Data 2.0 Hackathon
Web Scraping Workshop

Sophie Janaskie
Yale School of Forestry and Environmental Studies

March 17, 2018

What is web scraping?

- A method of extracting information from websites programmatically
- A way to convert semi-structured web data into usable, structured data that we can analyze

Why is it useful, and when to use it?

- No API or readily available data
- Programmatically pull data from the web to save time from manual collection
- Useful for projects and capstones!

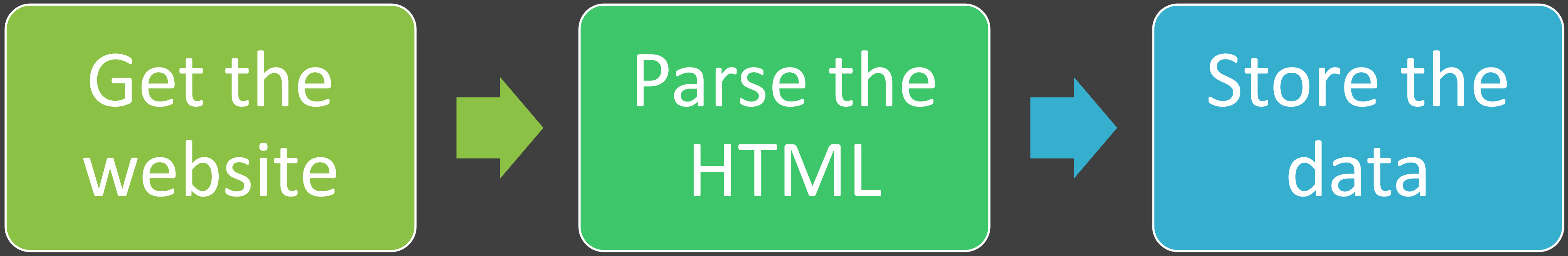
“The Internet is one giant API – with a really terrible interface.”

- Ryan Mitchell, author of Web Scraping with Python

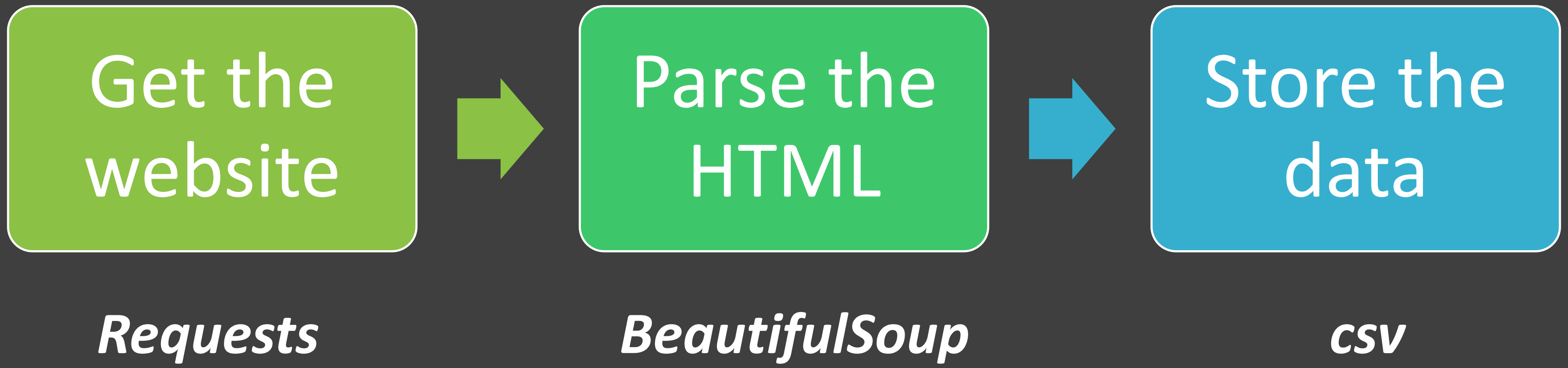
Agenda

1. Scraping workflow
2. HTML Basics
3. Pulling the web page with Requests
4. Parsing the HTML with BeautifulSoup
5. Storing your data
6. Other considerations

