

Your starting point...

js/helper.js

Within helper.js, you'll find a large collection of strings containing snippets of HTML. Within many snippets, you'll find placeholder data in the form of `%data%` or `%contact%`. Each string has a title that describes how it should be used. For instance, `HTMLworkStart` should be the first `<div>` in the Work section of the resume. `HTMLSchoolLocation` contains a `%data%` placeholder which should be replaced with the location of one of your schools.

Your process:

The resume has four distinct sections: work, education, projects and a header with biographical information. You'll need to:

1. Build four JavaScript objects, each one representing a different resume section. The objects that you create (including property names and the data types of their values) need to follow the schema below exactly. All properties should be included and contain a value of the type specified unless the property is marked 'optional'. Property values may contain real or fake data. Property names are case-sensitive. Make sure your JavaScript objects are formatted correctly using jshint.com.

2.

3. `bio` contains:

4.

```
5.     name : string
6.     role : string
7.     contacts : an object with
8.         mobile: string
9.         email: string
10.            github: string
11.            twitter: string (optional)
12.            location: string
13.        welcomeMessage: string
14.        skills: array of strings
15.        biopic: url
16.        display: function
```

17.

18. `education` contains:

```
19.     schools: array of objects with
20.         name: string
21.         location: string
22.         degree: string
23.         majors: array of strings
24.         dates: string (works with a hyphen between them)
25.         url: string (optional)
26.     onlineCourses: array of objects with
27.         title: string
28.         school: string
```

```

29.         dates: string (works with a hyphen between them)
30.         url: string
31.     display: function

```

32. **work** contains

```

33.         jobs: array of objects with
34.             employer: string
35.             title: string
36.             location: string
37.             dates: string (Can be 'in progress')
38.             description: string
39.     display: function

```

40. **projects** contains:

```

41.         projects: array of objects with
42.             title: string
43.             dates: string (works with a hyphen between them)
44.             description: string
45.             images: array with string urls
46.     display: function
47.

```

1. Iterate through each JavaScript object and append its information to index.html in the correct section.

1. First off, you'll be using jQuery's **selector.append()** and **selector.prepend()** functions to modify index.html. **selector.append()** makes an element appear at the end of a selected section. **selector.prepend()** makes an element appear at the beginning of a selected section.

2. Pay close attention to the ids of the **<div>**s in index.html and the HTML snippets in helper.js.

They'll be very useful as jQuery selectors for **selector.append()** and **selector.prepend()**

2. You'll also be using the JavaScript method **string.replace(old, new)** to swap out all the placeholder text (e.g. **%data%**) for data from your resume JavaScript objects.

3. Here's an example of some code that would add the location of one your companies to the page:

```

0. var formattedLocation = HTMLworkLocation.replace("%data%",
work.jobs[job].location);

```

```

1. $(".work-entry:last").append(formattedLocation);

```

1. Use the mockup at the bottom of this document as a guide for the order in which you should append elements to the page.

4. The resume includes an interactive map. Do the following to add it.

0. In resumeBuilder.js, append the googleMap string to **<div id="mapDiv">**.

1. In index.html, uncomment the Google script element: **<script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?libraries=places"></script>**

2. In helper.js, at the bottom of the file, uncomment code to initialize map and set fitBounds.

5. All of your code for adding elements to the resume should be contained within functions.

6. As described in the JavaScript object schema, each **display** function should be encapsulated within the JavaScript object it displays in the resume. For instance, your **display** function for appending 'work' experience data to the resume should be encapsulated within the **work** JavaScript object. The **display** function can be encapsulated within the **work** JavaScript object definition in the same way other properties are defined there, or it can be encapsulated later in the file using dot notation. For example: **work.display =**

7. It's possible to make additional information show up when you click on the pins in the map. Check out line 174 in helper.js and the Google Maps API to get started.