day02

目标

- 课程介绍
- 复习
- 爬虫入门

课程介绍

- 本月:爬虫
- 下月: 小实训
- 大实训1/大实训2

要求

- **正常上课**
- **作业完成**

发短信: 给家长发短信提醒!

复习

- 爬虫框架 requests `pip insall xxx`
- 爬虫代码步骤

```
#1. 定义地址       # 找url地址!
#2. 发送请求       request.get()  request.post()
request.post(url,headers=headers,date=POST请求参数,params=GET请求参数)

get请求参数如何传递?
post请求参数如何传递?
如何添加请求头?   header中的user-agent用于模拟浏览器!

# 3. 获取响应结果
  获取为文本/json/解决中文乱码

 print(resposne.status_code) # 响应的状态码 200 表示请求网站成功
resposne.encoding='utf-8' # 设置相应的字符编码
print(resposne.text) # 获取响应文本
# print(resposne.json()) # 获取响应json数据
```

```
print(resposne.headers) # 获取响应的头资料!
print(resposne.content) #获取响应内容

# 4.存储
  文件/数据库
```

网络爬虫入门

> 网络爬虫也叫"网络蜘蛛" ,用于爬取网络资源的一个技术!
>
> 百度和谷歌是最强大的爬虫, 一单用户网站有数据更新,百度和谷歌就可以爬取到!
>
> 网络爬虫/网络蜘蛛: WebSpider

爬虫核心知识点介绍

- 爬虫原理和爬虫组件
- 解析数据(正则re,xpath,bs4.json....)
- 动态html处理(反扒)
- 框架Scrapy
- 案例/分布式爬虫

简介

- 大数据时代公司数据的来源:

```
# 1. 各种大公司台网站(百度指数,阿里指数,腾讯指数...)  公司内部数据
# 2. 政府公开数据
# 3. 第三方数据平台(聚合数据.BAT三家公司都有)
# 4. 数据服务公司
# 重要: 通过爬虫技术自己从网上爬!!!!
```

- 爬取网络中数据技术：网络爬虫/网络蜘蛛

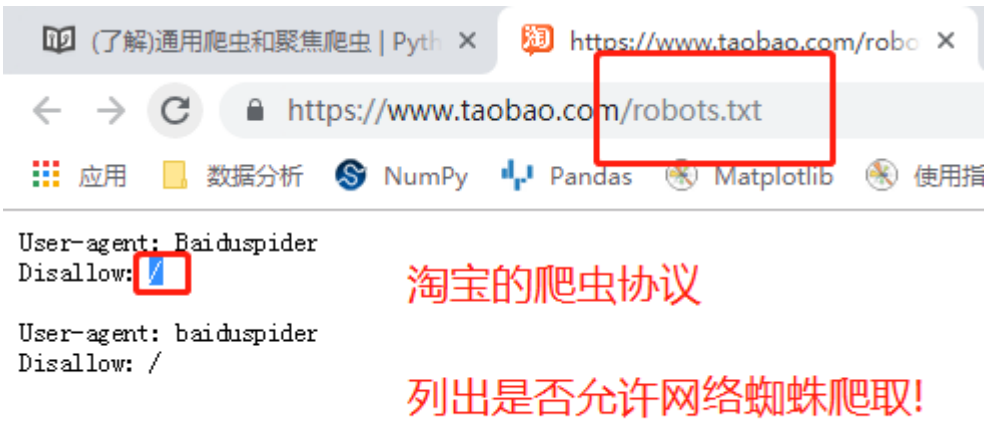
网络爬虫（又称为网页蜘蛛，网络机器人，在FOAF社区中间，更经常的称为网页追逐者），是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。 WebSpider

面试问题: 会写脚本么?

```
# py 文件就是python脚本文件!
#1. 回答: 会,我写过很多方面脚本, 做服务器运维的,做web的,处理数据的,爬虫相关的...
#2. 问: 咱公司写脚本处理什么业务啊?
```

