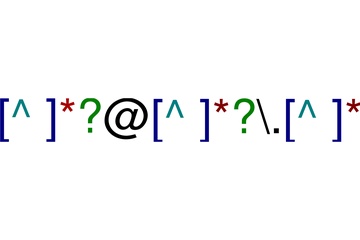
# BeautifulSoup 解析网页: 正则表达

作者: **莫烦**编辑: **莫烦 2017-12-29**

学习资料:

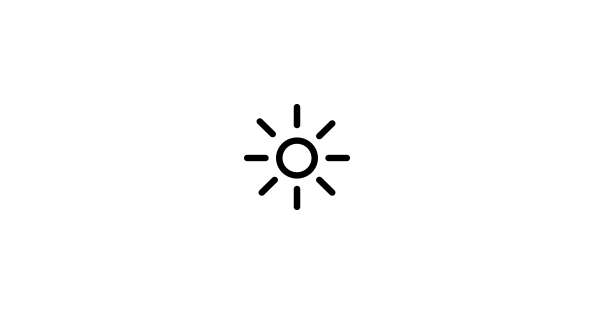
* [本节学习代码](https://github.com/MorvanZhou/easy-scraping-tutorial/blob/master/notebook/2-3-beautifulsoup-regex.ipynb)
* 本节使用的爬虫[测试网页](https://morvanzhou.github.io/static/scraping/table.html)
* 我的完整[正则表达式教程](https://morvanzhou.github.io/tutorials/python-basic/basic/13-10-regular-expression/)

正则表达式, 是处理文本信息的重要工具, 除了 Python, 在其他的程序语言中, 也有十分重要的地位. 如果将正则表达式 + BeautifulSoup, 岂不是完美中的完美, 哈哈. 我们今天就来看看, 在 BeautifulSoup 中如何使用正则表达式, 获取更有难度的信息.

[](https://morvanzhou.github.io/static/results/scraping/2-3-1.jpg)

## [正则表达式](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/2-03-beautifulsoup-regex/#%E6%AD%A3%E5%88%99%E8%A1%A8%E8%BE%BE%E5%BC%8F)

正则表达式很厉害, 它能用简单的规则匹配到多样化的文本信息. 在做爬虫教程之前, 我特地做了一个[正则表达式的教程](https://morvanzhou.github.io/tutorials/python-basic/basic/13-10-regular-expression/), 为爬虫做铺垫. 所以有兴趣了解使用正则表达式的朋友, 都可以看看这个非常全的正则教程.

[](https://morvanzhou.github.io/static/results/scraping/2-3-2.png)

这次的教程有一些表格形式的 HTML, 在表格中, 有一些信息的格式类似, 我们先用 BeautifulSoup 筛选一些, 然后完全可以用正则给匹配出来. 比如你想下载这个页面的图片, 我们就可以将图片形式的 url 个匹配出来. 之后再下载就简单多了.

## [正则匹配](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/2-03-beautifulsoup-regex/#%E6%AD%A3%E5%88%99%E5%8C%B9%E9%85%8D)

我们先读取这个[网页](https://morvanzhou.github.io/static/scraping/table.html). 导入正则模块 re.

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

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

**import** re

# if has Chinese, apply decode()

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

我们发现, 如果是图片, 它们都藏在这样一个 tag 中:

<td>

<img src="https://morvanzhou.github.io/static/img/course\_cover/tf.jpg">

</td>

所以, 我们可以用 soup 将这些 <img> tag 全部找出来, 但是每一个 img 的链接(src)都可能不同. 或者每一个图片有的可能是 jpg 有的是 png, 如果我们只想挑选 jpg 形式的图片, 我们就可以用这样一个正则 r'.\*?\.jpg' 来选取. 把正则的 compile 形式放到 BeautifulSoup 的功能中, 就能选到符合要求的图片链接了.

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

img\_links = soup.find\_all("img", {"src": re.compile('.\*?\.jpg')})

for link in img\_links:

print(link['src'])

"""

https://morvanzhou.github.io/static/img/course\_cover/tf.jpg

https://morvanzhou.github.io/static/img/course\_cover/rl.jpg

https://morvanzhou.github.io/static/img/course\_cover/scraping.jpg

"""

又或者我们发现, 我想选一些课程的链接, 而这些链接都有统一的形式, 就是开头都会有https://morvan., 那我就将这个定为一个正则的规则, 让 BeautifulSoup 帮我找到符合这个规则的链接.

course\_links = soup.find\_all('a', {'href': re.compile('https://morvan.\*')})

for link in course\_links:

print(link['href'])

"""

https://morvanzhou.github.io/

https://morvanzhou.github.io/tutorials/scraping

https://morvanzhou.github.io/tutorials/machine-learning/tensorflow/

https://morvanzhou.github.io/tutorials/machine-learning/reinforcement-learning/

https://morvanzhou.github.io/tutorials/data-manipulation/scraping/

"""

学习了这么多实用的方法, 我们[接下来](https://morvanzhou.github.io/tutorials/data-manipulation/scraping/2-04-practice-baidu-baike/)就来做一个小实战, 让我们的爬虫在百度百科上自由爬行, 在各个百科网页上跳来跳去.