Beautiful Soup
PIP
Requests

PIP

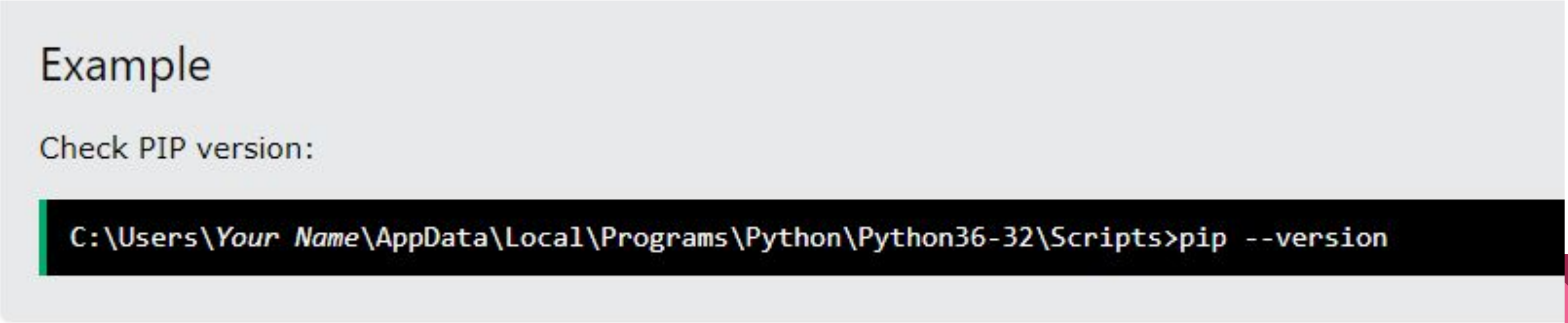
PIP is a package manager for Python packages, or modules.

What is a Package:

- A package contains all the files you need for a module.
- Modules are Python code libraries you can include in your project.

Navigate your command line to the location of Python's script directory, and type the following:

If you do not have PIP installed, you can download and install it from this page: pip · PyPI

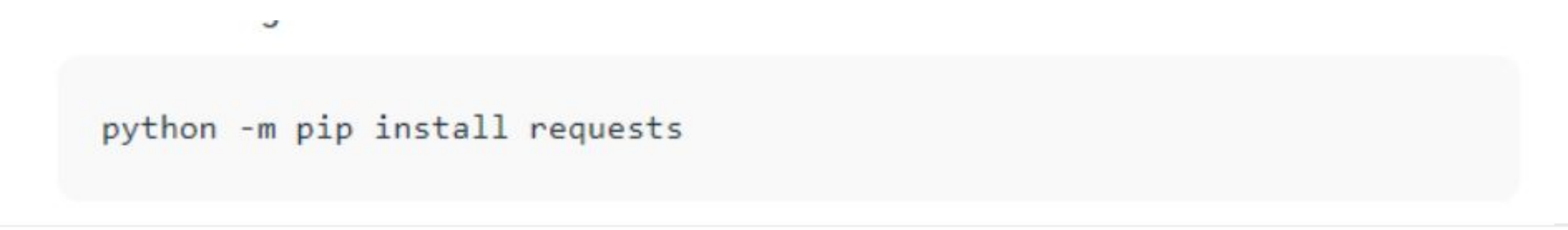


Example

Check PIP version:

```
C:\Users\Your Name\AppData\Local\Programs\Python\Python36-32\Scripts>pip --version
```

PIP install requests



```
python -m pip install requests
```

OR

pip install requests

Location to install modules→**C:\Users\User\AppData\Local\Programs\Python\Python310\Lib**

```
import requests

x = requests.get('https://www.google.com')

print(x.text)
```

The `requests` module allows you to send HTTP requests using Python.

