

# Webscrapping

- Harshal Chaudhari



# What is scraping?

- Extract data from websites.
- Extract data from images/videos.
- Extract data from pdf files.

---

# What is scraping?

- Extract data from websites.

---

# Disclaimer

- Prohibited by Terms of Conditions of most websites.
- Can result into DoS attacks.
- Can affect quality of service provided by website. (e.g. Uber API)

Web administrators make it difficult to scrape websites, their intellectual property.

---

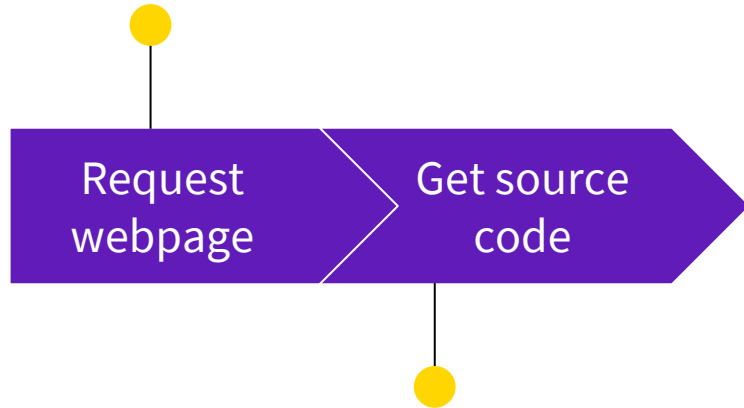
# Procedure



Your browser requests a  
webpage using HTTP(s)  
protocol.



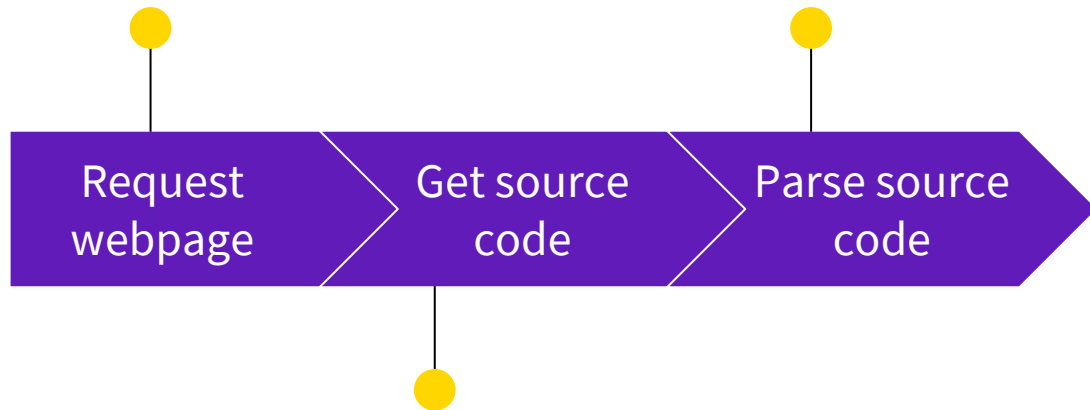
Your browser requests a webpage using HTTP(s) protocol.



Webpage is typically coded in HTML language and its variants.

Your browser requests a webpage using HTTP(s) protocol.

