深圳技术大学 大数据与互联网学院
SHENZHEN TECHNOLOGY UNIVERSITY College of Big Data and Internet

2023-2024秋季课程：数据科学与大数据导论

Introduction to Data Science and Big data

第三章：大数据处理基础

Chapter 3: Big Data Analytics Fundmentals

曹劲舟 博士 助理教授

深圳技术大学 大数据与互联网学院

caojinzhou@sztu.edu.cn

2023年9月

Outline

- Data Types and Sources 数据模型
- **Data Collection 数据采集**
- Data Preprocessing 数据预处理
- Exploratory Data Analysis 数据探索性分析

数据模型与数据采集

- 数据采集案例
- 考虑一个场景：请你思考你要采集哪些数据来支撑你的分析？

数据模型与数据采集

□你要采集哪些数据来支撑你的分析？

- ■产品数据库（关系数据）
  - 例如：iPhone不同型号，及在不同销售地的定价
- ■系统日志（文本数据）
  - 例如：用户在苹果官网搜索、购买iPhone及其周边的历史
- ■文档数据（Word, Excel, PDF, CSV）
  - 例如：销售渠道汇总来的表格数据
- ■多媒体数据（视频、音频、图片）

数据模型与数据采集

□你要采集哪些数据来支撑你的分析？

□外部数据

■网页数据

2018Q2中国市场手机市场份额：

China Smartphone Shipment Market Share (%)	Q2 2017	Q2 2018	YoY Growth
HUAWEI	20%	26%	22%
OPPO	19%	19%	-9%
vivo	17%	18%	-1%
Xiaomi	13%	13%	-10%
Apple	8%	9%	0%
Others	23%	16%	-37%
TOTAL	100%	100%	-7%

华为依然是中国市场的老大，主要得益于子品牌荣耀多渠道分销策略带来的快速增长，而且华为是唯一一家能够实现同比增长的制造商，出货量暴涨了 22%，其余均不同程度下降，小米出货量跌幅达到 10%，“其他”类别暴跌 37%，说明小厂商几乎已无法生存。就出货量占比而言，华为出货量达到 26% 的份额，其次是 OPPO 的 19%， vivo 的 18%，小米的 13% 和苹果的 9%。

数据模型与数据采集

- 你要采集哪些数据来支撑你的分析？
- 外部数据
  - 网页数据
  - Web API

数据模型与数据采集

- 你要采集哪些数据来支撑你的分析？
- 外部数据
  - 网页数据
  - Web API
  - 开放数据（Open Data）

哪一些网站提供中国的开放数据(open data)?

国内资源不完全统计：

北京 bjdata.gov.cn/
上海 datashanghai.gov.cn/
浙江省 data.zjzwfw.gov.cn/
武汉 http://wuhandata.gov.cn
青岛 data.qingdao.gov.cn/
杭州 114.215.249.58/
襄阳 datagy.cn/
无锡 opendata.wuxi.gov.cn/
湛江 data.zhanjiang.gov.cn/
宁波海曙 data.haishu.gov.cn/hs_m...
佛山南海 data.nanhai.gov.cn/
深圳罗湖 szlh.gov.cn/opendata/
深圳质量监管 szscjg.gov.cn/fz/openda...
深圳住建 szjs.gov.cn/fzlm/openda...

中国气象开放服务平台 openweather.weather.com.cn...
中国专利数据 patdata.sipo.gov.cn/
国家数据 data.stats.gov.cn/

数据采集

- 无时无刻产生数据，获得数据的方式多种多样

网页

测量

数据库

监控

传统媒体

数据采集

□数据的分类

数据

结构化数据

半结构化数据

非结构化数据

China Smartphone Shipment Market Share (%)	Q2 2017	Q2 2018	YoY Growth
HUAWEI	20%	26%	32%
OPPO	19%	19%	9%
vivo	17%	18%	3%
Xiaomi	13%	13%	-1%
Apple	8%	9%	6%
Others	23%	16%	-37%
TOTAL	100%	100%	2%

Counterpoint

6***m
PLUS会员
手机还行，信号不怎么好。。
银色 公开版 256GB 2018-12-03 10:43

旧***昔
用起来还好，还是很相信京东的！
深空灰色 公开版 256GB 2018-12-04 17:18

0***b
商品很好。信号很差
深空灰色 公开版 64GB 2018-12-14 18:45

h***8
PLUS会员
物流速度快，信号是有点问题！
深空灰色 公开版 256GB 2018-10-03 07:00

全球各地的评论媒体对 iPhone Xs 和 iPhone Xs Max 进行了测试。下面是他们做出的一些评论：

Mashable
“再度改进的摄像头硬件结合了新的‘智能 HDR’自动技术，由神经网络引擎和 A12 仿生的图像信号处理器再添动力，意味着你可以充分享用先进的摄像头光学技术和计算摄影技术带来的益处。”

