- **Selenium**
    - Pros:
        - Can mimic a browser and allows for automated user-like actions (ex. scrolling, clicking)
        - Works for dynamically loaded websites
            - i.e. websites that require you to scroll down to load more content
        - Relatively low learning curve
    - Cons:
        - Not built for grabbing data from PDFs
        - Slower performance
        - Resource intensive

- **MechanicalSoup**
    - Pros:
        - Works for websites that do not provide APIs
        - Lighter-weight than Selenium
    - Cons:
        - Does not support asynchronous loading
        - Does not support JavaScript

Selenium looks to be the best tool available given that it can mimic user interactions and work on dynamic websites. While MechanicalSoup may be lighter-weight, it cannot handle asynchronous requests nor can it simulate user interactions like pressing buttons or scrolling. Selenium does not handle PDFs however. Below are a couple of cursory PDF parsers I found:

- PDFminer
    - Python package for extracting text from PDFs
    - Simple to install and use
    - Doesn't seem to be maintained recently (latest release was in 2019)

- PyPDF
    - Can extract text and metadata
    - Recently maintained (latest release was 2024)
    - Can attempt to keep the layout of the original PDF

- Tabula-py
    - Useful for extracting tables from a PDF
    - Not designed to extract pure text data
    - Decently maintained (latest release was November 2023)