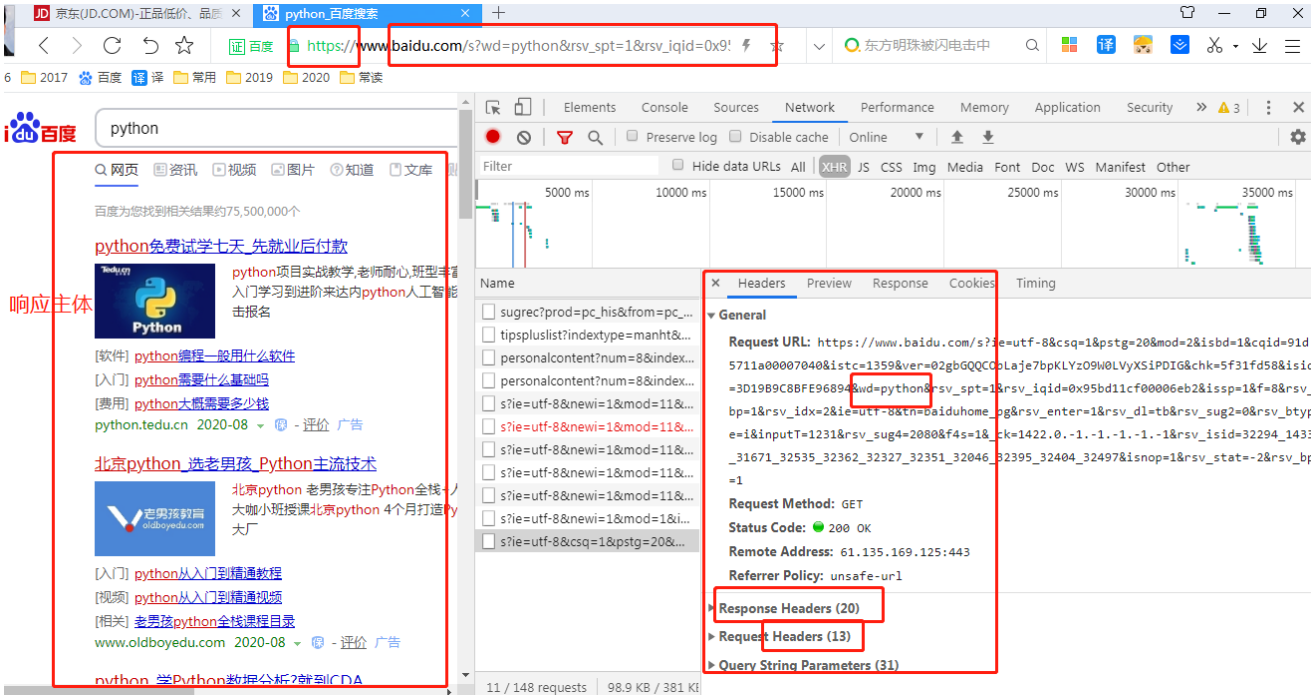
- 爬虫分类
  - 通用爬虫(百度,谷歌等搜索引擎)
  - **聚焦爬虫** (自己写的针对某个网站的爬虫)

  聚焦爬虫:只抓取与需求相关的网页信息,就是我们自己写的针对某个网站的爬虫
- 百度爬取数据原理/步骤
  - 1. 爬网页 **任何网站必须遵循Robots爬虫协议,在网站根目录加robots.txt**
  - 2. 存储
  - 3. 预处理
  - 4. **提供检索服务，网站排名**



http协议

> 互联网信息传输遵循的是http协议
>
> 浏览器输入一个网站(www.baidu.com) 得到服务器的响应(能看到百度网页),
>
> http协议内部只带很多信息!
>
> 比如: 请求头,响应头,参数列表等....

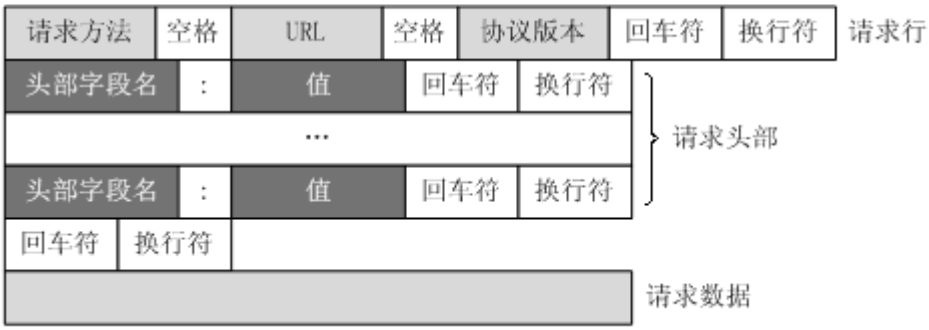


- http协议端口: 30 https协议端口:443 !

