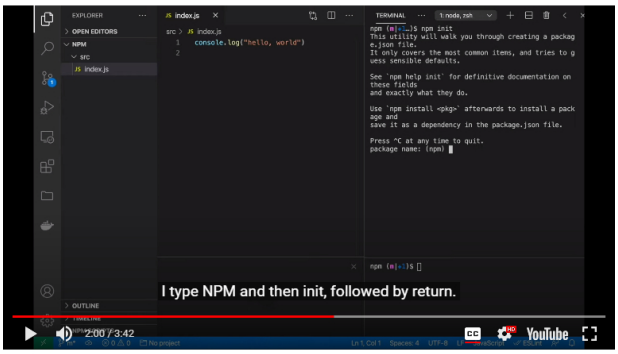
What Is Node Package Manager (NPM)?



NPM

NPM is both a **tool for managing project dependencies** via command line and a **website hosting more than 1 million third-party packages** that can be used for your project.

- Modules are shared as packages
- Packages extend the functionality of your app
- Modules are stored in the app's `node_modules` folder
- Core modules include `path`, `Filesystem`, and more

Initializing `npm` and Creating a `package.json` file

Initializing npm will create a package.json within the root of your application folder containing general information about the project.

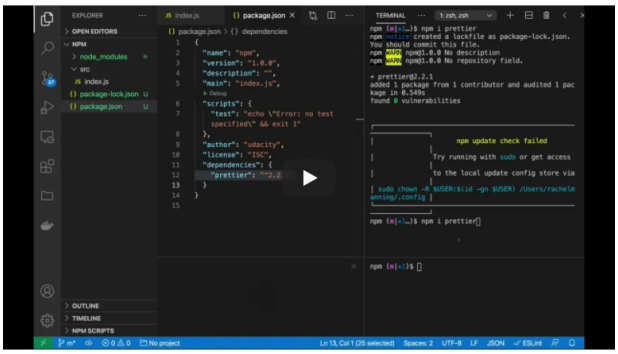
To initialize `npm` and go through all of the settings use:

```
$ npm init
```

To automatically select all defaults use `-y`

```
$ npm init -y
```

Adding Dependencies



Applications will either include both dependencies and devDependencies or just dependencies. It is dependent on the team setting up the project. `devDependencies` are thought of as dependencies that are only necessary for development whereas `dependencies` are those dependencies used in both development and production. An example would be needing TypeScript added as a dependency for development, but since it compiles to standard JavaScript to be used in production, TypeScript is not needed for production and therefore could be just a `devDependency`. Many teams find little use in separating but when learning, it can be a helpful practice to determine which dependencies are only being used in development vs which are also needed for production.

```
$ npm i module-name // install module to dependencies
$ npm i --save-dev module-name // install to dev dependencies
$ npm i --save-dev module-name@1.19 // install a specific version (1.19 here) of module
```

Installing dependencies adds the dependency to your package.json file in the format:

```
  "devDependencies": {
    "prettier": "^2.2.1"
  }
```

Pay special attention to the version listed. The format is as follows.

- First number = major version
- Second number = minor release
- Third number = patch

The version states what was installed, but it also clarifies how it can be updated should you remove the node_modules and package-lock.json files and reinstall all dependencies with `$ npm install`.

The additional included characters (or lack thereof) tell npm how to maintain your dependencies.

- `*` means that you'll accept all updates
- `^` means that you'll only accept minor releases

- `~` means that you'll only accept patch releases
- `>`, `>=`, `<=`, `<` are also valid for saying you'll accept versions greater/less/equal to the listed version
- `||` allows you to combine instructions `"prettier": "2.2.1 || >2.2.1 < 3.0.0"` which says use prettier greater than 2.2.1 and less than version 3.0.0
- You can also leave off a prefix and only accept the listed version

package-lock.json

`package-lock.json` contains all of the information for the dependencies of the modules you have installed.

It is best practice to add `package-lock.json` as well as `/node_modules` to your `.gitignore` file when using a repository. The `node_modules` folder can grow rapidly, containing thousands of files. It is best to clone a repository without `node_modules` and run `npm install` to reinstall all dependencies of the project directly from npm.

npm update

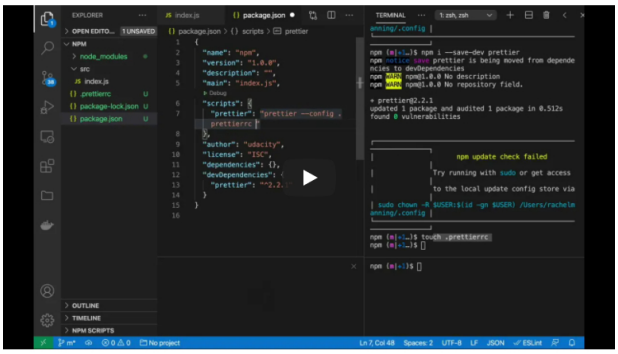
running `$ npm update` will update all of your dependencies based on the specifications given in your package.json file.

scripts

To run a script that you have added to your package.json file, simply run $ npm run argument with the name of the script as the argument.

```
$ npm run prettier
```

Using Prettier



Prettier is a code formatter that will ensure you're keeping your code consistent. It's commonly added to projects to ensure all members on a team are formatting in a consistent way such as always using semicolons, trailing commas, and single quotes. It can be configured to the preferred settings of the team and works well with additional tools like linting.

We are able to add it to a project with NPM by doing the following:

- Locate prettier on npmjs.com to get the install script and other information.
- Run the install script `npm i --save-dev prettier`.
- Add a prettier script to your `package.json` file. The script you choose can vary dramatically depending on the project. The one below will only overwrite files located in the src directory that are js files. You may need a **different script** depending on the project.

```
// example config file, path structure to check, and write fixes
"prettier": "prettier --config .prettierrc 'src/**/*.js' --write"
// or
"prettier": "prettier --config .prettierrc \"src/**/*.js\" --write"
```

- Create a `.prettierrc` file for any custom configurations.
- Run `npm run prettier` to run prettier (or whatever you named your script).

> NOTE: It's common to encounter deprecation warnings when working with NPM packages. Packages may have multiple dependencies. If one updates before the other, you may encounter one of these warnings. They are typically take care of within the next 2 updates of the package. It's best to look them up when you find them to see if someone is actively working to repare the issue or to see is a better solution.

New Terms

Term	Definition
dependencies	Dependencies used in both development and production
devDependencies	Dependencies that are only necessary for development
Node Package Manager (npm)	A tool for managing project dependencies via command line as well as a website hosting more than 1 million third-party packages that can be used for your project
package-lock.json	A JSON file that contains all of the information for the dependencies of an app's installed modules
package.json	A JSON file that acts as a manifest for your project including name, author, version, description, license, dependencies, scripts, etc.
Prettier	A code formatting package that can be integrated into projects to improve code consistency and readability

Further reading

Check out the **full documentation** on **Prettier** to see how it's capable of improving your projects.

NEXT