



# 温州大学瓯江学院

WENZHOU UNIVERSITY OUJIANG COLLEGE

## 《爬虫综合实验》

题    目：\_\_\_\_\_ 爬虫 \_\_\_\_\_

分    院：\_\_\_\_\_ 数信分院 \_\_\_\_\_

班    级：\_\_\_\_\_ 16 计算机科学与技术二班 \_\_\_\_\_

姓    名：\_\_\_\_\_ 陈哲 \_\_\_\_\_

学    号：\_\_\_\_\_ 16219111221 \_\_\_\_\_

完成日期：\_\_\_\_\_ 2019 年 6 月 12 日 \_\_\_\_\_

温州大学瓯江学院教务部

二〇一二年十一月制

## 目录

1. 项目需求分析.....	1
1.1. 需求分析.....	1
2. 开发工具与编程语言.....	1
2.1. Python 技术简介.....	1
2.2. SQLite 技术简介.....	1
2.3. MYSQL 技术简介.....	1
3. 相关技术.....	2
3.1. Selenium.....	2
3.2. Scrapy.....	2
4. 编码实现.....	3
4.1. 豆瓣电影爬虫.....	3
4.2. 京东手机爬虫.....	4
4.3. Scrapy 爬虫项目.....	6
4.4. 12306 爬虫项目.....	7
4.5. 百度百科爬虫项目.....	8
4.5.1. 深度优先.....	8
4.5.2. 广度优先.....	9
5. 结果展示.....	10
6. 本文完成的工作.....	13

# 1. 项目需求分析

## 1.1. 需求分析

- 豆瓣电影爬虫：需要爬取榜单电影的名称、时间、国家、类型、评分。
- 京东手机爬虫：需要爬取手机的名字、价格、评价数、超链接和图片链接。
- Scrapy 爬虫项目：利用框架定向爬取新闻网站。
- 12306 爬虫项目：支持自动化订票。
- 百度百科爬虫项目：分别用深度优秀和广度优先。

# 2. 开发工具与编程语言

## 2.1. Python 技术简介

Python 是一种面向对象的解释型计算机程序设计语言，由荷兰人 Guido van Rossum 于 1989 年发明，第一个公开发行版发行于 1991 年。Python 是纯粹的自由软件，源代码和解释器 CPython 遵循 GPL（GNU General Public License）许可。Python 语法简洁清晰，特色之一是强制用空白符（white space）作为语句缩进。Python 具有丰富和强大的库。它常被昵称为胶水语言，能够把用其他语言制作的各种模块（尤其是 C/C++）很轻松地联结在一起。常见的一种应用情形是，使用 Python 快速生成程序的原型（有时甚至是程序的最终界面），然后对其中有特别要求的部分，用更合适的语言改写，比如 3D 游戏中的图形渲染模块，性能要求特别高，就可以用 C/C++ 重写，而后封装为 Python 可以调用的扩展类库。需要注意的是在您使用扩展类库时可能需要考虑平台问题，某些可能不提供跨平台的实现。7 月 20 日，IEEE 发布 2017 年编程语言排行榜：Python 高居首位。

## 2.2. SQLite 技术简介

SQLite 是遵守 ACID 的关系数据库管理系统，它包含在一个相对小的 C 程式库中。它是 D.Richard Hipp 建立的公有领域项目。不像常见的客户端/服务器结构范例，SQLite 引擎不是个程序与之通信的独立进程，而是连接到程序中成为它的一个主要部分。所以主要的通信协议是在编程语言内的直接 API 调用。这在消耗总量、延迟时间和整体简单性上有积极的作用。整个数据库（定义、表、索引和数据本身）都在宿主主机上存储在一个单一的文件中。它的简单的设计是通过在开始一个事务的时候锁定整个数据文件而完成的。

## 2.3. MYSQL 技术简介

MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 旗

下产品。MySQL 最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，它分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。由于其社区版的性能卓越，搭配 PHP 和 Apache 可组成良好的开发环境。

