Node dev-dependencies

https://stackoverflow.com/questions/10068592/how-do-i-update-devdependencies-in-npm

- http://scottksmith.com/blog/2014/06/25/8-npm-tips-for-better-node-development/

Dev Dep...

https://stackoverflow.com/questions/6480549/install-dependencies-globally-and-locally-using-package-json

global vs local

- https://nodejs.org/en/blog/npm/npm-1-0-global-vs-local-installation/

learn from issues

- https://github.com/dylang/npm-check

Node Tutorials!!!

- GURU99: https://www.guru99.com/node-js-tutorial.html

Node Hero

- https://blog.risingstack.com/node-hero-tutorial-getting-started-with-node-js/

Node HTTP stream

google"great tutorial on http get listen"

- Stream basics
- Learn Server-side JavaScript
- Creating HTTP Server in Node

```
> npm -g install http-server
```

  Just launch http-server command line and this will launch a http server serving ./ on localhost:3000

- Other Colin's Articles
- Colin's GitHub

- Another guy: https://www.sitepoint.com/author/jhibbard/
- https://www.sitepoint.com/author/jvandemoere/
- JavaScript Visualization: https://www.sitepoint.com/author/rlubushkin/

Express package for Node

- http://evanhahn.com/understanding-express-3/

Intro to Node.js and other Tutorials

- Print:

https://www.sitepoint.com/deeper-dive-javascript-promises/

https://www.sitepoint.com/implementing-memoization-in-javascript/

https://www.sitepoint.com/stream-your-webcam-to-a-browser-in-javascript/

https://www.sitepoint.com/http-authentication-in-node-js/

https://www.sitepoint.com/web-scraping-in-node-js/

https://www.sitepoint.com/headless-webkit-and-phantomjs/

https://www.sitepoint.com/a-detailed-breakdown-of-the-script-tag/

https://www.sitepoint.com/profiling-page-loads-with-the-navigation-timing-api/

- Read:

https://www.sitepoint.com/accessing-the-file-system-in-node-js/

https://www.sitepoint.com/the-basics-of-the-shadow-dom/

https://www.sitepoint.com/modern-javascript-development-hard/

https://www.sitepoint.com/fun-with-javascript-numbers/

https://www.sitepoint.com/back-to-basics-javascript-hoisting/

https://www.sitepoint.com/an-introduction-to-node-js/

https://www.sitepoint.com/back-to-basics-javascript-object-syntax/

- Printed:

https://www.sitepoint.com/solve-global-npm-module-dependency-problem/

https://www.sitepoint.com/exceptional-exception-handling-in-javascript/

https://www.sitepoint.com/javascript-closures-demystified/

https://www.sitepoint.com/back-to-basics-array-extras/

https://www.sitepoint.com/creating-a-http-server-in-node-js/

https://www.sitepoint.com/anatomy-of-a-modern-javascript-application/

https://www.sitepoint.com/basics-node-js-streams/

https://www.sitepoint.com/simplifying-asynchronous-coding-async-functions/

https://www.sitepoint.com/overview-javascript-promises/

https://www.sitepoint.com/making-http-requests-in-node-js/

- Reference:

Proxy Wiki cute... reference

Unix Pipe and Streams

- Reactive Programming, some crazy funky shit

https://gist.github.com/staltz/868e7e9bc2a7b8c1f754

- Stream Updates with Server Sent Events

https://www.html5rocks.com/en/tutorials/eventsource/basics/

- Only if you wan to study other ppls code

https://www.sitepoint.com/angular-2-mockbackend/

Node Manual

- https://nodejs.org/docs/v0.4.2/api/modules.html

requiring modules in Node.js

- https://medium.freecodecamp.org/requiring-modules-in-node-js-everything-you-need-to-know-e7fbd119be8

Tutorial Full-Stack: JS, Node, React, MongoDB

- https://www.lynda.com/Express-js-tutorials/Course-overview/533304/557601-4.html#tab

Node.js modules

https://www.codementor.io/ashish1dev/list-of-useful-nodejs-modules-du107mcv3

"Async", "Browserify", "Bower", "Backbone", "Csv", "Debug", "Express", "Forever", "Grunt", "Gulp", "Grunt", "Gulp", "Hapi", "Http-server", "Inquirer", "Jquery ", "Jshint ", "Koa", "Lodash", "Less", "Moment", "Mongoose", "MongoDB", "Npm", "Nodemon", "Nodemailer", "Optimist", "Phantomjs", "Passport", "Q", "Request", "Socket.io", "Sails ", "Through", "Underscore", "Validator", "Winston", "Ws", "Xml2js", "Yo", "Zmq "

Node Core Module names

"http", "events", "util", "domain", "cluster", "buffer", "stream", "crypto", "tls", "fs", "string_decoder", "path", "net", "dgram", "dns", "https", "url", "punycode", "readline", "repl", "vm", "child_process", "assert", "zlib", "tty", "os", "querystring"

how to obtain list of all available Node.js modules

- https://stackoverflow.com/questions/35725976/how-to-obtain-a-list-of-all-available-node-js-modules

export-require

https://stackoverflow.com/questions/38660022/curly-brackets-in-node-require-statement