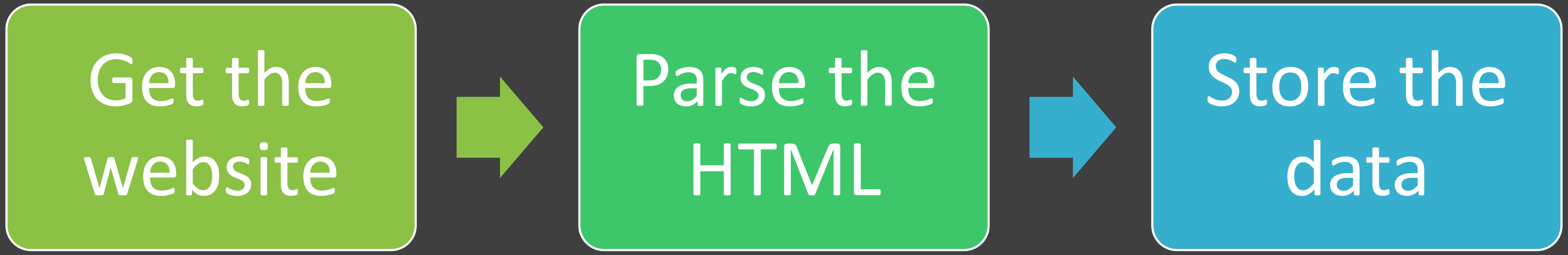
Web Scraping Workflow



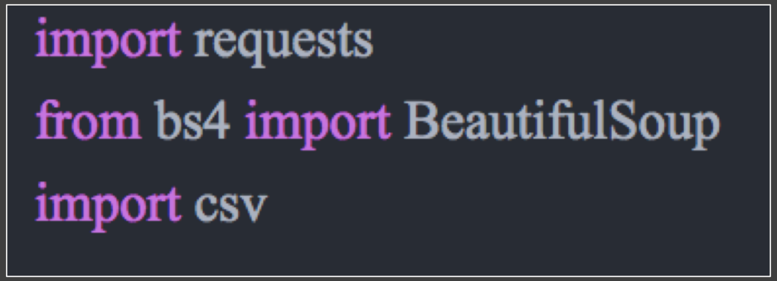
Web Scraping Workflow



Web Scraping Workflow



Requests *BeautifulSoup* *csv*

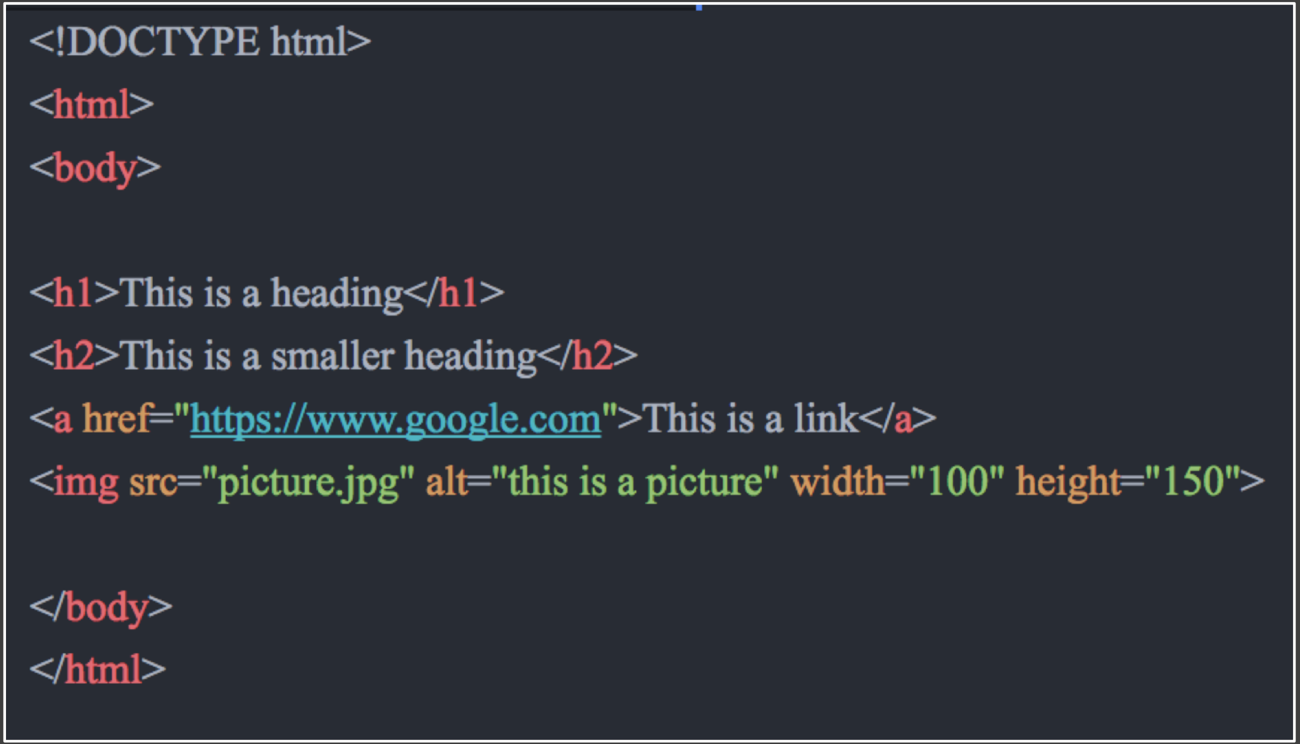


```
import requests
from bs4 import BeautifulSoup
import csv
```

HTML Basics

HTML

