



like and share

# npm vs npx





# npm

Node Package **Manager**, manages dependencies in Node.js projects, simplifying the process of **adding**, **updating**, and **sharing** packages.

## Managing Dependencies:

Use `npm install package-name` to fetch and install **project-specific dependencies** locally in the `node_modules` directory.

## Package.json:

The `package.json` file tracks project metadata and dependencies, ensuring version consistency.





# npX

Node Package **eXecute**, executes Node.js packages, streamlining the running of binaries or **commands** without global installations.

## Executing Binaries:

**Run** tools **without global installation**, ensuring a clean development environment.

## Running Commands:

Execute commands from packages on the fly, ideal for occasional use or one-off scripts.





## npm vs npx

NPM is a package manager used to install, delete, and update Javascript packages on your machine.

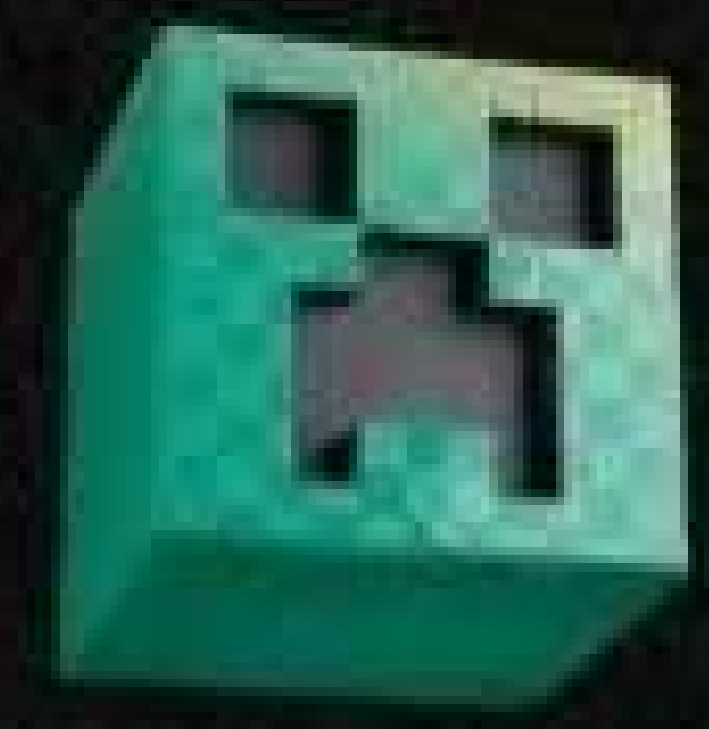


NPX is a package executor, and it is used to execute javascript packages directly, without installing them.





## npm vs npx



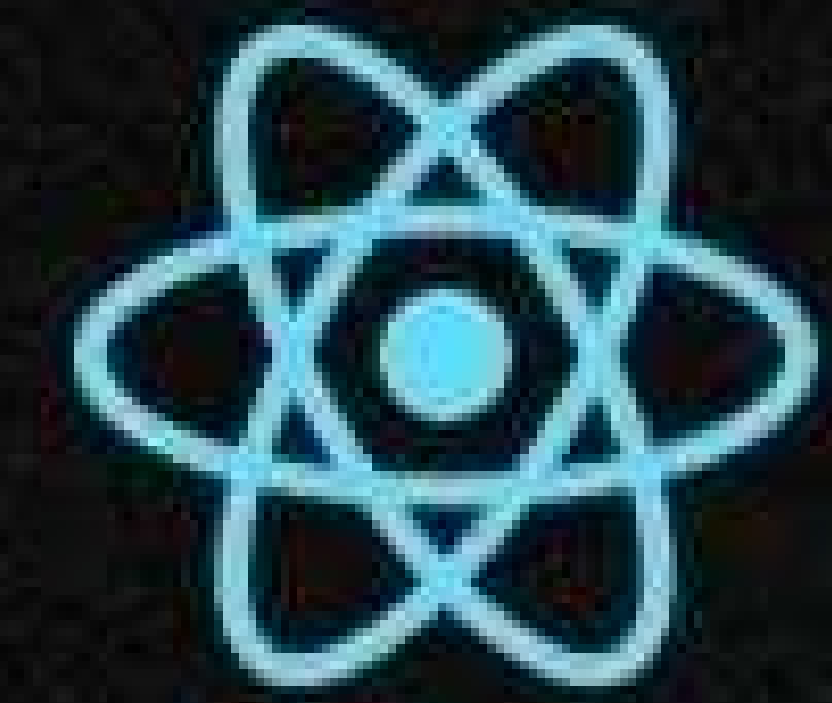
NPM installs packages globally, which means that your **machine may be polluted by packages** that are not required anymore in the long run.

NPX does not install packages, so package pollution on the machine is not a concern.





## npm vs npx



To use create-react-app using NPM, we would first have to install it globally, and then run it, which makes using NPM in such cases redundant.

The most common application of NPX is the create-react-app command. Since we only need to use it once, i.e., while initializing the project, **we do not install it.**







codewith**sloba**.com

Get a weekly digest of my tips and  
tutorials by subscribing now.