

[Python]网络爬虫（12）：爬虫框架Scrapy的第一个爬虫示例入门教程 - 汪海的实验室 - 博客频道

分类：

爬虫（18）



Python（26）



版权声明：本文为博主原创文章，未经博主允许不得转载。

（建议大家多看看官网教程：[教程地址](#)）

我们使用[dmoz.org](#)这个网站来作为小抓抓一展身手的目标。首先先要回答一个问题。问：把网站装进爬虫里，总共分几步？答案很简单，四步：

- 新建项目（Project）：新建一个新的爬虫项目
- 明确目标（Items）：明确你想要抓取的目标
- 制作爬虫（Spider）：制作爬虫开始爬取网页
- 存储内容（Pipeline）：设计管道存储爬取内容

好的，基本流程既然确定了，那接下来就一步一步的完成就可以了。1. 新建项目（Project）在空目录下按住Shift键右击，选择“在此处打开命令窗口”，输入一下命令：

其中，tutorial为项目名称。

可以看到将会创建一个tutorial文件夹，目录结构如下：下面来简单介绍一下各个文件的作用：

- scrapy.cfg：项目的配置文件
- tutorial/：项目的Python模块，将会从这里引用代码
- tutorial/items.py：项目的items文件
- tutorial/pipelines.py：项目的pipelines文件
- tutorial/settings.py：项目的设置文件
- tutorial/spiders/：存储爬虫的目录

2. 明确目标（Item）

在Scrapy中，items是用来加载抓取内容的容器，有点像[Python](#)中的Dic，也就是字典，但是提供了一些额外的保护减少错误。

一般来说，item可以用scrapy.item.Item类来创建，并且用scrapy.item.Field对象来定义属性（可以理解成类似于ORM的映射关系）。

接下来，我们开始来构建item模型（model）。首先，我们想要的内容有：

- 名称（name）
- 链接（url）
- 描述（description）

修改tutorial目录下的items.py文件，在原本的class后面添加我们自己的class。因为要抓dmoz.org网站的内容，所以我们可以将其命名为DmozItem：

```
1. # Define here the models for your scraped items
2. #
3. # See documentation in:
4. # http://doc.scrapy.org/en/latest/topics/items.html
5. from scrapy.item import Item, Field
6. class TutorialItem(Item):
7.     # define the fields for your item here like:
8.     # name = Field()
9.     pass
10. class DmozItem(Item):
11.     title = Field()
12.     link = Field()
13.     desc = Field()
```

刚开始看起来可能会有些看不懂，但是定义这些item能让你用其他组件的时候知道你的 items到底是什么。

可以把Item简单的理解成封装好的类对象。

3. 制作爬虫（Spider）制作爬虫，总体分两步：先爬再取。也就是说，首先你要获取整个网页的所有内容，然后再取出其中对你有用的部分。3.1爬

Spider是用户自己编写的类，用来从一个域（或域组）中抓取信息。他们定义了用于下载的URL列表、跟踪链接的方案、解析网页内容的方式，以此来提取items。要建立一个Spider，你必须用scrapy.spider.BaseSpider创建一个子类，并确定三个强制的属性：

- name：爬虫的识别名称，必须是唯一的，在不同的爬虫中你必须定义不同的名字。
- start_urls：爬取的URL列表。爬虫从这里开始抓取数据，所以，第一次下载的数据将会从这些urls开始。其他子URL将会从这些起始URL中继承性生成。
- parse()：解析的方法，调用的时候传入从每一个URL传回的Response对象作为唯一参数，负责解析并匹配抓取的数据（解析为item），跟踪更多的URL。

这里可以参考宽度爬虫教程中提及的思想来帮助理解，教程传送：[\[Java\] 知乎下巴第5集：使用HttpClient工具包和宽度爬虫](#)。

也就是把Url存储下来并依此为起点逐步扩散开去，抓取所有符合条件的网页Url存储起来继续爬取。下面我们来写第一只爬虫，命名为dmoz_spider.py，保存在tutorial\spiders目录下。

dmoz_spider.py代码如下：

```
1. In [1]: sel.xpath('//title')
2. Out[1]: [<Selector xpath='//title' data=u'<title>Open Directory - Computers: Program
r'>]
3. In [2]: sel.xpath('//title').extract()
4. Out[2]: [u'<title>Open Directory - Computers: Programming: Languages: Python: Bo
oks</title>']
```

```

5. In [3]: sel.xpath('//title/text()')
6. Out[3]: [<Selector xpath='//title/text()' data=u'Open Directory - Computers: Programming:'>]
7. In [4]: sel.xpath('//title/text()').extract()
8. Out[4]: [u'Open Directory - Computers: Programming: Languages: Python: Books']

9. In [5]: sel.xpath('//title/text()').re('(\w+):')
10. Out[5]: [u'Computers', u'Programming', u'Languages', u'Python']