Hyper Text Markup Language provides structure for web pages.



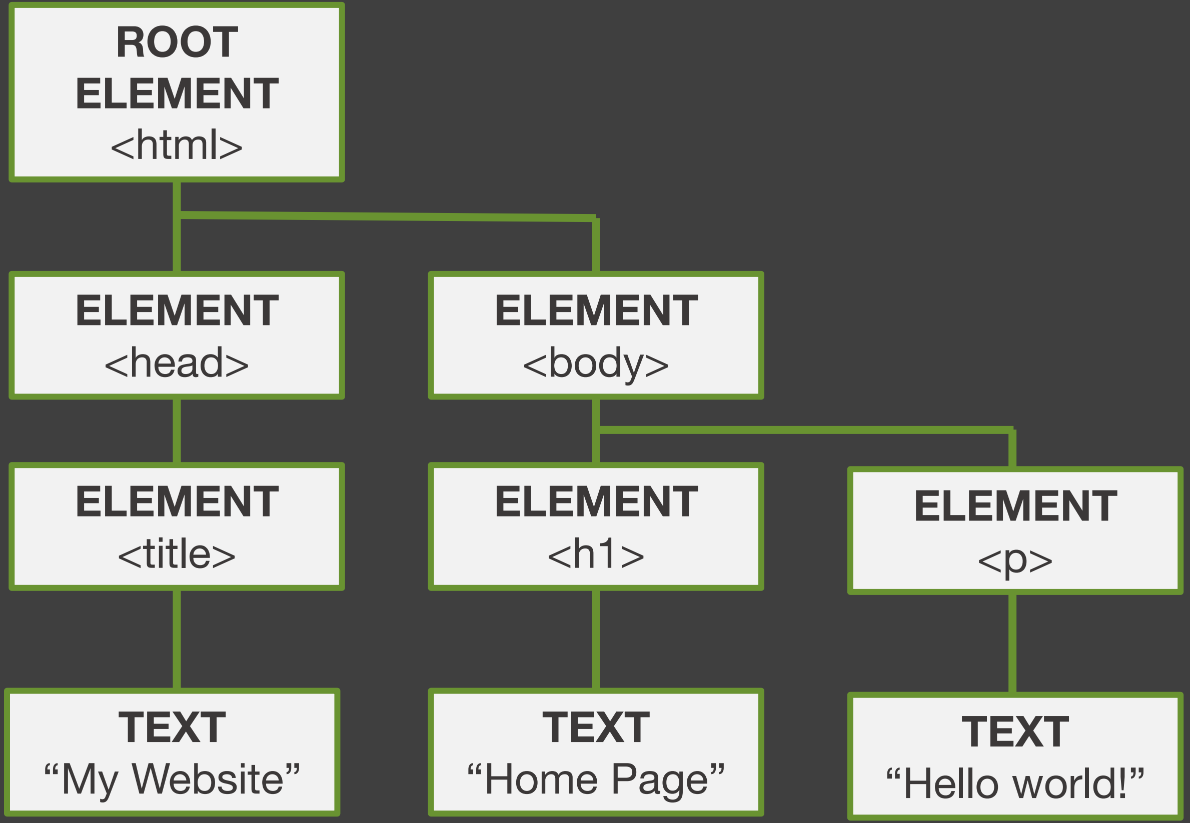
```
<!DOCTYPE html>
<html>
<body>

<h1>This is a heading</h1>
<h2>This is a smaller heading</h2>
<a href="https://www.google.com">This is a link</a>
<img src="picture.jpg" alt="this is a picture" width="100" height="150">

</body>
</html>
```

