网络爬虫简单介绍及Robots协议

# 爬虫

**网络爬虫**（又被称为**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，自动的抓取万维网信息的程序或者脚本。**

**网络未来的发展趋势： The Website is the API......**



爬虫需要**遵守Robots协议**，Robots.txt 网络爬虫排除标准，一般存储在网站的根目录下。

# Python爬虫需要学习的模块

## Python基础知识：基本语法；

## Python的requests库或urllib/urllib2库的使用；

requests库比较好用。

## Python的正则表达式re库的使用；

## Python的BeautifulSoup库的使用（bs4库）；（煲汤）

## Python的专业爬虫框架Scrapy；

## Python爬虫的更高级的功能，如爬取JavaScript文件等。

# 完成爬虫的技术路线

### requests+re+BeautifulSoup

### urllib+re+BeautifulSoup

由于urllib版本混乱，完全可以用requests库替代。

### Scrapy框架（5+2结构）

# 模块具体介绍

## Python基础知识

## Python urllib和urllib2 库的用法或requests库

urllib和urllib2库是学习Python爬虫最基本的库，利用这个库我们可以得到网页的内容，并对内容用正则表达式提取分析，得到我们想要的结果。这个在学习过程中我会和大家分享的。

## Python正则表达式

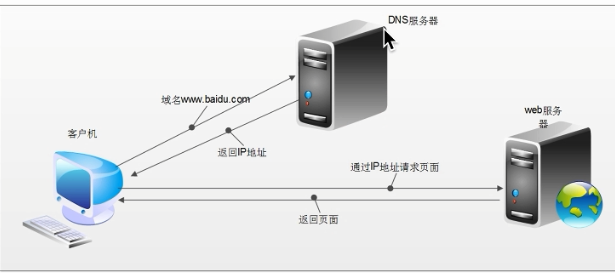
**Python正则表达式**是一种用来匹配字符串的强有力的武器。它的设计思想是用一种描述性的语言来给字符串定义一个规则，凡是符合规则的字符串，我们就认为它“匹配”了，否则，该字符串就是不合法的。

