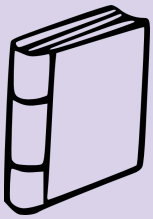
Coding & Cocktails Session 12:

Intro to NPM (Node Package Manager)

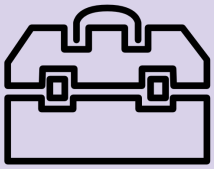
Overview

Tonight's objective is to understand the purpose of NPM and become comfortable using it.



Don't know a term?

Check out our glossary
bit.ly/CnCgloss



First time at C&C ?

Install recommended tools
bit.ly/CnCTools

Project

Dependencies are references to code that someone else wrote. People publish this code by creating node packages. Node packages are open-source, which means you don't have to pay to use it. Most websites and apps are built with some kind of dependencies.

The mission tonight is to create a simple project and add dependencies to it with NPM. We'll use each new dependency in some small way.

Part 0: Verify NodeJS is installed

In order to use NPM, you have to install NodeJS. (We won't be writing a NodeJS application, but the two are essentially installed together.)

1. Open your terminal / command line

Helpful tip:

The "terminal" and "command line" (aka CLI, command line interface) are the same thing.

- On Windows, this is **Git Bash**
- On Mac, this is **iTerm2**

2. Type: **node -v**
3. If you get a version number, you're good. Proceed to Part 1.

Command not found

If your system doesn't recognize the node command, it's probably not installed. You can get it from **http://nodejs.org**

Part 1: Create a project folder

When you're starting from scratch, an empty project is simply an empty folder. You could do that visually (using your file explorer) but let's do it on the command line.

Command line woes?

Revisit the command line worksheet from March:
bit.ly/cmdln

1. Navigate to your CodingAndCocktails folder: **cd <yourHomeDirectory>/CodingAndCocktails**

Helpful tips:

- Your home directory is:
  - Mac: /users/<yourUsername>
  - Windows: C:/Users/<yourUsername>
- Command to change folders: cd <folderToGoTo>
- Command to make a folder: mkdir <newFolder>
- Most command line applications are not case sensitive, but a few of them are!

2. Make a new folder called packagesproject: **mkdir packagesproject**
3. Move into that new folder: **cd packagesproject**

Helpful tip:

- If you start typing the name of a folder or file, hit tab and it will autocomplete