Convert source code into DOM (Document Object Model Structure) parse tree

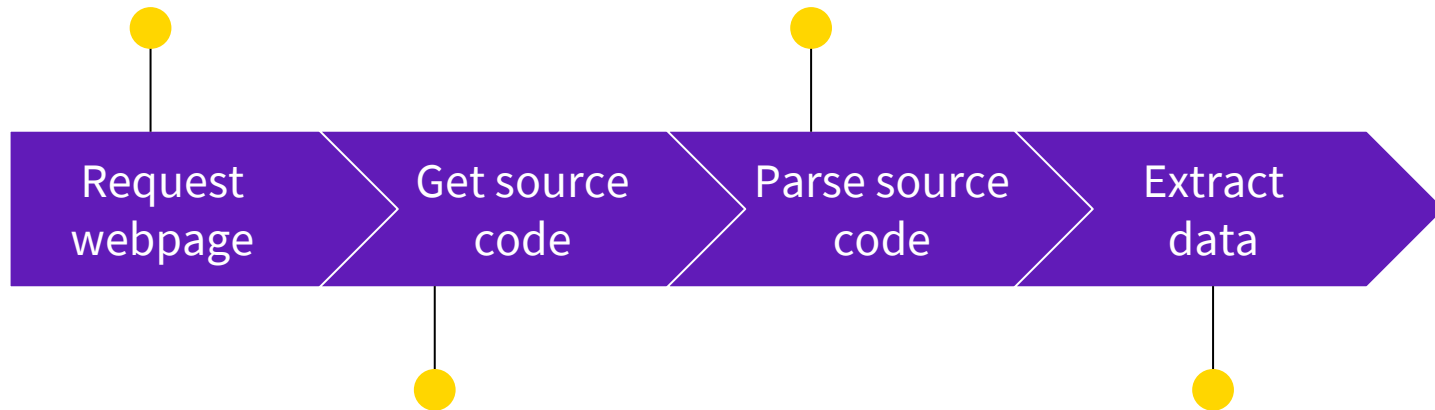


Webpage is typically coded in HTML language and its variants.



Your browser requests a webpage using HTTP(s) protocol.

Convert source code into DOM (Document Object Model Structure) parse tree



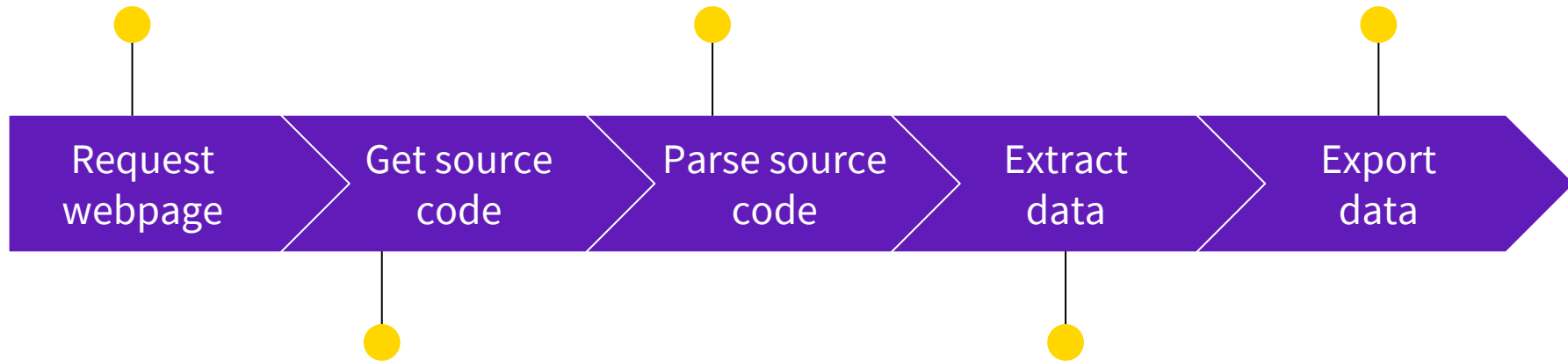
Webpage is typically coded in HTML language and its variants.

Navigate, search, modify parse tree to extract relevant data

Your browser requests a webpage using HTTP(s) protocol.

Convert source code into DOM (Document Object Model Structure) parse tree

Export data tables to databases



Webpage is typically coded in HTML language and its variants.

