Case study: Weather!

Downloading weather data

We now know enough to proceed with extracting information about the local weather from the National Weather Service website!

The local weather of Boulder, CO is: https://forecast.weather.gov/MapClick.php?lat=40.0466&lon=-105.2523#.YwpRBy2B1f0

Time to Start Scraping!

We now know enough to download the page and start parsing it. In the below code, we will:

- Download the web page containing the forecast.
- Create a BeautifulSoup class to parse the page.
- Find the div with id seven-day-forecast, and assign to seven_day
- Inside seven_day, find each individual forecast item. Extract and print the first forecast item.

```
import requests
from bs4 import BeautifulSoup

page = requests.get("https://forecast.weather.gov/MapClick.php?lat=40.0466&lon=-10
soup = BeautifulSoup(page.content, 'html.parser')
seven_day = soup.find(id="seven-day-forecast")
forecast_items = seven_day.find_all(class_="tombstone-container")
print(forecast_items)
```

```
[<div class="tombstone-container">
<p class="period-name">Today<br/><br/></p>
<p><img alt="Today: Sunny, with a high near 88. Northwest wind 9 to 13 mph, w
<p class="period-name">Tonight<br/><br/></p>
<p><img alt="Tonight: Mostly clear, with a low around 59. Northeast wind 8 to
<p class="period-name">Sunday<br/><br/></p>
<p><img alt="Sunday: A 20 percent chance of showers and thunderstorms after 1
<p class="period-name">Sunday<br/>Night</p>
<p><img alt="Sunday Night: A 20 percent chance of showers and thunderstorms b
<p class="period-name">Monday<br/><br/></p>
<p><img alt="Monday: Sunny, with a high near 87. Light and variable wind beco
<p class="period-name">Monday<br/>Night</p>
<p><img alt="Monday Night: Mostly clear, with a low around 58." class="foreca
<p class="period-name">Tuesday<br/><br/></p>
<p><img alt="Tuesday: A 10 percent chance of showers and thunderstorms after
<p class="period-name">Tuesday<br/>Night</p>
<p><img alt="Tuesday Night: A slight chance of showers and thunderstorms befo
<p class="period-name">Wednesday<br/><br/></p>
<p><img alt="Wednesday: A slight chance of showers and thunderstorms after 1p
```

```
tonight = forecast_items[0]
print(tonight.prettify())
```

```
<div class="tombstone-container">
 <p class="period-name">
  Today
  <br/>
  <br/>
 </p>
 <p>
  <img alt="Today: Sunny, with a high near 88. Northwest wind 9 to 13 mph, wi
 </p>
 <p class="short-desc">
  Sunny
 </p>
 <p class="temp temp-high">
  High: 88 °F
 </p>
</div>
```

Extracting information of tonight

As we can see, inside the forecast item tonight is all the information we want. There are four pieces of information we can extract:

- The name of the forecast item — in this case, Tonight.
- The description of the conditions — this is stored in the title property of img.
- A short description of the conditions — in this case, Sunny and hot.
- The temperature hight — in this case, 98 degrees.

We'll extract the name of the forecast item, the short description, and the temperature first, since they're all similar:

```
period = tonight.find(class_="period-name").get_text()
short_desc = tonight.find(class_="short-desc").get_text()
temp = tonight.find(class_="temp").get_text()
print(period)
print(short_desc)
print(temp)
```

```
Today
Sunny
High: 88 °F
```

Now, we can extract the title attribute from the img tag. To do this, we just treat the BeautifulSoup object like a dictionary, and pass in the attribute we want as a key:

```
img = tonight.find("img")
desc = img['title']
print(desc)
```

```
Today: Sunny, with a high near 88. Northwest wind 9 to 13 mph, with gusts as
```

Extract all nights!

Now that we know how to extract each individual piece of information, we can combine our knowledge with CSS selectors and list comprehensions to extract everything at once.

In the below code, we will:

Select all items with the class period-name inside an item with the class tombstone-container in seven_day. Use a list comprehension to call the get_text method on each BeautifulSoup object.

```
period_tags = seven_day.select(".tombstone-container .period-name")
periods = [pt.get_text() for pt in period_tags]
periods
```

```
['Today',
 'Tonight',
 'Sunday',
 'SundayNight',
 'Monday',
 'MondayNight',
 'Tuesday',
 'TuesdayNight',
 'Wednesday']
```

As we can see above, our technique gets us each of the period names, in order.

We can apply the same technique to get the other three fields:

```
short_descs = [sd.get_text() for sd in seven_day.select(".tombstone-container .sho
temps = [t.get_text() for t in seven_day.select(".tombstone-container .temp")]
descs = [d["title"] for d in seven_day.select(".tombstone-container img")]

print(short_descs)
print(temps)
print(descs)
```

```
Sunny', 'Mostly Clear', 'Sunny thenSlight ChanceT-storms', 'Slight ChanceT-sto
High: 88 °F', 'Low: 59 °F', 'High: 88 °F', 'Low: 57 °F', 'High: 87 °F', 'Low:
Today: Sunny, with a high near 88. Northwest wind 9 to 13 mph, with gusts as h
```

Deal with data

We can now combine the data into a Pandas DataFrame and analyze it. A DataFrame is an object that can store tabular data, making data analysis easy. If you want to learn more about Pandas, check out our free to start course here.

In order to do this, we'll call the DataFrame class, and pass in each list of items that we have. We pass them in as part of a dictionary.

Each dictionary key will become a column in the DataFrame, and each list will become the values in the column:

```
import pandas as pd
weather = pd.DataFrame({
    "period": periods,
    "short_desc": short_descs,
    "temp": temps,
    "desc":descs
})
weather
```

	period	short_desc	temp	desc
0	Today	Sunny	High: 88 °F	Today: Sunny, with a high near 88. Northwest w...
1	Tonight	Mostly Clear	Low: 59 °F	Tonight: Mostly clear, with a low around 59. N...
2	Sunday	Sunny thenSlight ChanceT-storms	High: 88 °F	Sunday: A 20 percent chance of showers and thu...
3	SundayNight	Slight ChanceT-storms thenMostly Clear	Low: 57 °F	Sunday Night: A 20 percent chance of showers a...
4	Monday	Sunny	High: 87 °F	Monday: Sunny, with a high near 87. Light and ...
5	MondayNight	Mostly Clear	Low: 58 °F	Monday Night: Mostly clear, with a low around 58.
6	Tuesday	Sunny thenSlight ChanceT-storms	High: 89 °F	Tuesday: A 10 percent chance of showers and th...
7	TuesdayNight	Slight ChanceT-storms	Low: 60 °F	Tuesday Night: A slight chance of showers and ...
8	Wednesday	Sunny thenSlight ChanceT-storms	High: 86 °F	Wednesday: A slight chance of showers and thun...

Now let's save it to CSV.

```
weather.to_csv('data/Boulder_Weather_7_Days.csv')
```

Your Task

Now use your location, and repeat the process!