Web Scraping without Programming Tipsheet

NICAR 2012, St. Louis
Chris Keller, madison.com
Michelle Minkoff, The Associated Press

As journalists, we find ourselves relying on data more and more. One of our challenges, though, is that getting data into a structured format, making it possible to manipulate, analyze and present said information can often be very tough. Information can live on the Web in a variety of forms, and pulling down from the Internet , and into a structure is a process called Web scraping. When the tasks are really complicated, programming can help. But for some tasks, there are other tools -- which don't require such skills, which can help us out.

Here's a rundown of some of our favorite tools, and links that we've found helpful.

Nice general roundup of state of non-programmatic scrapers by Reporters' Lab Tyler Dukes - http://www.reporterslab.org/needlebase-and-the-future-of-scrapers/

Spreadheet Formulas:
Easy to use formulas found in Excel and Google Docs are actually a great low-hanging fruit way to grab data and information from websites and lend structure to them, especially data tables. Some you might use often are:

=ImportHtml(URL, query, index)
Example: Chicago Swimming Locations: http://bit.ly/zpLFDu
Learn More: http://bit.ly/AzWhfk

=ImportData(URL)
Example to import: Failed Banks csv: http://1.usa.gov/yg9mHT
*Example of import***:** http://bit.ly/zD86Jb

=ImportFeed(URL, query, headers, numItems)
Example: http://bit.ly/wlmlzy
Learn More: http://bit.ly/AoDWlk

Google Refine:
http://code.google.com/p/google-refine/
(Downloaded application that runs in the browser)

Best known for it's ability to clean up data through a variety of filters and facets, Google Refine is also a powerful tool to build CSVs and spreadsheets from JSON APIs that Twitter and other services use. An easy-to-follow recipe for this can be found at http://bit.ly/AhaRD3.

Yahoo Pipes:
http://pipes.yahoo.com/pipes/

(Browser-based service)

Pipes bills itself as “a tool to aggregate, manipulate, and mashup content from around the web.” With it you can combine many feeds into one, then sort, filter and translate it, and then grab the output as RSS, JSON, KML or XML.

Firebug/Chrome Developer Tools:
http://getfirebug.com/
(Included automatically in Chrome)

Look at a visual representation of the HTML that makes up the page. Right click on what you want to scrape, and hit Inspect Element to get a clear delineation of the pattern of what comes before/after your desire data. That pattern will help you create your scraper, with or without programming.

Firequark:
Download for Firefox 3: http://www.quarkruby.com/firequark-1.2.xpi
Read about how it works:
http://www.quarkruby.com/2007/9/5/firequark-quick-html-screen-scraping

Supercharge Firebug so you have a “selector” for the piece of data you want to scrape. Instead of targeting data by looking before and after it, we look at how to best identify it that piece of data. The selector is what’s also used in CSS to color a certain item red, for example. Right-click on desired data, and grab a unique CSS selector, or one that applies to a general group.

Scraper Chrome Extension:
https://chrome.google.com/webstore/detail/mbigbapnjcgaffohmbkdlecaccepngjd

Right-click on one example of an item you would like to scrape. Select Scrape Similar. The extension will pull a table of everything that is like it on a page. Great for getting a list into a table form, when you don’t want to copy/paste individual items. Can’t handle multiple fields of data at once.

WP-Web Scraper:
http://wordpress.org/extend/plugins/wp-web-scrapper/

Use the selector you found with Firebug, and enter it and the URL into a simple Wordpress tag. That will dynamically pull a piece of data into your Wordpress page. You’ll need to install this plugin before you can use it, but once you do, just use the shortcode:
[wpws url=“” selector= “”]

Needlebase and Dapper:

http://needlebase.com/
http://open.dapper.net/

Needlebase and Dapper both have the ability to take a variety of different types of information and turn them into a data feed. Could be an RSS, or it could iterate through a multiple-page searchable database.

Needlebase has been bought by Yahoo, and the current website is set to be retired on June 1, 2012, but will remain fully functional until that point, according to a note on the website.

As for Dapper, getting past the first step of "creating a dapp" (what they call one of their scrapes) doesn't seem to function, as of Feb. 2012. The project has also been bought by Yahoo.

Junar:
http://www.junar.com/

A relatively new project that's great for grabbing data from, and keeping track of, HTML tables on Web pages. Point the tool at a Webpage you want to track, highlight the table, and name the stream. You can embed that on a webpage, or with a bit of programming, access that table through Junar's API. As changes are made to the table, Junar will update its data stream, which will in turn, update on your site. More robust developments are planned for this relatively new project, according to its developers.

Outwit Hub:
http://www.outwit.com/products/hub/

A fairly robust and nuanced tool. With a single click, can grab all links, images, tables, etc. from a page. You can also write your own custom scraper by defining what HTML comes before and after your data, and then run it. You can apply this scraper to a series of links that show up in the link auto-detection tab on the left of the interface.

This has recently been made into a downloadable, stand-alone app for Mac and Windows, and remains a robust Firefox extension.

It's important to note that the free version of Outwit (Hub) only supports exporting 100 rows of data at a time. Be sure you're aware when you're working with a truncated data set. (You can upgrade to Outwit Pro to remove this limitation, automatically export lists of documents, unlock other features, etc.)

ScraperWiki:
https://scraperwiki.com/

ScraperWiki aims more toward basic programming skills, and as such, the learning curve is a bit steeper. But fear not. IRE and ScraperWiki are hosting a 12-hour data liberation marathon

during the CAR Conference from 6 p.m. Thursday, Feb. 23, to 6 a.m. on Friday, Feb. 24. It begins with an opening session on how to scrape using Scraperwiki, and the liberation will begin.

Scrape responsibly:
Scraping relies on the structure of a website not changing, and standard rules of ethics still apply. It should be a last resort. Try to request your data first, work with representative from the organization to get the information you want.

“Take only what you need. When screen scraping it's tempting to just grab everything, downloading more and more pages just because they are there. But it’s a waste of their bandwidth and your time, so make a list of the data you want to extract and stick to it.”

--Will Larson, programmer, http://lethain.com/an-introduction-to-compassionate-screenscraping/

Words of encouragement:

Every tutorial, walkthrough and tip sheet offers you practice and an excuse to fail. But the fun thing is, eventually it will start to make sense and your imagination will run wild with the possibilities.

Ultimately, you will come to a data set you want to grab that is too advanced for these tools. Perhaps it’s a paginated database, or you need to enter 100 different inputs in a search form, and scrape THOSE results. Then, it may be time to start looking into programming tools, or collaborating with a programmer.

These tools are not the only options, but sometimes they are faster than other methods. And most importantly, they help us understand what scraping is, and why we might want to do it. Getting structured data mostly means figuring out the “columns” in Excel, and defining patterns that help us get there. Understanding that fact is the most valuable “tool” you can have -- and it will be very helpful to you, whether you’re using Excel or Python to scrape.