TechCrunch
“谈到中央处理器性能，这款开创性的规模化 7 纳米架构已带来显著成效。iPhone Xs 拥有可媲美笔记本电脑的运行速度和远超 iPhone X 的处理性能，其架构的成效由此可见一斑。”

Daring Fireball
“iPhone 镜头和感光元件的品质无法与体积更大的专业相机相比，甚至相差较远。这是由于物理定律的限制。但是，传统的相机企业在定制化芯片和软件方面却逊色于 Apple，他们的相机无法像 iPhone 一样便于随身携带，也无法随时连接互联网进行分享。从长期考虑，明智的投资应当用于芯片和软件。”

数据模型与数据采集

- 按数据源类型进行分类
  - 来自CSV文件
  - 来自JSON文件
  - 来自网页Web Pages
  - 来自关系数据库（如MySQL）
  - 来自HDFS
  - 来自Web API
  - 来自Open Data网站

掌握

可选掌握

了解

数据采集主要方法

- 数据检索
- 公开数据
- 批量数据获取
  - 网络爬虫
  - WEB API

数据采集：数据检索

- 最简单、最灵活的数据获取方式就是依靠检索
- 学会使用搜索引擎
  - 百度：适合于搜索中文信息
  - Google：更适合搜索英文信息

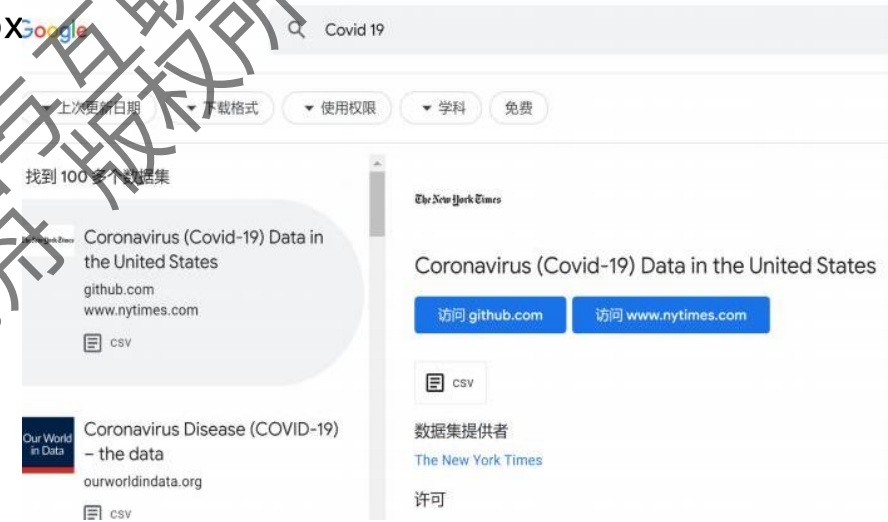
数据采集：数据检索

- 最简单、最灵活的数据获取方式就是依靠检索
- 学会使用搜索引擎
  - Google Dataset Search （Google 数据集搜索）
  - 网址： https://toolbox.google.com/datasetsearch

支持中文搜索，但中国大陆的用户想要使用需要 “梯子”

数据采集：数据检索

- 最简单、最灵活的数据获取方式就是依靠检索
- 学会使用搜索引擎
  - Google Dataset Search （Google 数据集搜索）
  - 网址： https://toolbox



数据采集：公开数据

- 国内常见公开数据渠道
  - 国家相关部门统计信息
  - 中国国家统计局

- 代表性公开数据集
  - 1400万的图像数据
    - http://www.image-net.org/
  - Amazon从2008年开始就为开发者提供几十TB的开发数据
    - http://aws.amazon.com/datasets
  - YouTube视频的统计与社交网络数据
    - http://netsg.cs.sfu.ca/youtubedata/

数据采集：公开数据

□代表性公开数据集

- ■用户评分MovieLens：https://grouplens.org/datasets/movielens/
- ■文本数据-头条： https://github.com/aceimnorstuvwxz/toutiao-text-classfication-dataset
- ■金融数据-股票： https://github.com/asxinyu/Stock
- ■网络数据-Large scale network：https://snap.stanford.edu/data/
- ■教育数据:
  ASSISTmentsData-学业:
  https://sites.google.com/site/assistmentsdata/home/
  BASEGroup: https://github.com/bigdata-ustc/EduData
- ■阿里天池数据-数据平台： https://tianchi.aliyun.com/dataset/
  公开大数据竞赛的数据： KDDCup ， NeurIPS Challenge