## Python框架Scrapy

如果你是一个Python高手，基本的爬虫知识都已经掌握了，那么就寻觅一下**Python框架**吧，我选择的框架是**Scrapy框架**。

# 网络爬虫的基本原理

## B/S与C/S模式

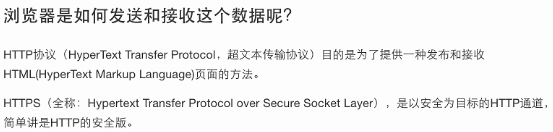


## 网络爬虫



首先是把**网页的大体结构代码**发过来，对于音乐、图片或视频资源类的文件，等到浏览器解析到img等时，浏览器会自动再次发起请求，获取这些资源。这样

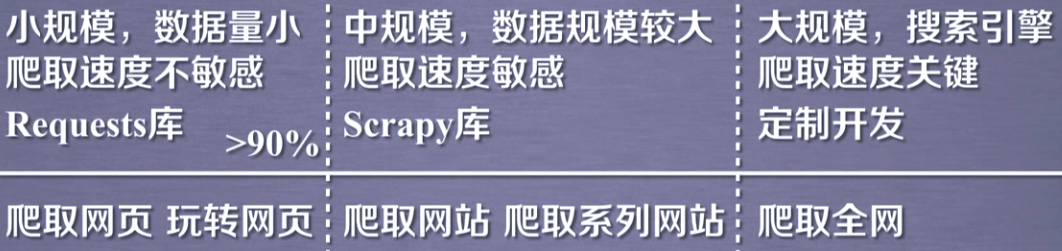
就加快了显示网页的速度。



# 北京理工大学嵩老师教学计划

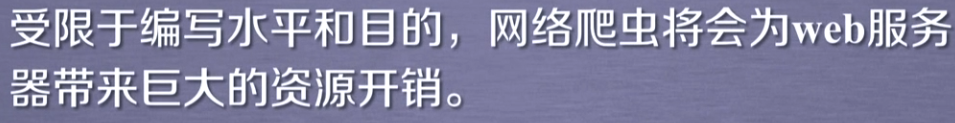


# 网络爬虫的尺寸规模



# 网络爬虫带来的问题

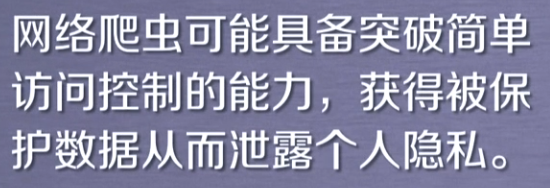
## 网络爬虫的“骚扰”



1. **网络爬虫的法律风险**



1. **网络爬虫泄露隐私**

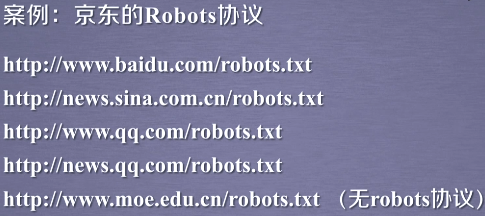


# Robots协议

## Robots协议



## 网站的robots.txt文件一般放在网站的根目录下。

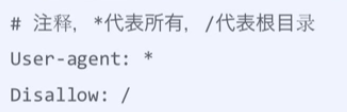




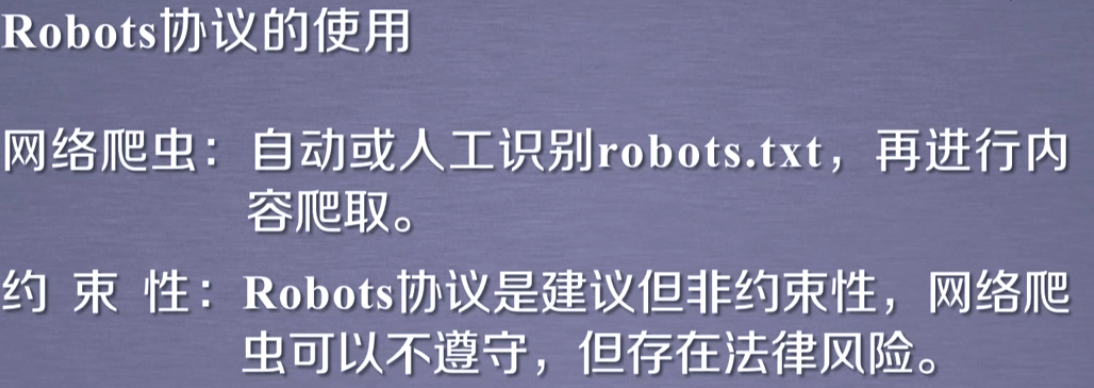
## Robots协议的基本语法

**Robots利用User-agent和Disallow来约束网络爬虫。**

**robots文件中一般只具有： User-agent和Disallow。**



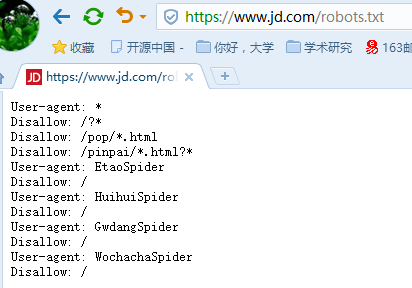
## Robots协议的使用



注意：当利用爬虫是为了获取网页信息，但是访问量很小，接近真人操作，这时候可以不用遵守**robots协议**，不会对网站造成危害。

## 示例：京东网站的robots协议

**访问 http://www.jd.com/robots.txt**



## Scrapy对robots.txt协议的支持设置

### Scrapy框架默认支持Robots协议

**Scrapy框架**在爬去网站时，首先检查该网站的Robots.txt，判断哪些可以爬取，哪些不可以爬取。

可以在**settings.py**文件中，禁止Scrapy遵循Robots协议。

关闭scrapy自带的**ROBOTSTXT\_OBEY**功能，在setting找到这个变量，设置为False即可解决。 默认情况下**ROBOTSTXT\_OBEY = True**

