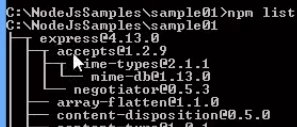
* To find the existing package in npm:







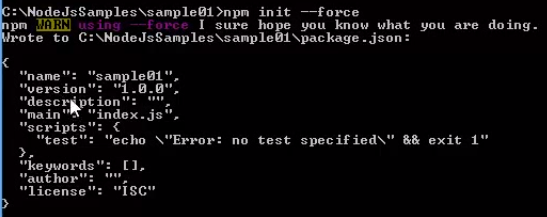
* To list the version of all node packges for my app “sample01”



* To install particular version of node package



* To create ‘package.json’:



* To update all the node packages



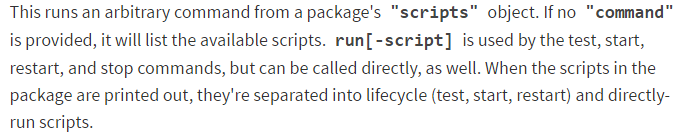
* To update particular package:



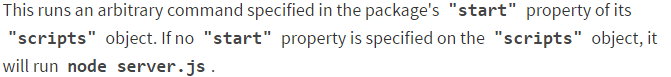
* List of packages available globally:



* “npm run <command>”



* “npm start”



* “npm test”



npm supports the "scripts" property of the *package.json* script, for the following scripts:

* **prepublish**: Run BEFORE the package is published. (Also run on local **npm install** without any arguments.)
* **publish**, **postpublish**: Run AFTER the package is published.
* **preinstall**: Run BEFORE the package is installed
* **install**, **postinstall**: Run AFTER the package is installed.
* **preuninstall**, **uninstall**: Run BEFORE the package is uninstalled.
* **postuninstall**: Run AFTER the package is uninstalled.
* **preversion**, version: Run BEFORE bump the package version.
* **postversion**: Run AFTER bump the package version.
* **pretest**, **test**, **posttest**: Run by the **npm test** command.
* **prestop**, **stop**, **poststop**: Run by the **npm stop** command.
* **prestart**, **start**, **poststart**: Run by the **npm start** command.
* **prerestart**, **restart**, **postrestart**: Run by the **npm restart** command. Note: **npm restart** will run the stop and start scripts if no **restart** script is provided.

npm has a great subset of functionality decidated to running tasks to facilitate in a packages lifecycle - in other words, it is a great tool for build scripts.