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

2022-2023秋季课程:数据科学与大数据导论

Introduction to Data Science and Big data

Chapter 3: Big Data Analytics Fundmentals

曹劲舟 助理教授

深圳技术大学 大数据与互联网学院

caojinzhou@sztu.edu.cn

2022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%	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%

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

阳***替
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

O***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

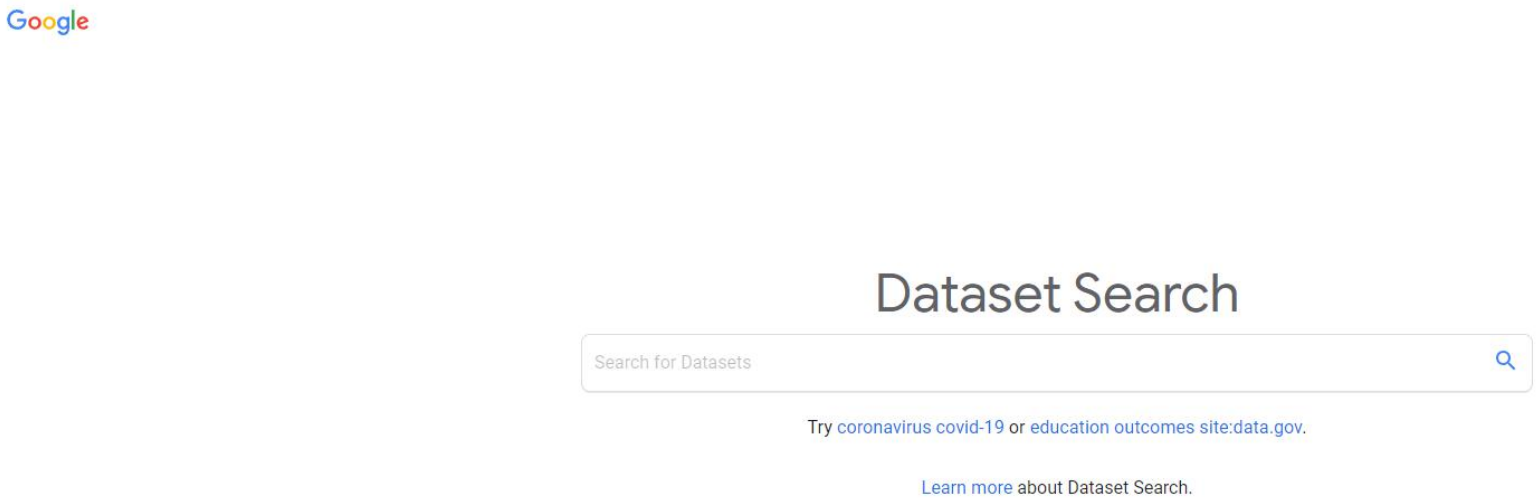
- 数据检索
- 公开数据
- 批量数据获取
  - 网络爬虫
  - WEB API
- 数据筛选

