**Project Overview**

In this project you will write the JavaScript that powers your own online resume.

This project is meant to be completed while taking the [**JavaScript Basics course**](https://classroom.udacity.com/courses/ud804). Throughout the course you'll build your resume by writing JavaScript that will combine your personal information with pre-written HTML and CSS templates to generate your resume.

**Note**

If you have successfully completed the project for the JavaScript Basics course in the past (which entails having graduated from the course and having access to your course certificate), simply email us at frontend-project@udacity.com with your passing evaluation and we'll give you credit for this project.

**Why this Project?**

Once you've mastered the skills of a front end web developer you'll want to make a great first impression. You need a resume that stands out.

The resume you build will not only help you build important skills, but will also make it easy to show employers why you’re perfect for the job. As you progress through this nanodegree you can update this resume with your new skills and projects.

**What will I Learn?**

You will learn basic JavaScript syntax, which includes manipulating data, building loops and creating functions. At the same time, you’ll learn some simple jQuery DOM manipulation methods to build your resume the moment a user opens your website.

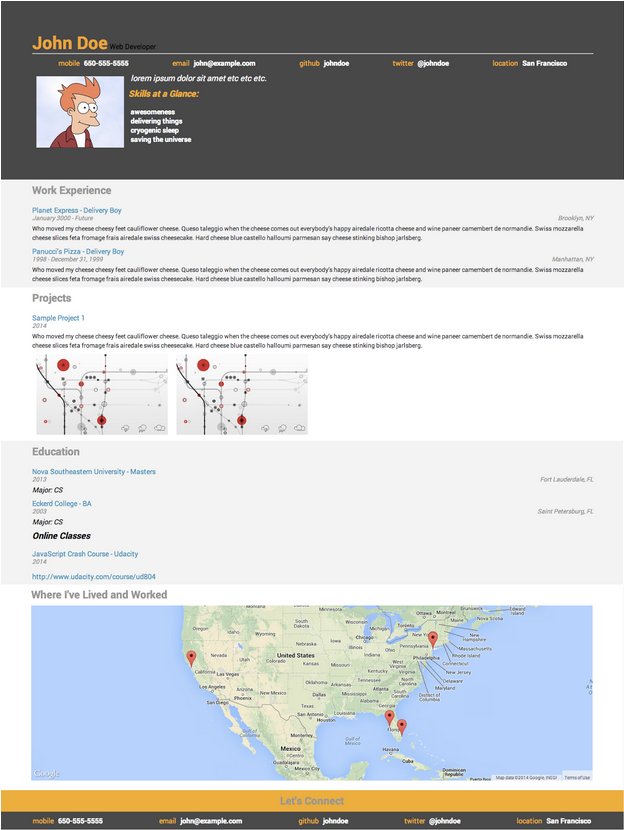
**How does this Help my Career?**

* It's a resume. Resumes help you get jobs.
* JavaScript is **the** language of web development.
* JavaScript lets you turn static web pages into dynamic applications.
* Since it runs on normal web browsers, JavaScript is one of the most accessible and flexible programming languages.

**How do I complete this project?**

Review the Online Resume [**Project Rubric**](https://review.udacity.com/?_ga=1.189245867.12280332.1465333852#!/projects/2962818615/rubric).

1. In this project, you will store your resume data in four JavaScript objects according to the schema given below. As is often the case when leveraging an API, the objects must follow the schema exactly. All properties must be present and have real or fake values. The names must match those in the schema (note that object and property names are case-sensitive). All property values should be of the data-type given for the property in the schema. For example, if the data type is given as an array, it is not acceptable to use a string as a value for that property.
2. Once you've created your JavaScript objects, you will write the code needed to dynamically display all of the resume data contained within these objects in your resume. Your resume should be capable of displaying any resume that provides the data according to the given schema. Note that [**'forEach' or 'for'**](https://udacity.github.io/frontend-nanodegree-styleguide/javascript.html#for-in-loop) (rather than 'for in') loops must be used to display the **skills**, **jobs**, **projects**, project **images**, **schools** and **onlineCourses** arrays. Your solution should work for any number of items in those arrays.
3. All of the HTML code needed to build the resume is stored in **js/helper.js** variables. The variable names indicate their function. You will replace substrings in these variable string values such as **%data%** and **#** with the data in your JavaScript objects, and append or prepend the formatted result to your resume in the appropriate location.
4. If you need a refresher on JavaScript syntax, go to the [**JavaScript Basics**](https://classroom.udacity.com/nanodegrees/nd001/parts/0011345406/modules/296281861575460/lessons/1946788554/concepts/25505685350923) course; if you would like to understand how this project is manipulating and traversing the DOM, review [**Intro to jQuery**](https://classroom.udacity.com/nanodegrees/nd001/parts/0011345406/modules/296281861575461/lessons/3314378535/concepts/33166386820923).
5. Go through the videos and assignments in this course to learn the JavaScript necessary to build your resume.
6. Fork the project repo from [**GitHub**](https://github.com/udacity/frontend-nanodegree-resume) and begin building your resume.
7. If you are prompted to do so, you may want to get a [**Google Maps API key**](https://developers.google.com/maps/documentation/javascript/get-api-key), and include it as the value of the key parameter when loading the Google Maps API in **index.html**: <script src="http://maps.googleapis.com/maps/api/js?libraries=places&key=[YOUR\_API\_KEY]"></script> You may have some initial concerns with placing your API key directly within your JavaScript source files but rest assured this is perfectly safe. All client-side code must be downloaded by the client; therefore, the client must download this API key - it is not intended to be secret. Google has security measures in place to ensure your key is not abused. **It is not technically possible to make anything secret on the client-side.**
8. Check your work against the [**Project Rubric**](https://review.udacity.com/?_ga=1.189245867.12280332.1465333852#!/projects/2962818615/rubric).
9. When you are satisfied with your project, submit it according to the Submission Instructions on the next page.

