NPM CLI - Commands and Flags

This cheat sheet covers the most common and helpful commands and command-line options for version 8 of the npm command-line interface, which installs Node.js packages.

BASIC USAGE

-v, --version

Print the current version of npm you are using.

<command> -h, --help

Quick help on the specified command.

-l

Display full usage info.

npx <command>

Run <command> from an npm module (either one installed locally, or fetched remotely). npx prompts you if the module you are trying to run is not installed yet.

CREATE A NEW NODE.JS APPLICATION

npm init [--yes|-y|--scope]

Initialize a new Node.js project and create a package.json file. Options include:

--yes, -y	Skip the init questionnaire.
--scope	Create a scoped application/module.

INSTALL NODE.JS MODULES

npm install <module_name>

Install the specifed module as a dependency.

--save-dev	Install the specifed module as a development dependency.
-g	Install the module globally.

npm install [--production]

Install the modules specified in the package.json file. The --production flag omits the installation of development dependencies.

npm ci

Similar to npm install, but for use in automated environments such as test platforms, continuous integration, and deployment.

npm uninstall <module>

Remove a module.

npm outdated

Check for outdated modules.

npm update

Update modules to the latest version, in accordance with semamtic versioning (version number incrementation).

npm ls [--all]

List the installed top-level modules. The --all option lists all installed modules.

npm prune

Remove extraneous modules.

npm link

Create a symbolic link in the global folder to the package in the local folder.

NPM SCRIPTS

npm start

Run a predefined command specified in the start property of a package's scripts object.

npm run <script>:: Run a predefined command specified in the <script> property of package's scripts object.

PUBLISH A NODE MODULE

npm whoami

Display the npm username.

npm login

Log in to the registry.

npm version [options]

Bump up the version of a module. Options include:

<newversion>	Bump up the version to <newversion>. Example: npm version 1.1.0.
major	Increment the major version number. Example: npm version major changes v1.1.0 to v2.1.0.
minor	Increment the minor version number. Example: npm version minor changes v1.1.0 to v1.2.0.
patch	Bump up the patch version by 1. Example: npm version patch changes v1.1.0 to v1.1.1.
premajor	Bump up the major version by 1, but as a prerelease. Example: npm verison premajor changes v1.1.0 to v2.1.0-0.
preminor	Bump up the minor version by 1, but as a

	prerelease. Example: npm verison preminor changes v1.1.0 to v1.2.0-0.
prepatch	Bump up the patch version by 1, but as a prerelease. Example: npm verison prepatch changes v1.1.0 to v1.1.1-0.
prerelease	Bump up the prerelease version. Example: npm version prerelease changes v1.1.0 to 1.1.0-1.
--preid= <prerelease-id>	Use a custom prerelease id tag. Example: npm version prelease --preid=beta changes v1.1.0``1.1.0-beta.0.
from-git	Use the latest Git tag as the version.

npm shrinkwrap

Lock down dependency versions for publication

npm publish [--dry-run]

Publish a module. The --dry-run option show a preview of what will be published.

npm deprecate

Deprecate a version of a module.

npm unpublish

Remove a module from the registry.

npm dist-tag

Modify package distribution tags.

npm logout

Log out of the registry.

WORKSPACES

Support for managing multiple packages from a single top-level, root package.

```
{
  "name": "workspace-example",
  "version": "1.0.0",
  "workspaces": {
    "packages": [
      "packages/*"
    ]
  }
}
```

MISC

npm audit

Run a security audit.

npm bugs

Display a module's bugs in a web browser.

npm docs

Display a module's documentation in a web browser.

npm fund

Retrieve funding information.