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

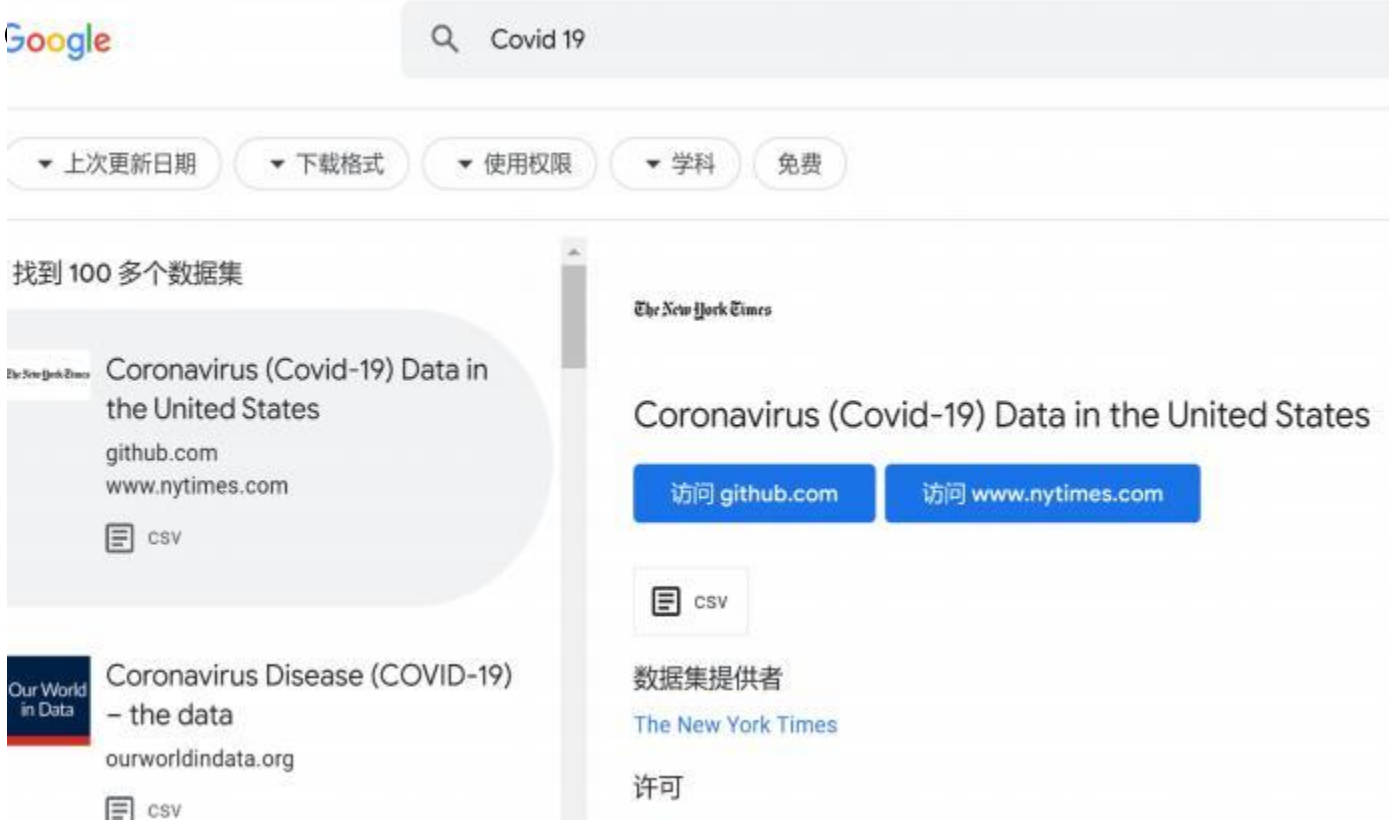
- 最简单、最灵活的数据获取方式就是依靠检索
- 学会使用搜索引擎
  - Google Dataset Search（Google 数据集搜索）
  - 网址： https://toolbox.google.com/datasetsearch



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.go



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：网络爬虫

- 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程序或者脚本。
    - 请求网站并提取数据的自动化程序
  - 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