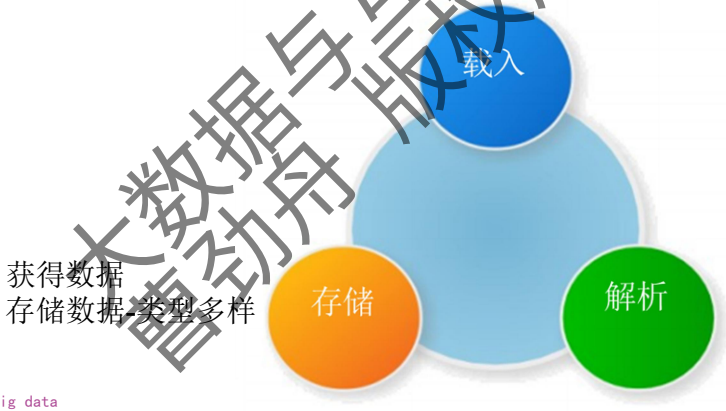
数据采集：批量网络数据获取

- 大量数据的获取难以手动实现，需借助爬虫程序
  - 也有可能通过交易(购买)“数据”而得
- 网络爬虫是一个自动在网上抓取数据的程序
  - 爬虫本质上就是下载特定网站网页的HTML/JSON/XML数据 ， 并对数据进行解析、提取与存储
  - 通常先定义一组入口URL，根据页面中的其他URL ， 深度优先或广度优先的遍历访问，逐一抓取数据

数据采集：批量网络数据获取

□网络爬虫是什么？

- ■网络爬虫(又被称为网页蜘蛛，网络机器人，网页追逐者)，是一种按照一定的规则，自动的抓取万维网信息的自动化程序。
- ■爬虫的行为可以划分为： 载入、解析、存储，最复杂的部分为载入



获得数据
存储数据-类型多样

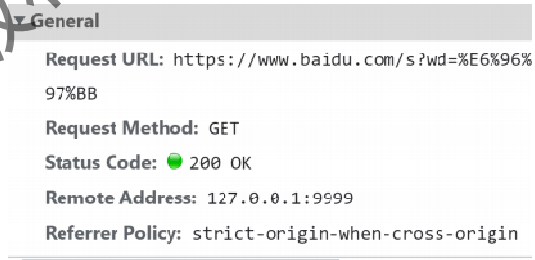
解析内容
提取数据-类型多样

数据采集：批量网络数据获取

- 载入：将目标网站数据下载到本地
  - 网站数据主要依托于网页（html等）展示
  - 爬虫程序向服务器发送网络请求 Request，获取相应的网页，服务器response信息(html等)
  - 网站常用网络协议：http，https
  - 数据常用请求方式：get，post

网络爬虫：载入

- 载入：将目标网站数据下载到本地
  - 数据常用请求方式：get ， post
    - get：参数常放置在URL中
      - http://www.adc.com?p=1&q=2&r=3，问号后为参数
      - 例如， https://www.baidu.com/s?wd=图片



请求头

网络爬虫：载入

- 载入：将目标网站数据下载到本地
  - 数据常用请求方式：get ， post
    - post：参数常放置在一个表单中
      - 在向目标URL发送请求时，将参数放置在一个网络请求的报文头中
      - 相比于Get，多了请求体（Form Data）部分
      - 更安全：登录操作常用(账号密码信息不会放在URL后面)

网络爬虫：载入

- 示例操作：抓取一个静态网页步骤
  - 首先，确定URL，例如： http://www.baidu.com
  - 其次，确定请求方式及相关参数：
    - 直接用浏览器实现： 使用chrome, firefox浏览器抓包工具，详见 http://jingyan.baidu.com/article/3c343ff703fee20d377963e7.html
    - 或者抓包工具： charles等，详见 http://blog.csdn.net/jiangwei0910410003/article/details/41620363/
  - 最后，在代码中按照特定请求方式(get 或post)向URL发送参数，即可收到网页返回的结果

网络爬虫：载入

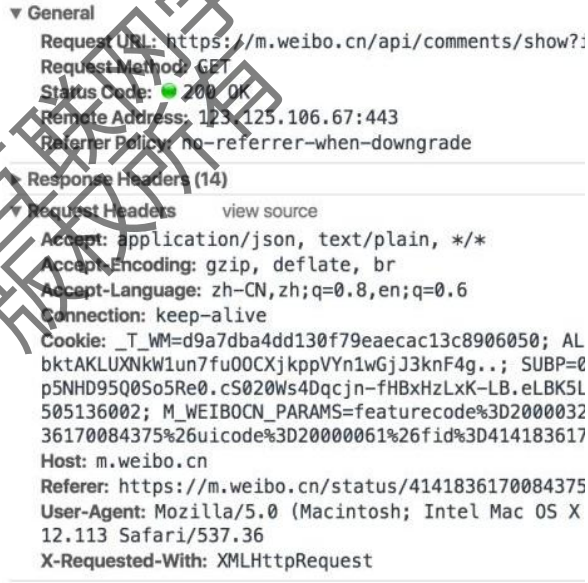
- 但部分页面的数据是动态加载的
  - Ajax异步请求
    - 网页中的部分数据需要浏览器渲染（JavaScript调用接口获取数据）
    - 用户的某些点击、下拉的操作触发才能获得
    - 解决方案：
      - 借助抓包工具，分析Ajax某次操作所触发的请求，通过代码实现相应的请求
        - 有技术难度，但抓取速度快。
      - 利用智能化的工具：selenium webdriver
        - 用程序控制驱动浏览器，模拟浏览器
        - 可以模拟实现人的所有操作
        - 操作简单，但是速度慢
          - 因为爬虫需要启动浏览器，浏览器需要渲染页面， 所以速度比较慢
      - 其他： Splash，Pyv8等

