2025

4Geeks Academy: data science cohort 12

DAY 7: WEB SCRAPING WITH PYTHON

TODO

WEB SCRAPING	Use case, tools and strategies.
SQL PROJECT	Submit SQL project: Global Life Institute Data Detectives (Intro to SQL module) if you haven't done so already.
WEB SCRAPING PROJECT	Work on Web Scraping with Python (web scraping module) plan to finish before class on Friday.

TOPICS

01 WEB SCRAPING

02 GETTING HTML

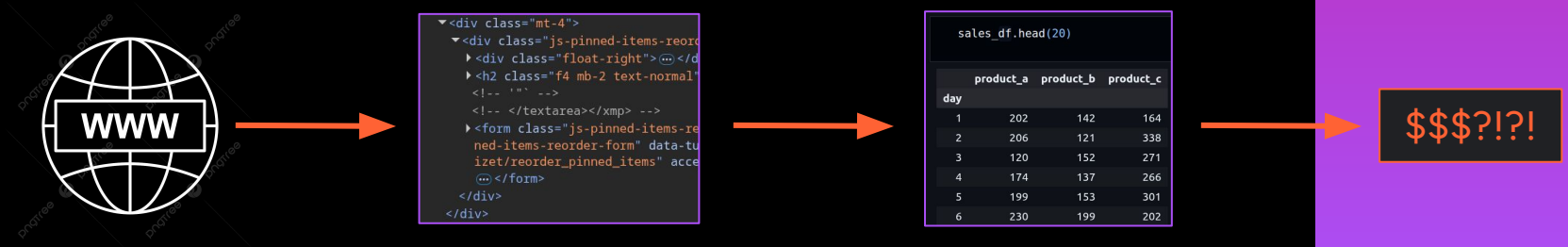
03 PARSING HTML

WEB SCRAPING

WHAT Downloading web pages as HTML to extract data.

WHY Allows access to (theoretically) any data on any website, even if it is not available through an API or downloadable file.

HOW Use Python to access a URL, recover the page HTML and parse it to extract target data.



GETTING HTML

TOOLS

- `wget`: non-interactive GNU file download utility
- **Requests**: simple HTTP library for Python
- **Selenium**: web browser automation framework

GOTCHAS

JavaScript

- Dynamically rendered content not present in GET response
- Compare `wget` to inspect element, look for `<script>` tags
- Use Selenium instead of requests

Bot detection/CAPTCHA/login etc.

- Include web-browser-like header in request
- Use Selenium
- Don't be a jerk

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

<h2>A basic HTML table</h2>

<table style="width:100%">
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>

<p>Some text about the table.</p>

</body>
</html>
```

PARSING HTML

TOOLS

- **`pandas.read_html()`**: can read HTML (tables only) directly into DataFrame
- **beautifulsoup4**: Python library for parsing HTML

WORKFLOW

- Look at target HTML with inspect element
- Find target data in HTML
- Note unique tags/classes/ids in target HTML
- Use Python loops and logic as necessary

```python
soup = BeautifulSoup(html_string, 'html.parser')
table = soup.find('table')
rows = table.find_all('tr')

for row in rows:
    cells = row.find_all(['th', 'td'])

    for cell in cells:
        value = cell.get_text(strip=True)
```

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

<h2>A basic HTML table</h2>

<table style="width:100%">
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>

<p>Some text about the table.</p>

</body>
</html>
```