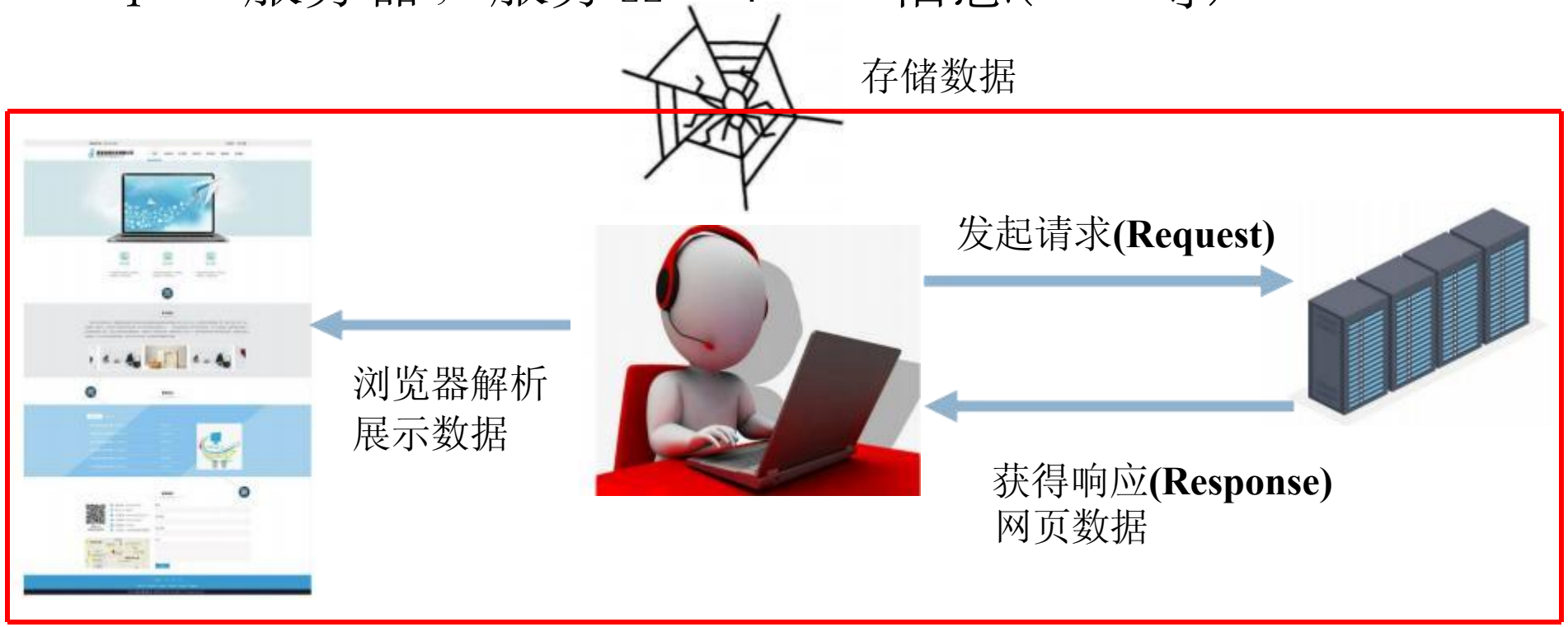
获得数据
存储数据-类型多样

解析内容
提取数据-类型多样

数据采集：网站数据

◻访问网页示例

- ■网站数据主要依托于网页 (html, 超文本标记语言)展示
- ■用户Request服务器，服务器response信息(html等)



网络爬虫：载入

- 载入：将目标网站数据下载到本地
  - 爬虫程序向服务器发送网络请求 Request，获取相应的网页
    - 网站常用网络协议：http，https
    - 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
      - post：参数常放置在一个表单中(报文头(header))
        - 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
        - 更安全

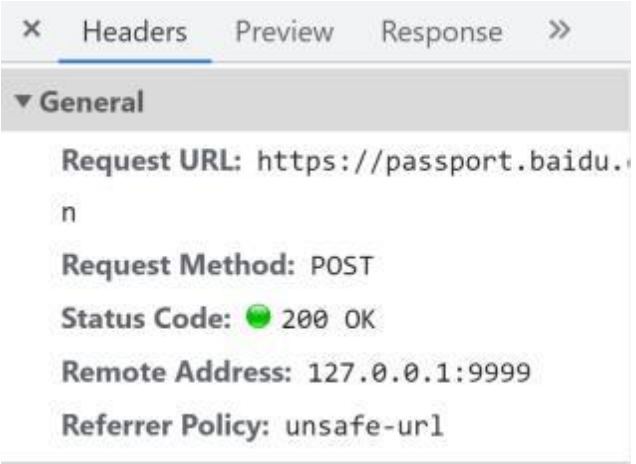
网络爬虫：载入

- 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Form Data部分(请求体)
      - 更安全：登录操作常用(不会放在URL后面)



请求体

网络爬虫：载入

- 实际操作：抓取一个静态网页步骤
  - 首先确定URL，例如： http://www.baidu.com
  - 其次确定请求的方式以及相关参数：
    - 直接用浏览器实现： 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的请求方式(get ， post)向URL发送参数，即可收到网页的结果

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：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

