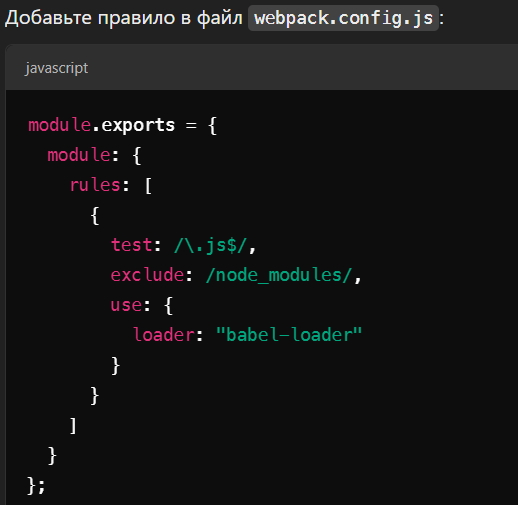
1. **npm**
   1. Какие основные функции выполняет npm (Node Package Manager)? – дефолтный пакетный менеджер для JavaScript, работающий на Node.js. Позволяет работать с пакетами: устанавливать, удалять, обновлять, работать с зависимостями, публиковать итд
   2. Какова роль файла package.json в проекте, использующем npm? – Каждый проект в JavaScript – может быть скопирован как npm-пакет с собственным описанием и файлом package.json. Он содержит основную информацию о пакете (название, версию, описание, автора итд)
   3. Как можно установить пакет (package) с использованием npm? – npm install <package-name>. Используя разные флаги и вариации, можно управлять местом установки пакета (глобальная установка), версиями, зависимостями
   4. Какие команды можно использовать для управления зависимостями в проекте с помощью npm? – устанавливаются командой npm install и записываются в файле package.json в разделе dependencies
   5. Как можно обновить установленные пакеты с помощью npm? – npm update. Она указывает менеджеру пакетов, что надо проверить список зависимостей в package.json и установить все доступные обновления, если нет ограничения.
   6. Что такое локальные и глобальные пакеты в npm? –
      * **Локальные** – устанавливаются для работы конкретно в проекте. Сохраняются в папке node\_modules внутри проекта.
      * **Глобальные** – устанавливаются на уровень всей системы и доступны из любой папки компьютера
   7. Какие возможности предоставляет npm для создания сценариев (scripts) в проекте? – В файле package.json можно указать скрипты и сложные команды, которые должны запускаться для сборки проекта или вывода ошибок. Перед командой надо добавить название скрипта, с помощью которого можно будет запустить его выполнение. Далее можно запускать скрипт через его имя.
   8. Как можно удалить установленный пакет с помощью npm? – npm uninstall <package-name>. манипулируя командой и различными флагами можно удалять глобально установленные пакеты и devDependencies.
   9. Какова роль файла .npmrc и какие настройки можно задать с его помощью? – файл может оптимизировать и изменять настройки Node.js-окружения. Автоматизация команд, изменение стандартного реестра npm
   10. Как можно узнать информацию о конкретном пакете, включая его зависимости и использование, с помощью npm? –
       * npm list package-name - Показывает, установлен ли пакет в проекте, его версию и где он находится.
       * npm view package-name dependencies – покажет конкретное поле о пакете (тут – dependencies)
2. **Babel**
   1. Что такое Babel и какую роль он играет в разработке JavaScript-приложений? – инструмент транспиляции (переписывания кода) для JavaScript, который позволяет разработчикам использовать последние возможности языка JavaScript, даже если они не поддерживаются в текущей версии браузера или среды выполнения Node.js. Babel преобразует современный код JavaScript в совместимый с более старыми версиями JavaScript, которые могут быть успешно выполнены в различных окружениях.
   2. Как можно настроить Babel в проекте? Какие файлы и конфигурационные опции используются? – установить Babel и создать конфигурационный файл
   3. Как можно подключить Babel к проекту, использующему сборщики модулей, такие как Webpack? –



1. **Source maps**
   1. Что такое source maps и какая их роль в разработке JavaScript приложений? – это файлы, которые сопоставляют скомпилированный или транспилированный код с его исходным кодом. Они позволяют разработчикам видеть оригинальный код и его структуру во время отладки, даже если исполняемый файл в браузере выглядит иначе.
   2. Какие преимущества предоставляют source maps при разработке и отладке JavaScript кода? –
      * Улучшение процесса отладки.
      * Упрощение работы с транспиляцией и минификацией.
      * Возможность работы с исходниками на других языках (TypeScript, SASS).
   3. Как source maps помогают в исследовании ошибок и упрощении процесса отладки? –
      * Ошибки и breakpoints привязываются к исходнику
      * Удобно отлаживать, т.к. код читаем
      * Работает с препроцессорами
      * Интегрирован с системами сборщиков ошибок
2. **lock файл**
   1. Какова роль lock-файла в проекте, использующем менеджер пакетов, например, npm или Yarn? – это файл, который фиксирует точные версии установленных зависимостей и их транзитивных зависимостей (зависимостей зависимостей). Он служит для обеспечения согласованности и повторяемости установки пакетов в проекте.
      * В проектах на npm используется файл package-lock.json
   2. Какие данные содержатся в lock-файле и как они связаны с зависимостями проекта? –
      * Повторяемость установки:  
        Lock-файл фиксирует точные версии всех пакетов и их зависимостей. Это гарантирует, что каждый раз при установке зависимостей проект будет получать одинаковые версии пакетов.
      * Изоляция изменений:  
        Если кто-то обновляет одну из зависимостей в проекте, lock-файл изменяется, что позволяет остальным разработчикам или серверу сборки получать обновления. Это исключает неожиданные изменения, которые могут возникнуть из-за обновлений в npm-репозитории.
      * Управление транзитивными зависимостями:  
        Lock-файл сохраняет не только прямые зависимости, но и все их зависимости. Это позволяет точно контролировать, какие версии транзитивных зависимостей используются, минимизируя вероятность конфликтов.
      * Оптимизация установки:  
        Lock-файл ускоряет установку зависимостей, поскольку npm не нужно каждый раз вычислять, какие версии пакетов и их зависимостей необходимо установить — все это уже зафиксировано в lock-файле.
3. **npm scripts**
   1. Что такое npm-scripts и какую роль они играют в проекте? — это механизм в npm, который позволяет автоматизировать выполнение различных задач в процессе разработки и эксплуатации проекта. npm-scripts задаются в файле package.json и могут быть использованы для выполнения команд, таких как сборка проекта, запуск тестов, запуск сервера и другие задачи.
   2. Какие команды можно выполнять с помощью npm-scripts? —
      * Сборка проекта.
      * Запуск сервера.
      * Тестирование кода.
      * Линтинг (проверка кода на ошибки).
      * Обновление зависимостей.
      * Деплой на сервер
      * Итд.
   3. Каким образом можно определить собственные npm-scripts в проекте? — npm-scripts задаются в файле package.json в разделе scripts