网络爬虫：载入

- 反爬虫：随着网络爬虫对目标网站访问频率的加大，网站禁止爬虫程序继续访问Ajax异步请求
  - 常见反爬手段：
    - 出现用户登录界面，需要验证码
    - 禁止某个固定帐号或ip一段时间内访问网站
    - 更有甚者，直接返回错误的无用数据
  - 应对措施：
    - 优化爬虫程序，尽量减少访问次数， 尽量不抓取重复内容
    - 使用多个cookie （网站用来识别用户的手段，每个用户登录会 生成一个cookie）
    - 使用多个ip （可以用代理实现）

网络爬虫：抓取微博评论

抓包工具
获取请求



网络爬虫：抓取微博评论

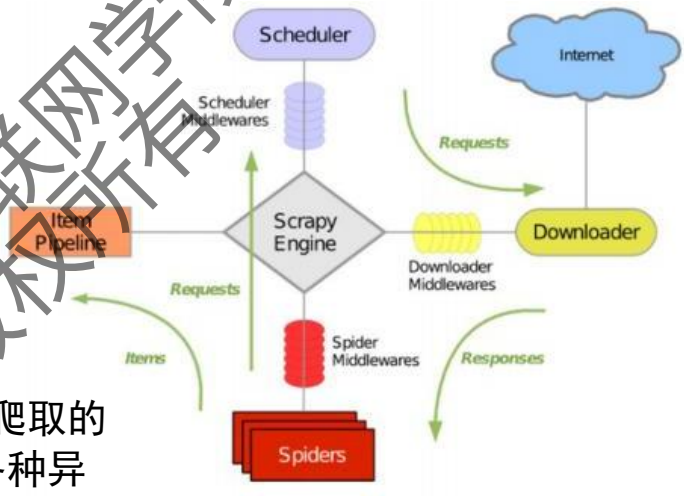
获得评论的**json**格式

解析出需要的字段

用户id	时间	内容
陈赫	08-18	天霸
邓超	08-18	我们都很好，谢谢大家
邓超	08-18	我也不知道
贼亮zl	08-17	迪丽热巴
... ...	... ...	... ...

网络爬虫：现有技术

- 基于Python的工具
  - urllib库
  - Requests
  - Beautiful-soup
  - Scrapy



现有的爬虫框架很成熟，能够合理的控制爬取的过程，并有效的 处理爬取过程中出现的各种异常，推荐使用Scrapy

网络爬虫：注意事项

- 注意网站规定
- 注意法律规定
- 2021年6月1日，《中华人民共和国数据安全法》
- 注意数据使用规范

Requests

Requests库是第三方模块，需要额外进行安装。安装方式与numpy安装方式相同，直接执行pip install requests即可。

In

```
import requests

r = requests.get('https://www.baidu.com')
r.encoding=requests.utils.get_encodings_from_content(r.text)
# 注意get_encodings_from_content的参数是字符串，所以要用r.text而不是r.content
print(r.text)
```

```
<!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 rel=stylesheet type=text/css href=https://ss1.bdstatic.com/5eN1bjq8AAUYm2zgoY3K/r/www/cache/bdorz/baidu.min.css><title>百度一下，你就知道</title></head> <body link=#0000cc> <div id=wrapper> <div id=head> <div class=head_wrapper> <div class=s_form> <div class=s_form_wrapper> <div id=lg> <img hidefocus=true src=//www.baidu.com/img/bd_logo1.png width=270 height=129>
</div> <form id=form name=f action=//www.baidu.com/s class=fm> <input type=hidden name=bdorz_come value=1> <input type=hidden name=ie value=utf-8> <input type=hidden name=f value=8> <input type=hidden name=rsv_bp value=1> <input type=hidden name=rsv_idx value=1> <input type=hidden name=tn value=baidu><span class="bg s_ipt_wr"><input id=kw name=wd class=s_ipt value maxlength=255 autocomplete=off autofocus=autofocus></span><span class="bg s_btn_wr"><input type=submit id=su value=百度一下 class="bg s_btn" autofocus></span> </form> </div> </div>
<div id=ul> <a href=http://news.baidu.com name=tj_trnews class=mnav>新闻</a> <a href=https://www.hao123.com name=tj_trhao123 class=mnav>hao123</a> <a href=http://map.baidu.com name=tj_trmap class=mnav>地图</a> <a href=http://v.baidu.com name=tj_trvideo class=mnav>视频</a> <a href=http://tieba.baidu.com name=tj_trtieba class=mnav>贴吧</a> <noscript> <a href=http://www.baidu.com/bdorz/login.gif?login&amp;tpl=mn&amp;u=http%3A%2F%2Fwww.baidu.com%2f%3fbdorz_come%3d1 name=tj_login class=lb>登录</a> </noscript> <script>document.write('<a href="http://www.baidu.com/bdorz/login.gif?login&tpl=mn&u='+ encodeURIComponent(window.location.href+ (window.location.search === "" ? "?" : "&")+ "bdorz_come=1")+ '" name="tj_login" class="lb">登录</a>');
                </script> <a href=//www.baidu.com/more/ name=tj_briicon class=bri style="display: block;">更多产品</a> </div> </div> </div> <div id=ftCon> <div id=ftConw> <p id=lh> <a href=http://home.baidu.com>关于百度</a> <a href=http://ir.baidu.com>About Baidu</a> </p> <p id=cp>&copy;2017 Baidu <a href=http://www.baidu.com/duty/>使用百度前必读</a>  <a href=http://jianyi.baidu.com/ class=cp-feedback>意见反馈</a> 京ICP证030173号  <img src=//www.baidu.com/img/gs.gif> </p> </div> </div> </div> </body> </html>
```