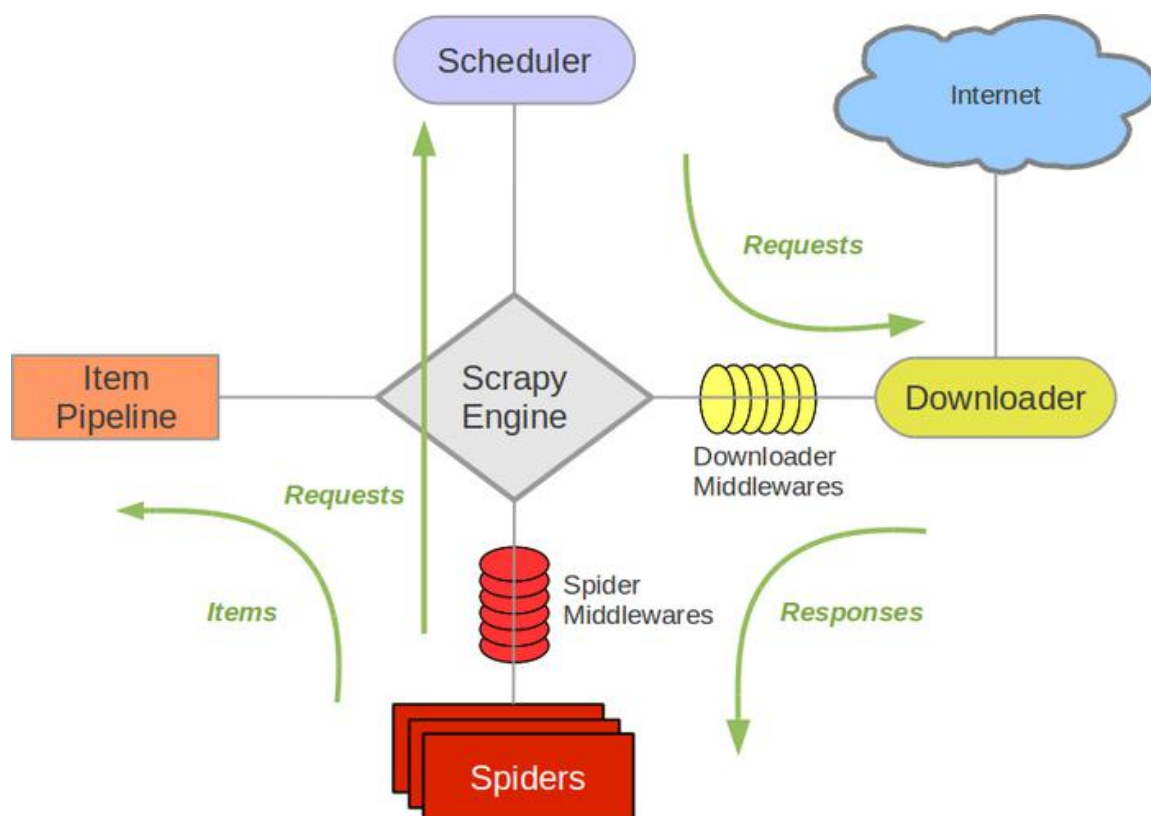
## 3. 相关技术

### 3.1. Selenium

Selenium 是什么？一句话，自动化测试工具。它支持各种浏览器，包括 Chrome，Safari，Firefox 等主流界面式浏览器，如果你在这些浏览器里面安装一个 Selenium 的插件，那么便可以方便地实现 Web 界面的测试。换句话说叫 Selenium 支持这些浏览器驱动。  
成果展示

### 3.2. Scrapy

Scrapy 是用 Python 实现的一个为了爬取网站数据、提取结构性数据而编写的应用框架。Scrapy 常应用在包括数据挖掘，信息处理或存储历史数据等一系列的程序中。通常我们可以很简单的通过 Scrapy 框架实现一个爬虫，抓取指定网站的内容或图片。