GET() request

The `get()` method sends a GET request to the specified url.

```
import requests

#the required first parameter of the 'get' method is the 'url':

x = requests.get('https://www.google.com/')

#print the response text (the content of the requested file):

print(x.text)
```

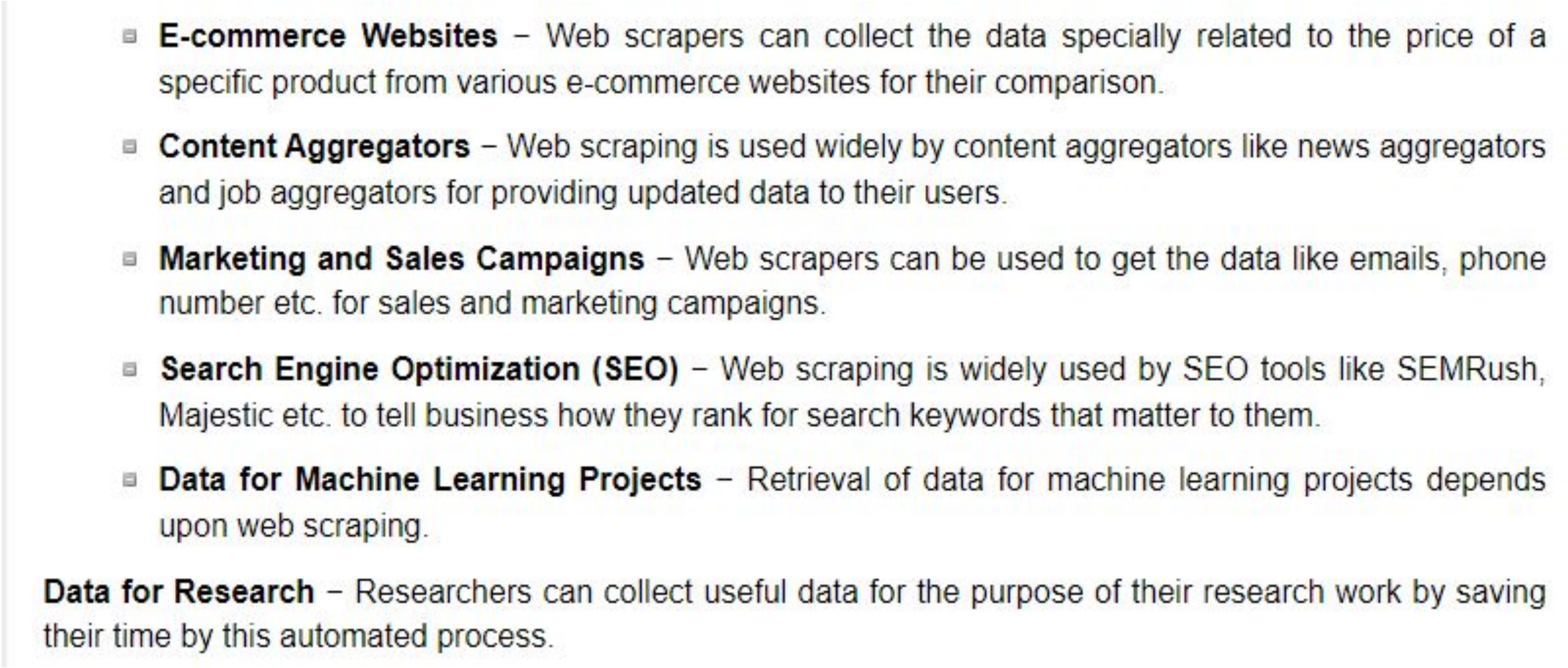
BeautifulSoup bs4 **module**

Data is absolutely necessary for any programmer and the basic requirement of every programming project is the large amount of useful data.

Web scraping, also called web data mining or web harvesting, is the process of constructing an agent which can extract, parse, download and organize useful information from the web automatically. In other words, we can say that instead of manually saving the data from websites, the web scraping software will automatically load and extract data from multiple websites as per our requirement

The information extracted using web scraping can be used to replicate in some other website or can be used to perform data analysis. For example the data elements can be names, address, price etc.

