

# 7 npm commands

*you should know about!*





# create package.json



```
npm init
```

```
# generate and answer yes to all queries
```

```
npm init --yes
```

```
# short hand
```

```
npm init -y
```



## install package



```
# if you have a package.json with dependencies specified  
npm install
```

```
# short hand  
npm i
```

```
# install a specific package local to current folder  
npm i <package-name>
```

```
# install a package with a specific version  
npm i <package-name>@<version>
```

```
# install a package globally  
npm i -g <package-name>
```



## update package



```
# update all production packages in current folder
```

```
npm update
```

```
# update a specific package
```

```
npm update <package-name>
```

```
# update a package globally
```

```
npm update -g <package-name>
```



## list packages



# list all packages in the current directory

```
npm list
```

# short hand

```
npm ls
```

# list packages installed globally

```
npm list -g
```

# list packages in dev dependencies

```
npm list --dev
```

# list packages in production

```
npm list --prod
```



## check security issues



```
# scan your project for security issues
```

```
npm audit
```

```
# get report in json format
```

```
npm audit --json
```

```
# scan issues and fix them
```

```
npm audit fix
```

```
# if you want to skip dev dependencies
```

```
npm audit fix --only=prod
```



## remove extraneous packages



```
npm prune
```

When you run `prune`, the **npm CLI will run through your `package.json` and compare it to your project's `/node_modules` directory.** It will print a list of modules that aren't in your `package.json`.

The `npm prune` command then **strips out those packages, and removes any you haven't manually added to `package.json` or that were npm installed without using the `--save` flag.**



## lock dependency versions



```
npm shrinkwrap
```

using shrinkwrap in your project, **generates an `npm-shrinkwrap.json` file**. This allows you to **pin the dependencies of your project to the specific version you're currently using within your `node_modules` directory**. When you run `npm install` and there is a `npm-shrinkwrap.json` present, it will override the listed dependencies and any semver ranges in `package.json`





*Let me know if you found it  
useful :)*