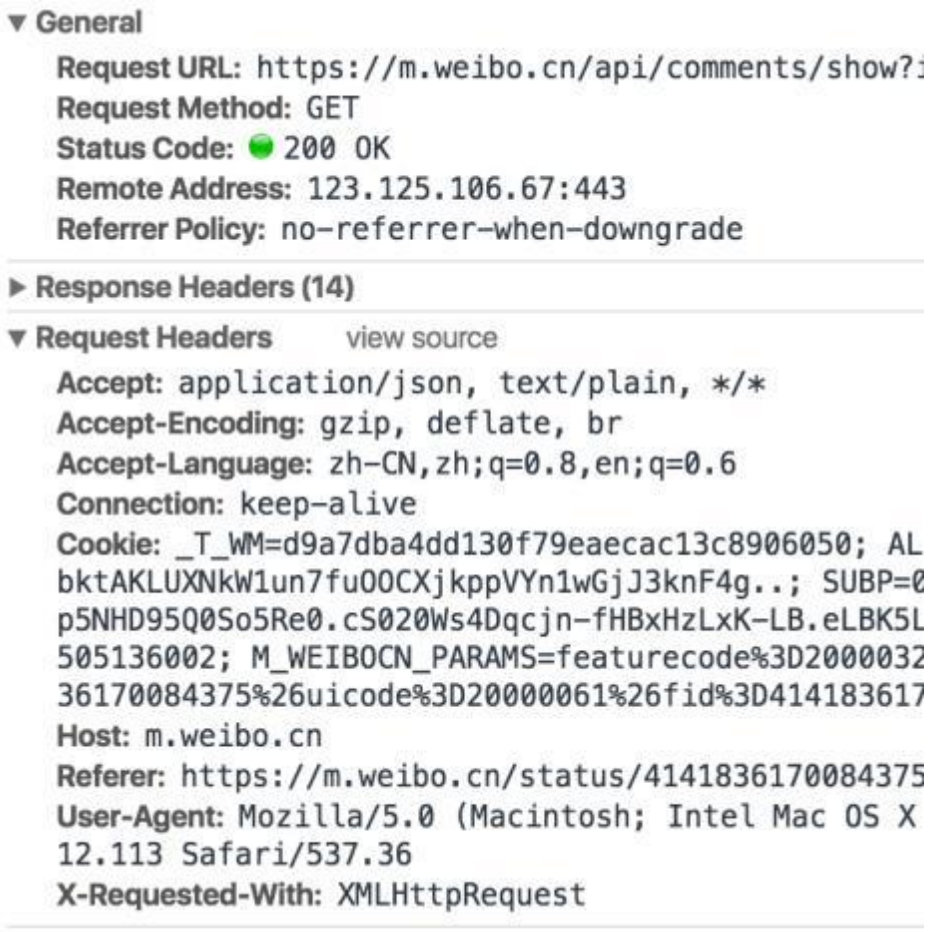
```
<?xml version="1.0" encoding="utf-8"?>
<country>
  <name>中国</name>
  <province>
    <name>黑龙江</name>
    <cities>
      <city>哈尔滨</city>
      <city>大庆</city>
    </cities>
  </province>
  <province>
    <name>广东</name>
    <cities>
      <city>广州</city>
      <city>深圳</city>
      <city>珠海</city>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

京ICP备15025187号-1 邮箱: service@json.cn

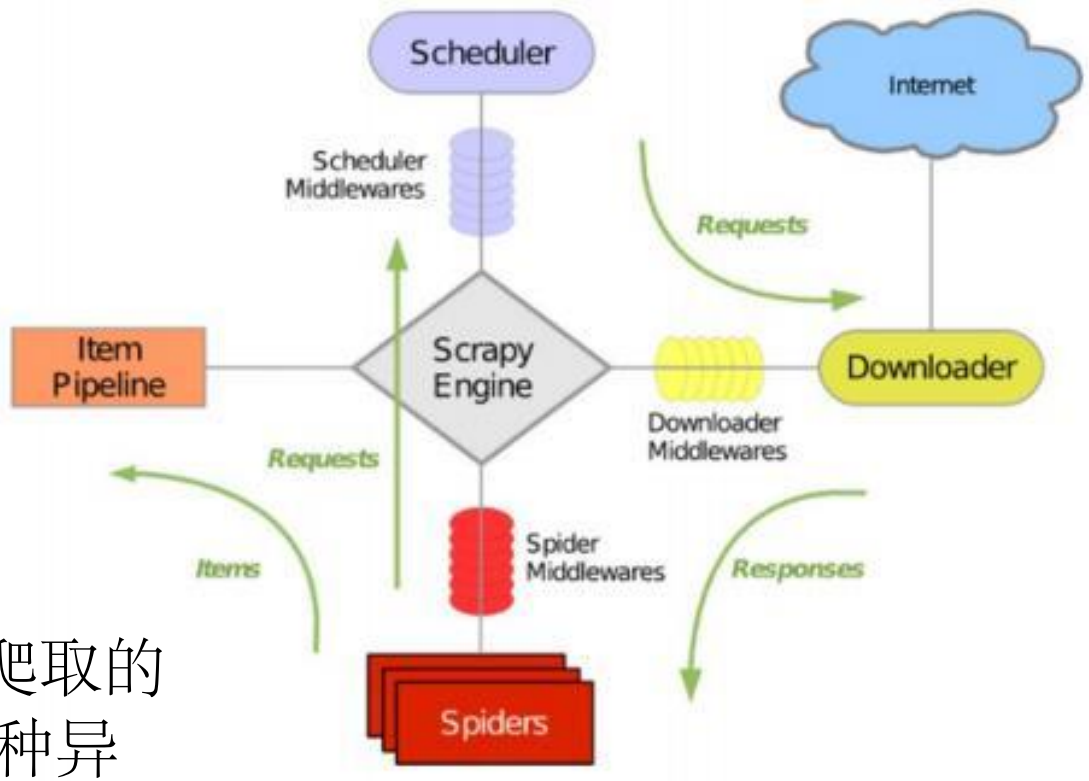