将其修改为**ROBOTSTXT\_OBEY = False** 即可。

### 问题描述

由于淘宝网站禁止了所有的爬虫，所以Scrapy在爬去之前，首先下载robots.txt协议，发现禁止爬取，则就不会继续爬行。报出DEBUG：

2017-08-08 09:05:43 [scrapy.core.engine] **DEBUG**: Crawled (200) <GET **https://s.taobao.com/robots.txt**> (referer: None)

2017-08-08 09:05:43 [scrapy.downloadermiddlewares.robotstxt] **DEBUG**: Forbidden by robots.txt: <GET <https://s.taobao.com/search?q=ipad+pro&rs=up&rsclick=2&preq=ipad>>

### 解决办法

**解决办法**：关闭scrapy自带的**ROBOTSTXT\_OBEY**功能，在setting找到这个变量，设置为**False**即可解决。 默认情况下ROBOTSTXT\_OBEY = **True**

将其修改为ROBOTSTXT\_OBEY = **False** 即可。

### 其他

**robots.txt文件**中规定了本站点允许的爬虫机器爬取的范围（比如你不想让百度爬取你的页面，就可以通过**robot**来限制），因为**默认scrapy遵守robots协议**，所以会先请求这个文件查看自己的权限，而我们现在访问这个url得到

User-agent: \*

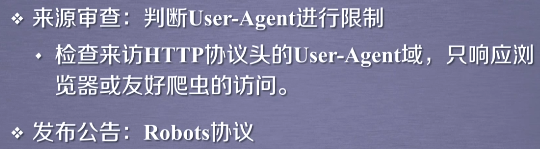
Disallow: /

可以看见，淘宝disallow根目录以下所有页面。所以scrapy就停止了之后的请求和页面解析。 我们在setting改变**ROBOTSTXT\_OBEY为False**，让scrapy不要遵守robot协议，之后就能正常爬取了。

### 其他限制爬虫方法

**对于使用robot协议的站点，只需要我们的爬虫不遵守该协议，就可以了，但是对于防止爬虫爬取，站点还有检查请求头、检查ip等等手段，还需要其他的相应处理。**

# 对网络爬虫的限制



### user-agent限制

对于user-agent ，可以通过Request的headers参数修改，如

req = scrapy.Request(url="http://www.xicidaili.com/nn/%s"%i,callback=self.parse,**headers={'user-agent':'Mozilla/5.0'}**)

### ip的限制

有些网站对ip地址进行检查禁用等，这时候需要利用**代理ip**进行爬取，方法类似，需要参数**meta**： request.meta[‘proxy’]=”http://\*\*\*.\*\*\*.\*\*\*.\*\*\*”

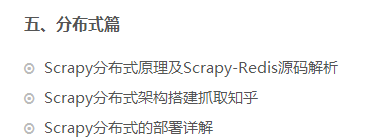
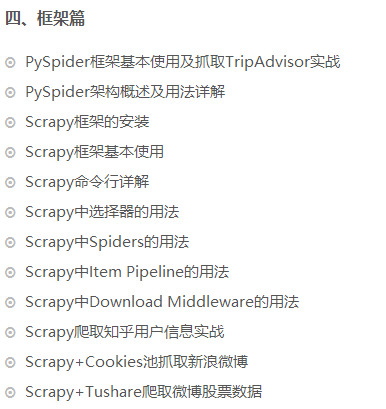


### robots协议限制

对于**Robots.txt**，直接忽视掉就可以。对于Scrapy框架，需要禁止其遵守Robots协议。

# Python3爬虫教程





# 爬虫展望

