:\xx\爬虫\实验\dzy\redis\slave.py'
开始分布爬虫
http://mat1.gtimg.com/www/images/qc2012/gswj2015.jpg
已经获取图片 http://mat1.gtimg.com/www/images/qc2012/gswj2015.jpg
http://mat1.gtimg.com/www/images/qc2012/cxrz5.png
已经获取图片 http://mat1.gtimg.com/www/images/qc2012/cxrz5.png
http://mat1.gtimg.com/www/images/qc2012/wmlogo.gif
已经获取图片 http://mat1.gtimg.com/www/images/qc2012/wmlogo.gif
http://mat1.gtimg.com/www/images/qc2012/buliang.png
已经获取图片 http://mat1.gtimg.com/www/images/qc2012/buliang.png
http://mat1.gtimg.com/www/images/qc2012/ind36.gif
已经获取图片 http://mat1.gtimg.com/www/images/qc2012/ind36.gif
http://mat1.gtimg.com/www/qc2018/imgs/default_b.png
已经获取图片 http://mat1.gtimg.com/www/qc2018/imgs/default_b.png
http://mat1.gtimg.com/www/qc2018/imgs/default_b.png
已经获取图片 http://mat1.gtimg.com/www/qc2018/imgs/default_b.png
```



### 3.4 解决中文乱码

#### 1. 获取网站的中文显示乱码

R. encoding='gb2312'



```

1.py
1.py ▾ ...
1 import requests
2 from bs4 import BeautifulSoup
3
4 url='http://w3school.com.cn/'
5 r=requests.get(url)
6 r.encoding='gbk'
7 soup=BeautifulSoup(r.text,"lxml")
8 xx=soup.find('div',id='d1').h2.text
9 print(xx)

问题 输出 调试控制台 终端

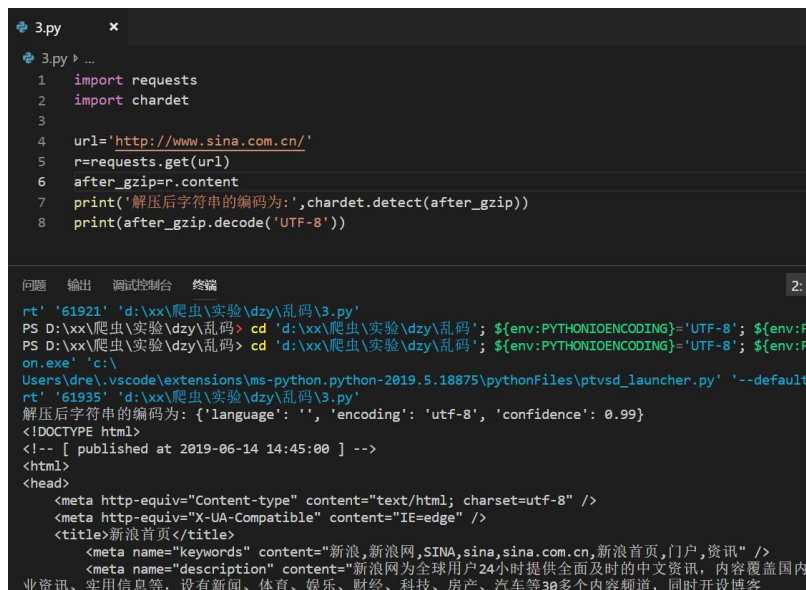
Windows PowerShell
Active code page: 65001
PS D:\xx\爬虫\实验\dzy\乱码> cd 'd:\xx\爬虫\实验\dzy\乱码'
PS D:\xx\爬虫\实验\dzy\乱码> cd 'd:\xx\爬虫\实验\dzy\乱码'
on.exe' 'c:\
Users\dre\.vscode\extensions\ms-python.python-2
rt' '61880' 'd:\xx\爬虫\实验\dzy\乱码\1.py'
领先的 Web 技术教程 - 全部免费
PS D:\xx\爬虫\实验\dzy\乱码>
  
```

#### 2. 非法字符抛出异常

Str1.decode('GBK','ignore')

#### 3. 网页使用 gzip 压缩

R.content



```

3.py
3.py ▾ ...
1 import requests
2 import chardet
3
4 url='http://www.sina.com.cn/'
5 r=requests.get(url)
6 after_gzip=r.content
7 print('解压后字符串的编码为:',chardet.detect(after_gzip))
8 print(after_gzip.decode('UTF-8'))

问题 输出 调试控制台 终端

rt' '61921' 'd:\xx\爬虫\实验\dzy\乱码\3.py'
PS D:\xx\爬虫\实验\dzy\乱码> cd 'd:\xx\爬虫\实验\dzy\乱码'
PS D:\xx\爬虫\实验\dzy\乱码> cd 'd:\xx\爬虫\实验\dzy\乱码'
on.exe' 'c:\
Users\dre\.vscode\extensions\ms-python.python-2019.5.18875\pythonFiles\ptvsd_launcher.py' '--default
rt' '61935' 'd:\xx\爬虫\实验\dzy\乱码\3.py'
解压后字符串的编码为: {'language': '', 'encoding': 'utf-8', 'confidence': 0.99}
<!DOCTYPE html>
<!-- [ published at 2019-06-14 14:45:00 ] -->
<html>
<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <title>新浪首页</title>
  <meta name="keywords" content="新浪,新浪网,SINA,sina,sina.com.cn,新浪首页,门户,资讯" />
  <meta name="description" content="新浪网为全球用户24小时提供全面及时的中文资讯,内容覆盖国内
  业资讯、实用信息等,设有新闻、体育、娱乐、财经、科技、房产、汽车等30多个内容频道,同时开设博客
  
```

## 4. 读写文件的中文乱码