Uses

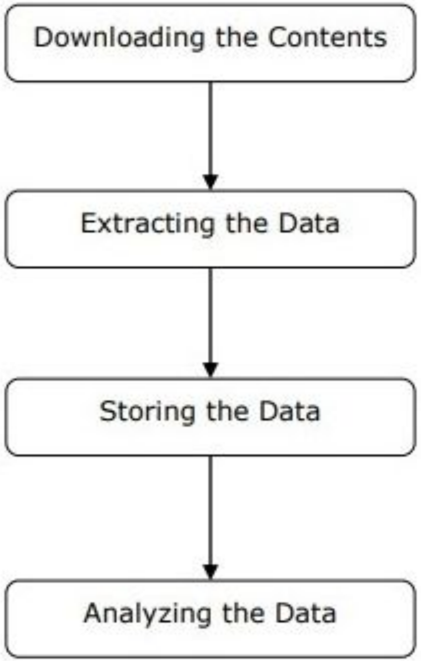


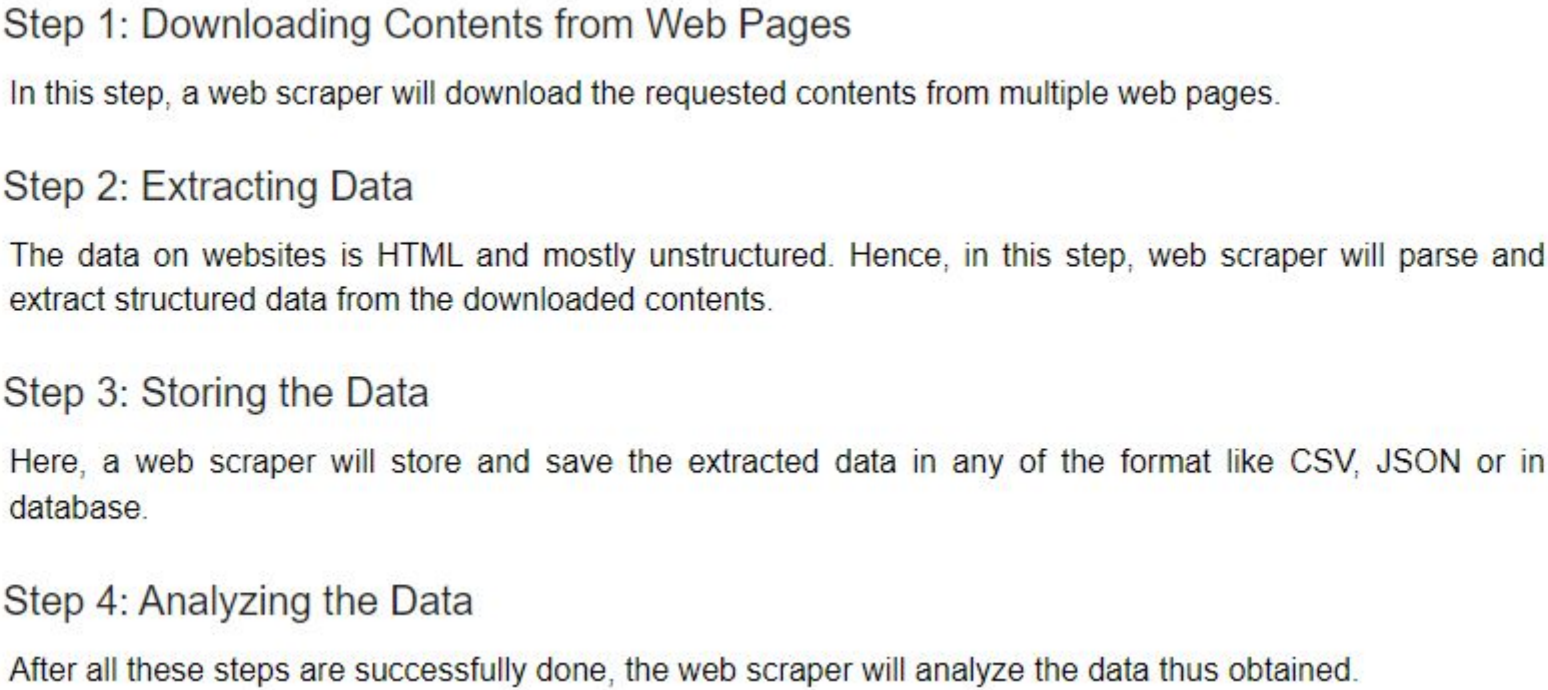
- **E-commerce Websites** – Web scrapers can collect the data specially related to the price of a specific product from various e-commerce websites for their comparison.
- **Content Aggregators** – Web scraping is used widely by content aggregators like news aggregators and job aggregators for providing updated data to their users.
- **Marketing and Sales Campaigns** – Web scrapers can be used to get the data like emails, phone number etc. for sales and marketing campaigns.
- **Search Engine Optimization (SEO)** – Web scraping is widely used by SEO tools like SEMRush, Majestic etc. to tell business how they rank for search keywords that matter to them.
- **Data for Machine Learning Projects** – Retrieval of data for machine learning projects depends upon web scraping.