- 请求详细组成,通过浏览器F12工具需要会看!!

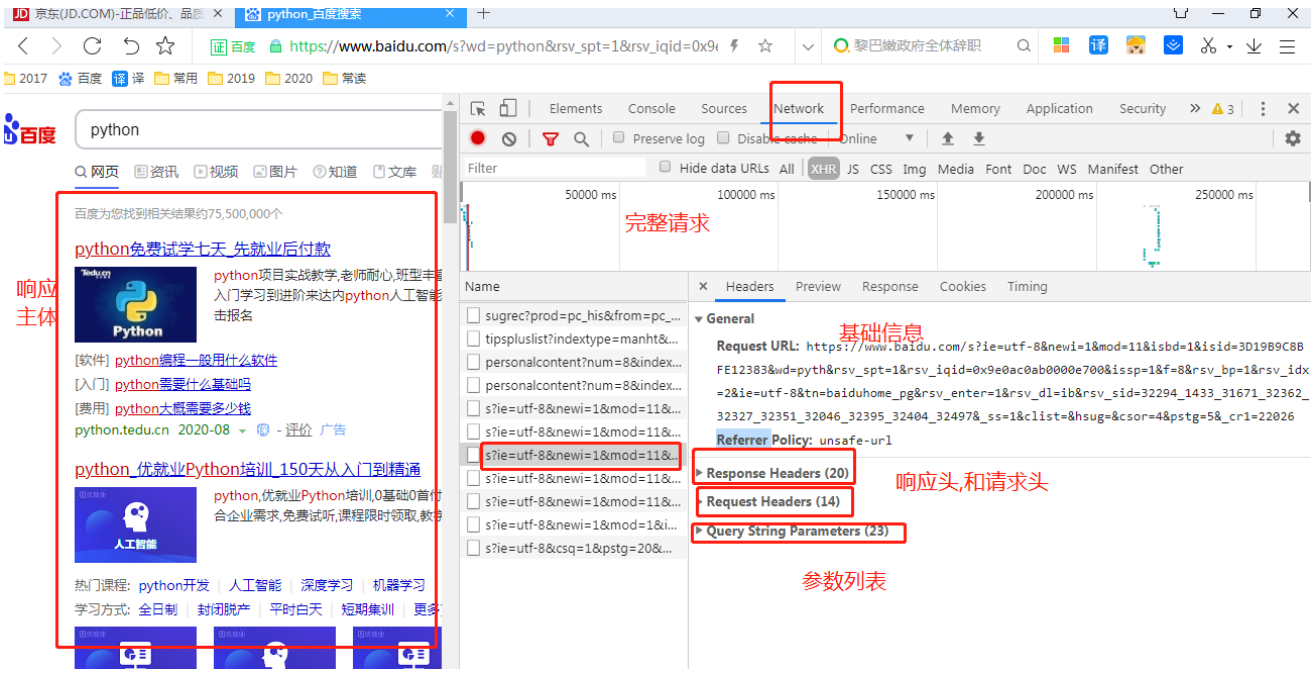
请求行 、 请求头部 、 空行 、 请求数据

四个部分组成，下图给出了请求报文的一般格式。

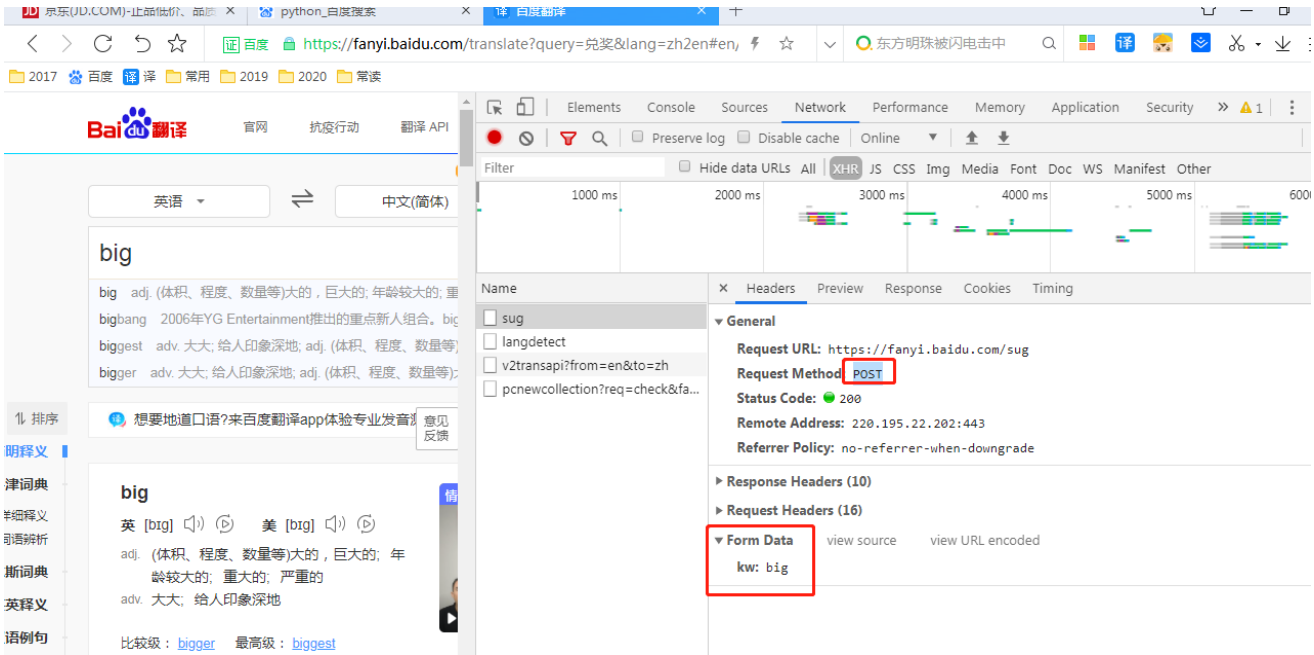


请求对象重点:

```
#1.  请求头中包含的浏览器信息
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/78.0.3904.108 Safari/537.36
#2. 请求方法
get/post
# 3.请求参数
request(url,headers,data=post参数,params=get参数)
```



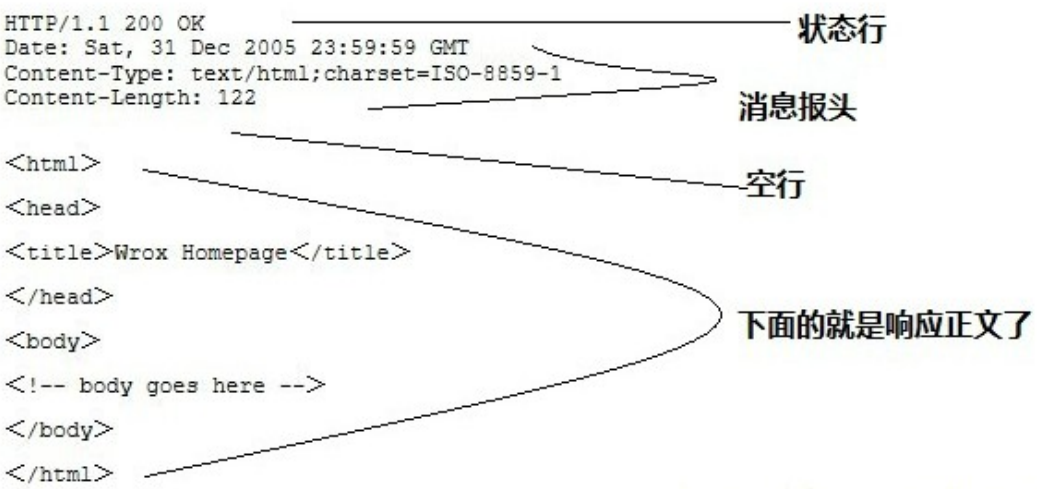
post请求参数:form Data



Response对象组成

服务端HTTP响应

HTTP响应也由四个部分组成，分别是： 状态行 、 消息报头 、 空行 、 响应正文




```
# 响应对象相应中重点
1. 响应类型        html/text  或 json
2. 获取相应主体内容    response.content    response.text/json()
3. 相应状态码:       response.status_code
      200表示成功      4xx:资源找不到   5xx: 程序内部错误

4. session或cookie信息
```

以上所有的请求和相应详细资料:通过 浏览器内置的调试工具F12查看到!

更加专业的: 抓包工具Fiddler, 可以帮我们抓APP数据!

