Brandon Selway

I started by creating a generic scraper function. This function takes a url, applies a selector using a serverside DOM/jQuery emulator (jsdom), and appends the results to a specified variable.

Next I worked out the jQuery selectors necessary each tier on the path from root URL to final date data.

Then I went even further to test/show off some features of Node.js. I programmed the option to process http requests in parallel rather than serially (tamejs) and I wrapped the scraper program into a live stream web-server (dnode). These improvements highlight the performance strength and potential for innovation of this language. (Open the link above in multiple browsers simultaneously and initiate a request!)

What I would improve:

-The web interface does not work in IE. IE does not fully implement Javascript standards so this could be fixed by finding a JS library to add missing functionality.

-Due to the nature of the process, the output from parallel is in arbitrary order. This is solved by converting the date strings into numerically sortable data, and using this to order the array.

-I would improve the code legibility and logic by making better use of utility libraries (like underscore.js) and breaking out more code chunks into functions.