**# 爬虫概念、工具和HTTP**

**### 1.什么是爬虫**

* 爬虫就是：模拟客户端（浏览器）发送网络请求’，获取响应，按照规则提取数据的程序
* ‘模拟客户端（浏览器）发送网络请求’：照着浏览器发送一模异样的请求，获取和浏览一模一样的数据

**## 2 。爬虫的数据去哪儿了**

—呈现出来：展示在网页上，或者是展示在APP上

—进行分析：从数据中寻找一些规律

**## 3. 需要的原件和环境**

—Python3

—黑马Python基础班15天视频

：<httP://yun.itheima.ocm/course/214.html>

—基础语法 （字符串，列表，字典，判断循环）

—函数（函数的创建和调用）

—面向对象（如何创建一个类，如何使用这个类）

—pycharm

—Python编程

—Chrome浏览器

—分析网络请求用的

**## 4. 浏览器的请求**

—url

—在Chrome点击右键选择检查，点到network,

—url = 请求的协议 + 网站的域名 + 资源的路径 + 参数

—浏览器请求url 地址

—当前url对应的响应 + js + css + 图片 - - -》 elements中的内容

—爬虫请求url地址

—当前url对应的响应

—elements的内容和爬虫获取到的url地址的响应不同，爬虫中需要以当前url地址对应的  
响应为准提取数据

—当前url地址对应的响应在哪里

—从network中找到当前的url地址，点击response

—在页面上右键显示网页源码

**## 5. 认识HTTP、HTTPS**

-- HTTP:超文本传输协议

--以明文的形式传输

--效率高，但是不安全

--HTTPS:HTTP + SSL（安全套接字层）

--传输之前数据先加密，之后解密获取内容

--效率低，但是安全

—get请求和post请求区别

—get请求没请求体，post有，get请求把数据放到url地址中

—post请求常用于登录注册，

—post请求携带的数据量比get请求大，多，常用于传输大文本的时候

—HTTP协议之请求

—1.请求行

—2.请求头

—3.User-Agent：用户代理：对方服务器能够通过user\_agent知道当前请求对方资源的是什么浏览器

—如果我们需要手机版的浏览器发送请求，对应的，就需要

把user\_agent改成手机版

—Cookie：用来存储用户信息的，没吃请求会被携带上发送给对法的浏览器

—要获取登录后才能访问的页面

—对方的服务器会通过Cookie来判断我们是一个爬虫

—3. 请求体

—携带体

—get请求没有请求体

—post请求有请求

—HTTP协议之响应

—响应头

—Set-Cookie： 对方服务器通过该字段设置cookie到本地

—响应体

—url地址对应的响应

**## request模块学习**

**## 使用事前**

—pip install reqeusts

**### 发送get ， post请求，获取响应**

—response = requests.get(url) # 发送get请求，请求url地址对应的响应

—response = requests.pos（url ，data = {请求体的字典}） ## 发送post请求

**### response 的方法**

—response . text

—该方式往往会出现乱码，出现乱码使用response .encoding=“utf-8”

—response.content.decode()

—把响应的二进制字节流转化为str类型

—response.requests.url # 发送请求的url地址

—response.url #response 响应的url地址

—response.requests.headers #请求头

—response.headers #响应请求

**### 获取网页源码的正确打开方式（通过下面三种方式一定能够获取到网页的正确解码之后的字符串）**

* 1.、response.content.decode()
* 2 、response.content.decode(“gbk”)
* 3 、response.text

**### 发送带header的请求**

—为了模拟浏览器，获取和浏览器一模一样的内容

、、、Python

Headers = {"**User-Agent”: “**

Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.84 Safari/537.36“}

response = requests.get（url ，headers=headers）

**## 使用超时参数**

—response.get(rul ,headers=headers,timeout=3) # 3秒必须返回响应，否则会报错

**### 处理cookie相关的请求**