

# # [ npm Commands ] ( CheatSheet )

## 1. Getting Started

- `npm init`: Initialize a new Node.js project.
- `npm init -y`: Skip questionnaire and create a new Node.js project with default settings.

## 2. Installing Packages

- `npm install <package_name>`: Install a package locally.
- `npm install -g <package_name>`: Install a package globally.
- `npm install <package_name>@<version>`: Install a specific version of a package.
- `npm install <package_name> --save`: Install and add the package to the project's dependencies.
- `npm install <package_name> --save-dev`: Install and add the package to the project's dev dependencies.

## 3. Updating Packages

- `npm update`: Update all packages in your project.
- `npm update <package_name>`: Update a specific package.
- `npm outdated`: List outdated packages.

## 4. Removing Packages

- `npm uninstall <package_name>`: Remove a package from your project.

## 5. Listing Packages

- `npm list`: List installed packages.
- `npm list -g --depth 0`: List globally installed packages.

## 6. Package Information

- `npm view <package_name>`: View registry info for a package.
- `npm view <package_name> versions`: View all available versions of a package.

## 7. Running Scripts

- `npm run <script_name>`: Run a script defined in the package.json.
- `npm start`: Start a package.
- `npm test`: Run test scripts.

## 8. Publishing Packages

- `npm publish`: Publish a package.
- `npm login`: Log in to the npm registry.
- `npm whoami`: Display npm user name.

## 9. Miscellaneous Commands

- `npm cache clean --force`: Clean the npm cache.
- `npm config list`: List npm configuration.
- `npm help <term>`: Search for help on a term.

## 10. Troubleshooting

- `npm doctor`: Check your environment for common issues.
- `npm rebuild`: Rebuild a package.

## 11. Version Handling

- `npm version major`: Bump major version.
- `npm version minor`: Bump minor version.
- `npm version patch`: Bump patch version.

## 12. Exploring Packages

- `npm explore <package_name> -- npm run <script>`: Explore and optionally run a script in a package.

## 13. Security

- `npm audit`: Audit your project for vulnerabilities.
- `npm audit fix`: Auto-fix vulnerabilities.

## 14. Linking Packages

- `npm link`: Symlink a package folder.
- `npm link <package_name>`: Use a locally developed package in a local install.

## 15. Configuring NPM

- `npm set init-author-name 'Your Name'`: Set author name for npm init.
- `npm config set registry <registry_url>`: Set a custom registry.

## 16. NPM CI/CD

- `npm ci`: Clean install based on package-lock.json.

## 17. NPM Cache

- `npm cache verify`: Verify the contents of the cache.

## 18. Deprecating Packages

- `npm deprecate <package_name> "<message>"`: Deprecate a package with a message.

## 19. Peer Dependencies

- `npm install <package_name> --save-peer`: Add a package as a peer dependency.

## 20. Optional Dependencies

- `npm install <package_name> --save-optional`: Add a package as an optional dependency.

## 21. Package Lock

- `npm shrinkwrap`: Create an npm-shrinkwrap.json file.

## 22. NPM for Organizations

- `npm access public <package_name>`: Set package access to public.
- `npm access restricted <package_name>`: Set package access to restricted.

## 23. Checking for Engine Compatibility

- `npm engine-strict`: Enforce engine compatibility.

## 24. NPM Funding

- `npm fund`: Display information about funding.

## 25. NPM Hooks

- `npm hook add <url> <package_name>`: Add a hook for a package.

## 26. NPM Orgs

- `npm org create <orgname> <user>`: Create an organization.
- `npm team create <org:team> <user>`: Create a team within an organization.

## 27. NPM Tokens

- `npm token create`: Create an authentication token.
- `npm token list`: List all tokens.

## 28. NPM Token Management

- `npm token revoke <token>`: Revoke a token.

## 29. Scoped Packages

- `npm install @<scope>/<package_name>`: Install a scoped package.
- `npm publish --access=public`: Publish a scoped package publicly.

## 30. Git Dependencies

- `npm install <git-host>:<git-user>/<repo-name>`: Install a package from a git repository.
- `npm install <github username>/<repository>#<branch>`: Install directly from a GitHub repository.

## 31. Handling Global Packages

- `npm ls -g --depth=0`: List globally installed packages without dependencies.
- `npm root -g`: Find the path of globally installed packages.

## 32. NPM Scripts and Custom Commands

- `npm set-script <name> "<command>"`: Add custom scripts to your `package.json`.

## 33. Environment Variables

- `npm config set <key> <value>`: Set an environment variable in npm.
- `npm config get <key>`: Get the value of an environment variable.

## 34. NPM Pack and Link

- `npm pack`: Pack a package into a `.tgz` file.
- `npm link <package>`: Link a local package to use it in another project.

## 35. NPM Diff

- `npm diff`: Compare the dependencies or versions.

## 36. NPM Dist Tags

- `npm dist-tag add <package>@<version> [<tag>]`: Add a tag to a package version.
- `npm dist-tag rm <package> <tag>`: Remove a tag from a package.

## 37. NPM Search

- `npm search <term>`: Search for packages related to the term.

## 38. NPM Teams

- `npm team add <team> <user>`: Add a user to a team.
- `npm team remove <team> <user>`: Remove a user from a team.

## 39. NPM Version Management

- `npm set version <version>`: Force npm to use a specific version.

- `npm view <package_name> version`: Check the latest version of a package.

## 40. NPM Registry Interaction

- `npm ping`: Check if you can communicate with the registry.
- `npm whoami`: Check the username logged in to the registry.

## 41. Package Restoration

- `npm dedupe`: Reduce duplication in the `node_modules` folder.

## 42. Handling Local Packages

- `npm link`: Create a symbolic link from a local package to `node_modules`.
- `npm unlink <package>`: Remove the symbolic link.

## 43. Shrinkwrapping

- `npm shrinkwrap`: Lock down the versions of a package's dependencies.

## 44. NPM Bugs and Support

- `npm bugs <package>`: Open the issues page for a package.
- `npm support <package>`: Open the support page for a package.

## 45. NPM Cache Control

- `npm cache add <tarball file>`: Add a package tarball to your cache.

## 46. NPM Networks and Proxies

- `npm config set proxy <url>`: Set a proxy for npm.
- `npm config set https-proxy <url>`: Set an HTTPS proxy for npm.

## 47. NPM and Docker

- `npm prune`: Remove extraneous packages.

## 48. NPM Package.json Fields

- `npm pkg set <field>=<value>`: Directly set a field in `package.json`.

## 49. NPM Completion

- `npm completion`: Tab completion for npm commands.

## 50. NPM Registry Management

- `npm set registry <url>`: Change the registry URL.