BeautifulSoup

使用BeautifulSoup库前要安装该库：pip install beautifulsoup4。
创建BeautifulSoup库的对象，需从bs4库导入：from bs4 import BeautifulSoup。

```
In [2]: from bs4 import BeautifulSoup
   ...: soup = BeautifulSoup(html, 'html.parser')
   ...: soup
Out[2]:
```

BeautifulSoup

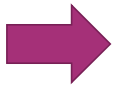
```
  Out[2]:
  <!DOCTYPE html>

  <!--STATUS OK-->
  <html>
  <head>
  <meta content="text/html;charset=utf-8" http-equiv="content-type"/>
  <meta content="IE=Edge" http-equiv="X-UA-Compatible"/>
  <meta content="always" name="referrer"/>
  <meta content="#2932e1" name="theme-color"/>
  <link href="/favicon.ico" rel="shortcut icon" type="image/x-icon"/>
  <link href="/content-search.xml" rel="search" title="百度搜索"
type="application/opensearchdescription+xml"/>
  <link href="//www.baidu.com/img/baidu_85beaf5496f291521eb75ba38eacbd87.svg" mask="" rel="icon"
sizes="any"/>
  <link href="//s1.bdstatic.com" rel="dns-prefetch"/>
  <link href="//t1.baidu.com" rel="dns-prefetch"/>
  <link href="//t2.baidu.com" rel="dns-prefetch"/>
  <link href="//t3.baidu.com" rel="dns-prefetch"/>
  <link href="//t10.baidu.com" rel="dns-prefetch"/>
  <link href="//t11.baidu.com" rel="dns-prefetch"/>
  <link href="//t12.baidu.com" rel="dns-prefetch"/>
  <link href="//b1.bdstatic.com" rel="dns-prefetch"/>
  <title>百度一下，你就知道</title>
  <style id="css_index" index="index" type="text/css">html,body{height:100%}
  html{overflow-y:auto}
  ……
  <div class="s_tab" id="s_tab">
  <div class="s_tab_inner">
  <b>网页</b>
  <a href="//www.baidu.com/s?rtt=1&amp;bsst=1&amp;cl=2&amp;tn=news&amp;word=" onmousedown="return
c({'fm':'tab','tab':'news'})" sync="true" wdfield="word">资讯</a>
  <a href="http://tieba.baidu.com/f?kw=&amp;fr=wwwt" onmousedown="return
c({'fm':'tab','tab':'tieba'})" wdfield="kw">贴吧</a>
  <a href="http://zhidao.baidu.com/q?ct=17&amp;pn=0&amp;tn=ikaslist&amp;rn=
10&amp;word=&amp;fr=wwwt" onmousedown="return c({'fm':'tab','tab':'zhidao'})" wdfield="word">知道</a>
    ……
```

网络爬虫：解析

- 获取html网页以后
  - 从文本数据中抽取结构化信息

For years, Microsoft Corporation CEO Bill Gates was against open source. But today he appears to have changed his mind. "We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access. "

Richard Stallman, founder of the Free Software Foundation, countered saying…



PEOPLE

Name	Title	Organization
Bill Gates	CEO	Microsoft
Bill Veghte	VP	Microsoft
Richard Stallman	Founder	Free Soft..

```
Select  Name
From   PEOPLE
Where Organization = 'Microsoft'
```

```
Bill Gates
Bill Veghte
```

网络爬虫：解析

- 解析：在载入的结果中抽取特定的数据，载入的结果主要分成三类html 、json 、xml
  - html
    - Python工具包： beautifulSoup等
  - json
    - Python工具包： json、demjson等
  - Xml
    - Python工具包： xml、libxml2等