- **Scrapy Engine(引擎)**: 负责 Spider、ItemPipeline、Downloader、Scheduler 中间的通讯，信号、数据传递等。
- **Scheduler(调度器)**: 它负责接受引擎发送过来的 Request 请求，并按照一定的方式进行整理排列，入队，当引擎需要时，交还给引擎。
- **Downloader（下载器）**: 负责下载 Scrapy Engine(引擎)发送的所有 Requests 请求，并将其获取到的 Responses 交还给 Scrapy Engine(引擎)，由引擎交给 Spider 来处理，
- **Spider（爬虫）**: 它负责处理所有 Responses,从中分析提取数据，获取 Item 字段需要的数据，并将需要跟进的 URL 提交给引擎，再次进入 Scheduler(调度器)。
- **Item Pipeline(管道)**: 它负责处理 Spider 中获取到的 Item，并进行后期处理（详细分析、过滤、存储等）的地方。
- **Downloader Middlewares（下载中间件）**: 你可以当作是一个可以自定义扩展下载功能的组件。
- **Spider Middlewares（Spider 中间件）**: 你可以理解为是一个可以自定义扩展和操作引擎和 Spider 中间通信的功能组件（比如进入 Spider 的 Responses;和从 Spider 出去的 Requests）

## 4. 编码实现

### 4.1. 豆瓣电影爬虫

获取 html 对象

返回指定 URL 的网页源代码并引用 lxml 解析为 lxml.html.HtmlElement 对象

```
res = requests.get(url).content  
  
text = html.fromstring(res)
```

解析数据

通过 xpath 语法来解析文本获取需要的数据

例如选取电影的名字可通过以下语句

```
i.xpath('div[@class="hd"]/a/span[@class="title"]/text()')[0]
```

获取分页

分析 URL 中 “start” 元素的值等于当前页列表第一位的 TOP 序号减 1。一共 10 页，一页 25 项。分页请求的写法用一个 for 循环来完成：

```
for i in range(0, 10):  
  
    text = my_spider.get_page(i*25)  
  
    top250 += my_spider.parseItem(text)  
  
    print("正在爬取第%d 页"%(i))
```

持久化数据

这里分别写了 MongoDB, Sqlite3 和 MySQL 连接数据库的函数，分别调用了

```
import pymongo, sqlite3, pymysql
```

为了在博客页面上显示，所以需要将展示内容组合成 HTML 代码写入数据库

## 4.2. 京东手机爬虫

每页的商品共有 60 个，页面先加载前 30 个，下滑滚动时再加载后 30 个。

故用 selenium 模拟浏览器下滑操作，再将页面源码给 bs4 进行解析抽取。

初始化浏览器

```
def __init__(self,url):  
  
    #driver 配置  
  
    firefox_options = Options()  
  
    firefox_options.add_argument('--headless')  
  
    firefox_options.add_argument('--disable-gpu')  
  
    self.driver = webdriver.Firefox(firefox_options=firefox_options)
```

```
#等待浏览器加载
```

```
self.driver.implicitly_wait(3)
```

```
self.driver.get(url)
```

动态加载数据

```
def scroll(self):
```

```
#动态加载数据
```

```
self.driver.execute_script("window.scrollTo(0,document.body.scrollHeight);")
```

```
time.sleep(1)#等待数据加载
```

```
soup=BeautifulSoup(self.driver.page_source,'html.parser')
```

```
return soup
```

抽取数据

```
def ParseItem(self,soup):
```

将各个数据封装在字典中并以字典列表的形式返回

```
item = {"title":title,"price":price,"img":img,"comments":comments,"url":url}  
lt.append(item)
```

模拟点击下一页按钮

```
def TurnToNextPage(self):
```

```
self.driver.find_element_by_class_name('pn-next').click()
```

```
self.driver.implicitly_wait(3)
```

通过循环调用解析函数，并将结果累加

```
for i in range(0,2):
```

```
jd.TurnToNextPage()
```

```
goodsList += jd.ParseItem(jd.scroll())
```