Document Object Model

The DOM creates an object-oriented model of the information to enable easier manipulation.

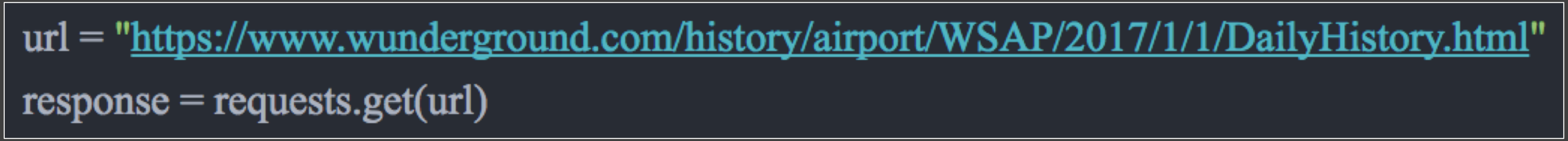


Get the website

Using the Requests library

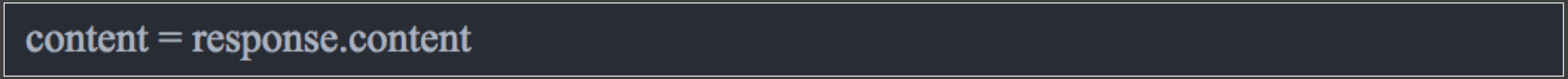
Requests library

- An easy way to send an HTTP request



```
url = "https://www.wunderground.com/history/airport/WSAP/2017/1/1/DailyHistory.html"
response = requests.get(url)
```

- Access the response’s body of content



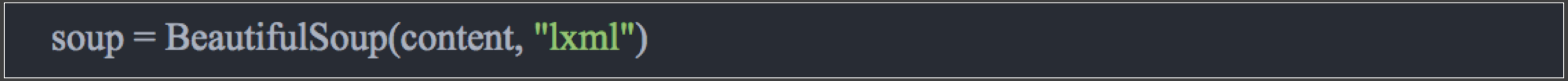
```
content = response.content
```

Parse the HTML

Using the BeautifulSoup library

BeautifulSoup

Make the Soup



```
soup = BeautifulSoup(content, "lxml")
```

Navigate the Tree

- Using .title, .contents, .children, .parent, .next_sibling, .text etc.

