

## Web scrapping/MongoDB

we will use cheerio npm as our web scrapping library

**\*\*Before you start webscrapping you should look at how the page you want to scrap is structure in order to construct the JS correctly.**

### Cheerio -

in line 31 of 1-Scrapping/server.js we use '\$' to stay consistent with jQuery

*Side note (go over the comments inside the server files. they really broke it down*

```
1  $("figure.site > a > img").each(function(i, element) {
2
3      /* Cheerio's find method will "find" the first matching child element in a parent.
4       *   We start at the current element, then "find" its first child a-tag.
5       *   Then, we "find" the lone child img-tag in that a-tag.
6       *   Then, .attr grabs the imgs src value.
7       *   So: <figure> -> <a> ->  -> "link" */
8      // var imgLink = $(element).find("a").find("img").attr("src");
9      var imgLink = $(element).attr("src");
10
11     // Push the image's URL (saved to the imgLink var) into the result array
12     result.push({ Link: imgLink });
13 });
```

### Activity

skeleton-cheerio file

## MongoDB review

see mongo\_exercise.txt. (answers at the bottom of file)

## Mongojs

we will “never” use mongo in shell form. we will work with it in places like node ... so of course someone has made an npm for it ... mongojs

### Activity

skeleton-mongojs file

## Last Activity

Create a site that brings mongojs and query together

*Side note(theory did his very cool function technique and helped me out with the result)*

*skeleton-combo file —> app.js (where the activity was)*