香港中文大學
The Chinese University of Hong Kong

CSCI2720 - Building Web Applications

Lecture 11: Package Managers

Dr Colin Tsang

Outline

- Packages and libraries
- Package managers: npm, yarn, pnpm
- npm
- npx
- create-react-app

Packages and Libraries

- It is possible for web development to be without any external tools, yet…
  - Are you willing to write all the HTML, CSS, JS code directly?
- Modern development: incorporating building blocks for efficiency and convenience
  - Packages, libraries, frameworks, …
  - E.g., bootstrap for CSS , jQuery for JS, React for apps, …
  - Engineers are lazy …….
- An important issue: *dependency*
  - Do different versions of things work together?

Package manager

- Finding correct package JS files
- Checking to ensure absence of *vulnerabilities*
- Downloading, managing, and putting them in *proper locations*
- Including packages in the development application
- Handling *sub-dependencies*
- *Clearing file* for unnecessary packages
- See: https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Understanding_client-side_tools/Package_management

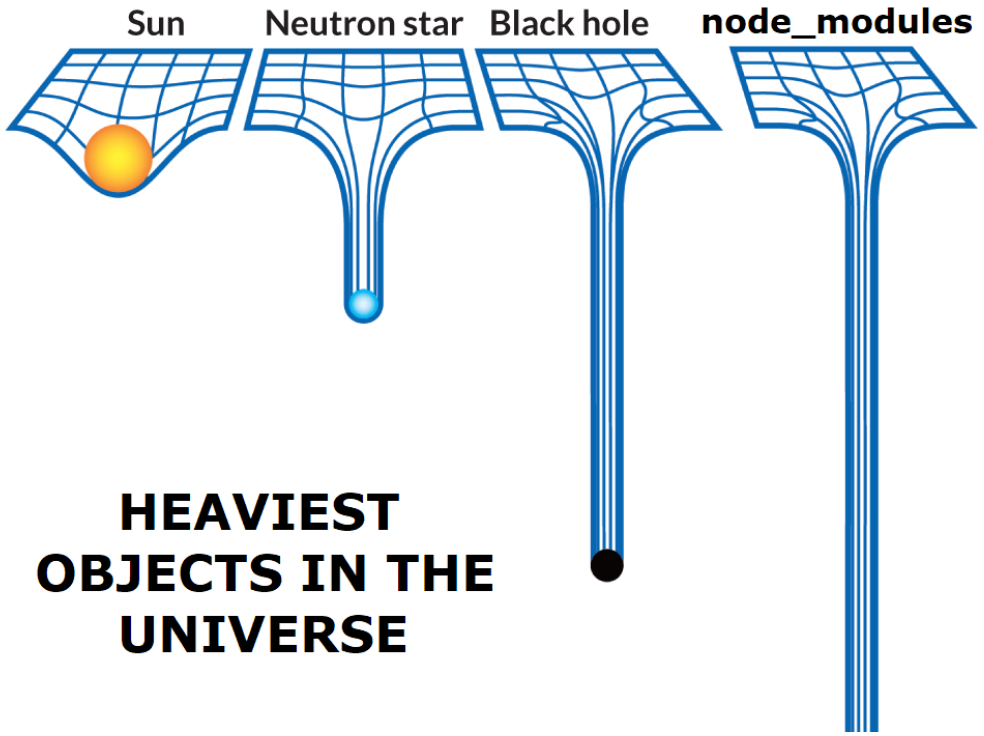
NPM, YARN, PNPM, …

- Generally, the package managers are executed in the command line interface
  - More sophisticated option settings
  - More convenient for scripting and automation
- npm: https://www.npmjs.com/
- Yarn: https://yarnpkg.com/
- pnpm: https://pnpm.js.org/
- Which one to use? Many comparisons on internet…
  - https://blog.logrocket.com/javascript-package-managers-compared/

NPM

- Part of *Node.js*, originally as the *"Node Package Manager"*.
  - Since January 2010
- To obtain npm, download for your platform with *Node.js*
  - https://nodejs.org/en/download/
- Local packages: in a folder **node_modules** in project
- Global packages: in system folders if you have admin rights
- Most packages are working on the *Node.js* environment for backend development
  - We will use Node.js in the later part of the course.

node_modules memes ……



- See: https://tsh.io/blog/reduce-node-modules-for-better-performance/
- In assignments and project, do not submit the **node_modules** folder.

NPM commands

- Initializing a project
  - **npm init**
- Installing a local package
  - **npm install {package name}**
- Installing a global package (admin rights required)
  - **npm install –g {package name}**
- Installing all packages as defined in *package-lock.json*
  - **npm install**

NPX

- While npm only manages the packages, a supplementary tool **npx** allows you to:
  - execute Node packages without installing
  - E.g., npx create-react-app my-app

- See more: https://blog.npmjs.org/post/162869356040/introducing-npx-an-npm-package-runner

CREATE-REACT-APP

- A very common way to prepare a React app (which you find in a lot of tutorials) is to use **create-react-app**
  - After installing *Node.js* and *npm*, only this command is needed to build the skeleton of a simple React app
    - **npx create-react-app theAppName**
  - Then, this command will transpile the code, build the app, and start a web server on the local machine in development mode
    - **npm start**
  - To build the app for production, run this command for optimized performance with HTML/CSS/JS in the build folder
    - **npm run build**
- See: https://create-react-app.dev/docs/getting-started

In the upcoming lab…

- Open in Google Chrome: http://stackblitz.com/edit/node-inpmqg
  - A react app is already created for you
  - Fork the project to edit and save a copy. (you can do it with or without a StackBlitz account)
- In the terminal of StackBlitz, enter the following commands:

```
cd demo-app/  # enter app folder
npm install react-router-dom # install react-router-dom
npm start# start server
```

- After the command **npm start**, you should be able to see the web app rendered inside StackBlitz.

Further readings

- Beginner’s guide to npm
  - https://www.sitepoint.com/npm-guide/

- npm CLI commands
  - https://docs.npmjs.com/cli/v8/commands