```
{
    "mod_type":"mod/pagelist",
    "previous_cursor":"",
    "next_cursor":"",
    "card_group":[
        {
            "id":4142016554789113,
            "created_at":"08-18 08:46",
            "source":"柔光自拍vivo X7",
            "user":Object{...},
            "text":"回复<a
href="/n/%E9%82%93%E8%B6%85">@邓超</a>:不管是谁,
请大家记住陈赫的话，他们很好，感情都很好。恳请各家粉丝不要
戏太多就好<i class="face face_1 icon_1">[微笑]</i>
没准你们那么嫌弃骂的那么难听，人家正主还是感情好的时不时去
吃火锅呢，你们不累吗？别用自己对他的爱去给他造成困扰",
            "reply_id":4142015488402958,
            "reply_text":"<a
href='/u/'5187664653>@邓超</a>: 我也不知道<i
class="face face_1 icon_20">[doge]</i>",
            "like_counts":10811,
            "liked":false,
            "mod_type":"mod/single/infobox"
```


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
  - Scrapy
  - Beautiful-soup



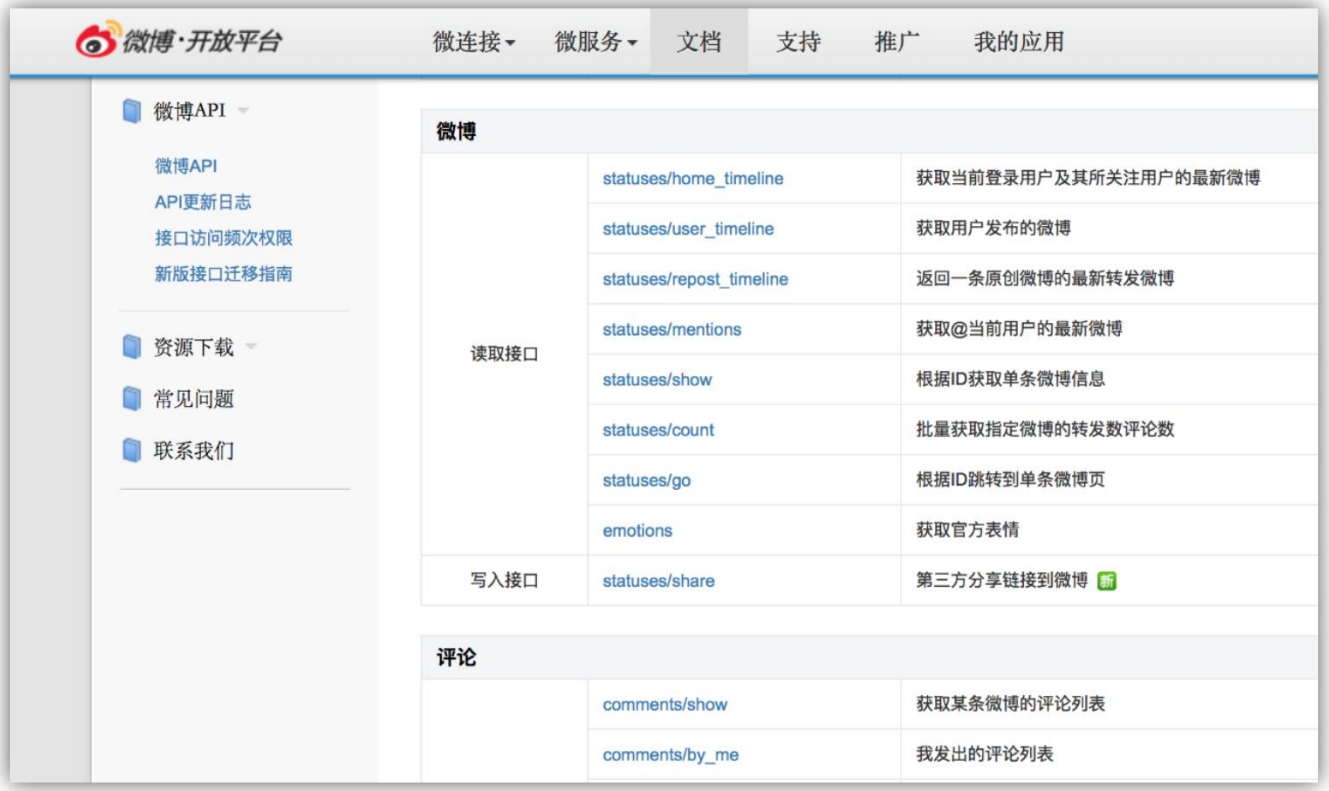
现有的爬虫框架很成熟，能够合理的控制爬取的过程，并有效的 处理爬取过程中出现的各种异常，推荐使用Scrapy

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

Web API

- 外部数据
  - Web API



数据读取-Python

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-Python

◻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


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


employee.json的内容

数据采集——python

◻读取网页数据

■requests库

