**Module #2 - Assignment #2**

Read through the ["So You Wanna Be A Web Developer..."](https://the-coding-bootcamp.gitbooks.io/pre-work-book/content/chapter2.html) chapter of Pre-Work. Then, visit a job site like [Indeed](http://www.indeed.com/) or [Dice](http://www.dice.com/). Search for any of the following job positions: Web Developer, Frontend Web Developer, Backend Developer, Javascript Developer, Node Developer

Read the job postings for at least 10 of the listings. As you are reading, take note of at least one technology, tool, or term, with which you are unfamiliar, from each listing (e.g., Angular, API). Then, search online for a definition or description of that technology or term. Try to understand what role it plays in the context of how web applications were described in that chapter. Open a Word or Text file and paste in your description for each of the technologies you researched. You should have a total of 10 descriptions listed.

**AJAX**

* **A**synchronous **J**avaScript **A**nd **X**ML
* Uses a combination of:
  + A browser built-in XMLHttpRequest object - to request data from a web server
  + Javascript and HTML DOM (Document Object Model) - to display or use the data
* AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
* How it works:

1. An event occurs in a web page (the page is loaded, a button is clicked)
2. An XMLHttpRequest object is created by JavaScript
3. The XMLHttpRequest object sends a request to a web server
4. The server processes the request
5. The server sends a response back to the web page
6. The response is read by JavaScript
7. Proper action (like page update) is performed by JavaScript

**Task runners (Grunt and Gulp.js)**

* JavaScript-based workflow management tools
* Build tools that tackle repetitive but essential code-related tasks in front-end development, such as:
  + Optimization
  + Minifying
  + Testing
  + CSS pre-processing
  + Deploying
* **Grunt**
  + Automates tasks like testing, compilation, and minification
  + Lets programmers build customized workflows with Javascript
  + ‘The kitchen sink of plug-ins’, which process multiple tasks
  + Best for small projects
  + Favors configuration over code and gives developers more decision-making power
  + Runs on OS X, Windows, and Linux
  + Develops build flows with JavaScript
  + Manages deployments
  + Uses JSON-like data configuration files
  + Developers do not need to be proficient in Node.js
  + Examples of sites that use Grunt: Twitter, Adobe, and jQuery
* **Gulp.js**
  + A streaming build system that automates and streamlines tasks using Node.js’s streams and modules
    - Streaming build system:
      * An asynchronous approach to tasks that involves processing files independently, then piping them downstream to the next plug-in
      * No waiting, just a steady flow of processing tasks
      * A *file stream* (a Unix-and Node.js philosophy) is created for related files of code. This modified stream is fed into another stream, where files are *concatenated* (a task that links these related files together). In the next stream, concatenated files are then *minified*, and you’re left with modified code that’s ready for a production server.
    - Stream approach gives developers more control over the workflow.
    - Fast - echoing AJAX’s concurrent approach: The output of one stream is pushed into the input of another stream (no need for the program to temporarily write files, saving disk space and time.
    - Cross-platform: supports Linux, OS X, and Windows
    - Natural match for developers who know Node
    - Gulp API is simple, elegant, and easy to use
    - Gulpfiles are simply JavaScript files (write/use Node modules without requiring a Gulp plug-in)
    - Faster than Grunt
    - Gulp focuses on the code, while Grunt focuses on the task
    - Orchestrator enables developers to set up task dependencies, allowing for more control over the sequence of plug-ins

**gRPC**

* **R**emote **P**rocedure **C**all
* Modern open source high performance RPC framework that can run in any environment
* Can efficiently connect services in and across data centers with pluggable support for load balancing, tracing, health checking and authentication
* Applicable in last mile of distributed computing to connect devices, mobile applications and browsers to backend services
* Examples: Square, Netflix, Cisco

**Jenkins**

* Open source automation server written in Java
* Can be installed through any native system packages, Docker, or even run standalone by any machine with a Java Runtime Environment (JRE) installed
  + Docker: Open platform for developers and sysadmins to build, ship and run distributed applications, whether on laptops, data center VMs, or the cloud

**Waterfall**

* Sequential (non-iterative) design process in which progress is seen as flowing steadily downwards through the phases of:
  + Conception
  + Initiation
  + Analysis
  + Design
  + Construction
  + Testing
  + Production/Implementation
  + Maintenance
* Places emphasis on documentation
* Best suited for projects where requirements and scope are fixed, the product itself is firm and stable, and the technology is clearly understood

**LESS**

* CSS preprocessor, which allows writing clean CSS in a programming construct instead of static rules
* Extends CSS with dynamic behavior such as variables, mixing, operations and functions
  + Variables: Defined with @ sign. Variable assignment is done with a colon (:). During translation, the values of the variables are inserted into the output CSS document.
  + Mixins: Allow embedding all the properties of a class into another class by including the class name as one of its properties, thus behaving as a sort of constant or variable. Can also behave like functions, and take arguments.
    - CSS does not support Mixins.
    - Any repeated code must be repeated in each location.
    - Allow for more efficient and clean code repetitions, as well as easier alteration of code.
  + Operations: Allow addition, subtraction, division and multiplication of property values and colors, which can be used to create complex relationships between properties.
  + Functions: Map one-to-one with JavaScript code, allowing manipulation of values.
* Runs on both the server-side (with Node.js and Rhino) or client-side
* Originally written in Ruby, but has since been deprecated and replaced by JavaScript
* Main difference between Less and other CSS precompiles - Less allows real-time compilation via less.js by the browser

**React**

* Open-source (with caveats) JavaScript library for building user interfaces.
* Maintained by Facebook, Instagram and a community of individual developers and corporations
* Allows developers to create large web-applications that use data that can change over time, without reloading the page
* Aims to provide speed, simplicity and scalability
* React Fiber (introduced Apr 2017) will become the foundation of any future improvements and feature development of the React framework.

**jQuery**

* Cross-platform JavaScript library designed to simplify the client-side scripting of HTML
* Free, open-source software using permissive MIT License
* Most widely deployed JavaScript library by a large margin
* Syntax is designed to make it easier to:
  + Navigate a document
  + Select DOM elements
  + Create animations
  + Handle events
  + Develop AJAX applications
* Principles of developing with jQuery:
  + Separation of JavaScript and HTML
  + Brevity and clarity
  + Elimination of cross-browser incompatibilities
  + Extensibility (see below)
* Plug-ins can be created on top of the JavaScript library, enabling developers to create abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets. (modular approach)
* Microsoft and Nokia bundle jQuery on their platforms

**JSON**

* **J**ava**S**cript **O**bject **N**otation
* Open-standard file format that uses human-readable text to transmit data objects consisting of attribute-value pairs and array data types (or any other serializable value)
* Common data format used for asynchronous browser/server communication
* Replacement for XML in some AJAX-style systems
* Use file extension .json
* Originated out of the need for stateful (designed to remember preceding events/interactions), real-time server-to-browser communication protocol without using browser plugins such as Flash or Java applets.

**Bootstrap**

* Free and open-source front-end web framework for designing websites and web applications.
* Contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.
* Only applicable to front-end development.
* Began as a collaboration among Twitter developers.
* Mobile first approach, supporting responsive web design.
* Modular and consists of a series of Sass (formerly Less) stylesheets that are compiled into a bundle and included in web pages, but individual components can be included or removed.