```

4.py
4.py ▸ ...
1  import json
2
3  result=open('test_ANSI.txt','r').read()
4  print(result)
5
6  re=open('test_utf8.txt','r',encoding='UTF-8').read()
7  print(re)
8
9  title='我们'
10 with open('title.txt','a+',encoding='UTF-8') as f:
11     f.write(title)
12     f.close()
13
14 t='我们love你们'
15 with open('t.json','w',encoding='UTF-8') as f:
16     json.dump([t],f,ensure_ascii=False)

```

问题 输出 调试控制台 终端

```

rt' '61962' 'd:\xx\爬虫\实验\dzy\乱码\4.py'
PS D:\xx\爬虫\实验\dzy\乱码> cd 'd:\xx\爬虫\实验\dzy\乱码'; ${env:
PS D:\xx\爬虫\实验\dzy\乱码> cd 'd:\xx\爬虫\实验\dzy\乱码'; ${env:
on.exe' 'c:\
Users\dre\.vscode\extensions\ms-python.python-2019.5.18875\pythonF
rt' '61978' 'd:\xx\爬虫\实验\dzy\乱码\4.py'
PS D:\xx\爬虫\实验\dzy\乱码> cd 'd:\xx\爬虫\实验\dzy\乱码'; ${env:
on.exe' 'c:\
Users\dre\.vscode\extensions\ms-python.python-2019.5.18875\pythonF
rt' '61991' 'd:\xx\爬虫\实验\dzy\乱码\4.py'
abc中文
abc中文

```

t - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)  
 ["我们love你们"]

title - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)  
 我们

## 3.5 百度百科

## 1. 深度优先的递归爬虫

定义 exist\_url 列表，用于存放已经爬取的网页，scrappy(url,depth=1)为爬虫的函数，在获取页面的 html 源代码后，可以使用正则表达式提取所有词条链接(link\_list)，并且用 list(set(link\_list))-set(exist\_url)去掉那些已经爬取的链接和重复的链接，得到 unique\_list，每一个新获取的链接都要先保存到 TXT 文件中，再使用递归函数调用，在 scrappy 函数中调用递归 scrappy 访问一条没有访问过的词条链接，直至深度大于或等于 2 为止。

sd - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

No. 1	Depth:1	->体育奖项
No. 2	Depth:1	->动物
No. 3	Depth:1	->交通
No. 4	Depth:1	->地形地貌
No. 5	Depth:1	->科研机构
No. 6	Depth:1	->体育组织
No. 7	Depth:1	->电影
No. 8	Depth:1	->互联网
No. 9	Depth:1	->美容
No. 10	Depth:1	->电视剧
No. 11	Depth:1	->文化人物
No. 12	Depth:1	->旅游
No. 13	Depth:1	->美术
No. 14	Depth:1	->政治
No. 15	Depth:1	->行政区划
No. 16	Depth:1	->体育项目
No. 17	Depth:1	->民族
No. 18	Depth:1	->曲艺
No. 19	Depth:1	->经济人物
No. 20	Depth:1	->自然资源
No. 21	Depth:1	->植物
No. 22	Depth:1	->自然现象
No. 23	Depth:1	->体育设施

## 2. 广度优先的多线程爬虫

首先定义 Crawler 类, 其参数 url 是爬虫的初始词条, threadnum 代表线程数, 然后调用 Crawler 类中的 crawl 函数。

在 Crawler 类的 crawl 函数中, 首先定义 depth 深度为 1, 然后将 url 加入 g\_queueURL 等待爬取的 url 链接列表中。接着进入循环, 当 depth<3 时, 先用 self.downloadAll() 函数使用多线程下载 g\_queueURL 中所有页面的词条链接, 当完成某一层深度所有节点的爬取后, 使用 self.updateQueueURL() 将新下载的所有词条链接加入 g\_queueURL 中。

在 downloadAll() 函数中, 有 URL 可以爬虫的时候, 其中代码的循环会不断创建线程, 知道达到线程数的最大值或爬取了 g\_queueURL 中所有的链接为止。

g\_existURL 会保存爬取过的链接, g\_pages 会保存刚刚爬过网页的 html 源代码, 在 updateQueueURL() 函数中, 从 g\_pages 中获取所有的词条链接, 然后使用 g\_queueURL=list(set(newUrlList)-set(g\_existURL)) 去除重复和已经爬取过的链接, 完成之后, 将深度 depth 加 1, 然后再循环中爬取更深一层的词条链接, 直到等于最大深度为止。

gd - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

No. 1	Thread0	->体育组织
No. 2	Thread0	->各国历史
No. 3	Thread0	->演出
No. 4	Thread0	->医学
No. 5	Thread0	->娱乐
No. 6	Thread0	->政治
No. 7	Thread0	->美容
No. 8	Thread0	->文化人物
No. 9	Thread0	->体育设施
No. 10	Thread0	->虚拟人物
No. 11	Thread0	->话题人物
No. 12	Thread0	->旅游
No. 13	Thread0	->体育奖项
No. 14	Thread0	->法律
No. 15	Thread0	->航空航天
No. 16	Thread0	->曲艺
No. 17	Thread0	->自然资源
No. 18	Thread0	->小说
No. 19	Thread0	->美术
No. 20	Thread0	->地形地貌
No. 21	Thread0	->摄影
No. 22	Thread0	->动物
No. 23	Thread0	->历史著作
No. 24	Thread0	->文物考古
No. 25	Thread0	->行政区划
No. 26	Thread0	->动漫
No. 27	Thread0	->时尚
No. 28	Thread0	->经济