For example (functions.js):

```
module.exports = {
   func1,
   func2
}
```

is included in your file:

```
const { func1, func2 } = require('./functions')
```

Now you can call them individually,

```
func1()
func2()
```

as opposed to:

```
const Functions = require('./functions')
```

that are called using dot notation:

```
Functions.func1()
Functions.func2()
```

npm init

- Initialize NPM on a new project

Use your Command Line and navigate to the root folder of your project and enter

```
> npm init
```

This command will ask you some questions to generate a package.jsonfile in your project route that describes all the dependencies of your project. This file will be updated when adding further dependencies during the development process, for example when you set up your build system.

beefy

- https://www.npmjs.com/package/beefy

a local development server designed to work with browserify.

```
> npm install -g beefy
> npm install -g browserify
```

package managers

- Setup Dependency Managers

Usually I use three different package managers. Bower is used to manage frontend dependencies like jQuery plugins and third party libraries, Composer keeps track on PHP dependencies and NPM (Node Package Manger) is our tool to manage the dependencies for Node.js which will be used to setup our build automation.

electron app packager

- https://github.com/electron-userland/electron-packager

This module requires Node.js 4.0 or higher to run.

for use in npm scripts

```
> npm install electron-packager --save-dev
```

for use from cli

```
> npm install electron-packager -g
```

Note: You need the -g flag to install it globally which makes the command electron-packager available in your PATH. If you do not do a global install, you will need to run ./node_modules/electron-packager/cli.js instead.

gulp

Setup Build System

local-web-server package

- https://www.npmjs.com/package/local-web-server

package.json properties

- https://nodesource.com/blog/the-basics-of-package-json-in-node-js-and-npm/

npm - list installed packages

e.g. Protractor Official Tutorial

```
npm list -g --depth=0
webdriver-manager update
```

- geckodriver v0.18.0
- chromedriver 2.31

- node 6.10.2
- npm 3.10.10
- browser-sync 2.18.13
- protractor 2.18.13
- jasmine 2.7.0
- jasmine-core 2.7.0

npm - list locally installed packages

e.g. Protractor DEMO from github

```
> cd protractor-demo
> npm list --depth=0
> webdriver-manager update (local from node_modules/.bin)
```

- selenium-server-standalone 2.45.0
- chromedriver 2.15

- express 3.4.8
- fire-fox profile 0.3.4
- mkdirp 3.5
- protractor 2.1.0
- q 1.0.0

How to use locally installed package

https://stackoverflow.com/questions/9679932/how-to-use-package-installed-locally-in-node-modules

where does npm install packages

https://stackoverflow.com/questions/5926672/where-does-npm-install-packages

What is Node.js

https://stackoverflow.com/questions/1884724/what-is-node-js

Overview

- Node.js allows the creation of Web servers and networking tools using JavaScript and a collection of "modules" that handle various core functionality. Modules are provided for file system I/O, networking (DNS, HTTP, TCP, TLS/SSL, or UDP), binary data (buffers), cryptography functions, data streams and other core functions. Node.js's modules use an API designed to reduce the complexity of writing server applications.
- Node.js applications can run on Linux, macOS, Microsoft Windows, NonStop, and Unix servers. They can alternatively be written with CoffeeScript (a JavaScript alternative), Dart or Microsoft TypeScript (strongly typed forms of JavaScript), or any other language that can compile to JavaScript.
- Node.js is primarily used to build network programs such as Web servers. The biggest difference between Node.js and PHP is that most functions in PHP block until completion (commands execute only after previous commands have completed), while functions in Node.js are designed to be non-blocking (commands execute in parallel, and use callbacks to signal completion or failure).

Platform architecture

- Node.js brings event-driven programming to web servers, enabling development of fast web servers in JavaScript. Node.js connects the ease of a scripting language (JavaScript) with the power of Unix network programming.
- Node.js was built on the Google V8 JavaScript engine since it was open-sourced under the BSD license, extremely fast, and proficient with internet fundamentals like HTTP, DNS, TCP. Also, JavaScript was a well-known language, making Node.js immediately accessible to the entire web development community.

Industry support

- The open-source community has developed server frameworks to accelerate the development of applications. Such frameworks include **Connect, Express.js, Socket.IO, Koa.js, Hapi.js, Sails.js, Meteor, Derby,** and many others.
- Modern desktop IDEs provide editing and debugging features specifically for Node.js applications. Such IDEs include **Atom, Brackets, JetBrains WebStorm, Microsoft Visual Studio (with Node.js Tools for Visual Studio, or TypeScript with Node definitions), NetBeans, Nodeclipse Enide Studio (Eclipse-based) and Visual Studio Code**.
- Certain online web-based IDEs also support Node.js, such as Codeanywhere, Codenvy, Cloud9 IDE, Koding and the visual flow editor in Node-RED.