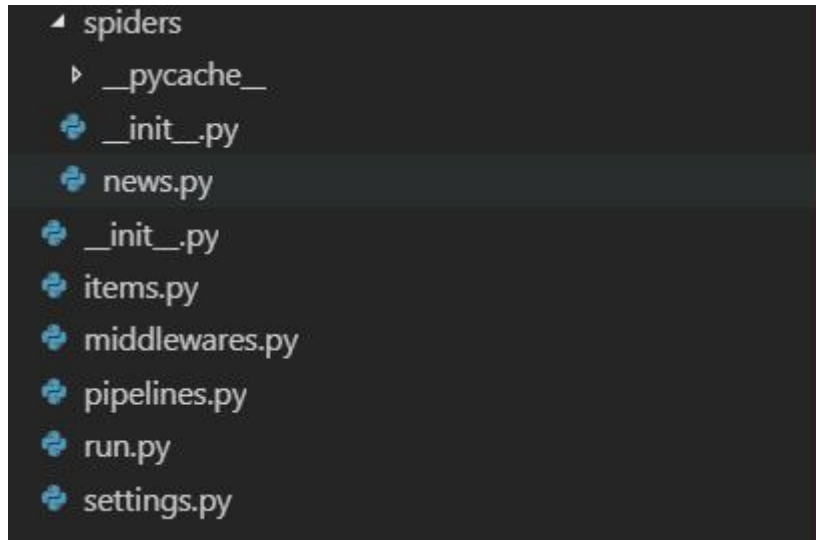
持久化数据

```
#jd.Save2Mongo(goodsList)
```

```
jd.Save2MySql(goodsList)
```

### 4.3. Scrapy 爬虫项目

目录结构如下：



将每一条有效的 Item 插入到 MongoDB 中

```
def process_item(self, item, spider):
    postItem = dict(item) # 把 item 转化成字典形式
    self.coll.insert(postItem) # 向数据库插入一条记录
    self.client.close()
```

定义要爬取信息的关键字段

```
class GetnewsItem(scrapy.Item):
    title = scrapy.Field()
    # 建立 keywd 存储网页关键词
    keywd = scrapy.Field()
    # 建立 keywd 存储网页链接
    link = scrapy.Field()
    # 建立 content 存储网页内容
    content = scrapy.Field()
```

在 settings 文件启用自定义中间件用于随机选择 user-agent

```
DOWNLOADER_MIDDLEWARES = {
    'getNews.middlewares.getNewsUserAgentMiddleware' :400
}
```

并启用 redis 组件用于链接去重，并重新指定 scrapy 的调度器

```
#Redis 组件(去重和调度器组件)
DUPEFILTER_CLASS = "scrapy_redis.dupefilter.RFPDupeFilter"
SCHEDULER = "scrapy_redis.scheduler.Scheduler"
```

启用持久化数据，可以在下次启动时继续爬取

```
SCHEDULER_PERSIST = True
```

编写 Spider 文件 NewsSpider，这里用 self.redis\_key 来替代 self.start\_urls，后续在 redis 中通



过 lpush 命令来推送起始链接。根据 Rule 来指定爬取链接的格式，通过 follow 来判断是否跟进这个链接，如果通过正则匹配，则会执行回调函数

```
class NewsSpider(RedisCrawlSpider):
def __init__(self,rule):
    self.name=rule["name"]
    self.rule=rule
    self.allowed_domains = [rule["allowed_domains"]]
    #self.start_urls = [rule["start_urls"]]
    self.redis_key = "NewsSpider:start_urls"
rule_list = []
rule_list.append(Rule(LinkExtractor(allow=rule["rule"]),
callback='parse_item', follow=True,))
self.rules = tuple(rule_list)
super(NewsSpider, self).__init__()
configure_logging({'LOG_FORMAT': '%(levelname)s: %(message)s'})
```

处理每个 item

```
def parse_item(self, response):
    i = GetnewsItem()
    i['title'] = self.getTitle(response)
    i['keywd'] = self.getKeywd(response)
    i["link"] = response.url
    i['content'] =self.getContent(response)
    if i['content'] is not None:
        yield i
    else:
        pass
```

通过 xpath 来寻去爬取内容

```
def getTitle(self,response):
    title = response.xpath(self.rule["title"]).extract()
    if len(title):
        title=title[0]
    else:
        title = ""
    return title.strip()
```

## 4.4. 12306 爬虫项目

指定 header

```
headers = {
    "User-agent": "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:43.0)
    Gecko/20100101 Firefox/43.0",
    "Accept-Language":
    "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2",
    "Accept-Encoding": "gzip, deflate, br",
```

```
"DNT": "1",
"Content-Type": "application/x-www-form-urlencoded; charset=UTF-8",
"Origin": "https://kyfw.12306.cn",
}
```

