Web Scraping Know & Tell

Haley Johnson

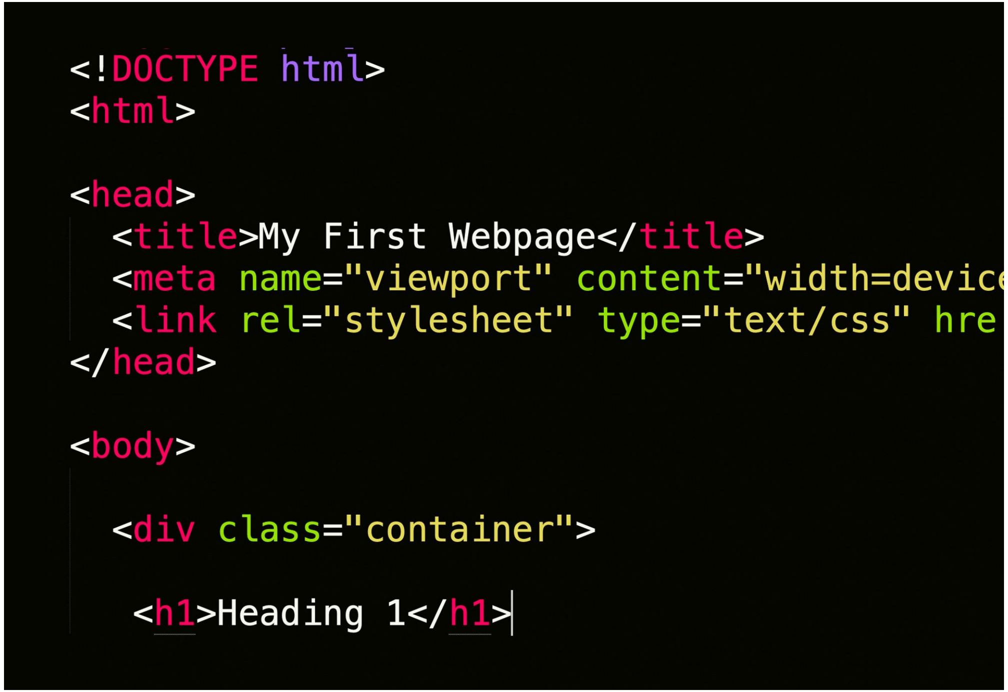
https://github.com/haleyejohnson/web_scraping_101

Web Scraping

Get information from live web pages

There are many methods - this presentation will focus on BeautifulSoup

- Uses structure of hypertext markup language (HTML)



