**Scrapy**

Scrapy is an open source collaborative framework for extracting the data from the websites what we need. Its performance is ridiculously fast and it is one of the most powerful libraries available out there. One of the key advantages of scrapy is that it is built on top of Twisted, an asynchronous networking framework, that means scrapy uses the non-blocking mechanism while sending the requests to the users.

*The asynchronous requests follows non-blocking I/O calls to the server. It is having much more advantages than synchronous requests.*

The key features of Scrapy are —

1. Scrapy has built-in support for extracting data from HTML sources using XPath expression and CSS expression.
2. It is a portable library i.e(written in Python and runs on Linux, Windows, Mac, and BSD)
3. It can be***Easily Extensible.***
4. It is faster than other existing scraping libraries. It can able to extract the websites with 20 times faster than other tools.
5. It consumes a lot less memory and CPU usage.
6. It can help us to build a Robust, and flexible application with a bunch of functions.
7. It has good community support for the developers but the documentation is not that much great for the beginners because it is not having a beginner friendly documentation.

**Beautiful Soup**

When it comes to Beautiful soup, it is really a beautiful tool for web scrappers because of its core features. It can help the programmer to quickly extract the data from a certain web page. This library will help us to pull the data out of HTML and XML files. But the problem with Beautiful Soup is it can’t able to do the entire job on its own. this library requires specific modules to work done.

The dependencies of the Beautiful soup are —

1. A library is needed to make a request to the website because it can’t able to make a request to a particular server. To overcome this issue It takes the help of the most popular library named ***Requests***or ***urlib2.***these libraries will help us to make our request to the server.
2. After downloading the HTML, XML data into our local Machine, Beautiful Soup require an External parser to parse the downloaded data. The most famous parsers are — lxml’s XML parser, lxml’s HTML parser, HTML5lib, html.parser.

The advantages of Beautiful soup are —

1. It is easy to learn and master. for example, if we want to extract all the links from the webpage. It can be simply done as follows —

**from** **bs4** **import** BeautifulSoup  
soup = BeautifulSoup(html\_doc, 'html.parser')  
  
**for** link **in** soup.find\_all('a'): # It helps to find all anchor tag's  
 **print**(link.get('href'))

In the above code, we are using the***html.parser*** to parse the content of the ***html\_doc.***this is one of the strongest reason for developers to use Beautiful soup as a web scraping tool.

2. It has good comprehensive documentation which helps us to learn the things quickly.

3. It has good community support to figure out the issues that arise while we are working with this library.

**Selenium**

Finally, when it comes to Selenium for web scraping! first of all, you should need to remember that Selenium is designed to automate test for Web Applications. It provides a way for the developer to write tests in a number of popular programming languages such as C#, Java, Python, Ruby, etc. This framework is developed to perform browser automation. Let’s have a look at the sample code that automates the browser.

**# Importing the required Modules.from** **selenium** **import** webdriver  
**from** **selenium.webdriver.common.keys** **import** Keys  
  
driver = webdriver.Chrome()  
driver.get("http://www.python.org")  
**assert** "Python" **in** driver.title  
elem = driver.find\_element\_by\_name("q")  
elem.send\_keys("selenium")  
elem.send\_keys(Keys.RETURN)  
**assert** "Google" **in** driver.title  
driver.close()

From the above code, we can conclude that API is very beginner-friendly, you can easily write code with Selenium. That is why it is so popular in the developer community. Even Selenium is mainly used to automate tests for web applications, it can also be used to develop web spider, many people have done this before.

The Key feature of Selenium is —

1. It can easily work with core Javascript concepts(DOM)
2. It can easily handle AJAX and PJAX requests.

**Choosing the Appropriate Library**

When it comes to the selection of a particular library to perform web scraping operation we need to consider various key factors because every library has it’s own pros and cons so In this selection criteria we will discuss the various factors that we need to consider while we are selecting a library for our project. The key factors that we must point out are —

**Extensibility**

**Scrapy:**The architecture of ***Scrapy*** is well designed to customize the middleware to add our own custom functionality. This feature helps us our project to be more Robust and flexible.

One of the biggest advantages of Scrapy is that we can able to migrate our existing project to another project very easily. So for the large/Complex projects, Scrapy is the best choice to work out.

If Your project needs proxies, data pipeline, then Scrapy would be the best choice.

**Beautiful Soup:**When it comes to a small project, Or low-level complex project Beautiful Soup can do the task pretty amazing. It helps us to maintain our code simple and flexible.

If you are a beginner and if you want to learn things quickly and want to perform web scraping operations then Beautiful Soup is the best choice.

**Selenium:** When you are dealing with Core Javascript featured website then Selenium would be the best choice. but the Data size should be limited.

**Performance**

**Scrapy:**It can do things quickly because of its built-in feature i.e usage of asynchronous system calls. The Existing libraries out there not able to beat the performance of Scrapy.

**Beautiful Soup:**Beautiful Soup is pretty slow to perform a certain task but we can overcome this issue with the help of Multithreading concept but However the programmer need to know the concept of multithreading very effectively. This is the downside of Beautiful Soup.

**Selenium:**It can handle up to some range butn’t equivalent to Scrapy.

**EcoSystem**

**Scrapy:**It has a good ecosystem, we can use proxies and VPN’s to automate the task. This is one of the reasons for choosing the library for complex projects. we can able to send multiple requests from the multiple proxy addresses.

**BeautifulSoup:** This library has a lot of dependencies in the ecosystem. This is one of the downsides of this library for a complex project

**Selenium:** It has a good ecosystem for the development but the problem is we can’t utilize the proxies very easily.

*From the above three common factors, you need to decide which one should be the right choice for your next project.*

**Conclusion**

I hope you got a clear understanding of Scrapy, Selenium, and Beautiful Soup. I discussed pretty much everything about the most popular web scraping libraries in a detailed manner. But the Selection of the library is really a big task. But I would suggest —

***Scrapy:*** *if you are dealing with complex Scraping operation that requires huge speed and with low power consumption*

**Beautiful soup**: If you’re new to programmer want to work with web scraping projects, this option is the easiest to learn.

**Selenium**: *When you want to deal with Core Javascript based web Applications and want to make browser automation with AJAX/PJAX Requests.*