

[**Junior** **4**](#_Toc11255)

[1 Что такое React? 4](#_Toc11256)

[2 Какие основные преимущества есть в React? 4](#_Toc11257)

[3 Какие есть ограничения в React? 4](#_Toc11258)

[4 Что такое JSX? 4](#_Toc11259)

[5 Что такое Virtual DOM в React? 4](#_Toc11260)

[6 Что такое Props? 4](#_Toc11261)

[7 Что такое state и как он используется? 5](#_Toc11262)

[8 Что такое refs в React? 5](#_Toc11263)

[9 Что такое JEST? 5](#_Toc11264)

[10 Когда следует использовать Class компоненты, а когда функциональные? 5](#_Toc11265)

[11 Что происходит, когда вы вызываете setState? 5](#_Toc11266)

[12 В чем разница между state и props? 5](#_Toc11267)

[13 Когда следует делать асинхронные запросы на сервер в React? 6](#_Toc11268)

[14 В чем смысл специального атрибута key? 6](#_Toc11269)

[15 Что значит компонент mounted? 6](#_Toc11270)

[16 Назовите разницу между контролируемым и неконтролируемым компонентом 6](#_Toc11271)

[17 Что такое фрагменты? 6](#_Toc11272)

[**Middle** **6**](#_Toc11273)

[18 Как React обрабатывает пользовательские события? 6](#_Toc11274)

[19 Что такое Redux? 6](#_Toc11275)

[20 Назовите основные этапы жизненного цикла компонента 7](#_Toc11276)

[21 В setState можно передавать объект или функцию. В чем разница и что лучше использовать? 7](#_Toc11277)

[22 Назовите разницу между Презентационным и Контейнер компонентом? 7](#_Toc11278)

[23 Что такое Context? 7](#_Toc11279)

[24 Что такое Higher-Order компоненты? 7](#_Toc11280)

[25 Что делает shouldComponentUpdate и почему он важен? 8](#_Toc11281)

[26 Что такое store в Redux? 8](#_Toc11282)

[27 Что такое action в Redux? 8](#_Toc11283)

[28 Что нельзя делать в методе render? 8](#_Toc11284)

[29 Какие типы middleware есть в redux для работы с асинхронностью? 8](#_Toc11285)

[30 Что такое Pure Components? 8](#_Toc11286)

[31 Почему не стоит изменять state напрямую? 9](#_Toc11287)

[32 Как изменить state используя динамический ключ? 9](#_Toc11288)

[33 Что такое Error Boundaries в React? 9](#_Toc11289)

[34 Что такое React Hooks? 9](#_Toc11290)

[35 В чем разница между useRef и createRef? 9](#_Toc11291)

[36 Что такое useState? 9](#_Toc11292)

[37 Что такое prop drilling и как этого избежать? 10](#_Toc11293)

[38 Как валидировать props в React? 10](#_Toc11294)

[**Senior** **10**](#_Toc11295)

[39 Зачем делать eject? 10](#_Toc11296)

[40 Что такое reducer? 10](#_Toc11297)

[41 Разница между Flux и MVC? 10](#_Toc11298)

[42 Что не так с этим кодом? 10](#_Toc11299)

[43 Какой второй опциональный параметр можно передать в метод setState и за что он отвечает? 11](#_Toc11300)

[44 Что такое mapStateToProps и mapDispatchToProps? 11](#_Toc11301)

[45 Что такое React Fiber? 11](#_Toc11302)

[46 Разница между Flow и PropTypes? 11](#_Toc11303)

[47 Правда ли, что React делает ре-рендер всех компонентов и дочерних компонентов каждый раз когда вызывается setState? 11](#_Toc11304)

[48 Как можно улучшить производительность React приложения? 11](#_Toc11305)

[49 Зачем нужен Redux Thunk? 12](#_Toc11306)

[50 В чем ключевое отличие между React и Angular? 12](#_Toc11307)

# Junior

## 1 Что такое React?

* React - JavaScript фронтенд библиотека, разработанная Facebook в 2011
* В ядре - компонентный подход, позволяющий создавать переиспользуемые UI блоки
* Служит, для создания сложных интерактивных UI для web и мобильной разработки

## 2 Какие основные преимущества есть в React?

* Увеличивает производительность отрисовки приложений
* Может использоваться и на клиенте и на сервере
* Из-за JSX читаемость кода увеличивается
* Легко интегрировать с другими фреймворками
* Легко писать unit тесты

## 3 Какие есть ограничения в React?

* React - всего лишь библиотека
* Требуется некоторое время на освоение
* Может быть немного сложным для начинающих
* Код по-началу может выглядеть сложным из-за инфраструктуры и JSX

## 4 Что такое JSX?

JSX - ярлык для JavaScript XML. Это специальный синтаксис, который расширяет

JavaScript возможностью писать HTML внутри

Это позволяет интегрировать шаблоны компонентов прямо в JavaScript, что делает разработку проще

## 5 Что такое Virtual DOM в React?

Virtual DOM - легковесный JavaScript объект, который представляет копию реального DOM дерева. Нужен для оптимизации взаимодействия с DOM

## 6 Что такое Props?

Сокращенно от Properties. Входящие свойства в компонент. Они только для чтения и их нельзя менять. Всегда идут от родителя к ребенку.

