Outline

Checkpoint 7

Installing packages

One of the most wonderful aspects of the web development community is the amount of open-source content that's available. When something is *open source*, it means that someone has built a utility that can be viewed, downloaded, and edited by anyone else.

The JavaScript community has created a number of packages that you can download and use. These packages are built with JavaScript. They may be useful additions to your JavaScript projects, or they may help you perform particular tasks in your development environment. The npm tool can help you install and manage these packages.

By the end of this checkpoint, you will be able to do the following:

- Install packages from the web into your local JavaScript project

Viewing packages

There are thousands of packages available for you to use, some of which are better than others. All packages can be found on **the npm** website through the search bar.

In this program, you will learn how to use certain popular packages. However, it is important to keep in mind that there are always more packages out there.

Moment

Take a look at the Moment package. This page shows you information about how to install the package, how it can be used, and various statistics on the package's usage. For example, this package describes itself as "a lightweight JavaScript date library for parsing, validating, manipulating, and formatting dates."

You also can see that this package has been downloaded often and has been updated recently. When you're assessing whether or not to use a package, it is a good idea to look at these statistics.

Reading documentation

You learned that the Moment package can help parse and format dates in JavaScript. How does it do this? To answer this question, you will need to read the documentation.

So far, you have likely only seen documentation on the official JavaScript language from the Mozilla Developer Network. MDN is a great resource, but it doesn't contain documentation on packages.

Now, you will need to look instead at the specific package and its documentation. Thankfully, Moment has fantastic documentatio[illegible]hich can be found at Moment's website.

Reading documentation is its own skill, and it can be challenging at first. As you come across phrases and terms that you don't understand, you will need to be patient and look up new concepts.

Installing

You can install a package with a simple npm command:

```
npm install <package-name>
```

Running this command (while changing `<package-name>` to the name of the package) will do the following:

- If a `node_modules/` folder *doesn't* already exist, a `node_modules/` folder will be created in the current directory.
- If a `package-lock.json` file *doesn't* already exist, a `package-lock.json` file will be created in the current directory.
- If a `package.json` file *does* already exist, the package and its version number will be recorded in the `package.json` file.
- The package will be downloaded from the web to your computer.

In general, you always want to have a `package.json` file before running any `install` commands.

Note: You can use `npm i` as a shorthand for `npm install`.

Do this

Create a JavaScript project

Outline

First, complete the following steps:

1. Create a new directory called `installing-packages`.
2. Then, create a `package.json` file and a `main.js` file.
3. In the `package.json` file, create a `start` script that runs your `main.js` file with Node.
4. Finally, include the following in your `main.js` file:

```
console.log("The current date is: " + Date.now()
```

When you run your script, you will see something like the following show in your terminal:

```
The current date is: 1592329893846
```

Install the Moment package

The above date is not very useful! Instead, try using the Moment package. Install Moment using the following command:

```
npm install moment
```

Then, require the Moment package in your `main.js` file:

```
let moment = require("moment");
```

Outline

Run your code and confirm that the output looks the same as it did before.

Use Moment

Now it is time to actually use Moment. Before looking at the solution below, try reading the Moment documentation yourself, and look for how you might display a more human-readable date format.

If you're having a hard time, try looking at the display format section of the documentation. As you can see, there are actually several ways to display dates with Moment.

If you're still having trouble, you can replace your `console.log()` statement with the following:

```
console.log("The current date is: " + moment().format
```

Dependencies

Whenever you install a package with `npm install` or `npm i`, the installation will be recorded in your `package.json` file under a key titled `dependencies`. It will look something like this:

```
"dependencies": {
  "moment": "^2.26.0"
}
```

The string value next to the package name is the version numb
that package. If you are interested in the version number syntax, you

may optionally read more about it in the package.json dependencies documentation.

Developer dependencies

In the future, you may also see a command like this:

```
npm install <package-name> --save-dev
```

The `--save-dev` flag will be included for packages that are necessary only for the development of the project, not for running it. Packages installed with that flag will show up under the `devDependencies` key instead of `dependencies`.

Outline

The reason behind this will be explained whenever it is asked of you. For now, just keep in mind that you may see this flag.

The `package-lock.json` file

The `package-lock.json` file is automatically created whenever you install a package. Similarly to `package.json`, `package-lock.json` is a configuration file.

The difference is that you will never need to edit this file. The `package-lock.json` file is essentially metadata for `package.json`. Although it is required, it does not require any management by you.

You can read more about the `package-lock.json` file in the package-lock.json documentation.

Node modules

When you install a package, that package's code will be added to a `node_modules` folder. Modules that aren't created by you or installed in Node will be contained in this folder.

As you'll see, the installed packages also have a `package.json` file. If the installed package has its own dependencies, those packages will also be installed.

You can delete the `node_modules` folder at any time and reinstall the packages by using the `npm install` command without providing a package name. By default, `npm install` will install everything listed in your `package.json` file.

Outline

Do this

Browse the `node_modules/` folder

Take a look inside the `node_modules/` folder inside your `installing-packages/` directory. You will see a single folder: `moment/`.

Look around the folder briefly, and try to find the `package.json` file. As you can see, Moment is just another Node project, like the one that you have!

Complete example

A completed example from this checkpoint can be found here:

- JavaScript on your machine: Installing packages

The assignment below includes a quiz to test your knowledge. Here is the answer key; feel free to use it to check your answers.

Checkpoint

This checkpoint will be autograded. Please click the link below to open your assignment in a new tab. Once you complete the assignment, you will see a button allowing you to submit your answers and move on to the next checkpoint.

Your work

03.18.21 **Approved**

Outline

Completed

Next checkpoint

How would you rate this content?

Report a typo or other issue

Go to Overview

Outline