网络爬虫：解析(对比JSON与XML)

```
{
  "name": "中国",
  "province": [{
    "name": "黑龙江",
    "cities": {
      "city": ["哈尔滨", "大庆"]
    }
  },
  {
    "name": "广东",
    "cities": {
      "city": ["广州", "深圳", "珠海"]
    }
  },
```

对象，成员：键值对

```
<?xml version="1.0" encoding="utf-8"?>
<country>
  <name>中国</name>
  <province>
    <name>黑龙江</name>
    <cities>
      <city>哈尔滨</city>
      <city>大庆</city>
    </cities>
  </province>
  <province>
    <name>广东</name>
    <cities>
      <city>广州</city>
      <city>深圳</city>
      <city>珠海</city>
    </cities>
  </province>

  .....
</country>
```

网络爬虫：解析(对比JSON与XML)

- 可读性
  - Json简洁， XML规范， XML比较好
- 可扩展性
  - 均很好
- 数据体积
  - JSON数据量少，传输快。 XML数据量大，传输慢
- 编码解码
  - JSON容易， XML复杂(树结构，父子节点)
- 数据描述
  - XML数据描述更好
- 数据交互
  - JSON与JavaScript交互更方便，易于解析。 XML更适合跨平台共享

网络爬虫：解析html

□读取网页数据：中文网页

- ■String的split
- ■切割新闻主体内容

```
temp = mytext.split("<div class=\"content-article\">")[1]
temp = temp.split("<div id=\"Status\"></div>")[0]
print(temp)
```

```
<!--导语-->
<p class="one-p"><img class="content-picture" src="//inews.gtimg.com/newsapp_bt/0/14167142397/1000">
</p>
<p class="one-p">《自然·物理》以封面文章形式发表成果论文。 中科院高能所 供图</p>
<p class="one-p">中新网北京11月11日电 (记者 孙自法)记者11日从中国科学院高能物理研究所(中科院高能所)获悉，北京谱仪III(BESIII)作为北京正负电子对撞机核心科研装置之一，其国际合作组最近已实现对中子电磁结构精确测量，从而揭开困扰学界20多年的光子-核子相互作用之谜。</p>
<p class="one-p">北京谱仪III国际合作组最新完成的对中子的类时电磁形状因子进行精确测量，实验结果不仅解决了长期存在的光子-核子耦合反常的问题，还观测到中子电磁形状因子随质心能量变化的周期性振荡结构。这项重要物理实验结果论文，近日以封面文章形式在国际学术期刊《自然·物理》发表。</p>
<p class="one-p">据中科院高能所实验物理中心介绍，中子和质子统称为核子，它们是构成可见物质世界的主要成分。迄今为止核子的内部结构仍有许多未解之谜，长达20余年的光子-核子相互作用之谜即是其中之一。1998年意大利芬尼斯(FENICE)实验首次测量了中子的类时电磁形状因子，其结果表明光子-中子相互作用强于光子-质子相互作用，与参考模型预期不符。</p>
<p class="one-p"><img class="content-picture" src="//inews.gtimg.com/newsapp_bt/0/14167142399/1000">
</p>
```

切割以后的结果

网络爬虫：解析html

□读取网页数据：中文网页

- ■String的replace
- ■去除无关html标记

```
temp = temp.replace("<p class=\"one-p\">", "")
print(temp)
```

```
temp = temp.replace("</p>", "")
print(temp)
```

```
<!--导语-->
<img class="content-picture" src="//inews.gtimg.com/newsapp_bt/0/14167142397/1000">

《自然·物理》以封面文章形式发表成果论文。 中科院高能所 供图
中新网北京11月11日电 （记者 孙自法）记者11日从中国科学院高能物理研究所(中科院高能所)获悉，北京谱仪III(BESIII)作为北京正负电子对撞机核心科研装置之一，其国际合作组最近已实现对中子电磁结构精确测量，从而揭开困扰学界20多年的光子-核子相互作用之谜。
北京谱仪III国际合作组最新完成的对中子的类时电磁形状因子进行精确测量，实验结果不仅解决了长期存在的光子-核子耦合反常的问题，还观测到中子电磁形状因子随质心能量变化的周期性振荡结构。这项重要物理实验结果论文，近日以封面文章形式在国际学术期刊《自然·物理》发表。
据中科院高能所实验物理中心介绍，中子和质子统称为核子，它们是构成可见物质世界的主要成分。迄今为止核子的内部结构仍有许多未解之谜，长达20余年的光子-核子相互作用之谜即是其中之一。1998年意大利芬尼斯(FENICE)实验首次测量了中子的类时电磁形状因子，其结果表明光子-中子相互作用强于光子-质子相互作用，与夸克模型预期不符。
```

网络爬虫：解析html

□读取网页数据：中文网页

- ■正则表达式
- ■去除无关html标记<img>

```
import re
s = temp
replaced = re.sub('<img .*>', '', s)
print (replaced )
```