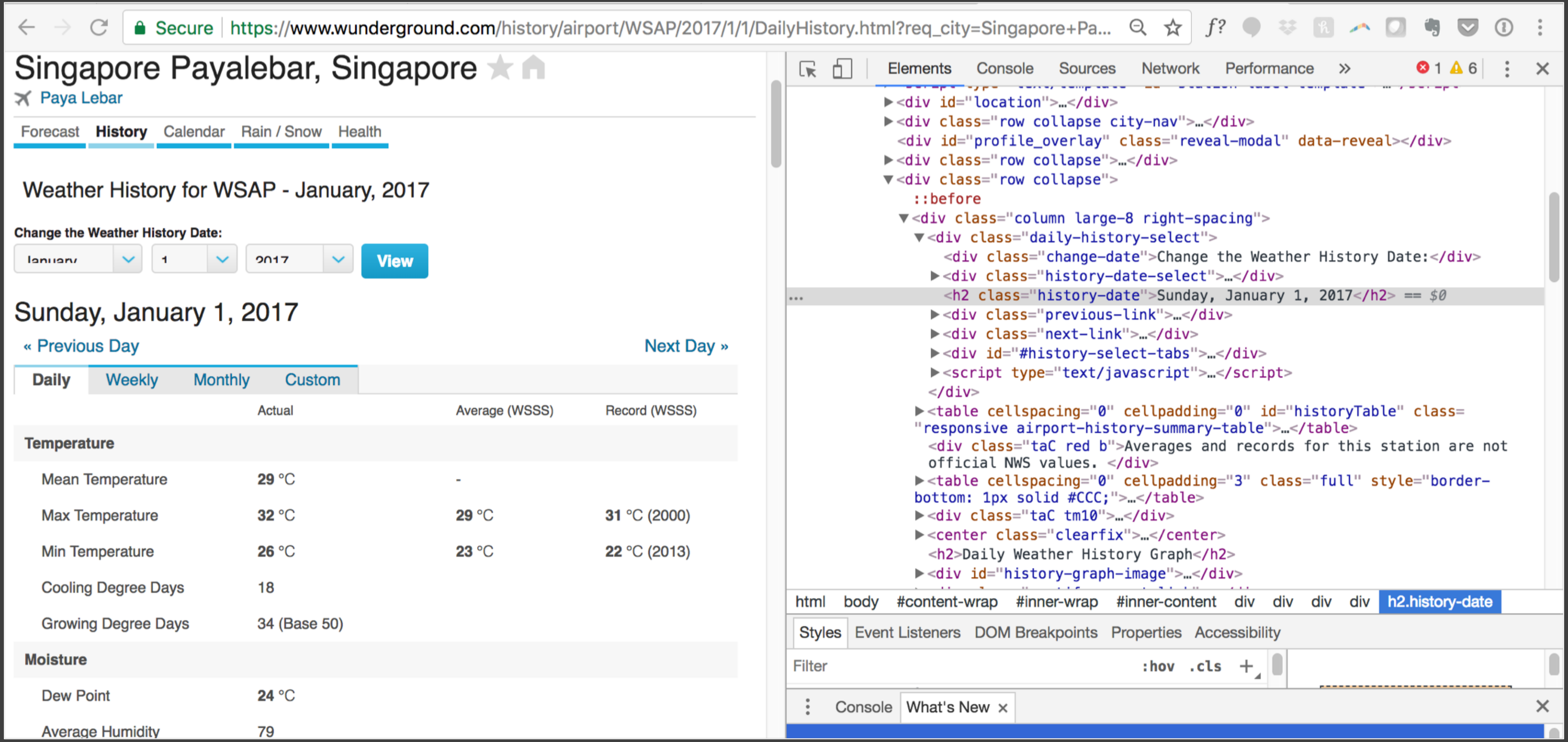
Search the Tree

- Using find*(tag),* find_all*(tag),* etc.

Modify the Tree

- Using decompose( ), insert( ), NavigableString( ), etc.

Using Inspect Element



Secure https://www.wunderground.com/history/airport/WSAP/2017/1/1/DailyHistory.html?req_city=Singapore+Pa...

