Introduction to Web Scraping

By Eric Schles

Background knowledge

Data Structures - understanding the structure of an html page / understanding the "document" data structure generally.

Data Structures of the web:

html, xml, json, packets, headers

Background knowledge

Networking - understanding how pages are requested and sent, the server client model for a network, and latency issues. Why do pages fail to load? How do you actually get content from the web?

Background Knowledge

String manipulation - taking “unstructured” data and turning it into something meaningful.

Introduction to Data Structures

Disclaimer: Here we will be doing a very fast introduction to data structures. I am teaching you everything you need to know to understand data structures on the web. That’s it.

Introduction to Data Structures

What is a data structure? Well, it’s a structure for data.

Definition: In computer science, when we use the term data, we are always implicitly referring to data stored in the machine. In data science, we are usually referring to the semantic meaning of that data, but they are the same thing at the end of the day.

Introduction to Data Structures

So how is data stored in your machine?

As integers. If I’m being pedantic, as a discrete subset of the integers, that turing referred to as computable numbers.

The computer translates these numbers into other symbols like letters, other numbers and instructions that the machine can execute.

Introduction to Data Structures

Python has two methods that expose you to this fact of basic numbers: ord() and chr()

ord() -> turns letters into numbers

chr() -> turns numbers into letters

[Example 1 - example1.py]

Introduction to Data Structures

So for our purposes, integers, strings, floats, negative numbers are all the data there is.

Definition: Data Structure:

A data structure takes collections of these more primitive data types and stores them in some way.

Introduction to Data Structures

The simplest example of a Data Structure is a node.

Definition: A node is a storage unit for some collection of primitive data and/or another node.

[example 2 - example2.py]

Introduction to Data Structures

A slightly more complex Data Structure and our first “used” Data Structure is the linked list.

Definition:

A linked list is a collection of sequentially accessible nodes.

[example 3 - example3.py]

Introduction to Data Structures

The most advanced Data Structure we care about is the tree data structure.

Definition:

A tree is a data structure where each node points to at least two nodes, and potentially more.

[example 4 - example4.py]

Introduction to Data Structures

Json, xml, and html can all be stored as tree data structures and have a hierarchical nature.

Headers and packets are similar to node data structures, storing collections of primitive data types.

Introduction to Networking

In networking we are looking at the connections between machines. Here the data structures are packets and headers. The packets are sent along a wire as electrical signals from one machine to the next.

Every time you type in a url to your browser you are requesting a number of packets from far off machine, which sends them to you. Regardless of what content is being sent across the wire, all of it must behave the same way in transmission.

Introduction to Networking

The OSI model of a computer:

The idea of the OSI model is to completely abstract how data is sent from point A to point B so that computer programmers don’t need to understand electrical engineering in order to do web programming.

Introduction to Networking

There are seven layers in the OSI model, but we will only care about Layer 7 - the application layer and Layer 4 - the transport layer.

At each layer of the OSI model there are a different set of rules and behaviors that computers will take, called protocols.

Introduction to Networking

At Layer 7 we will care about the http protocol
At Layer 4 we will care about the TCP protocol

Informally, a protocol is a set of methods and a set of standards for the types of data structures those methods act on.

Introduction to Networking

For the http protocol:

Data Structures:

- JSON, HTML, XML

Methods:

- Get, Put, Post

Introduction to Networking

For the TCP Protocol:

Data Structures:

    Header, Packet

Methods:

    We don’t actually care about any of the methods.

Introduction to Networking

Client - Your computer (usually). The one accessing the content.

Server - the computer “serving” (sending) the content.

Introduction to Networking

Get - gets data from the server and displays it to the client.

Put - sends data from the client to the server.

Post - sends data from the client to the server.

Explanation in detail if you care:

http://stackoverflow.com/questions/630453/put-vs-post-in-rest

Introduction to Networking

JSON is stored the same way as a python dictionary (they are exactly the same)

HTML is stored as a series of tags. If a tag is inside another tag, it is “deeper” in the tree.

XML works the same as HTML, except different tags are used.

Introduction to Networking

At Layer 4 we worry about things like reliability of connection and how to send the packets across the wire.

A header the describes the packets that will be sent from the server to the client.

Introduction to Networking

The html, xml or json documents are then split up into discrete “chunks” and turned into packets to be sent across the wire from server to client.

Introduction to Networking

Once all the packets have been collected the TCP protocol is used to join all the packets back together and then the HTTP protocol is used to display the html, xml or json as something human friendly to read and view.

Introduction to Networking

To summarize:

Servers store html, xml and json files and then send them over a connection to a client to get and display the data.

What is web scraping really?

Web scraping is somewhat of a “hack”. It’s a clever way to manipulate the servers on the internet to use data in a way it wasn’t intended.

So whenever you “scrape” the web, you are purposefully taking content that was not intended to be viewed the way it is being viewed, and using it for your own purposes.

practical methods

lxml - parses the html document into a tree like structure so a computer can easily read it.

requests - performs http requests against a server to get you the content you want

Finishing out

At the end of the day, web scraping is usually about taking html, xml, and json and turning it into csv’s, pandas data frames, and data stored in a database.