数据解析

内容分类

一般来讲对我们而言，需要抓取的是某个网站或者某个应用的内容，提取有用的价值。内容一般分为两部分

结构化数据

- 先有结构 后有数据
    - json数据
        - 转为python格式进行键值对取值(推荐)
        - jsonpath(不推荐)
        - 正则
    - XML文件(比较少见)
        - 转为python类型进行操作
        - xpath
        - css选择器
        - 正则

非结构化

- 先有数据 后有结构
    - 文本 电话号码 邮箱地址
        - 正则
    - HTML文件
        - 正则
        - xpath(推荐)
        - css选择器

不同类型的数据，我们需要采用不同的方式来处理。(哪种方式能取到值 就可以用哪种 你喜欢哪种就用哪种)

JSON

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式，它使得人们很容易的进行阅读和编写。同时也方便了机器进行解析和生成。适用于进行数据交互的场景，比如网站前台与后台之间的数据交互。

json模块

- import json
- 是内置模块
- json.dumps()
    - 将python的数据类型转成json字符串
    - skipkeys=True
        - 跳过异常过滤
    - ensure_ascii=False
        - 解决编码问题
- json.dump()
    - 将python类型转为JSON并写入文件
- json.loads()
    - 将JSON字符串转为python类型
- json.load()
    - 读取文件中json形式的字符串 转为python

正则

规则表达式

- . 匹配任意字符 除了换行符
- * 匹配0个或者多个
- +匹配一个或者多个
- ？ 匹配0个或一个
- \s 空白字符
- \d 数字
- match方法 查找字符串头部 一次匹配 要找到了一个匹配的结果就返回，而不是查找所有匹配的结果
- findall 查找所有匹配的结果 然后返回列表
- split 方法 按照能够匹配的子串将字符串分割后返回列表
- sub 方法用于替换

贪婪模式和非贪婪模式

- 贪婪模式
    - 在表达式匹配成功的前提下 尽可能多匹配
- 非贪婪模式
    - 在表达式匹配成功的前提下 尽可能少匹配
- 在python中 数量词默认是贪婪
- 量词后面跟? 表示非贪婪匹配 尽可能的少的匹配 +或*后面跟？表示非贪婪
- .*? 表示匹配任意数量的重复，但是在能使整个匹配成功的前提下使用最少的重复。

匹配中文

- 中文的Unicode的编码范围主要在[u4e00-u9fa5]
- 但是范围不是很完整 有些全角标点没有 基本够用

re.S

- 如果不使用re.S参数，则只在每一行内进行匹配，如果一行没有，就换下一行重新开始
- 而使用re.S参数以后，正则表达式会将这个字符串作为一个整体，在整体中进行匹配

XPath

lxml模块

- pip install lxml 下载命令
- lxml 是一个html/xml文件的解析器 主要的功能就是如何提取和解析HTML或者xml的数据

常用规则

- /　　从当前节点选取直接子节点
- //　　从当前节点选取子孙节点
- .　　选取当前节点
- ..　　选取当前节点的父节点
- @　　选取属性

- last() 选取最后一个
- *通配符 匹配任何元素节点
- @* 匹配任何属性节点。
- | 选取若干节点(多个规则去匹配元素)

使用

- from lxml import etree
- tree = etree.HTML(str1)
- tree.xpath(规则)

BS4

pip install beautifulSoup4

Beautiful Soup 是一个可以从HTML或XML文件中提取数据的Python库

BeautifulSoup 用来解析 HTML 比较简单，API非常人性化，支持CSS选择器，Python标准库中的HTML解析器，也支持 lxml 的 XML解析器。

lxml 只会局部遍历，而Beautiful Soup 是基于HTML DOM的，会载入整个文档，解析整个DOM树，因此时间和内存开销都会大很多，所以性能要低于lxml。

使用

- soup = BeautifulSoup(str, 'html.parser')
    - html.parser 是python自带的编辑器，
- 常用规则
    - soup.标签 获取标签
    - soup.标签.attrs 获取标签的属性 所有
    - soup.标签.attrs['指定属性']
    - soup.标签.get('属性值')
    - soup.标签.string
        - string得到标签下的文本内容，只有在此标签下没有子标签，或者只有一个子标签的情况下才能返回其中的内容，否则返回的是None
    - soup.标签名.get_text()

        - get_text()可以获得一个标签中的所有文本内容，包括子孙节点的内容
    - soup.标签名.text
        - 也是获取内容
    - soup.find_all('标签名',过滤条件)
        - 查找所有指定的元素
    - soup.select()
        - css选择器 匹配所有符合条件的数据
    - soup.select_one()
        - css选择器 匹配所有符合条件的第一条数据

爬取多页需要找到每一页的url

碰到翻页直接点下一页

请求 得到每一页的响应数据

请求多页 要生成每一页的url

- 一般是找规律
- 根据规律生成每一页的url

jsonpath模块 了解 基本不用

pip install jsonpath

信息抽取库

从JSON文档中抽取指定数据

`$` 根节点

`@`现行节点

`.`or`[]`取子节点

`..`就是不管位置，选择所有符合条件的数据

`*`匹配所有元素节点

`[]`迭代器标示（可以在里边做简单的迭代操作，如数组下标，根据内容选值等）

`?()` 支持过滤操作.

() 表达式的计算