4. Open the Sublime text editor from here: **subl .**

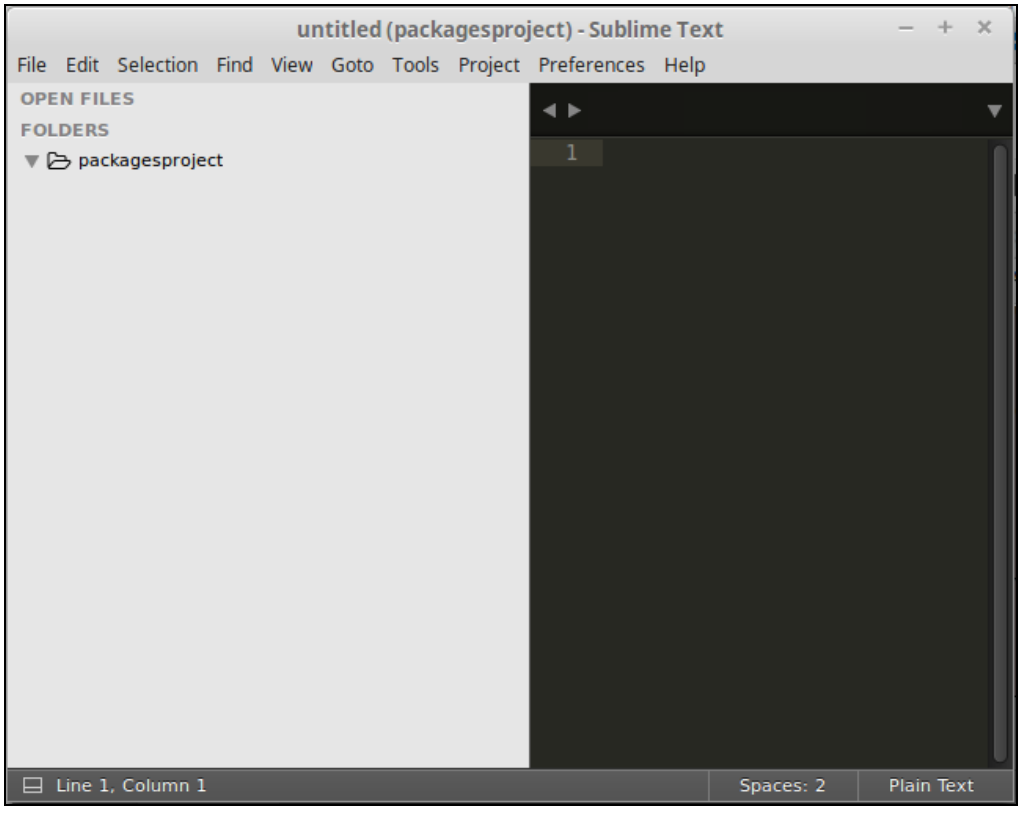
Command not found

If Sublime doesn't open (but you know it's installed), your system probably just doesn't recognize the sublime command.

You can configure your system to recognize the command (ask a mentor for help) or you can follow these steps to open it manually:

1. From your applications or start menu, open Sublime
2. In Sublime: **File -> Open Folder…**
3. Open the **packagesproject** folder
4. Click the **Open** button

Your Sublime application should look like this:



Part 2: Create the project files

All of these steps are performed inside Sublime.

1. Right click on the packagesproject folder and choose **New File**
2. Save the file (**Ctrl+S** or **Cmd+S**). This will prompt you for a file name.
3. Enter **index.html** and click the **Save** button.
4. Enter the following HTML and then save again:

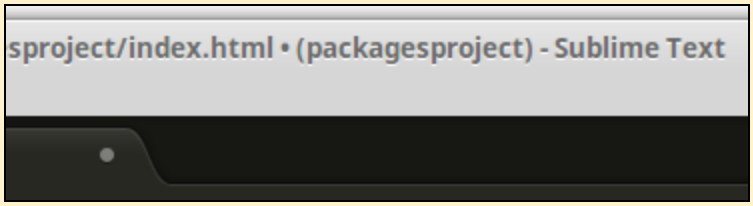
```
<!DOCTYPE html>
<html>
```

```
  <head>
    <link rel="stylesheet" type="text/css" href="main.css">
  </head>
  <body>
    <h2>
      NPM is awesome!
    </h2>
    <h4>
    </h4>
    <script src="bundle.js"></script>
  </body>
</html>
```

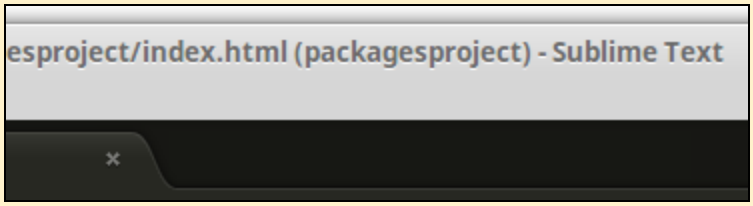
Helpful tip:

In Sublime, if a file is has been modified but not saved, the title bar and tab show a dot. Often, the answer to "Why isn't my code working?" is "Oh I didn't save the file".

Unsaved:



Saved:

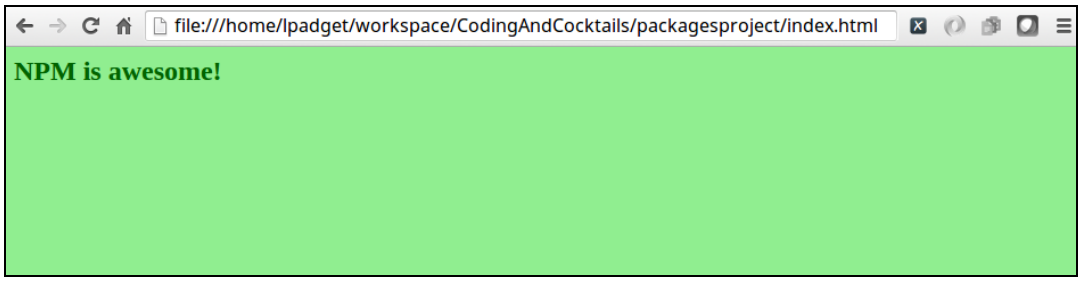


5. Repeat steps 1-3 to create a file called **main.css**
6. Enter the following CSS and then save again:

```
body {
    background-color: lightgreen;
}
h1, h2, h3, h4, h5, h6 {
  color: darkgreen;
}
```

7. Repeat steps 1-3 to create a file called **index.js** (it remains empty)
8. Now in your browser, navigate to your index.html file:
   file:///<yourHomeDirectory>/CodingAndCocktails/packagesproject/index.html

Your rendered HTML file should look like this:



Part 3: Create a package.json file

NPM provides an easy way to create a package.json file, which is where you specify the dependencies you want.