**[[](https://classroom.udacity.com/nanodegrees/nd001/parts/9d3a4440-1e65-4fc5-9ac1-b40e401ad069/modules/296281861575462/lessons/2962818615239847/concepts/29594685550923)](https://classroom.udacity.com/nanodegrees/nd001/parts/9d3a4440-1e65-4fc5-9ac1-b40e401ad069/modules/296281861575462/lessons/2962818615239847/concepts/29594685550923)**

**[By the end, your resume should look something like this.](https://classroom.udacity.com/nanodegrees/nd001/parts/9d3a4440-1e65-4fc5-9ac1-b40e401ad069/modules/296281861575462/lessons/2962818615239847/concepts/29594685550923)**

Your repository should include the following files:

* **index.html**: The main HTML document. Contains links to all of the CSS and JS resources needed to render the resume, including resumeBuilder.js.
* **js/helper.js**: Contains helper code needed to format the resume and build the map. It also has a few function shells for additional functionality. More on helper.js further down.
* **js/resumeBuilder.js**: This file is empty. You should write your code here.
* **js/jQuery.js**: The jQuery library.
* **css/style.css**: Contains all of the CSS needed to style the page.
* **README.md**: The GitHub README file.
* and some images in the images directory.

**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**](http://jshint.com/).
2. bio contains:
3. name : string
4. role : string
5. contacts : an **object** **with**
6. mobile: string
7. email: string
8. github: string
9. twitter: string (optional)
10. location: string
11. welcomeMessage: string
12. skills: array of strings
13. biopic: url
14. display: function
15. education contains:
16. schools: array of objects with
17. name: string
18. location: string
19. degree: string
20. majors: array of strings
21. dates: string (works with a hyphen between them)
22. url: string (optional)
23. onlineCourses: array of objects with
24. title: string
25. school: string
26. dates: string (works with a hyphen between them)
27. url: string
28. display: function
29. work contains
30. jobs: array of objects with
31. employer: string
32. title: string
33. location: string
34. dates: string (Can be 'in progress')
35. description: string
36. display: function
37. projects contains:
38. projects: array of objects with
39. title: string
40. dates: string (works with a hyphen between them)
41. description: string
42. images: array with string urls
43. display: function
44. Iterate through each JavaScript object and append its information to index.html in the correct section.
    * 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.
    * 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()
45. 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.
46. Here’s an example of some code that would add the location of one your companies to the page:
    * var formattedLocation = HTMLworkLocation.replace("%data%", work.jobs[job].location);
    * $(".work-entry:last").append(formattedLocation);
      + Use the mockup at the bottom of this document as a guide for the order in which you should append elements to the page.
47. The resume includes an interactive map. Do the following to add it.
    * In resumeBuilder.js, append the googleMap string to <div id=”mapDiv”>.
    * In index.html, uncomment the Google script element: <script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?libraries=places"></script>
    * In helper.js, at the bottom of the file, uncomment code to initialize map and set fitBounds.
48. All of your code for adding elements to the resume should be contained within functions.
49. 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 =
50. 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.

**Evaluation**

Your project will be evaluated by a Udacity reviewer according to the [**Online Resume Project Rubric**](https://review.udacity.com/#!/projects/2962818615/rubric). Be sure to review it thoroughly before you submit. All criteria must "meet specifications" in order to pass.

Please make note of the [**Udacity Front End JavaScript Style Guide**](https://udacity.github.io/frontend-nanodegree-styleguide/javascript.html) to make sure your project meets code quality and readability requirements.

**Submission**

1. The master branch is the default Github repository branch. If you wish to submit another branch, you'll need to set it as the [**new default branch**](https://help.github.com/articles/setting-the-default-branch/) inside your Github repository.
2. When you're ready to submit your project go back to your [**Udacity Home**](https://www.udacity.com/me), click on Project 2, and we'll walk you through the rest of the submission process. Due to the high volume of submissions we receive, please allow up to **7 business days** for your evaluation to be returned.
3. If you are having any problems submitting your project or wish to check on the status of your submission, please email us at **frontend-project@udacity.com** or visit us in the [**discussion forums**](http://discussions.udacity.com/).

**What's Next?**

You will get an email as soon as your reviewer has feedback for you. In the meantime, review your next project and feel free to get started on it or the courses supporting it!

**Useful Links**

* [**Udacity Front End JavaScript Style Guide**](https://udacity.github.io/frontend-nanodegree-styleguide/javascript.html)

Please make sure to completely fill out ALL sections of the resume including the **footer**section.

If you want to start now with good JavaScript practices, you can follow the [**Udacity JavaScript StyleGuide**](http://udacity.github.io/frontend-nanodegree-styleguide/javascript.html).

Project Submission

Once you've mastered the skills of a front end web developer you'll want to make a great first impression. You need a resume that stands out. The resume you build will not only help you build important skills, but will also make it easy to show employers why you’re perfect for the job. As you progress through this Nanodegree program you can update this resume with your new skills and projects.

**OPTIONAL Challenge Problems**

To help you keep practicing your JavaScript, we've created some optional challenge problems for you. These bonus problems are **not required** in order to complete the nanodegree. These problems will also **not be evaluated by instructors**, however we strongly urge students to try them out. Remember practice makes perfect!

**Where to find them**

You can find the optional challenge problems at the end of the **[Javascript Basics course](https://classroom.udacity.com/courses/ud804/lessons/3412099388/concepts/34278488440923" \t "_blank)**.