```python
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=u1> <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>
```

数据采集——python

◻读取网页数据：中文网页

■requests库

```python
import requests

url = "https://new.qq.com/omn/20211111/20211111A0AQ7700.html"
r = requests.get(url)
r.encoding='gb2312' # 根据网页编码设置

print(r.text)
mytext = r.text
```

```
<!DOCTYPE html>
<html lang="zh-CN" dir="ltr">
  <head>
    <title>北京谱仪精确测量中子电磁结构 揭开光子-核子相互作用之谜_腾讯新闻</title>
    <meta name="keywords" content="北京谱仪精确测量中子电磁结构 揭开光子-核子相互作用之谜,中科院高能所,光子,质子,中子,北京谱仪">
    <meta name="description" content="《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……">
    <meta name="apub:time" content="11/11/2021, 8:51:41 PM">
    <meta name="apub:from" content="default">
    <meta http-equiv="X-UA-Compatible" content="IE=Edge" />
<link rel="stylesheet" href="//mat1.gtimg.com/qqcdn/qqindex2021/qqdc/css/index.css" />
<!--[if lte IE 8]><meta http-equiv="refresh" content="0; url=/upgrade.htm"><![endif]-->
<!-- <meta name="sogou_site_verification" content="SYWy6ahy7s"/> -->
<meta name="baidu-site-verification" content="jJeIJ5X7pP" />
<link rel="shortcut icon" href="//mat1.gtimg.com/www/icon/favicon2.ico" />
<link rel="stylesheet" href="//vm.gtimg.cn/tencentvideo/txp/style/txp_desktop.css" />
<script src="//is.ag.qq.com/is/ag_common.is"></script>
```

数据采集

◻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，与夸克模型预期不符。</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

数据采集——python

◻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。
```

数据采集

- 从html网页解析（Parsing）内容
  - 可以选用如下python库
    - 正则表达式解析 re
    - Beautiful Soup (https://www.crummy.com/software/BeautifulSoup/)
    - lxml (http://lxml.de/)

数据采集——python

- 读取网页数据：中文网页
  - 正则表达式
  - 去除无关html标记<img>

```python
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。

数据采集

- 获取html网页以后
  - 可以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
```

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