Data for Research – Researchers can collect useful data for the purpose of their research work by saving their time by this automated process.

Working of a Web Scraper

Web scraper may be defined as a software or script used to download the contents of multiple web pages and extracting data from it.





Step 1: Downloading Contents from Web Pages

In this step, a web scraper will download the requested contents from multiple web pages.

Step 2: Extracting Data

The data on websites is HTML and mostly unstructured. Hence, in this step, web scraper will parse and extract structured data from the downloaded contents.

Step 3: Storing the Data

Here, a web scraper will store and save the extracted data in any of the format like CSV, JSON or in database.

Step 4: Analyzing the Data

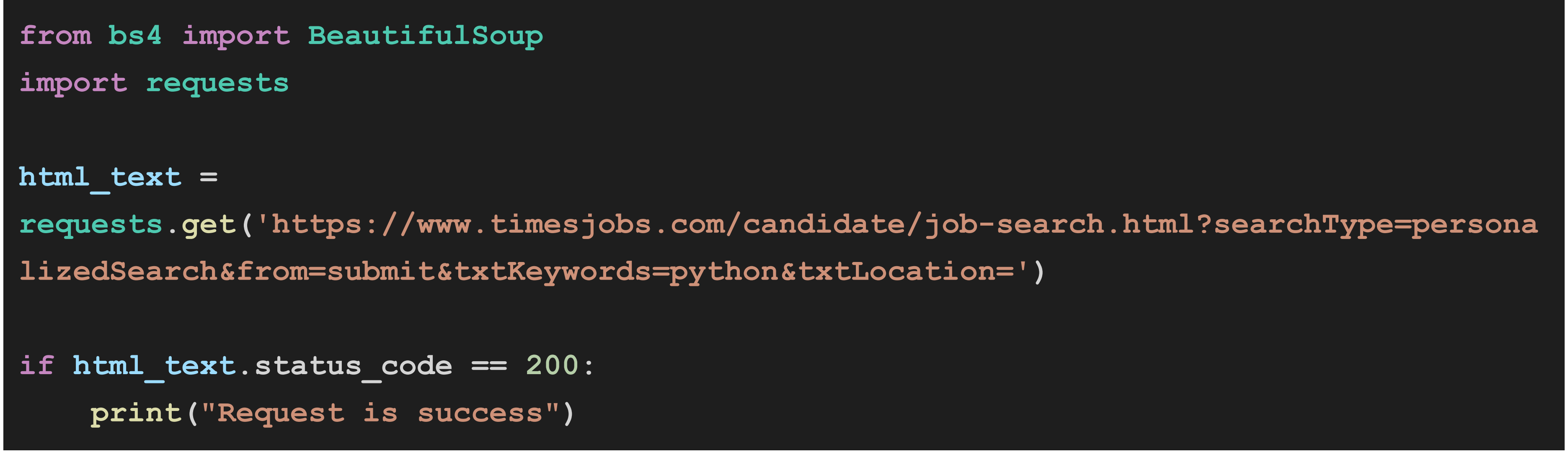
After all these steps are successfully done, the web scraper will analyze the data thus obtained.

Requests

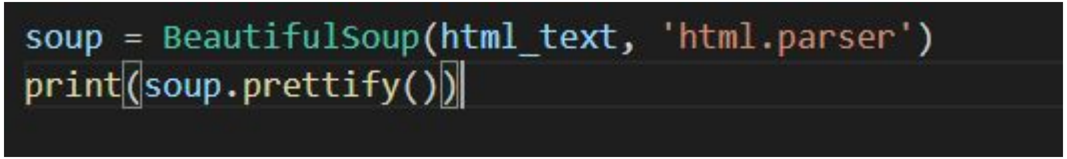
It is a simple python web scraping library. It is an efficient HTTP library used for accessing web pages. With the help of Requests, we can get the raw HTML of web pages which can then be parsed for retrieving the data