从配置文件中读取需要用到的请求参数

```
conf = configparser.ConfigParser()
conf.read("./conf/conf.ini", encoding="utf-8-sig")
```

从验证码接口:将验证码拉取到本地

"https://kyfw.12306.cn/passport/captcha/captcha-image?login\_site=E&module=login&rand=sjrand&0.6523880813900003"

向基于深度学习算法的验证码识别网站返回验证码结果, 这里采用 multipart/form-data 的格式 `url = "http://littlebigluo.qicp.net:47720/"`

验证码验证 接口:Post https://kyfw.12306.cn/passport/captcha/captcha-check 若自动识别验证码多次出错, 就调用手动验证

用户名密码认证 接口:Post https://kyfw.12306.cn/passport/web/login

获取 apptk 接口:Post https://kyfw.12306.cn/passport/web/auth/uamtk

获取登录结果 接口:Post https://kyfw.12306.cn/otn/uamauthclient 这里通过判断是否成功登录

余票查询 接口:Get https://kyfw.12306.cn/otn/leftTicket/query[A-Z]

车票信息确认 接口: Post https://kyfw.12306.cn/otn/leftTicket/submitOrderRequest

初始化预订票接口:Post https://kyfw.12306.cn/otn/confirmPassenger/initDc

获取乘客信息 接口:Post https://kyfw.12306.cn/otn/confirmPassenger/getPassengerDTOs

确认订单信息 接口:Post https://kyfw.12306.cn/otn/confirmPassenger/checkOrderInfo

获取排队信息 接口:Post https://kyfw.12306.cn/otn/confirmPassenger/getQueueCount

进入队伍 接口:Post https://kyfw.12306.cn/otn/confirmPassenger/confirmSingleForQueue

查询抢票结果 接口:Post https://kyfw.12306.cn/otn/confirmPassenger/queryOrderWaitTime

## 4.5. 百度百科爬虫项目

### 4.5.1. 深度优先

定义起始链接, 并显示深度为 5 层

```
start_url =
"https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/5162711"
baike = BaiKe(start_url)
baike.crawl(5)
```

定义先进后出的数据结构——栈

```
class LinkStack:
    def __init__(self):
        self.visited = []
        self.Unvisited = []
```

通过判断当前深度, 防止爬虫深度过深

```
while(self.current_depth < carwl_depth):
```

一次弹出一个待爬栈顶部的 url

```
if not self.linkStack.UnvisitedUrlsEmpty():  
    url_to_visit = self.linkStack.UnvisitedUrlStackPop()
```

获取超链接的 lemmaid, 防止出现同义词, 同时可以作为去重 key

```
lemmaid = url_to_visit[url_to_visit.rfind('/')+1:]
```

## 4.5.2. 广度优先

定义起始链接, 并显示深度为 3 层

```
start_url =  
"https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB/5  
162711"  
baike = BaiKe(start_url)  
baike.crawl(3)
```

定义队列数据结构

```
class LinkQueue:  
    def __init__(self):  
        self.visited = []  
        self.Unvisited = []
```

优先爬取同一深度的链接

```
while(self.current_depth <= crawl_depth):  
    print("当前爬取深度%d"%self.current_depth)  
    tempQueue = []  
    while not self.linkQueue.UnvisitedUrlsEmpty():  
        url_to_visit = self.linkQueue.UnvisitedUrlDeQueue()
```

## 5. 结果展示

Albert's Blog

[首页](#) [博客](#) [关于](#) [登录](#) [🔍](#)

### Selenium爬取京东手机商品列表

JdMobile · 2019年4月26日 08:00 · Albert · 0 评论 · 2 阅读

 <p>¥ 6999.00 三星 Galaxy S10+ 8GB+128GB炭晶黑 (SM- <a href="#">View</a> <a href="#">Edit</a> 2.4万+条评价</p>	 <p>¥ 1298.00 荣耀8X 千元屏霸 91%屏占比 2000万AI双摄 4GB+64GB 幻 <a href="#">View</a> <a href="#">Edit</a> 145万+条评价</p>	 <p>¥ 5699.00 Apple iPhone XR (A2108) 128GB 黑色 移动联通电信4G <a href="#">View</a> <a href="#">Edit</a> 95万+条评价</p>
--	---	--

