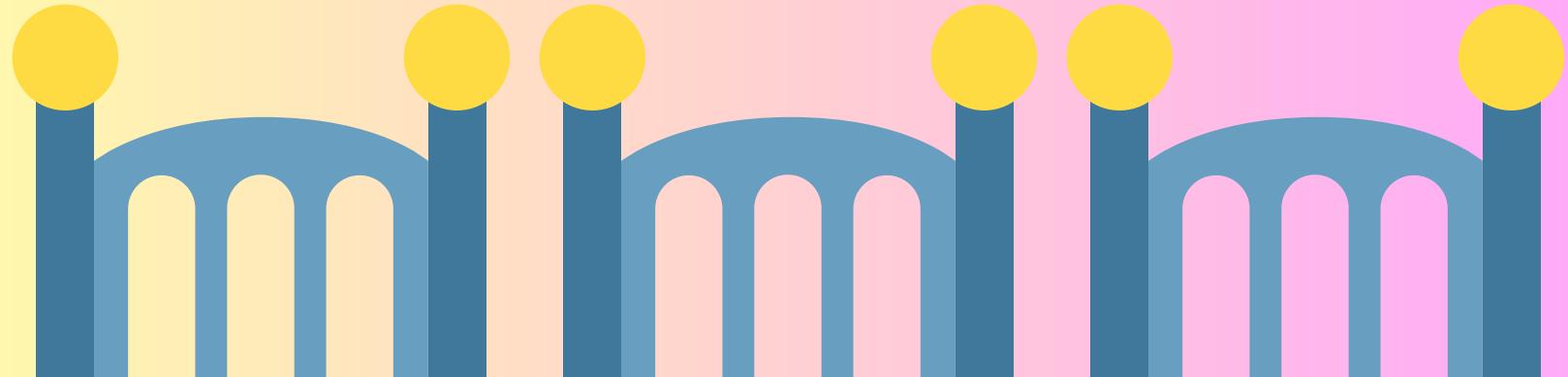




# NPM Vs YARN Vs PNPM



in Sonu Madheshiya



# NPM

NPM (Node Package Manager) is the default package manager for Node.js and has the largest repository of packages.

Basic Commands:

`npm install <package>`: Installs a package and its dependencies

- `npm update <package>`: Updates a package to its latest version

`npm uninstall <package>`: Uninstalls a package

- `npm run <script>`: Runs a script defined in the project's package.json file

# Yarn

Yarn (Yet Another Resource Negotiator) was created by Facebook as an alternative to npm with improved performance and security features.

You can install yarn using npm by running `npm install -g yarn` command and check it by running `yarn -v`.

Basic Commands:

- `yarn add <package>`: Installs a package and adds it to the project's dependencies
- `yarn upgrade <package>`: Upgrades a package to its latest version
- `yarn remove <package>`: Removes a package from the project
- `yarn run <script>`: Runs a script defined in the project's `package.json` file

# PNPM

PNPM (Fast, disk space efficient package manager) is focused on performance and efficiency and is designed to save disk space.

You can install pnpm using npm by running `npm install -g pnpm` command and check it by running `pnpm -v`.

Basic Commands:

`pnpm install <package>`: Installs a package and its dependencies

- `pnpm update <package>`: Updates a package to its latest version

`pnpm uninstall <package>`: Uninstalls a package

- `pnpm run <script>`: Runs a script defined in the project's package.json file

# Conclusion

| NPM                 | Yarn               | PNPM                 |
|---------------------|--------------------|----------------------|
| npm install         | yarn               | pnpm install         |
| npm install <pkg>   | yarn add <pkg>     | pnpm add <pkg>       |
| npm update <pkg>    | yarn upgrade <pkg> | pnpm update <pkg>    |
| npm uninstall <pkg> | yarn remove <pkg>  | pnpm uninstall <pkg> |
| npm run <script>    | yarn run <script>  | pnpm run <script>    |

These are just a few of the most common commands, and each package manager has more options and features that can be explored.

# Pros and Cons

NPM:

Pros: The default package manager for Node.js, has a large repository of packages, **easy to use** and **widely adopted**

Cons: **Can be slow** when dealing with **large projects** or installing **many packages**

Yarn:

Pros: Faster than npm, **good security features**, **efficient caching** of packages for offline usage

Cons: Lacks some features that npm has, may not be compatible with all packages in the npm registry

PNPM:

Pros: Saves disk space, fast and **efficient package resolution**, good performance for large projects

Cons: Not as widely adopted as npm or Yarn, may not be compatible with all packages in the npm registry

# What is the best?

It depends on the specific needs of a project. There is **no one "best"** package manager, as **each has its own strengths and weaknesses**, but they all serve the **same purpose** of managing dependencies for your projects.

In summary, it comes down to personal preference and the specific needs of a project. **Some developers** may prefer npm due to its widespread adoption, while **others** may prefer Yarn or pnpm for their **performance and efficiency**. It's important to consider the strengths and weaknesses of each and choose the one that fits the project's needs best.

***node\_modules is large....***