Singapore Payalebar, Singapore

Paya Lebar

Forecast History Calendar Rain / Snow Health

Weather History for WSAP - January, 2017

Change the Weather History Date:

January | 1 | 2017 | View

Sunday, January 1, 2017

« Previous Day Next Day »

Daily Weekly Monthly Custom

	Actual	Average (WSSS)	Record (WSSS)
Temperature			
Mean Temperature	**29** °C	-	
Max Temperature	**32** °C	**29** °C	**31** °C (2000)
Min Temperature	**26** °C	**23** °C	**22** °C (2013)
Cooling Degree Days	18		
Growing Degree Days	34 (Base 50)		
Moisture			
Dew Point	**24** °C		
Average Humidity	79		

Elements Console Sources Network Performance » ✖ 1 ⚠ 6

```
▶<div id="location">…</div>
▶<div class="row collapse city-nav">…</div>
 <div id="profile_overlay" class="reveal-modal" data-reveal></div>
▶<div class="row collapse">…</div>
▼<div class="row collapse">
   ::before
  ▼<div class="column large-8 right-spacing">
    ▼<div class="daily-history-select">
       <div class="change-date">Change the Weather History Date:</div>
      ▶<div class="history-date-select">…</div>
...    <h2 class="history-date">Sunday, January 1, 2017</h2> == $0
      ▶<div class="previous-link">…</div>
      ▶<div class="next-link">…</div>
      ▶<div id="#history-select-tabs">…</div>
      ▶<script type="text/javascript">…</script>
     </div>
    ▶<table cellspacing="0" cellpadding="0" id="historyTable" class=
    "responsive airport-history-summary-table">…</table>
     <div class="taC red b">Averages and records for this station are not
     official NWS values. </div>
    ▶<table cellspacing="0" cellpadding="3" class="full" style="border-
    bottom: 1px solid #CCC;">…</table>
    ▶<div class="taC tm10">…</div>
    ▶<center class="clearfix">…</center>
     <h2>Daily Weather History Graph</h2>
    ▶<div id="history-graph-image">…</div>
```

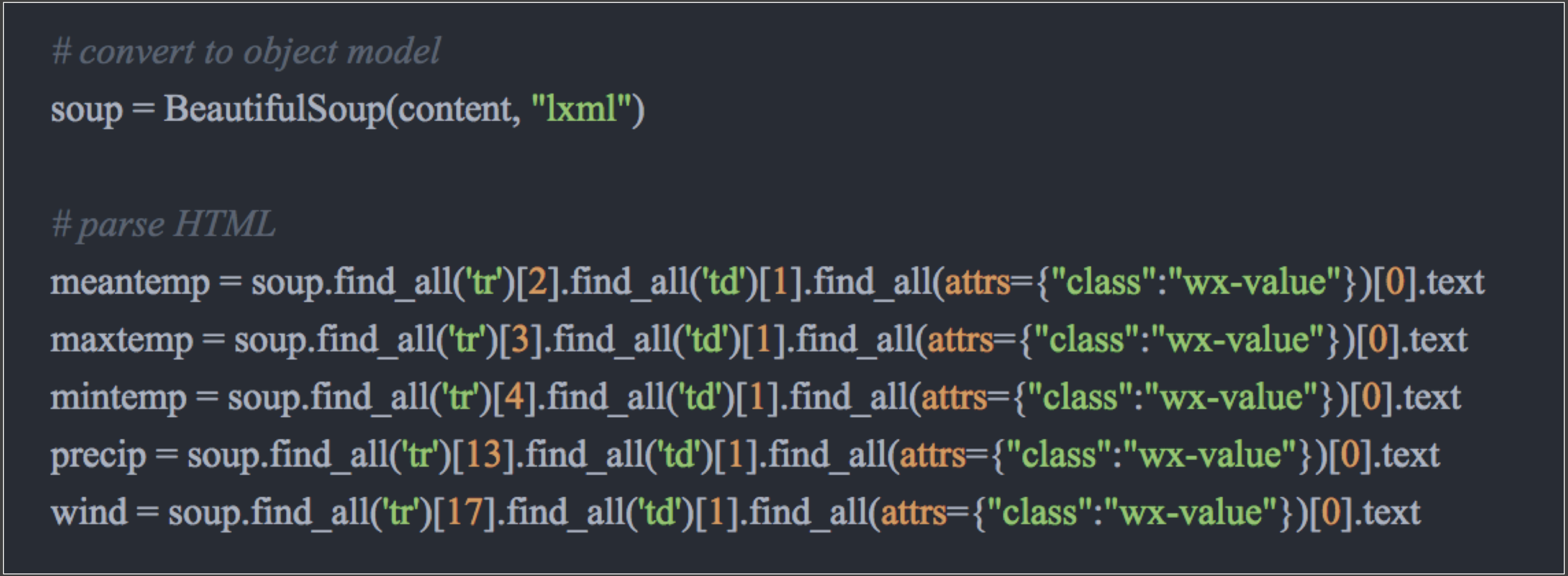
html body #content-wrap #inner-wrap #inner-content div div div div h2.history-date