### 文章目录

- [教程特点](#)
- [谁适合这个教程](#)
- [在线预览](#)
- [资源列表](#)
- [获取帮助](#)

### 最新文章

- > [Selenium爬取京东手机商品列表](#)
- > [code test](#)
- > [豆瓣电影 Top 250](#)
- > [...](#)

Albert's Blog

[首页](#) [博客](#) [关于](#) [登录](#) [🔍](#)

### 豆瓣电影 Top 250

doubanTop250 · 2019年4月24日 08:00 · Albert · 0 评论 · 3 阅读

肖申克的救赎
1994/美国/犯罪 剧情
9.6
霸王别姬
1993/中国大陆 香港/剧情 爱情 同性
9.6

### 文章目录

- [教程特点](#)
- [谁适合这个教程](#)
- [在线预览](#)
- [资源列表](#)
- [获取帮助](#)

### 最新文章

- > [Selenium爬取京东手机商品列表](#)
- > [code test](#)
- > [豆瓣电影 Top 250](#)
- > [...](#)

```

/* 1 */
{
  "id" : ObjectId("5bc5d024b0f09cf47ad3e037"),
  "allowed_domains" : "news.sina.com.cn",
  "start_urls" : "https://news.sina.com.cn/",
  "rule" : "https://news.sina.com.cn/.*/[0-9]{4}.[0-9]{2}.[0-9]{2}.doc.*?shtml",
  "title" : "/html/head/title/text()",
  "keywd" : "/html/head/meta[@name='keywords']/@content",
  "content" : "//div[@class='article']/p",
  "enable" : 1,
  "name" : "\\SinaNews\\"
}

/* 2 */
{
  "id" : ObjectId("5bd9545af85dc7e1dbeaff3d"),
  "allowed_domains" : "thepaper.cn",
  "start_urls" : "https://www.thepaper.cn/",
  "rule" : "www.thepaper.cn/newsDetail_forward_[0-9]{7}",
  "title" : "/html/head/title/text()",
  "keywd" : "/html/head/meta[@name='Keywords']/@content",
  "content" : "//div[@class='news_txt']/p",
  "enable" : 1,
  "name" : "\\thePaperNews\\"
}

/* 3 */
{
  "id" : ObjectId("5bd95a0af85dc7e1dbeaffa7"),
  "allowed_domains" : "new.qq.com",
  "start_urls" : "https://news.qq.com/ninjas/qnews_jinrihuati.htm",
  "rule" : "(.*)[0-9]{8}/[0-9]{8}[A-Z0-9]{6}.html",
  "title" : "/html/head/title/text()",
  "keywd" : "/html/head/meta[@name='keywords']/@content",
  "content" : "//div[@class='content-article']/p",
  "enable" : 0,
  "name" : "\\qqNews\\"
}

/* 4 */
{
  "id" : ObjectId("5bd98bf3f85dc7e1dbeb026f"),
  "allowed_domains" : "news.ifeng.com",
  "start_urls" : "https://news.ifeng.com",
  "rule" : "",
  "title" : "/html/head/meta[@property='og:title']/@content",

```

Mongo localhost:27017 spider

```

db.getCollection('news').find({})