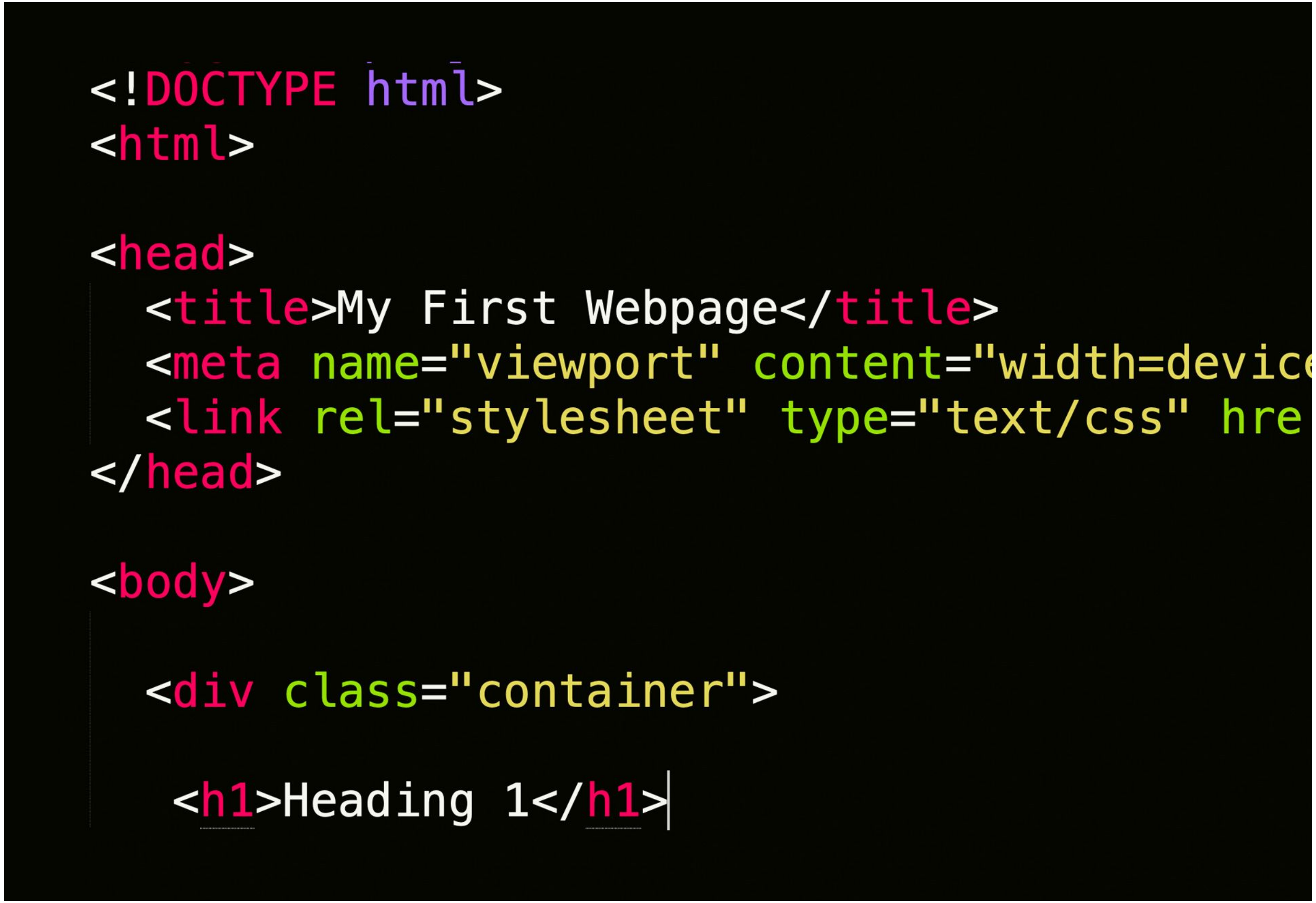
```
<!DOCTYPE html>
<html>

<head>
  <title>My First Webpage</title>
  <meta name="viewport" content="width=devic
  <link rel="stylesheet" type="text/css" hre
</head>

<body>

  <div class="container">

   <h1>Heading 1</h1>
```



```
<!DOCTYPE html>
<html>

<head>
  <title>My First Webpage</title>
  <meta name="viewport" content="width=device
  <link rel="stylesheet" type="text/css" hre
</head>

<body>

  <div class="container">

   <h1>Heading 1</h1>
```

Getting Data

To use the Beautiful Soup module for scraping, you need to create the Beautiful Soup object

1. Get the data from the url

```
r = requests.get(url)
```

2. Create a soup object using the data

```
soup = BeautifulSoup(r.text, ‘html.parser’)
```

Using Beautiful Soup

1. soup.find(‘tag’) will return the first tag that matches

2. soup.find_all(‘tag’) will return a list of all the tags that match

3. You can use find and find_all on the tag objects to find children tags!

4. Use the tag_object.attrs to obtain a dictionary of the attributes in a tag object

5. Use the tag_object.get(attr_name) or tag_object[attr_name] to get a specific attribute

What you see in the HTML	Tag description in code
<p>	soup.find_all('p')
<h3>	soup.find_all(‘h3’)
<div class=’comment’>	soup.find_all(‘div’, class_=’comment’)
<span style=’X5e72’>	soup.find_all(‘span’, style=’X5e72’)
<a class=’header-link’, href=’/nav’)	soup.find_all(‘a’, class_=’header-link’)

Ethics of Web Scraping

Some websites have anti-web scraping software embedded to keep you from grabbing their content

- AI crawlers have accelerated this trend and contributed to a more closed internet

Web scraping can have copyright implications