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

Console What's New ×

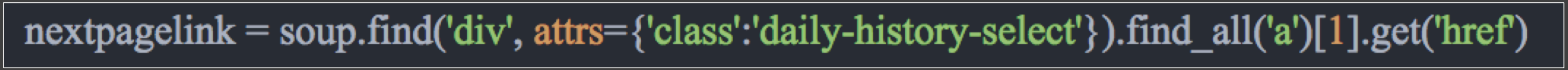
Example 1: Wunderground



```
# convert to object model
soup = BeautifulSoup(content, "lxml")

# parse HTML
meantemp = soup.find_all('tr')[2].find_all('td')[1].find_all(attrs={"class":"wx-value"})[0].text
maxtemp = soup.find_all('tr')[3].find_all('td')[1].find_all(attrs={"class":"wx-value"})[0].text
mintemp = soup.find_all('tr')[4].find_all('td')[1].find_all(attrs={"class":"wx-value"})[0].text
precip = soup.find_all('tr')[13].find_all('td')[1].find_all(attrs={"class":"wx-value"})[0].text
wind = soup.find_all('tr')[17].find_all('td')[1].find_all(attrs={"class":"wx-value"})[0].text
```

Example 2: Wunderground

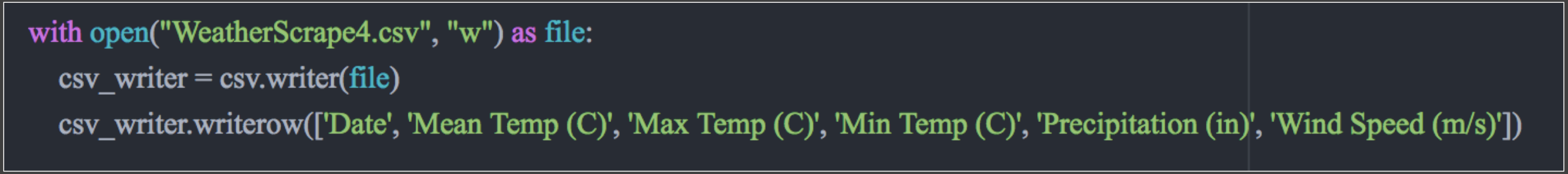


```
nextpagelink = soup.find('div', attrs={'class':'daily-history-select'}).find_all('a')[1].get('href')
```

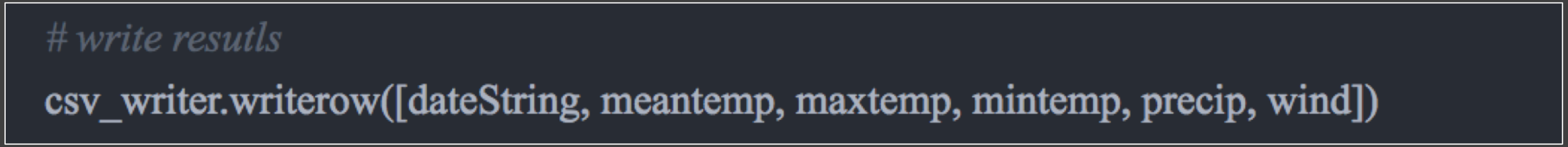
Store the results

Using the csv library

Example: Wunderground



```
with open("WeatherScrape4.csv", "w") as file:
    csv_writer = csv.writer(file)
    csv_writer.writerow(['Date', 'Mean Temp (C)', 'Max Temp (C)', 'Min Temp (C)', 'Precipitation (in)', 'Wind Speed (m/s)'])
```