<!--导语-->

《自然·物理》以封面文章形式发表成果论文。 中科院高能所 供图

中新网北京11月11日电 （记者 孙自法）记者11日从中国科学院高能物理研究所(中科院高能所)获悉，北京谱仪III(BESIII)作为北京正负电子对撞机核心科研装置之一，其国际合作组最近已实现对中子电磁结构精确测量，从而揭开困扰学界20多年的光子-核子相互作用之谜。

北京谱仪III国际合作组最新完成的对中子的类时电磁形状因子进行精确测量，实验结果不仅解决了长期存在的光子-核子耦合反常的问题，还观测到中子电磁形状因子随质心能量变化的周期性振荡结构，这项重要物理实验结果论文，近日以封面文章形式在国际学术期刊《自然·物理》发表。

据中科院高能所实验物理中心介绍，中子和质子统称为核子，它们是构成可见物质世界的主要成分。迄今为止核子的内部结构仍有许多未解之谜，长达20余年的光子-核子相互作用之谜即是其中之一。1998年意大利芬尼斯(FENICE)实验首次测量了中子的类时电磁形状因子，其结果表明光子-中子相互作用强于光子-质子相互作用，与夸克模型预期不符。

BeautifulSoup

采用soup.select()来筛选元素，返回类型是列表list。

（1）通过标签名查找。

```
In [6]: print(soup.select('title'))
        [<title>百度一下，你就知道</title>]

In [7]: print(soup.select('b'))
        [<b>网页</b>, <b>百度</b>]
```

(2) 通过类名查找。类名前加“ .”。

```
In [13]: print(soup.select('.c-tips-container'))
         [<div class="c-tips-container" id="c-tips-container"></div>]
```

（3）通过id名查找。id前加“#”。

```
In [16]: print(soup.select('#c-tips-container'))
   [<div class="c-tips-container" id="c-tips-container"></div>]
```

BeautifulSoup

（4）组合查找。

组合查找时，标签名与类名、id名进行单独查找方法一样，组合时只需用空格隔开。例如，查找div标签中，id等于s_qrcode_nologin的内容，二者需要用空格分开。

```
In [19]: print(soup.select('div #s_qrcode_nologin'))
```

```
[<div class="qrcode-nologin" id="s_qrcode_nologin"><div class="qrcode-layer icon-mask-wrapper"><img class="icon" src="http://ss.bdimg.com/static/superman/img/qrcode/qrcode@2x-daf987ad02.png"><img class="icon-hover" src="http://ss.bdimg.com/static/superman/img/qrcode/qrcode-hover@2x-f9b106a848.png"/></img></div><div class="tooltip qrcode-tooltip"><div class="text"><div class="login-text"><i class="c-icon login-icon">□</i>百度APP扫码登录</div><div class="login-info">有事搜一搜 没事看一看</div></div><div id="qrcode-login-wrapper"></div></div></div>]
```

直接子标签查找，标签之间加“>”

```
In [24]: print(soup.select("div > img"))
```

```
[<img class="index-logo-src" height="129" hidefocus="true" src="//www.baidu.com/img/dong1_dd071b75788996a161c3964d450fcd8c.gif" usemap="#mp" width="270"/>, <img class="index-logo-srcnew" height="129" hidefocus="true" src="//www.baidu.com/img/dong1_dd071b75788996a161c3964d450fcd8c.gif" usemap="#mp" width="270"/>]
```

BeautifulSoup

（5）属性查找。

查找时还可以加入属性元素，属性需要用中括号括起来，注意属性和标签属于同一结点，所以中间不能加空格，否则会无法匹配。

```
In [25]: print(soup.select('a[href="http://www.baidu.com/more/"]'))
[<a class="s-bri c-font-normal c-color-t" href="http://www.baidu.com/more/" name="tj_briicon"
target="_blank">更多</a>, <a class="c-color-gray2 c-font-normal" href="http://www.baidu.com/more/"
name="tj_more" target="_blank">查看全部百度产品 &gt;</a>, <a class="s-tab-item s-tab-more"
href="http://www.baidu.com/more/" onmousedown="return c({'fm':'tab','tab':'more'})">更多</a>]
```

同样，属性仍然可以与上述查找方式组合，不在同一结点的用空格隔开，同一结点的不加空格。

```
In [26]: print(soup.select('div a[href="http://www.baidu.com/more/"]'))
[<a class="s-bri c-font-normal c-color-t" href="http://www.baidu.com/more/" name="tj_briicon"
target="_blank">更多</a>, <a class="c-color-gray2 c-font-normal" href="http://www.baidu.com/more/"
name="tj_more" target="_blank">查看全部百度产品 &gt;</a>, <a class="s-tab-item s-tab-more"
href="http://www.baidu.com/more/" onmousedown="return c({'fm':'tab','tab':'more'})">更多</a>]
```