## 7 Что такое state и как он используется?

Обычный объект - источник данных. Содержит информацию по поведению и состоянию интерфейса. Можно мутировать

## 8 Что такое refs в React?

Сокращенно от References. Специальный атрибут, позволяющий получить доступ до конкретного DOM элемента Нужен для:

* Вызова анимаций
* Для задания фокуса или выделения текста ● Взаимодействия со сторонними библиотеками

## 9 Что такое JEST?

JavaScript фреймворк, для юнит тестирования на основе Jasmine. Разработал

Facebook. Очень удобен именно для React

## 10 Когда следует использовать Class компоненты, а когда функциональные?

Если нужны жизненные этапы компонента - используем class компоненты

Иначе для оптимизации лучше функциональные

## 11 Что происходит, когда вы вызываете setState?

Вначале React соединяет объект стейта с измененными полями. На основе нового состояния строит новое дерево React элементов и выясняет, какие именно части приложения должны быть изменены

Это нужно для наиболее производительного обновления интерфейса

## 12 В чем разница между state и props?

**state**​ - структура данных, необходимая для изменения и отслеживания пользовательских действий

**props**​ - набор конфигурации, поступающий от родительского элемента. Их нельзя изменять

## 13 Когда следует делать асинхронные запросы на сервер в React?

Для этого служит метод ​**componentDidMount**

Или ​**useEffect**​ с пустым набором зависимостей

## 14 В чем смысл специального атрибута key?

Атрибут позволяет React понимать, какие именно элементы в списке были модифицированы или удалены, что увеличивает производительность рендеринга.

Лучше всего использовать уникальные значения, такие как ID. Индексы использовать не рекомендуется

## 15 Что значит компонент mounted?

Шаблон компонента соединен с DOM деревом

## 16 Назовите разницу между контролируемым и неконтролируемым компонентом

* Контролируемый компонент обладает своим стейтом, управляемый React
* Неконтролируемые компоненты обладают внутренним стейтом (как пример значение тега textarea)

## 17 Что такое фрагменты?

Специальный элемент в React позволяющий возвращать группу элементов без дополнительного родительского DOM элемента

# Middle

## 18 Как React обрабатывает пользовательские события?

Добавляет один обработчик события на корневой элемент.

Объект события оборачивает в свою обертку - ​**SyntheticEvent**​ для кроссбраузерности

## 19 Что такое Redux?

Библиотека для работы с потоком данных в JavaScript

Позволяет добавить дополнительный слой для приложения, где состояние описано в

JavaScript объекте. Нужно для более удобного написания кода

## 20 Назовите основные этапы жизненного цикла компонента

* **componentWillMount**​ - перед рендерингом, в основном для настройки компонента
* **render**​ - процесс рендеринга
* **componentDidMount**​ - уведомляет, про то, что компонент соединен с DOM деревом
* **componentWillReceiveProps**​ - уведомляет, про то, что приходят новые входящие свойства в компонент
* **shouldComponentUpdate**​ - возвращает true или false и служит для оптимизации. Решает, нужно ли делать ре-рендеринг
* **componentWillUpdate**​ - уведомляет, что компонент будет обновлен
* **componentDidUpdate**​ - уведомляет, что компонент был обновлен
* **componentWillUnmount**​ - используется для удаления слушателей и очистки компонента. Вызывается перед удалением компонента

## 21 В setState можно передавать объект или функцию. В чем разница и что лучше использовать?

**props**​ и ​**state**​ могут изменяться асинхронно. Если мы передадим функцию, то мы точно будет знать, что стейт основывается на предыдущем состоянии

## 22 Назовите разницу между Презентационным и Контейнер компонентом?

* Презентационный - “как вещи выглядят”. Нужен для создания интерфейса.

Работает на входящих параметрах

* Контейнер - “как вещи работают”. Обладают состоянием, подключены к Flux или Redux

## 23 Что такое Context?

Context - позволяет передавать свойства от родителя к ребенку, избегая промежуточных компонентов

## 24 Что такое Higher-Order компоненты?

Higher-order component (HOC) - функции, у которых входящий параметр компонент. Возвращают новый компонент с добавленным поведением

Могут быть использованы в следующих случаях:

1. Переиспользование кода
2. Слой абстракции для state и взаимодействия с ним
3. Управление props

## 25 Что делает shouldComponentUpdate и почему он важен?

Этап жизненного цикла, который решает, будет ли ре-рендер, или нет

Позволяет оптимизировать приложение

## 26 Что такое store в Redux?

JavaScript объект, в котором содержится состояние приложения. Дополнительно отвечает за следующее:

1. state может быть получен через ​**getState()**
2. Изменять state можно через ​**dispatch(action)** 3. Регистрировать изменения через ​**subscribe(listener)**

## 27 Что такое action в Redux?

Объект, который обязательно должен содержать ключ ​**type**​. С помощью него Redux понимает, что именно нужно сделать со стейтом

## 28 Что нельзя делать в методе render?

Нельзя изменять состояние компонента (например вызывать ​**setState**​). Должен быть чистой (pure) функцией

## 29 Какие типы middleware есть в redux для работы с асинхронностью?

1. Redux Thunk
2. Redux Promise
3. Redux Saga

## 30 Что такое Pure Components?

Тоже самое, что и Component, кроме того, что автоматически за вас реализует метод **shouldComponentUpdate**​.

