1. **Объясни во всех деталях как работает стек вызовов в JavaScript.**

Стек вызовов

JavaScript — однопоточный язык программирования. Это означает, что у него один стек вызовов. Таким образом, в некий момент времени он может выполнять лишь какую-то одну задачу.  
  
Стек вызовов — это структура данных, которая, говоря упрощённо, записывает сведения о месте в программе, где мы находимся. Если мы переходим в функцию, мы помещаем запись о ней в верхнюю часть стека. Когда мы из функции возвращаемся, мы вытаскиваем из стека самый верхний элемент и оказываемся там, откуда вызывали эту функцию. Это — всё, что умеет стек.

1. **Что означают понятия development, building, deployment, production?**

**Процесс разработки**

**Процесс сборки**

**Процесс развертывания**

**Процесс работы**

1. **Зачем нужен TypeScript, если задачу можно решить на JavaScript? Какое самое главное преимущество он дает?**

TypeScript — это язык программирования со статической типизацией, позиционирующий себя как язык расширяющий возможности JavaScript.

TypeScript значительно сокращает время на устранение ошибок и выявление багов, которые порой не так просто отыскать в динамической среде JavaScript.

В случае, если для разработчика TypeScript является первым типизированным языком, следует знать, что его изучение значительно ускорит процесс профессионального роста поскольку типизированный мир открывает аспекты программирования неочевидные в динамических языках.

Помимо этого, TypeScript позволяет писать более понятный и читаемый код максимально описывающий предметную область, за счет чего архитектура приложения становится более выраженной, а разработка неявно увеличивает профессиональный уровень программиста.

1. **Почему знание и использование Linux считается преимуществом? Какие привелегии даёт это знание?**

**Преимущества**

* В операционной системе Linux все полностью бесплатно и не нужно ничего платить;
* Linux работает на каждой машине, которая находится в рабочем состоянии. Он будет работать очень хорошо на всех компьютерах, в отличие от MAC OS;
* он будет работать с максимальной скоростью на каждом компьютере, независимо от того, сколько лет системе, в отличие от Windows;
* Linux вне сферы действия вируса, поскольку Linux не подвергается атаке. Наоборот, он используется для атаки на Windows;
* под Linux можно легко выполнять хакерские дела.

1. **Что такое unit-тесты и зачем они нужны?**

Модульное тестирование, иногда блочное тестирование или юнит-тестирование ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *unit testing*) — процесс в [программировании](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), позволяющий проверить на корректность отдельные модули [исходного кода](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) программы, наборы из одного или более программных модулей вместе с соответствующими управляющими данными, процедурами использования и обработки.

Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к [*регрессии*](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), то есть к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок. Например, обновить используемую в проекте библиотеку до актуальной версии можно в любой момент, прогнав тесты и выявив несовместимости.

1. **Как начать использовать фичи из ES7, ES8 и т.д., если браузеры их еще не поддерживают?**

Поскольку ES6 все еще принимается, и не все браузеры поддерживают все, у них нет причин стремиться к ES7. Если вы хотите использовать >функции ES7, я бы предложил изучить Babel, так как есть способы использовать ES7 и выше и скомпилировать его обратно в ES5, чтобы даже IE поддерживал его.

1. **Назови 5 основных хуков React и приведи случаи, когда каждый из них может понадобиться.**

**Что такое хук?** Хук - это специальная функция, которая позволяет «прицепиться» к возможностям React.

useState - это хук, который позволяет добавлять состояние React к компонентам-функциям. Он принимает аргумент, который является начальным значением свойства состояния, и возвращает текущее значение свойства состояния и метод, способный обновить это свойство состояния.

UseEffect — один из трёх больших встроенных React Hooks и один из самых популярных хуков. Он даёт возможность создавать условные изменения, ссылающиеся на состояние программы внутри функционального компонента. Хук useEffect позволяет запускать дополнительный код после того, как React обновит DOM.

**useMemo**, чтобы избежать повторного выполнения дорогостоящих вычислений для генерации нового значения. Вызывает функцию при изменении зависимостей и memoizes (запоминает) результат функции между рендерами. Она возвращает мемоизированное значение. Возвращает не саму функцию, а ее значение.

**useCallback** Возвращает запомненный обратный вызов. Этот хук принимает функцию первым параметром, а вторым параметром массив зависимостей. При каждом вызове компонента, если хотя бы одно значение массива зависимостей не изменится, то этот хук вернет старую функцию

Хук **useRef** позволяет сохранить некоторый объект, который хранится в течение всей жизни компонента.

В качестве параметра функция **useRef ()** принимает начальное значение хранимого объекта. А возвращаемое значение - ссылка-объект, из свойства **тока** которого можно получить хранимое значение.

1. **Опиши весь процесс работы с Redux. Опиши каждое из этих понятий: store, reducer, dispatch, selector, action, action creator, middleware, thunk**

Redux – это библиотека js для управления состоянием приложения.

Redux состоит из следующих компонентов:

1. Action—— Это объект, используемый для описания того, что произошло.

В **Redux action creator** – **это** функция, которая возвращает объект **action**. **Action creator** может показаться лишним компонентом, но он повышает портативность и упрощает тестирование

1. Reducer—— Здесь можно определить, как изменится состояние.
2. Store—— Дерево состояний / объектов всей программы хранится в Магазине.

Store - это объект JavaScript, который может сохранять состояние программы и предоставлять некоторые методы для доступа к состоянию, планирования операций и регистрации слушателей

1. View—— Отображать только данные, предоставленные Магазином.

DISPATCH - Отправка действия, которая может вызвать изменение состояния приложения

Селекторы-это функции, которые принимают состояние Redux в качестве аргумента и возвращают некоторые данные для передачи компоненту.

**Redux Thunk это** middleware библиотека, которая позволяет вам вызвать action creator, возвращая при этом функцию вместо объекта.