```

当然title这个标签对我们来说没有太多的价值，下面我们就来真正抓取一些有意义的东西。使用火狐的审查元素我们可以清楚地看到，我们需要的东西如下：



我们可以用如下代码来抓取这个标签：当然，前面的这些例子是直接获取属性的方法。我们注意到xpath返回了一个对象列表，那么我们也可以直接调用这个列表中对象的属性挖掘更深的节点

```

sites = sel.xpath('//ul/li')
for site in sites:
    title = site.xpath('a/text()').extract()
    link = site.xpath('a/@href').extract()
    desc = site.xpath('text()').extract()
    print title, link, desc

```

3.4xpath实战我们用shell做了这么久的实战，最后我们可以把前面学习到的内容应用到dmoz_spider这个爬虫中。在原爬虫的parse函数中做如下修改：

```

1. from scrapy.spider import Spider
2. from scrapy.selector import Selector
3. class DmozSpider(Spider):
4.     name = "dmoz"
5.     allowed_domains = ["dmoz.org"]
6.     start_urls = [
7.         "http://www.dmoz.org/Computers/Programming/Languages/Python/Books/",
8.         "http://www.dmoz.org/Computers/Programming/Languages/Python/Resources/"
9.     ]
10.    def parse(self, response):
11.        sel = Selector(response)
12.        sites = sel.xpath('//ul/li')
13.        for site in sites:

```

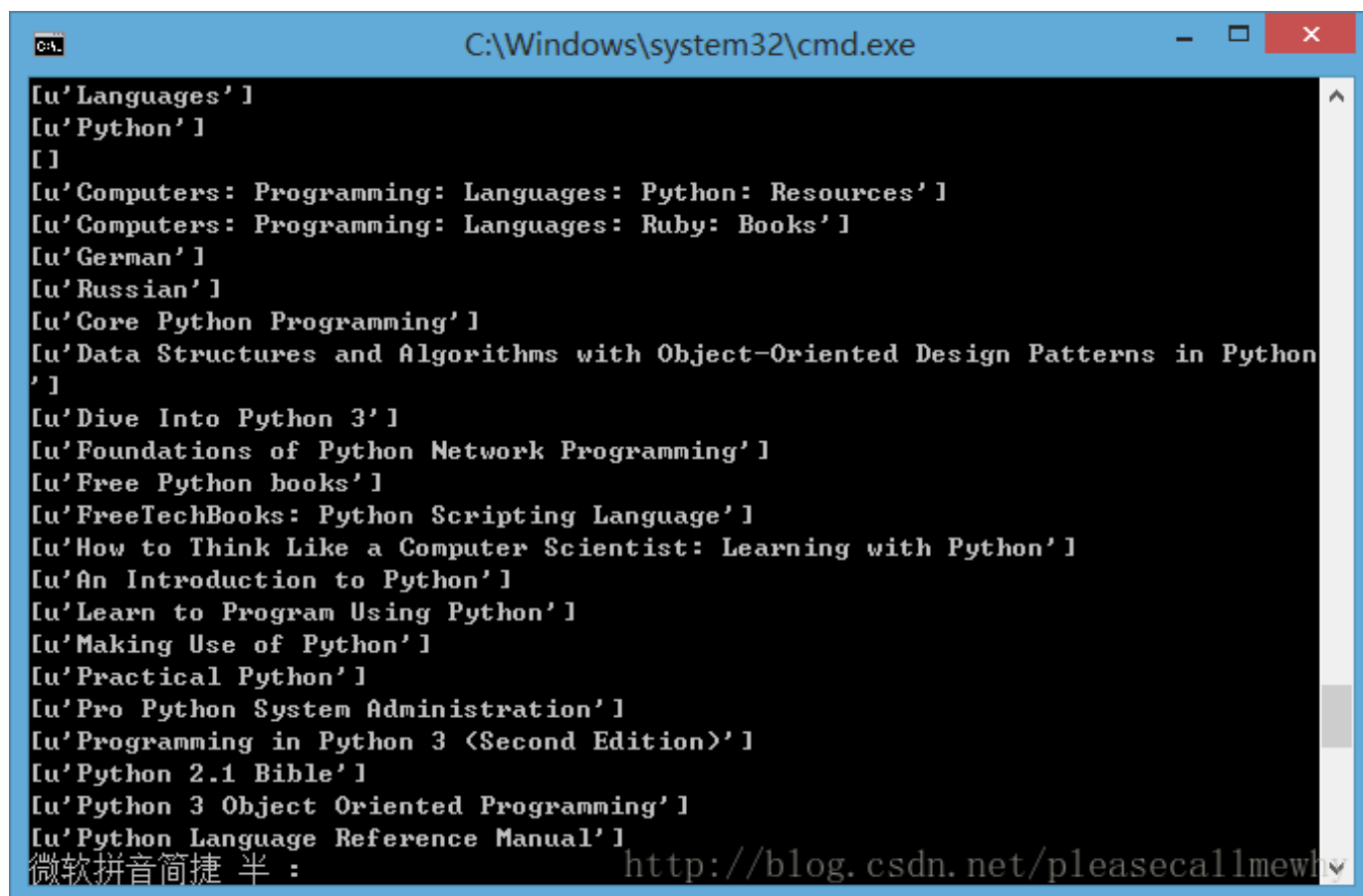
```
14.             title = site.xpath('a/text()').extract()
15.             link = site.xpath('a/@href').extract()
16.             desc = site.xpath('text()').extract()
17.             print title
```