小结

- 我们学的是爬取具体网站的(聚焦爬虫)
- 爬虫(WebSpider):就是把URL地址中指定的网络资源从网络流中读取出来，保存到本地。
- 要爬取数据,需要清晰http协议!(请求的组成,响应组成)
- F12浏览器调试工具,基本可满足日常开发

```
# 请求核心组成
url，请求方式，cookie/session，参数列表
# 响应核心组成
状态码，响应主体内容，响应内容类型,cookie和session. 响应数据类型(json输入如何查看)

F12工具,会在日常开发中经常用!

更加强大的爬虫转包工具(解析http请求和相应工具) Fiddler
```

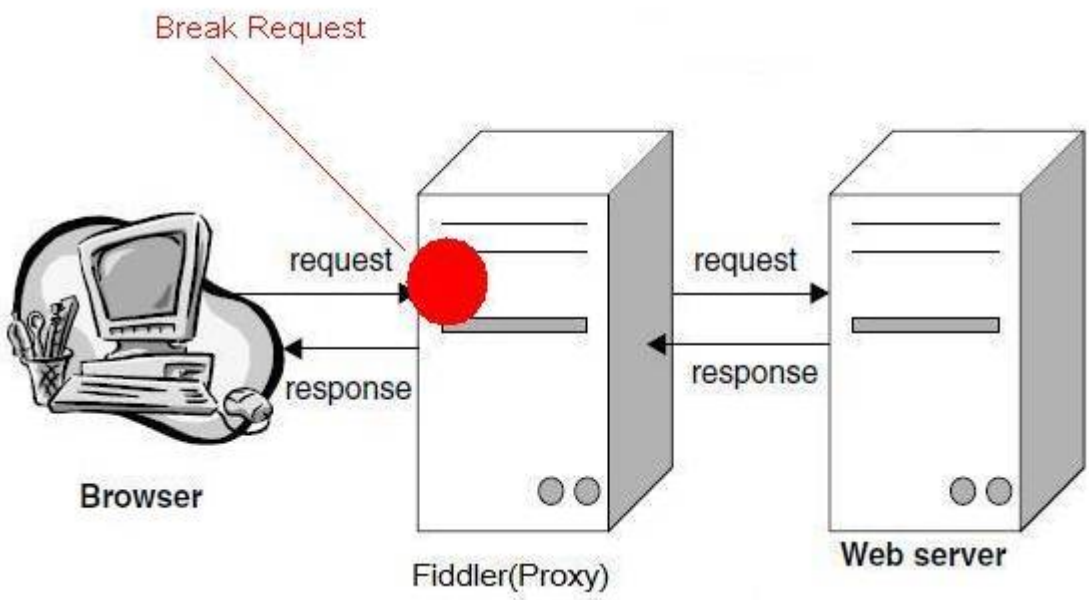
- Fiddler(抓包.抓App网络资源!)
  - 在电脑上,安装Android手机模拟器! (雷电模拟器、夜神模拟器、MUMU模拟器、逍遥模拟器、....)
  - 安装APP
  - 配置Fiddler...

Fiddler抓包工具

> 该工具可以方便解析http请求和相应的内容!

原理:代理

Fiddler 是以代理web服务器的形式工作的，它使用代理地址：127.0.0.1，端口：8888



```
1. 先配置fiddler
2. 下载谷歌插件
https://www.cnblogs.com/nicole-zhang/p/11955881.html
3. 配置chrome浏览器SwitchyOmega插件

原理: 在浏览器 输入 www.baidu.com---->谷歌插件自动找Fiddler---->请求百度服务器
```

各种爬虫框架

在Python中有很多库可以用来抓取网页

```
urllib 系列!
urllib
urllib2/3/4...

升级:
requests   [基于urllib]
```

- 下载对应爬虫库

```
pip install urllib
或
pip install  urllib3
或
pip install requests
```

- 使用步骤

```
# 所有的爬虫库使用步骤相同
1. 找url
2. 发出请求(url,请求头,参数列表,请求方式)
3. 获取响应解析
4. 存储
```

- 基础参考

1. 爬虫从入门到放弃系列博客!!!
2. 自己手册!
3. 爬虫小案例:
   https://blog.csdn.net/qq_40558166/article/details/102868801#13xpath_782

总结

爬虫基础原理

- 作业:

```
案例1：爬取百度产品列表
案例2：爬取新浪新闻指定搜索内容
案例3：爬取百度贴吧前十页（get请求）
案例4：爬取百度翻译接口
案例5：爬取有道翻译接口
```

- Fiddler软件配置!