```

news 0.004 sec.

```

/* 1 */
{
  "id" : ObjectId("5c77ee01ba01d7331cccc380e"),
  "title" : "今日话题 | 台湾名嘴赴瑞士安乐死: 安乐死发你想象的那么简单",
  "keywd" : "安乐死,台湾,医生,医患关系,瑞士",
  "link" : "https://new.qq.com/omn/20190228/20190228A14QS0.html",
  "content" : "\n\n特约作者 | 十立生\n台湾名嘴、体育主播傅达仁因晚年罹患胰腺癌，痛苦不堪，选择在瑞士安乐死。近日，他的临终画面曝光，家人陪
}

/* 2 */
{
  "id" : ObjectId("5c77ee03ba01d7331cccc380f"),
  "title" : "今日话题 | 选精生子，一种自由选择还是一颗定时炸弹？",
  "keywd" : "非婚生子女,代孕,冷冻卵子,户口,今日话题",
  "link" : "https://new.qq.com/omn/20190226/20190226A148DE.html",
  "content" : "\n\n特约作者 | 李勤余\n冷冻卵子、代孕等生育话题的争议至今仍未平息，“选精生子”又闯入了公众的视线。“国外一些被领养的小朋友会
}

/* 3 */
{
  "id" : ObjectId("5c77ee05ba01d7331cccc3810"),
  "title" : "今日话题 | 不得询问妇女婚育情况，能解决职场性别歧视吗？",
  "keywd" : "就业,性别歧视,今日话题,女性",
  "link" : "https://new.qq.com/omn/20190227/20190227A17UDF.html",
  "content" : "\n\n又 | 丁阳\n几天前，人社部教育部等九部门联合印发通知，要求企业在招聘时，“不得以性别为由限制妇女求职就业、拒绝录用妇女，不得
}

/* 4 */
{
  "id" : ObjectId("5c77ee07ba01d7331cccc3811"),
  "title" : "今日话题 | “不养狗才能买房”，是缓解人狗矛盾的有益尝试",
  "keywd" : "养狗,买房,物业,业主",
  "link" : "https://new.qq.com/omn/20190225/20190225A16VOD.html",
  "content" : "\n\n又 | 刘又昭\n从木糖醇到异烟肼，再到频繁发生的恶犬伤人……桩桩件件都在提醒人们，城市中人狗矛盾日益突出。怎么办？山西运城的
}

/* 5 */
{
  "id" : ObjectId("5c77ee09ba01d7331cccc3812"),
  "title" : "今日话题 | “无证卖泡面被罚5万”，如此执法真是为了食品安全？",
  "keywd" : "食品安全,罚款,泡面,葱油饼",
  "link" : "https://new.qq.com/omn/20190222/20190222A16IBT.html",
  "content" : "\n\n又 | 刘又昭\n到棋牌室打牌，饿了，买点零食或让店家做点点点，再寻常不过。可最近杭州的一家棋牌室向顾客卖泡面可乐，提供餐饮
}

/* 6 */

```

```

验证码图片下载成功
第1尝试自动识别验证码
验证码答案为:1,验证码校验成功
登录返回结果:
登录成功!
验证码是否登录,得到newapptk
{"apptk":null,"result_message":"验证通过","result_code":0,"newapptk":"TRe9ceHMX71bBL_x1c5LP7F3FRHKbWLF6kTa5_Yu4Yhu1110"}
检查unamuth客户端
{"apptk":"TRe9ceHMX71bBL_x1c5LP7F3FRHKbWLF6kTa5_Yu4Yhu1110","result_code":0,"result_message":"验证通过","username":""}
用户名:
初始化购票页面
购票页面初始化成功!
查询车票数据:出发地:温州南,目的地:杭州东,出发日期:2019-06-20
["CJA15iCPgPXfneUv4p7zUC980DqWaoiCuqFQ%28MTnsx6h5ry2pxbhly3HrEkK9j%28HBDvKJBFkgxLhQ%28Aum1aKH842%28rFVnu7HkepAtv%28FaSaZmPg10EAd1E92ZTtC3j2V6runLFPfgxTKMItOP%28qcGF6Mt3%28%28ZL0V2hZ8V0WzCq380jK26LDoq3JUSVdDUiaqYTC4Kk%28AIW7IX61ebqxcKA2L","预订","5j0006734203","67342","CEH","HGH","VRH","HGH","21:14","23:19","02:05","Y","f50xxdNICfJDX无","","","有","有","有","有","00900000","090M","1","0",""]
检查用户
check_user-->{"validateMessagesShowId": "_validatorMessage","status":true,"httpstatus":200,"data":{"flag":true},"messages":[],"validateMessages":{}}
开始进行订单检查...
{"validateMessagesShowId": "_validatorMessage","status":true,"httpstatus":200,"data":{"N"},"messages":[],"validateMessages":{}}
初始化订单数据
FE8FE6D4C8A52FDF1EF7F9AF50797C8DC3D35AF243A419340637AED
得到校验uid: 794cc9196bclac258228607fa7a45269
获取乘车人信息.....
检查订单信息
{"validateMessagesShowId": "_validatorMessage","status":true,"httpstatus":200,"data":{"ifShowPassCode": "N","canChooseBeds": "N","canChooseSeats": "Y","choose_Seats": "09M","is订单检查成功, 确认订单
检查余票...
{"validateMessagesShowId": "_validatorMessage","status":true,"httpstatus":200,"data":{"count": "0","ticket": "783,0","op_2": "false","count1": "0","op_1": "false"},"messages":[]
最后一次确认订单
{"validateMessagesShowId": "_validatorMessage","status":true,"httpstatus":200,"data":{"submitStatus":true},"messages":[],"validateMessages":{}}
{"validateMessagesShowId": "_validatorMessage","status": True, 'httpstatus': 200, 'data': {'queryOrderWaitTimeStatus': True, 'count': 0, 'waitTime': 4, 'requestId': 65438
{'validateMessagesShowId': '_validatorMessage', 'status': True, 'httpstatus': 200, 'data': {'queryOrderWaitTimeStatus': True, 'count': 0, 'waitTime': -1, 'requestId': 6543

