**BeautifulSoup 解析网页: 基础**

作者: **莫烦**编辑: **莫烦 2017-12-29**

学习资料:

* [本节学习代码](https://github.com/MorvanZhou/easy-scraping-tutorial/blob/master/notebook/2-1-beautifulsoup-basic.ipynb)
* BeautifulSoup [英文官网](https://www.crummy.com/software/BeautifulSoup/bs4/doc/), [中文官网](https://www.crummy.com/software/BeautifulSoup/bs4/doc.zh/)
* 本节使用的爬虫[测试网页](https://morvanzhou.github.io/static/scraping/basic-structure.html)

[上节内容](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/1-01-understand-website/), 我们了解了网页 (html) 的基本构架, 知道了爬网页就是在这个构架中找到需要的信息. 那么找到需要的信息时, BeautifulSoup 就是一个找信息好帮手. 它能帮你又快有准地找到信息. 大大简化了使用难度.

[](https://morvanzhou.github.io/static/results/scraping/2-1-1.jpg)

我们总结一下爬网页的流程, 让你对 BeautifulSoup 有一个更好的定位.

1. 选着要爬的网址 (url)
2. 使用 python 登录上这个网址 (urlopen等)
3. 读取网页信息 (read() 出来)
4. **将读取的信息放入 BeautifulSoup**
5. **使用 BeautifulSoup 选取 tag 信息等 (代替正则表达式)**

初学的时候总是搞不懂这些包是干什么的, 现在你就能理解这个 BeautifulSoup 到底是干什么的了.

[**安装**](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/2-01-beautifulsoup-basic/#%E5%AE%89%E8%A3%85)

等什么, 知道 BeautifulSoup 这么方便, 我们就赶紧装一个吧. 安装的步骤很简单, 用 pip 就能轻松安装.

# Python 2+

pip install beautifulsoup4

# Python 3+

pip3 install beautifulsoup4

注意在名字后面还有个 “4”, 可能是代表第4版吧. 如果你在安装的时候遇到任何问题, 请参考他们[官网](https://www.crummy.com/software/BeautifulSoup/bs4/doc.zh/#id5)上的解决方案.

[**简单实用方法**](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/2-01-beautifulsoup-basic/#%E7%AE%80%E5%8D%95%E5%AE%9E%E7%94%A8%E6%96%B9%E6%B3%95)

这次我们还是爬一爬上次爬的那个[基本网页](https://morvanzhou.github.io/static/scraping/basic-structure.html). BeautifulSoup 使用起来非常简单, 我们先按常规读取网页.

**from** bs4 **import** BeautifulSoup

**from** urllib.request **import** urlopen

# if has Chinese, apply decode()

html = urlopen("https://morvanzhou.github.io/static/scraping/basic-structure.html").read().decode('utf-8')

print(html)

回顾一下, 每张网页中, 都有两大块, 一个是 <head>, 一个是 <body>, 我们等会用 BeautifulSoup 来找到 body 中的段落 <p> 和所有链接 <a>.

<!DOCTYPE html>

<html lang="cn">

<head>

<meta charset="UTF-8">

<title>Scraping tutorial 1 | 莫烦Python</title>

<link rel="icon" href="https://morvanzhou.github.io/static/img/description/tab\_icon.png">

</head>

<body>

<h1>爬虫测试1</h1>

<p>

这是一个在 <a href="https://morvanzhou.github.io/">莫烦Python</a>

<a href="https://morvanzhou.github.io/tutorials/scraping">爬虫教程</a> 中的简单测试.

</p>

</body>

</html>

读取这个网页信息, 我们将要加载进 BeautifulSoup, 以 lxml 的这种形式加载. 除了 lxml, 其实还有[很多形式的解析器](https://www.crummy.com/software/BeautifulSoup/bs4/doc.zh/#id9), 不过大家都推荐使用 lxml 的形式. 然后 soup 里面就有着这个 HTML 的所有信息. 如果你要输出 <h1> 标题, 可以就直接 soup.h1.

soup = BeautifulSoup(html, features='lxml')

print(soup.h1)

"""

<h1>爬虫测试1</h1>

"""

print('\n', soup.p)

"""

<p>

这是一个在 <a href="https://morvanzhou.github.io/">莫烦Python</a>

<a href="https://morvanzhou.github.io/tutorials/scraping">爬虫教程</a> 中的简单测试.

</p>

"""

如果网页中有过个同样的 tag, 比如链接 <a>, 我们可以使用 find\_all() 来找到所有的选项. 因为我们真正的 link 不是在 <a> 中间 </a>, 而是在 <a href="link"> 里面, 也可以看做是 <a> 的一个属性. 我们能用像 Python 字典的形式, 用 key 来读取 l["href"].

"""

<a href="https://morvanzhou.github.io/tutorials/scraping">爬虫教程</a>

"""

all\_href = soup.find\_all('a')

all\_href = [l['href'] for l in all\_href]

print('\n', all\_href)

# ['https://morvanzhou.github.io/', 'https://morvanzhou.github.io/tutorials/scraping']

懂得这些还是远远不够的, 真实情况往往比这些复杂. BeautifulSoup 还有很多其他的选择”增强器”. 下次, 我们来了解一些 CSS 的概念, 用 BeautifulSoup 加上 CSS 来选择内容.