注意，我们从scrapy.selector中导入了Selector类，并且实例化了一个新的Selector对象。这样我们就可以像Shell中一样操作xpath了。

我们来试着输入一下命令运行爬虫（在tutorial根目录里面）：

```
scrapy crawl dmoz
```

运行结果如下：



```
C:\Windows\system32\cmd.exe
[ u'Languages' ]
[ u'Python' ]
[ ]
[ u'Computers: Programming: Languages: Python: Resources' ]
[ u'Computers: Programming: Languages: Ruby: Books' ]
[ u'German' ]
[ u-Russian' ]
[ u'Core Python Programming' ]
[ u'Data Structures and Algorithms with Object-Oriented Design Patterns in Python' ]
[ u'Dive Into Python 3' ]
[ u'Foundations of Python Network Programming' ]
[ u'Free Python books' ]
[ u'FreeTechBooks: Python Scripting Language' ]
[ u'How to Think Like a Computer Scientist: Learning with Python' ]
[ u'An Introduction to Python' ]
[ u'Learn to Program Using Python' ]
[ u'Making Use of Python' ]
[ u'Practical Python' ]
[ u'Pro Python System Administration' ]
[ u'Programming in Python 3 (Second Edition)' ]
[ u'Python 2.1 Bible' ]
[ u'Python 3 Object Oriented Programming' ]
[ u'Python Language Reference Manual' ]
微软拼音简捷 半 : http://blog.csdn.net/pleasecallmewhy
```

果然，成功的抓到了所有的标题。但是好像不太对啊，怎么Top, Python这种导航栏也抓取出来了呢？我们只需要红圈中的内容：

[Top](#): [Computers](#): [Programming](#): [Languages](#): [Python](#): **Books (22)**

See also:

- [Computers: Programming: Languages: Python: Resources](#) (5)
- [Computers: Programming: Languages: Ruby: Books](#) (7)

This category in other languages:

[German](#) (7)

[Russian](#) (3)

- [Core Python Programming](#) - By Wesley J. Chun; Prentice Hall PTR, 2001, ISBN 0130260363. For experienced developers to improve extant skills; pro handling, functions, classes, built-ins. [Prentice Hall]
- [Data Structures and Algorithms with Object-Oriented Design Patterns in Python](#) - The primary goal of this book is to promote object-oriented d oriented design patterns. A secondary goal of the book is to present mathematical tools just in time. Analysis techniques and proofs are presented as needed
- [Dive Into Python 3](#) - By Mark Pilgrim, Guide to Python 3 and its differences from Python 2. Each chapter starts with a real code sample and explains it f changes in Python 3
- [Foundations of Python Network Programming](#) - This book covers a wide range of topics. From raw TCP and UDP to encryption with TLS, and then understanding of each field and how to do everything on the network with Python.
- [Free Python books](#) - Free Python books and tutorials.
- [FreeTechBooks: Python Scripting Language](#) - Annotated list of free online books on Python scripting language. Topics range from beginner to advan
- [How to Think Like a Computer Scientist: Learning with Python](#) - By Allen B. Downey, Jeffrey Elkner, Chris Meyers; Green Tea Press, 2002; ISBN as subject language. Thorough, in-depth approach to many basic and intermediate programming topics. Full text online and downloads: HTML PDF PS I

看来是我们的xpath语句有点问题，没有仅仅把我们需要的项目名称抓取出来，也抓了一些无辜的但是xpath语法相同的元素。审查元素我们发现我们需要的具有class='directory-url'的属性，那么只要把xpath语句改成sel.xpath('//ul[@class="directory-url"]/li')即可将xpath语句做如下调整：