

# HW 04 - Course Catalog Analytics and Visualization

---

In this homework you will work with MIT's course catalog data. The URL address for the site is:

```
http://student.mit.edu/catalog/index.cgi
```

## Question 1. - Return an array with the links to every page

Return the address of all the html pages in the MIT course catalog - return a string array. For example, the first page for Course 1 is:

```
http://student.mit.edu/catalog/m1a.html
```

Sample Data:

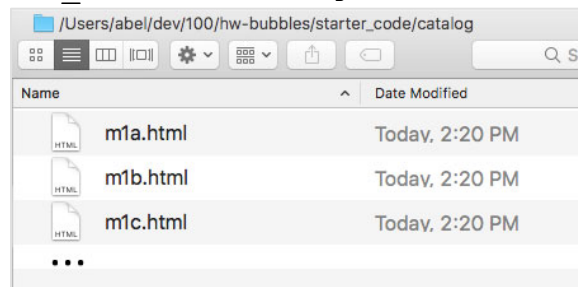
```
[ 1 [
  2   "http://student.mit.edu/catalog/m1a.html",
  3   "http://student.mit.edu/catalog/m1b.html",
  4   "http://student.mit.edu/catalog/m1c.html",
  5   "http://student.mit.edu/catalog/m2a.html",
  ... ]
```

## Question 2. - Download the Data

Download every course catalog page.

You can use the NPM package `request`. Or `curl` with the NPM package `shelljs`.

Save every page to `your_folder/catalog`:



### Question 3. - Combine all files into one

Combine all files into one, save to `your_folder/catalog/catalog.txt`

You can use the file system API, <https://nodejs.org/api/fs.html>

### Question 4. - Remove Whitespace

Remove line breaks and whitespaces from the file. Return a string of scrubbed HTML. In other words, HTML without line breaks or whitespaces.

You can use the NPM package `html-minifier`.

### Question 5. - Load data into DOM, get courses

Load your scrubbed HTML into the DOM. Use the DOM structure to get all the courses.

Return an array of courses.

You can use the NPM package `cheerio`.

### Question 6. - Get titles

Return an array of course titles.

You can use the NPM package `cheerio`.

### Question 7. - Clean titles

Filter out punctuation, numbers, and common words like "and", "the", "a", etc.

Return clean array.

### Question 8. - Make words array

Make an array of words from all the titles.

Return array of words.

### Question 9. - Count the word frequency.

Count the word frequency. Return a word count array.

## Question 10 - Graph the word frequency

The included zip file, `catalogSample.zip`, contains everything you need to graph your word frequency data.

### Example

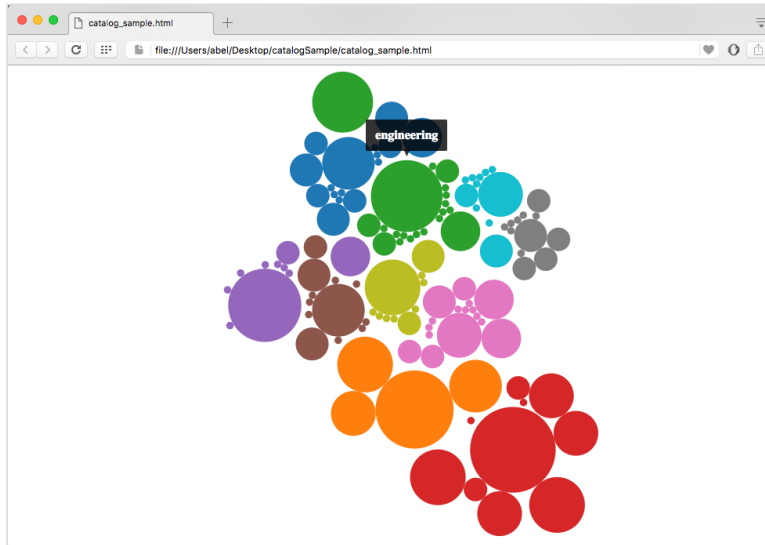
Sample word frequency data (`catalog_data.js`):

```
catalog_data.js
1 var scores = {"engineering":16,"computation":4,"and":29,"data":4,"science":3,"computer":2,"programming":1,"for":5,"scientific":1,"applications":1,"big":1,"small":1,"solutions":2,"with":1,"a":7,"large":1,"impact":1,"uncertainty":1,"in":5,"project":1,"evaluation":1,"management":1,"senior":1,"civil":5,"environmental":17,"design":10,"j":21,"of":8,"electromechanical":1,"robotic":1,"systems":7,"complex":1,"issues":1,"building":2,"communicating":1,"ideas":1,"fundamentals":3,"ecology":2,"i":8,"b":5,"ii":5,"principles":1,"energy":2,"water":2,"sustainability":1,"introduction":8,"to":8,"modeling":4,"simulation":2,"urban":1,"networks":1,"geomaterials":1,"geomechanics":1,"mechanics":7,"structures":2,"soils":1,"structural":3,"soil":1,"geotechnical":1,"new":2,"transportation":1,"buildings":1,"solid":1,"dynamics":4,"control":1,"concrete":1,"vibrations":1,"fluid":3,"transport":3,"processes":3,"the":2,"environment":2,"nonlinear":2,"continuum":1,"physical":1,"limnology":1,"turbulence":1,"hydrology":4,"resources":1,"global":1,"change":1,"groundwater":1,"analysis":5,"multivariate":1,"chemistry":3,"cancer":1,"risks":1,"prevention":1,"therapy":1,"ethics":1,"engineers":1,"health":2,"biology":3,"microbiology":2,"air":1,"pollution":1,"traveling":2,"research":3,"experience":2,"trex":2,"fieldwork":2,"communication":1,"aided":1,"teaching":1,"practicum":1,"laboratory":2,"software":2,"architecting":1,"pattern":1,"recognition":1,"computational":1,"geometry":1,"meng":1,"concepts":1,"practice":1,"wave":1,"propagation":1,"robust":1,"optimization":1,"economy":1,"module":1};
Line 1, Column 1 Tab Size: 4 JavaScript Next
```

Sample html file (`catalog_sample.html`):

```
catalog_sample.html
1 <!DOCTYPE html>
2 <html>
3 <script src="http://d3js.org/d3.v3.min.js"></script>
4 <script src="d3.tip.js"></script>
5 <script src="catalog_data.js"></script>
6 <script src="catalog_graph.js"></script>
7 <link rel="stylesheet" href="styles-example.css">
8 <body>
9 <div id="target"></div>
10 </body>
11 <script>
12
13 // visualize data
14 graph();
15
16 </script>
17
18 </html>
39 characters selected Tab Size: 4 HTML
```

Sample html file in browser (catalog\_sample.html):



### Question 11. - (OPTIONAL) Improve Graphing Logic

The graph logic is based on the scores you calculated for words. You can find the code in the graphing JavaScript file.

```
for (var word in scores) {  
  nodes.push({radius: radius(scores[word]),  
             color: color(word.length), word: word,  
             score: scores[word]});  
}
```

Can you improve the graph?