

Web Scrapping with Python

Garrett Dancik, PhD

Admitted Student Mock Class

Eastern Connecticut State University

Department of Computer Science

May 10, 2020

dancikg@easternct.edu

<http://gdancik.github.io/ASDD/>

Web development overview

- **Hypertext Markup Language (HTML)** describes the structure of a web page
 - HTML pages are composed of elements that are specified using tags
 - `<p> This is a paragraph </p>`
 - `<h1> This is a header </h1>`
- **Cascading style sheets (CSS)** describe how HTML
 - `<p style = "background-color:yellow"> This is a paragraph with yellow background </p>`
- Recommended tutorials at
 - <https://www.w3schools.com/html/default.asp>
 - <https://www.w3schools.com/css/default.asp>

Examples

- See example web page at:

<https://gdancik.github.io/CSC-301/data/notes/schedule.html>

- Let's use the Web Inspector to look at the structure of this page

Web Scraping Overview

- Web scraping is the process of retrieving web pages and extracting relevant data from them
- Why?
 - Search engines collect data to index web pages
 - Collecting weather and climate data for research
 - <https://www.sciencedirect.com/science/article/pii/S0168169909002348>
 - For businesses and consumers to keep track of products
 - For analyzing trends in job postings
 - <https://www.sciencedirect.com/science/article/abs/pii/S030643791630477X>
 - For research in economics
 - <https://www.aeaweb.org/articles?id=10.1257/jep.30.2.151>

Steps for web scraping

- Identify a page you want to scrape
- Check robots.txt for restrictions, e.g.
 - <https://www.travelocity.com/robots.txt>
- Understand the structure of the page (e.g., by using the Web Inspector)
- Write a script that
 - Retrieves the web page
 - From a URL using the Python requests library
 - From a file using *open*
 - Extracts information from the web page
 - Using Python's BeautifulSoup library

Thank You!

- Any questions?
- Contact:
 - dancikg@easternct.edu
 - <http://gdancik.github.io/>