lxml module

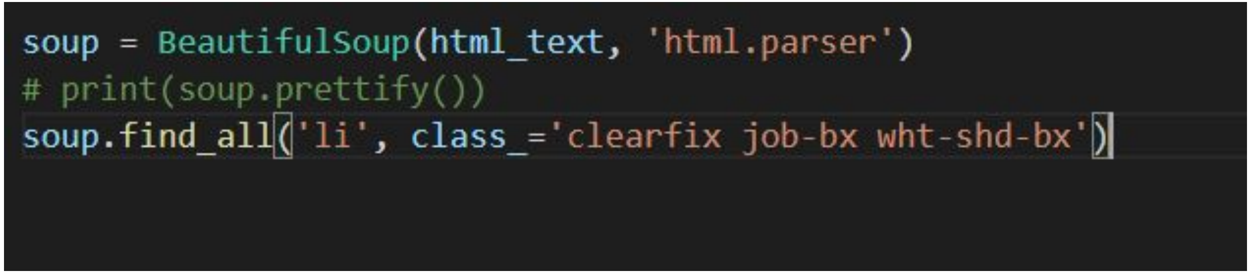
lxml is **a Python library which allows for easy handling of XML and HTML files, and can also be used for web scraping**.



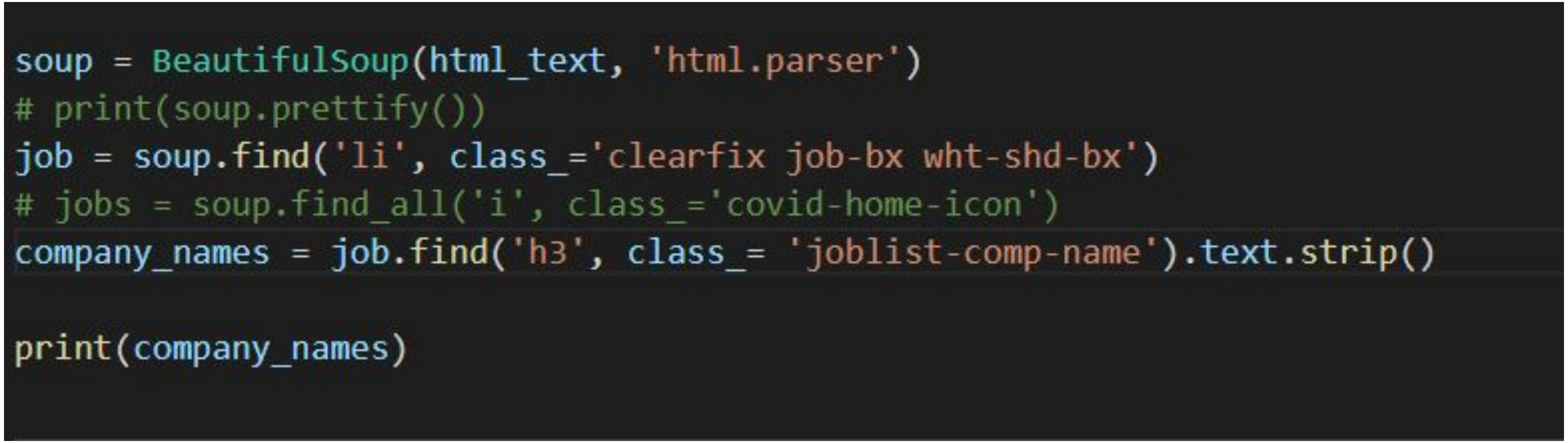
```
from bs4 import BeautifulSoup
import requests

html_text =
requests.get('https://www.timesjobs.com/candidate/job-search.html?searchType=persona
lizedSearch&from=submit&txtKeywords=python&txtLocation=')

if html_text.status_code == 200:
    print("Request is success")
```



```
soup = BeautifulSoup(html_text, 'html.parser')
print(soup.prettify())
```



```
soup = BeautifulSoup(html_text, 'html.parser')
# print(soup.prettify())
soup.find_all('li', class_='clearfix job-bx wht-shd-bx')
```



```
soup = BeautifulSoup(html_text, 'html.parser')
# print(soup.prettify())
job = soup.find('li', class_='clearfix job-bx wht-shd-bx')
# jobs = soup.find_all('i', class_='covid-home-icon')
company_names = job.find('h3', class_= 'joblist-comp-name').text.strip()

print(company_names)
```