1. On the command line, type: **npm init** This will give you a series of prompted questions.
2. You can hit enter for each and it will use the default values, which is fine.
3. When you're done, use Sublime to open the package.json file it created.

Nice work!

Take a break and grab another drink. You've earned it!

Part 4: Install Browserify (Globally)

Browserify will allow you to easily consume new dependencies with the javascript **require** statement.

1. On the command line, type: **sudo npm install -g browserify**
2. It will show a progress bar (of sorts). If it prompts you for a password, enter it.

Sudo warnings & passwords

The **sudo** prefix is like running an application as "root" or "administrator". On a mac, you might get a warning like "Improper use of the sudo command could lead to data loss…" We're not doing anything dangerous here, just installing the browserify tool via npm, which requires this level of system access.

Also, when you type your password on the command line, you won't be able to see it. This is normal. Just type it as you would normally and press enter.

3. When it's done, it should look something like this:

```
~/workspace/CodingAndCocktails/packagesproject $ sudo npm install -g browserify
/usr/local/bin/browserify -> /usr/local/lib/node_modules/browserify/bin/cmd.js
/usr/local/lib
└── browserify@13.1.0

~/workspace/CodingAndCocktails/packagesproject $
```

Part 5: Install a dependency via command line

Just like the init command, NPM provides an easy way to add a dependency via command line, too. In this part, we'll add a dependency on **lodash** using this method.

Helpful tip:

Lodash is a library that helps you manipulate JSON objects in javascript. This can be very handy if you want to:

- sort an array (alphabetically, or in some other way)
- replace words in a string
- Generate a random number

...and much, much more!

Lodash - along with hundreds of other libraries - are available via NPM. The URL for this is: **https://www.npmjs.com/package/lodash**

This is the "home" of the library on NPM, where you can find links to the author(s), documentation, support, etc. Try substituting the last part with another library, such as **jquery**.

1. On the command line, type: **npm install --save lodash**

NPM warnings

This command will warn that you don't have a description or repository filled out in your package.json file. You don't need to worry about this right now.

2. When it's done, notice the new dependency is listed in the package.json file. It also downloaded the lodash library to a folder called node_modules.

Part 6: Use the new dependency

Now we're ready to put the new dependency (lodash) to good use.

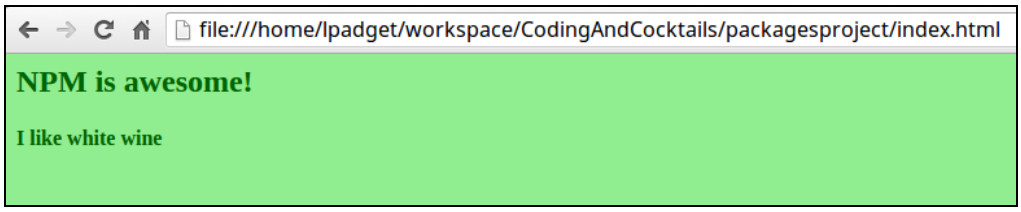
1. In Sublime, copy or type the following into **index.js**, then save again:

```
var _ = require('lodash');
var wine = "I like red wine"

var h4Heading = document.querySelector('h4');
h4Heading.textContent = _.replace(wine, 'red', 'white');
```

2. On the command line, type: **browserify index.js > bundle.js**
3. Refresh (or open again) index.html in your browser.

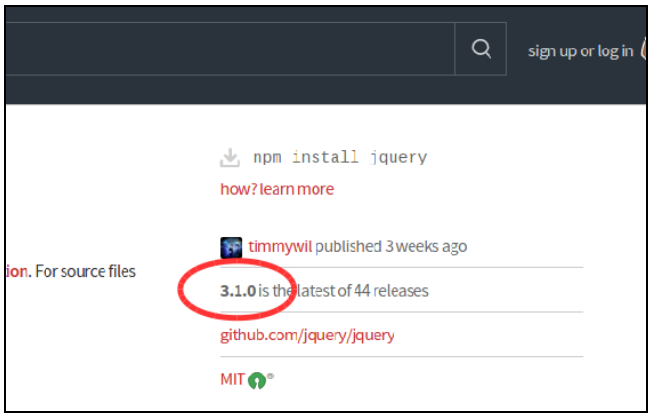
Your rendered HTML file should look like this:



Part 7: Install a dependency via package.json directly

You can also add a dependency to the package.json directly. In this part, we'll add a dependency on **jquery** using this method.