Node.js resources

- https://www.npmjs.com/package/conda

- Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine: conda install -c javascript nodejs=4.4.1
- https://anaconda.org/javascript/nodejs

Node.js & Protractor

- Node.js is an open source runtime environment to develop applications in JavaScript. WebDriverJS has been written in Node.js and Protractor has been written on top of WebDriverJS. Node.js will be executing the javascript code you've written in your spec and po files with its JavaScript Engine (Google V8).
- Protractor is a Node.js application... hence the Node requirement. Protractor is written in Javascript, so it requires a Javascript engine, which Node provides (Google V8), thus allowing it to run on a server.
- Node.js is package file we are using in protractor automation tool to run the angular.js based application.It contains selenium and other browser drivers to run our applications in different environments.

Anaconda JavaScript

- https://anaconda.org/javascript/repo
- https://anaconda.org/search?q=javascript
- https://anaconda.org/nbcio/typescript

Node.JS vs io.JS

- http://anandmanisankar.com/posts/nodejs-iojs-why-the-fork/

Difference JavaScript and Node.JS

- https://www.quora.com/What-is-the-difference-between-JavaScript-and-Node-js

Javascript is a programming language for writing code.

Node.js is not a programming language; it is rather an interpreter of Javascript code (due to the Google Chrome V8 engine) to the operating system installed on the server.

Node.js provides a purely evented, non-blocking infrastructure to script highly concurrent programs.

Javascript is designed specifically to be used with an event loop: anonymous functions, closures; only one callback at a time; I/O through DOM event callbacks. The culture of JavaScript is already geared towards evented programming.

- Java --> JRE --> JVM
- JavaScript --> Node --> V8

TypeScript with Node.JS

https://basarat.gitbooks.io/typescript/content/docs/quick/nodejs.html

IJavaScript

https://www.npmjs.com/package/ijavascript

- IJavascript is an npm package that implements a Javascript kernel for the Jupyter notebook (formerly known as IPython notebook). A Jupyter notebook combines the creation of rich-text documents (including equations, graphs and videos) with the execution of code in a number of programming languages.
- The execution of code is carried out by means of a kernel that implements the IPython/Jupyter messaging protocol. There are kernels available for Python, Julia, Ruby, Haskell and many other languages.

Easy to install & start

```
> npm install -g ijavascript
> ijs
```

IJavaScript Examples

- http://n-riesco.github.io/ijavascript/

ITypeScript

https://github.com/nearbydelta/itypescript

Easy to install & start

```
> npm install -g itypescript
> its
```

ITypeScript Examples

- https://github.com/nearbydelta/itypescript/blob/master/README.md
- https://github.com/nearbydelta/itypescript/blob/master/doc/hello.ipynb

Jupyter NodeJS¶

- https://github.com/notablemind/jupyter-nodejs

Protractor Test

http://www.protractortest.org/#/tutorial

- Protractor is an end-to-end test framework for AngularJS applications built on top of webdriverJS. Protractor can be run as a standalone binary runner or included into your tests as a library.

Protractor package

https://www.npmjs.com/package/protractor

- Protractor is an end-to-end test framework for Angular and AngularJS applications. Protractor is a Node.js program built on top of WebDriverJS. Protractor runs tests against your application running in a real browser, interacting with it as a user would.

Protractor Tutorial

http://www.protractortest.org/#/toc

- Protractor is a Node.js program. To run, you will need to have Node.js installed. You will download Protractor package using npm, which comes with Node.js. Check the version of Node.js you have by running node --version. Then, check the compatibility notes in the Protractor README to make sure your version of Node.js is compatible with Protractor.
- By default, Protractor uses the Jasmine test framework for its testing interface. This tutorial assumes some familiarity with Jasmine, and we will use version 2.4.
- This tutorial will set up a test using a local standalone Selenium Server to control browsers. You will need to have the Java Development Kit (JDK) installed to run the standalone Selenium Server. Check this by running java -version from the command line.

Protractor FAQ

https://github.com/angular/protractor/blob/master/docs/faq.md

Protractor debugging questions

https://stackoverflow.com/questions/tagged/protractor

Angular Discussion group

https://groups.google.com/forum/?fromgroups#!forum/angular

Angular Coursera

- https://www.coursera.org/learn/angular-js

Angular Gitter Chatroom

- https://gitter.im/angular/protractor

Angular Application Deployment

- https://stackoverflow.com/questions/41504330/packaging-angular-2-app-for-distribution-to-a-production-iis-server

Angular vs Angular2

https://www.quora.com/What-is-the-difference-between-AngularJs-and-Angular-2

Tutorials

- Angular2
- Node.js for beginners

Compare Packages

- https://stackoverflow.com/questions/35062852/npm-vs-bower-vs-browserify-vs-gulp-vs-grunt-vs-webpack
- https://hackernoon.com/how-it-feels-to-learn-javascript-in-2016-d3a717dd577f

Grunt

https://gruntjs.com/getting-started

Issues Grunt command not recognized

```
> npm install -g grunt-cli
```

Optional - Node vs PHP popularity

https://belitsoft.com/php-development-services/php7-vs-nodejs