

Web Scraping

KUNAAL NAIK

Lifeaholic Channel



LinkedIn profile of Kunaal Naik. The profile picture shows a man with glasses and a dark suit. The background of the profile banner is an abstract painting with vibrant colors like red, yellow, green, and black. The LinkedIn logo is visible in the top left corner of the profile card. To the right of the profile picture are three dots and a pencil icon.

Kunaal Naik
Analytics Practitioner, Lifeaholic Evangelist, Learner, YouTuber and Apprentice Philosopher
Brillio • Institute of Aeronautical Engineering
Bengaluru, Karnataka, India • 500+ 



fxexcel@gmail.com

Stay in touch!

2 Methods

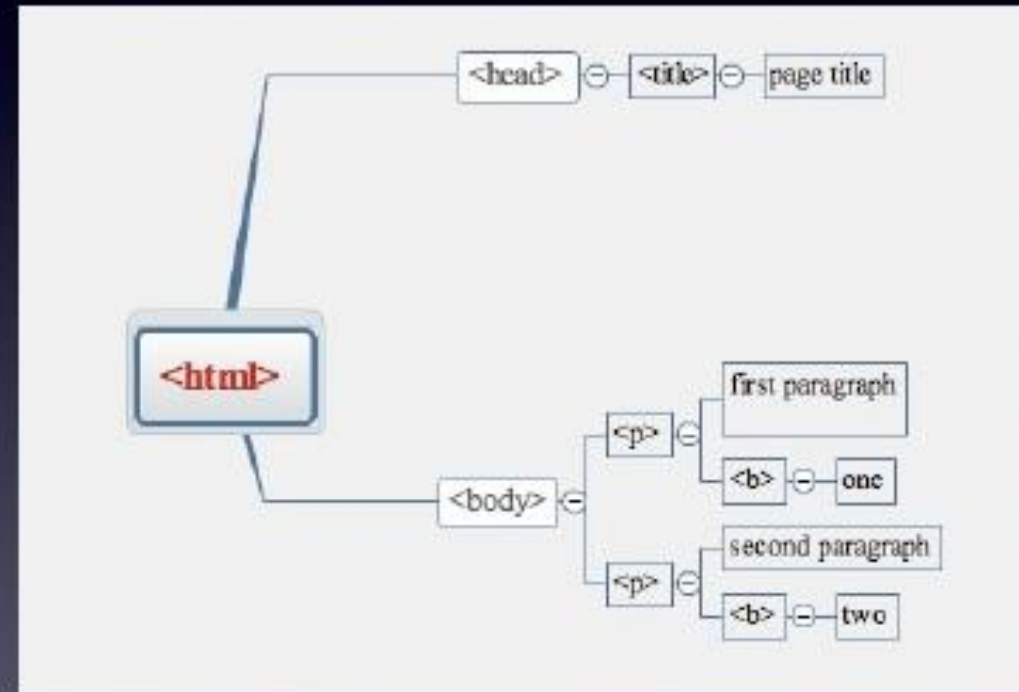
- ▶ BeautifulSoup
 - ▶ Used to pull data out of HTML and XML
 - ▶ Can get data from java or dynamically loaded pages
 - ▶ Easy to beign with and short learning curve
 - ▶ Requires requests, urllib2, lxml
- ▶ Scrapy
 - ▶ Fast high-level web crawling and web scraping framework
 - ▶ It is used to crawl websites and extract structured data from their pages
 - ▶ Limitations: data from java script or loading dynamically
 - ▶ Overcome by using splash, selenium
 - ▶ Steep Learning curve
 - ▶ Self Sufficient, does not require other packgaes

Differences

| Framework | BeautifulSoup | Scrapy |
|----------------|---|--|
| Learning Curve | Very easy to learn, beginner-friendly | Learning curve of <code>Scrapy</code> is much steeper, you need to read some Scrapy Tutorial or Scrapy Doc to get started, and work hard to become an Scrapy expert. |
| Ecosystem | Not many related projects or plugins | Many related projects, plugins on open source websites such as Github, and many discussions on StackOverflow can help you fix the potential issue. |
| Extensibility | Not very easy to extend the project | You can easily develop custom middleware or pipeline to add custom function, easy to maintain. |
| Performance | You need import <code>multiprocessing</code> to make it run quicker | Very efficient, web pages can be crawled in a short time, on the other hand, in many cases you need to set <code>download_delay</code> to avoid getting spider banned. |

What is Parsing?

```
<html>
  <head>
    <title>
      page title
    </title>
  </head>
  <body>
    <p id="firstpara" align="center">
      first paragraph
      <b>
        one
      </b>
    </p>
    <p id="secondpara" align="blah">
      second paragraph
      <b>
        two
      </b>
    </p>
  </body>
</html>
```



BeautifulSoup

► Some Start up codes

```
8 from bs4 import BeautifulSoup
9 import requests
10
11 with open('samplesite.html') as html_file:
12     soup = BeautifulSoup(html_file, 'lxml')
13
```

► Extract all to a database

```
40 for article in soup.find_all('div', class_='article'):
41     headline = article.h2.a.text
42     print(headline)
43     summary = article.p.text
44     print(summary) |
```

Scrapy – Exploring Shell

- ▶ To experiment
 - ▶ scrapy shell
- ▶ Fetch content of site
 - ▶ `fetch("http://quotes.toscrape.com/")`
- ▶ Check response
 - ▶ `view(response), print(response.text)`
- ▶ Extraction using css method(other is xpath)
 - ▶ `response.css(".title::text").extract()`
- ▶ Exit Shell
 - ▶ `exit()`

Scrapy – Scraping Codes

- ▶ Start Project
 - ▶ scrapy startproject quotes
- ▶ Create Spider
 - ▶ scrapy genspider quoteBot <http://quotes.toscrape.com/>
- ▶ Run Spider
 - ▶ scrapy crawl quoteBot
- ▶ Run Spider with export command
 - ▶ scrapy crawl quoteBot -o exp2.json
 - ▶ scrapy crawl quoteBot -o exp2.csv

Scrapy – Scraping Codes

```
ourfirstscrapper/  
├── ourfirstscrapper  
│   ├── __init__.py  
│   ├── items.py  
│   ├── middlewares.py  
│   ├── pipelines.py  
│   ├── settings.py  
│   └── spiders  
│       ├── __init__.py  
└── scrapy.cfg
```

Example

- ▶ BeautifulSoup
 - ▶ Basic: webscrape_bs4.py
 - ▶ CSV Extract: bs4example.py
- ▶ Scrapy
 - ▶ Shell demo
 - ▶ extractQuotes.py
 - ▶ extractQuoteswTags.py