BeautifulSoup

（6）通过find_all()函数查找。

```
findAll(name=None, attrs={}, recursive=True,
        text=None, limit=None, **kwargs)
```

返回一个列表，其中最重要的参数是name和keywords。

参数name匹配tags的名字，获得相应的结果集。有几种方法匹配name，最简单的用法是仅仅给定一个tag的name值。

① 搜索网页源码中所有b标签：soup.findAll('b')。

② 可以传一个正则表达式，下面的代码寻找所有以b开头的标签。

```
import re
tagsStartingWithB = soup.findAll(re.compile('^b'))
```

③ 可以传一个list或dictionary。查找所有的title和p标签，获得结果一样，但方法2更快一些。

方法1：`soup.findAll(['title', 'p'])`

方法2：`soup.findAll({'title' : True, 'p' : True})`

BeautifulSoup

输出如下：

```
[<title>百度一下，你就知道</title>,
 <p class="lh"><a class="text-color" href="//www.baidu.com/cache/setindex/index.html" target="_blank">设为首页</a></p>,
 <p class="lh"><a class="text-color" href="//home.baidu.com" target="_blank">关于百度</a></p>,
 <p class="lh"><a class="text-color" href="http://ir.baidu.com" target="_blank">About Baidu</a></p>,
 <p class="lh"><a class="text-color" href="https://isite.baidu.com/site/e.baidu.com/d38e8023-2131-4904-adf7-
a8d1108f51ef?refer=888" target="_blank">百度营销</a></p>,
 <p class="lh"><a class="text-color" href="//www.baidu.com/duty" target="_blank">使用百度前必读</a></p>,
 <p class="lh"><a class="text-color" href="//help.baidu.com/newadd?prod_id=1&amp;category=4" target="_blank">意见反馈
</a></p>,
 <p class="lh"><a class="text-color" href="//help.baidu.com" target="_blank">帮助中心</a></p>,
 <p class="lh"><a class="text-color"
href="http://www.beian.gov.cn/portal/registerSystemInfo?recordcode=11000002000001" target="_blank">京公网安备
11000002000001号</a></p>,
 <p class="lh"><a class="text-color" href="https://beian.miit.gov.cn" target="_blank">京ICP证030173号</a></p>,
 <p class="lh"><span class="text-color">©2021 Baidu </span></p>,
 <p class="lh"><span class="text-color">(京)-经营性-2017-0020</span></p>]
```

数据读取

从CSV文件读取数据

- CSV的全称是Comma-separated values，是一种用逗号分隔的方式来表示与存储表格数据的文件格式
- 使用Python **Pandas**读取CSV文件

```
import pandas as pd

df = pd.read_csv("./employee.csv", delimiter=",")
df.head()
```

	EMPID	FirstName	LastName	Salary
0	1001	Amal	Jose	100000
1	1002	Edward	Joe	100001
2	1003	Sabitha	Sunny	210000
3	1004	John	P	50000
4	1005	Mohammad	S	75000

数据读取

从JSON文件读取数据

- JSON是一种存储嵌套数据的文件格式（类似Python中的List, Dict）

```
df2 = pd.read_json("./employee.json")
df2.head()
```

	EMPID	FirstName	LastName	Salary
0	1001	Amal	Jose	100000
1	1002	Edward	Joe	100001
2	1003	Sabitha	Sunny	210000
3	1004	John	P	50000
4	1005	Mohammad	S	75000


```
[{"EMPID":1001,"FirstName":"Amal","LastName":"Jose","Salary":100000},
{"EMPID":1002,"FirstName":"Edward","LastName":"Joe","Salary":100001},
{"EMPID":1003,"FirstName":"Sabitha","LastName":"Sunny","Salary":210000},
{"EMPID":1004,"FirstName":"John","LastName":"P","Salary":50000},
{"EMPID":1005,"FirstName":"Mohammad","LastName":"S","Salary":75000}]
```


employee.json的内容

数据采集

□从关系数据库获取数据

■以MySQL数据库为例

- 创建连接
- 写SQL语句
- 执行SQL语句
- 解析结果
- 关闭连接

```python
import pymysql

# Open database connection
con = pymysql.connect(host='localhost',
                      user='root',
                      password='rootroot',
                      database='test1',
                      cursorclass=pymysql.cursors.DictCursor)

# prepare a cursor object using cursor() method
cursor = con.cursor()
sql = "select * from namelist"
# Execute the SQL command
cursor.execute(sql)

# Fetch all the rows in a list of lists.
results = cursor.fetchall()
for row in results:
    id = row['id']
    name = row['name']
    # Now print fetched result
    print ("id=%s,name=%s" %  (id, name))

# disconnect from server
con.close()
```

```
id=1,name=徐君
id=2,name=陈跃国
id=3,name=覃雄派
```

https://dev.mysql.com/doc/connector-python/en/connector-python-examples.html