npm init — для создания начального файла package.json;

npm i {название пакета} — для установки новых пакетов;

npm i — для установки всех пакетов, указанных в package.json.

npm i [redom@1.0.0](mailto:redom@1.0.0) - Установка пакета конкретной версии

npm un redom - Удаление пакета

npm uninstall redom - Удаление пакета

npm up - Вывод и обновление пакетов, для которых вышли новые версии

npm update - Вывод и обновление пакетов, для которых вышли новые версии

npm outdated - Команду стоит использовать в случае, когда вы хотите пройтись по всем зависимостям, требующим обновления, изучить их изменения и затем обновить вручную.

npm i package@latest - Для обновления конкретного пакета до последней версии

npm run {название команды} - С помощью npm run вы можете выполнить команду, которую вы сами определили в package.json.

{

    "scripts": {

        "build": "parcel build index.html",

        "dev": "parcel index.html"

    }

}

**Типы зависимостей:**

**dependencies** должны храниться только те зависимости, которые используются в исходном коде проекта, которые прямо импортировались в код приложения.

**devDependencies** должны прописываться и любые другие пакеты для тестирования кода, и инструменты для разработки. Чтобы поместить пакет в devDependencies, нужно установить его со специальным флагом -D или --save-dev.

npm i -D parcel

npm i --save-dev parcel

**peerDependencies**— равноправные зависимости, при включении которых в package.json сообщается человеку, устанавливающему ваш пакет, что ему нужна та же зависимость с указанной версией.

**optionalDependencies**— необязательные зависимости. Если во время установки с ними возникнут какие-то проблемы, то это не повлияет на удачное завершение всего установочного процесса.

**bundledDependencies** — это массив пакетов, которые объединяются с вашим пакетом. Они пригодятся, если вы захотите использовать стороннюю библиотеку, не входящую в npm, или включить некоторые проекты в качестве модулей.