## 31 Почему не стоит изменять state напрямую?

Не будет запущен процесс ре-рендеринга и интерфейс не поменяется. Корректно использовать метод ​**setState()**

## 32 Как изменить state используя динамический ключ?



## 33 Что такое Error Boundaries в React?

React - компонент, позволяющий обрабатывать ошибки в дочерних компонентах. Для это присутствует метод ​**componentDidCatch(error, info)**

## 34 Что такое React Hooks?

Функционал, добавленный в React 16.8. С помощью хуков, можно писать приложения, используя только функциональные компоненты, без классов.

С помощью хуков можно следить за стейтом, эмулировать жизненные этапы компонента, работа с ссылками и многое другое

## 35 В чем разница между useRef и createRef?

* **createRef**​ - всегда создает новую ссылку. Используется в class компонентах
* **useRef**​ - возвращает одинаковую ссылку на объект, которое были при начальном рендеринге

## 36 Что такое useState?

Встроенные React хук. Позволяет работать со стейтом в функциональных компонентах. Принимает начальное значение. Возвращает массив, состоящий всегда из 2х элементов (кортеж), где:

* первый элемент - само состояние
* второй элемент - функция, меняющая состояние

## 37 Что такое prop drilling и как этого избежать?

Передача свойств на прямую от родителя к ребенку через сложную и длинную иерархию компонентов.

Избежать можно используя ​**Context**​ или например ​**Redux**​ (Flux)

## 38 Как валидировать props в React?

Для этого есть дополнительная библиотека - ​**PropTypes**

# Senior

## 39 Зачем делать eject?

На случай, если необходимо модифицировать конфигурацию проекта (webpack, babel)

## 40 Что такое reducer?

Простая чистая функция, принимающая ​**state**​ и ​**action**​ и модифицирующая ​**state**​.

Должна возвращать новый объект

## 41 Разница между Flux и MVC?

MVC (model view controller) - парадигма, разделяющая отображение и данные, однако присутствуют следующие минусы:

* каскадная модель данных, сложно отслеживать состояние
* данные могут быть изменены где угодно. Как следствие непредсказуемое поведение UI

Flux позволяет решить проблему каскадной модели данных. Данные получаются из отдельного store и менять напрямую их нельзя.

## 42 Что не так с этим кодом?



С этим кодом все хорошо. Изменяем state на основе прошлого состояния и входящих параметров

## 43 Какой второй опциональный параметр можно передать в метод setState и за что он отвечает?

Функция, уведомляющая, что компонент закончил процесс ре-рендеринга.

## 44 Что такое mapStateToProps и mapDispatchToProps?

Функции в Redux, позволяющие приводить к более удобному формату данные из store в компонент

## 45 Что такое React Fiber?

Fiber - это новый механизм и базовый алгоритм для рендеринга в React 16. Основная цель - реализовать пошаговый рендеринг виртуального DOM для более быстрого рендеринга, работы с анимациями и дебагом.

## 46 Разница между Flow и PropTypes?

* **Flow**​ - статический инструмент для проверки типов. Использует аннотации и позволяет найти ошибки при компиляции (аналог TypeScript)
* **PropTypes**​ - проверяет типы входящих параметров в runtime

## 47 Правда ли, что React делает ре-рендер всех компонентов и дочерних компонентов каждый раз когда вызывается setState?

По умолчанию - да. Однако мы этим можем управлять в **shouldComponentUpdate(nextProps, nextState)**​.

## 48 Как можно улучшить производительность React приложения?

Избавиться от лишних рендеров (самой затратной операции).

Для этого можно использовать:

1. **shouldComponentUpdate**​ в класс компонентах
2. **PureComponent**​ для класс компонентов
3. **React.memo()**​ - для функциональных компонентов

## 49 Зачем нужен Redux Thunk?

Middleware позволяющая изменять состояние приложения в Redux в асинхронном режиме

## 50 В чем ключевое отличие между React и Angular?

React - библиотека для отрисовки приложения. Для другого функционала нужны другие решения (например для данных - Redux)

Angular - обширный фреймворк, где все решения есть в ядре (в коробке)

* Используя React - можно более гибко создавать приложения и более точечно управлять его частями
* Используя Angular - проще разрабатывать и поддерживать приложения