```

中国铁路12306

12306 CHINA RAILWAY

搜索车票/餐饮/常旅客/相关规章

我的12306 您好 退出

首页

车票

团购服务

会员服务

站车服务

商旅服务

出行指南

信息查询

当前位置: 个人中心 > 火车票订单

个人中心

订单中心

火车票订单

候补订单

餐饮-特产

保险订单

我的行程

会员中心

个人信息

查看个人信息

账号安全

手机核验

常用信息管理

常用联系人

车票快递地址

温馨提示

重点旅客预约

遗失物品查找

服务查询

投诉和建议

未完成订单

未出行订单

历史订单

车次信息

旅客信息

席位信息

票价

车票状态

订票日期: 2019-06-09

温州南→杭州东 G7342

2019-06-10 21:14 开

中国居民身份证

二等座 05车12C号

成人票 138.0元

待支付

取消订单

去支付

温馨提示

1.席位已锁定,请在指定时间内完成网上支付。

2.逾期未支付,系统将取消本次交易。

3.在完成支付或取消本订单之前,您将无法购买其他车票。

4.未尽事宜详见《铁路旅客运输规程》等有关规定和车站公告。



dfs @baike (root) - 表					
文件 编辑 查看 窗口 帮助					
导入向导 导出向导 筛选向导 网格查看 表单查看 备注					
id	key	lemmaid	summary		
69	网络爬虫	5162711	的名字还有蚂蚁、自动索引、模拟程序或者蠕虫。		
70	蜘蛛/813	8135707	网络爬虫（又被称为网页蜘蛛，网络机器人，在FC		
71	万维网/2	215515	WWW是环球信息网的缩写，（亦作“Web”、“WV		
72	蚂蚁/977	9770178	蚁科（Formicidae），通称蚂蚁，节肢动物门，多		
73	蠕虫/444	4454380	蠕虫是一种能够利用系统漏洞通过网络进行自我传		
74	网络爬虫	5162711	网络爬虫（又被称为网页蜘蛛，网络机器人，在FC		
75	蜘蛛	8135707	网络爬虫（又被称为网页蜘蛛，网络机器人，在FC		
76	万维网	215515	WWW是环球信息网的缩写，（亦作“Web”、“WV		
77	蚂蚁	9770178	蚁科（Formicidae），通称蚂蚁，节肢动物门，多		
78	蠕虫/444	4454380	蠕虫是一种能够利用系统漏洞通过网络进行自我传		
79	环球信息	9377238	WWW是环球信息网的缩写，（亦作“Web”、“WV		
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。另外一些不常使用的名字还有蚂蚁、自动索引、模拟程序或者蠕虫。					
SELECT * FROM `dfs` LIMIT 0, 100				第 1 条记录 (共 67 条) 于 1 页	