第一章：

http ://www.robotstxt .org

1、 robots.txt可以让爬虫了解爬取该网站时存在哪些限制。

2、robots.txts

关于robots.txt协议的更多信息

http : // www. robotstxt . org

#section 1

#robots.txt文件禁止用户代理为BadCrawler的爬虫爬取该网 站,不过这种写法可能无法起 到应有的作用，因为恶意爬虫根本不会遵从robots. txt 的要求

User-agent:BadCrawler

Disallow:

#sect ion 2 规定,无论使用哪种用户代理,都应该在两次下载请求之间给出5秒的抓取延迟,我们需要遵从该建议以避免服务器过载

#section 2

User-agent: \*

Crawl-delay:5

Disallow: trap /trap链接,用于封禁那些爬取了不允许链接的恶意爬虫

#sect ion 3定义了一个Sitemap文件

#section 3

Sitemap: <http://example.webscraping.com/sitemap.xml>

3、检查网站地图

获取网站地图标准的定义：

<https://www.sitemaps.org/protocol.html>

4、估算网站大小

目标网站的大小会影响我们如何进行爬取。

通过Goo gle搜索的s ite关键词过滤域名结果,从而获取网站大小

例如：搜索site: example.webscraping. com/view

5、识别网站所用技术

有一个十分有用的工具可以检查网站构建的技术类型----builtwith模块

>>> import builtwith

>>> builtwith.parse('http://example.webscriping.com')

>>> builtwith.parse('http://example.webscraping.com')

{u'javascript-frameworks': [u'jQuery', u'Modernizr', u'jQuery UI'], u'web-frameworks': [u'Web2py', u'Twitter Bootstrap'], u'programming-languages': [u'Python'], u'web-servers': [u'Nginx']}

>>>

6、寻找网站所有者

>>> import whois

>>> print whois.whois('appspot.com')

7、编写第一个爬虫网站

>>> def download(url):

... print 'downloading:',url

... try:

... html = urllib2.urlopen(url).read()

... except urllib2.URLError as e:

... print 'download error:', e.reason

... html = None

... return html

8、重试下载

查看HTTP错误的完成列表

<https://tools.ietf.org/html/rfc7231#section-6>

当函数遇到5XX错误码时，将会递归调用函数自身进行重试。

9、重试下载代码

>>> def download(url,num\_retries = 2):

... print 'Downloading:',url

... try:

... html = urllib2.urlopen(url).read()

... except urllib.URLError as e:

... print 'download error:', e.reasion

... html = None

... if num\_retries > 0:

... if hasattr(e,'code') and 500 <= e.code < 600:

... return download(url,num\_retries-1)

... return html

...

>>>

10、设置用户代理

设定一个默认的用户代理’wswp’

>>> def download(url,user\_agent='wswp',num\_retries=2):

... print 'Downloading:',url

... headers = {'User-agent',url}

... request = urllib2.Request(url,headers=headers)

... try:

... html = urllib2.urlopen(request).read()

... except urllib2.URLError as e:

... print 'Download error:',e.reason

... html = None

... if num\_retries > 0:

... if hasattr(e,'code') and 500 <= e.code < 600:

... return download(url,user\_agent,num\_retries-1)

... return html

...

>>>

1. python 迭代器

>>> import itertools

>>> natuals = itertools.count(1)

>>> for n in natuals:...

print n

count(1)会创建一个无限的迭代器。