**NPM**

Package manager for JavaScript -- Express, Gulp, Grunt …

Most commonly used for installing packages for Node.JS

NPM: Install, Update and Uninstall packages (bundled up SW – mainly JS, can include images, HTML, Coffee and CSS)

**Packages** for Node.JS projects are called **modules** (Express, Passport…)

npm Package Examples

* [Express](http://expressjs.com/)
* [Passport](http://passportjs.org/)
* [jQuery Plugins](https://www.npmjs.com/browse/keyword/jquery-plugin)
  + [tipso](https://www.npmjs.com/package/tipso)
* [Grunt](http://gruntjs.com/)
* [gulp.js](http://gulpjs.com/)
* [CoffeeScript](http://coffeescript.org/)
* [TypeScript](http://www.typescriptlang.org/)
* [Sass](http://sass-lang.com/)
* [Cordova](http://cordova.apache.org/)
* [PhoneGap](http://phonegap.com/)
* [Ionic](http://ionicframework.com/)
* [johnny-five](http://johnny-five.io/)
* [Espruino](http://www.espruino.com/)
* [Tessel](https://tessel.io/)

**Make sure you have Node and NPM installed by running simple commands to see what version of each is installed:**

* **Test Node.** To see if Node is installed, open the Windows Command Prompt, Powershell or a similar command line tool, and type node -v. This should print the version number so you’ll see something like this v0.10.35.
* **Test NPM.** To see if NPM is installed, type npm -v in Terminal. This should print the version number so you’ll see something like this 1.4.28
* **Create a test file and run it.** A simple way to test that node.js works is to create a simple JavaScript file: name it hello.js, and just add the code console.log('Node is installed!');. To run the code simply open your command line program, navigate to the folder where you save the file and type node hello.js. This will start Node.js and run the code in the hello.js file. You should see the output Node is installed!.

**NPM Command Line:**

> npm -- Shows all possible commands

Ex. Clear – clear out command line

Ex. install

> npm install –h (Help for the command)

For latest up to date version (do not need to include version number)

> npm install bcrypt (to get the latest version of bcrypt)

NPM Creates a folder called: **node\_modules**

**Include the npm code in your JS:**

Use the **require()** function with the name of the module to include in the **app.js** file:

var x = require(‘bcrypt’);

package.json -- provide file path to the package code. In key: “main”

“main”: “./bcrypt”,

**Global Package -- always available packages for any project**

> npm install http-server –g (simple Web server)

**Remember: all packages are stored in the ‘node\_modules’ folder**

**package.json : Keep Track -- Manage Package Dependencies -- keep packages up to date and**

package.json -- npm uses this to manage package dependencies [File in the ROOT of the project]

CREATE:

> npm init (to create the package.json file)

> npm install package name --save to install a package and save it as a dependency to package.json

> npm init

name: (name of folder currently in) ‘enter name of project here’

version: (1.0.0)

description:

entry point:

test command:

git repository:

keywords:

author:

license:

>>>> Creates the package.json file

**Add MORE NPM module to package.json: -- save**

> npm install colors --save

Add colors to the app.js file:

var colors = require(‘colors’);

**Add DEV DEPENDENCY: (Mocha Test Framework)**

> npm install mocha --save-dev

Requires ‘test’ directory

> npm test

**Set Development Environment**

Include environment variables before command

> NODE\_ENV=production npm install >>> only installs the dependencies needed to run, not the dev dependencies

**Updating Packages**

> npm install (installs depending on the symbols in package.json) ^update to latest minor

~update to latest patch

> npm outdated (overview of what is out of date)

> npm update (local up to date)

> npm update http-server –g (global package update)

Remember there is no package.json file for global package >> may go to Major new release

**To update NPM itself**

> npm update npm –g

**Uninstall**

> npm uninstall packagename >> removes from node\_modules folder

> npm install >> reinstall what is in package.json

Permanent Uninstall:

> npm uninstall packagename --save >> removed from node\_modules and package.json

Uninstall development dependency:

> npm uninstall mocha --save-dev

Uninstall a global package:

> npm uninstall packagename -g