1. In Sublime, open the **package.json** file.
2. In your browser, go to: https://www.npmjs.com/package/jquery
3. Notice the latest version of jQuery in the upper right corner:



4. In the dependencies section, add a comma after the lodash line, and then the following. Then save again:

```
"jquery": "^3.1.0"
```

> **Helpful tip:**
>
> For more information about the format of these versions (i.e. "^3.1.0"), checkout NPM documentation here:
>
> https://docs.npmjs.com/getting-started/semantic-versioning

5. On the command line, type: **npm install**
6. This will install any dependencies you just added to the package.json file.

Homework

The more you practice, the better you'll get. Reinforce what you've learned tonight with the following tutorial.

Hey Slacker!

Remember, we're here to help.
Join the KCWiT C&C Slack Channel:
kcwit.slack.com

Part 1: Use the new jQuery dependency

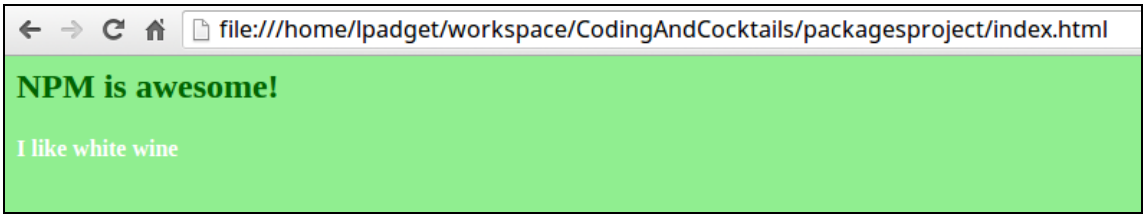
Your index.js file already contains one way of selecting an element (document.querySelector). Now that you have jQuery, you can do this another way:

1. In Sublime, open the **index.js** file and place the following code anywhere in the file:

```
var $ = require('jquery');
$("h4").css("color", "white");
```

2. On the command line, type the browserify command again: **browserify index.js > bundle.js**
3. Refresh (or reopen) index.html in your browser.

Your rendered HTML file should look like this:



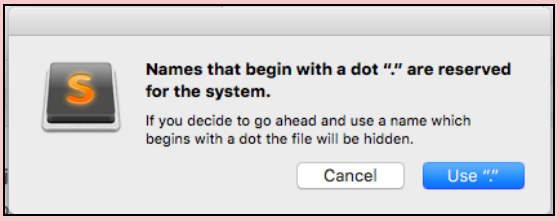
Part 2: Initialize your project as a git repository

Start tracking your progress with git. We're going to tell git what it should *and should not* care about.

1. On the command line, type the following: **git init**
2. Now type: **git status** This command lists the files and folders git thinks it should track.
3. In Sublime, create a new file called **.gitignore** (note, this file starts with a dot). See Part 2 of the project section above if you need help creating new files.

Files that begin with a dot

You may see a warning that file names that begin with a dot are reserved for the system. This is ok when we're talking about certain kinds of files, and **.gitignore** is one of them. Click **Use "."**

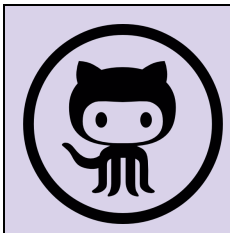


4. Add the following to the **.gitignore** file and save it:

```
/node_modules
bundle.js

#For macs
.DS_Store
```

5. Back on the command line, type this again: **git status** Notice that the node_modules folder and bundle.js file are no longer listed.
6. Stage and commit these files.



Don't remember git?

See the git version control worksheet from April:
bit.ly/CnCvers

Part 3: Continue to discover new dependencies and try them out!

The world is your oyster! Check out all of the packages available at **http://www.npmjs.com**.