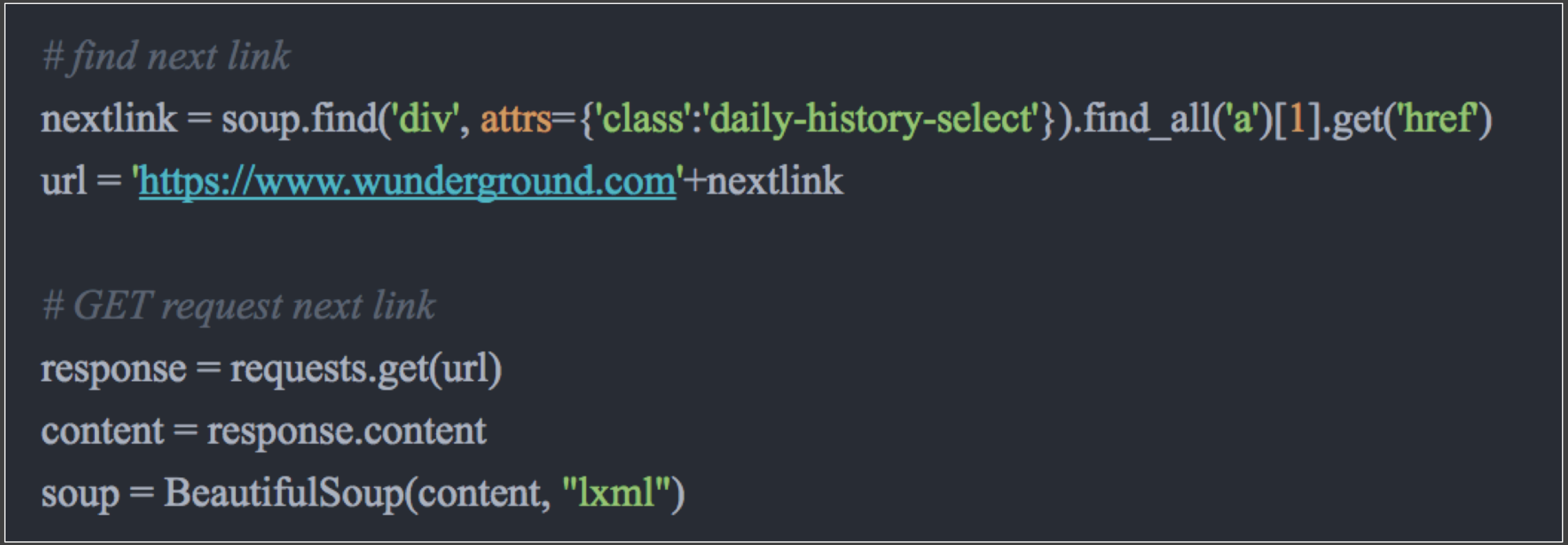


```
# write resutls
csv_writer.writerow([dateString, meantemp, maxtemp, mintemp, precip, wind])
```

Scaling up

Traversing links

- Handling URL generation / pagination



```
# find next link
nextlink = soup.find('div', attrs={'class':'daily-history-select'}).find_all('a')[1].get('href')
url = 'https://www.wunderground.com'+nextlink

# GET request next link
response = requests.get(url)
content = response.content
soup = BeautifulSoup(content, "lxml")
```

Handling exceptions

- Try/except and if statements
- Filling in N/A’s
- Consider special cases

Data Cleaning

- Unnecessary characters may be present in your scraped data
- Can use the .replace( ) function to remove these

Review

Thank you!

Resources

- Requests documentation
- BeautifulSoup documentation
- csv documentation