Navigate, search, modify parse tree to extract relevant data

# Requesting Webpage

Command line tools

- ‘Wget’ or ‘curl’

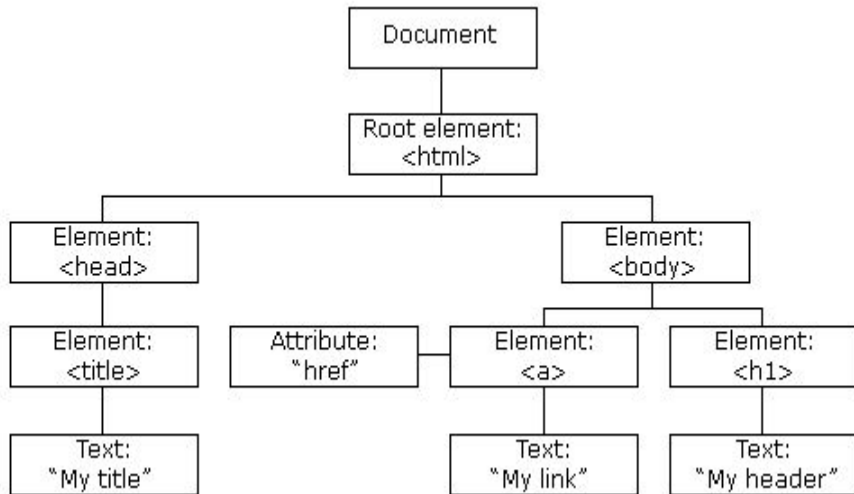
Network access libraries

- Urllib, Request, pyCurl

Use “Inspect elements” option in browser to observe the requests in browser, mimic them as closely as possible.

---

# Parsing Webpage



- Regular expressions
- Convert webpage source code into 'Document Object Model' (DOM).
- Traverse the DOM to extract relevant data.

# BeautifulSoup

```
html_doc = """
<html><head><title>The Dormouse's story</title></head>
<body>
<p class="title"><b>The Dormouse's story</b></p>

<p class="story">Once upon a time there were three little sisters;
and their names were
<a href="http://example.com/elsie" class="sister"
id="link1">Elsie</a>,
<a href="http://example.com/lacie" class="sister"
id="link2">Lacie</a> and
<a href="http://example.com/tillie" class="sister"
id="link3">Tillie</a>;
and they lived at the bottom of a well.</p>

<p class="story">...</p>
"""
```

## Documentation:

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html_doc, 'html.parser')
```

```
>> soup.title
```

```
# <title>The Dormouse's story</title>
```

```
>> soup.title.string
```

```
# u'The Dormouse's story'
```

```
>> soup.find_all('a')
```

```
# [<a class="sister" href="http://example.com/elsie"
id="link1">Elsie</a>,
# <a class="sister" href="http://example.com/lacie"
id="link2">Lacie</a>,
# <a class="sister" href="http://example.com/tillie"
id="link3">Tillie</a>]
```

# Storing & Exporting data

## Exporting data

- SQLite database
- CSV files
- JSON files

Preferable to store original webpage source code, especially for webpages that frequently change.

---

# Difficulties

- Websites identify automated access and block.
- HTML source codes change often, maintenance of 'parsers' is time-consuming activity.
- Modern websites rely on Javascript for dynamic loading.

# Tricks to prevent being blocked

- Random 'sleep' time between consecutive requests.
- Randomize 'User-Agent' for each request.
- Use HTTP / SOCKS proxy lists.
- Multi-processing with proxies.

---



# Sitemaps

- Major websites use ‘sitemap.xml’ to let search engines crawl their products / listings.
- Catalogue of listings.
- Robust to changes in source code.

Let's look at sitemap of Airbnb.

---

# AJAX Webpages

- Javascripts make requests to servers to fetch / display data.
- Source code obtained via code is not the same as the source code you see in browser.
- Example: Apple Jobs website  
<https://jobs.apple.com/us/search>

# AJAX Webpages

- Extended HTTP Requests (XHR).
- Automated Browser Instance.
- Use JavaScript to write scraper - Node.js library.

**Let's look at some  
scraping examples**

**Thank You!**