1) What is React?   
Библеотека для разработки пользовательскх интерсфейсоф для приложений написанных на JS

2) What are the major features of React?   
Reconsilation - ключевой алгоритм Реакт

3) What is JSX?    
Расширение JS для создания компонентов

4) Is it possible to use react without JSX?   
Да

5) What is the difference between Element and Component?   
Элемент - это древовидная система компонента. То как движок видит компоненты под капотом

6) How to create components in React?   
С помощью функций или классов

7) When to use a Class Component over a Function Component?   
Не думаю что есть обязательные кейсы. Сейчас функциональные компоненты покрывают всю необходимую область работы

8) What are Pure Components?    
При создании классовых компонентов, мы можем наследоваться от React.PureComponent и в нашем компоненте будет реализован метод жизненного цикла sholdComponentUpdate с неглубокой проверкой пропсов

9) If Pure Component is better from the optimization point of view, why don't we use it by default?   
В пюре компоненте реализована неглубокая сверка пропсов или состояние, поэтому, если объект не подходит под неглубокую сверку, то мы получим лишние сай эффекты.

10) What is state in React?   
Объект состояния компонента. Мы можем динамически менять эти данные с помощью setState

11) What are props in React?    
Объект с входными данными которые мы передаем от родительского компонента к потомку

12) What is the difference between state and props?   
Пропсы предназначены для чтения и их нельзя менять. Стейт можно менять

13) Why should we not update the state directly?    
Непонятный вопрос.    
ЕСли вопрос о том, почему бы не обновить как state = newState, то в таком случе мы можем получить сайд эффекты. Компонент не перерендерится

14) Is the state updated synchronously?   
Нет

15) What is the purpose of callback function as an argument of setState() - думаю этот вопрос должен быть HIGH . Его постоянно спрашивают   
Что бы получить доступ к предыдущему стейту.   
   
16) What is the difference between HTML and React event handling?   
В синтаксисе

17) How to bind methods or event handlers in JSX callbacks?   
Немного непонятьный вопрос. Через this. Байндим зыс в конструкторе

18) How to pass a parameter to an event handler or callback?   
Передать колбек и передать параметры

19) What are synthetic events in React?   
Экземпляры кросс-браузерной оболочки собственного события браузера

20) What are inline conditional expressions? --> ??   
Условный рендеринг. Условия, при которых произойдет перерендер страницы

21) What is "key" prop and what is the benefit of using it in arrays of elements?   
Если мы мапаем массив и рендерим данные то нам нужно передавать ключ. При удалении, изменении или добавлении, механизм реконсилейшн сравнивает ключи и перерендеривает конкретно измененнное место

22) What is the use of refs?   
Можем читать значение из DOM через рефы и это будет неуправляемый компонент

23) How to create refs?   
useRef() или React.createRef()

24) Difference between "createRef" and "useRef"(#)   
"createRef" -  для классовых компонентов   
"useRef"(#) - хук, для функциональных

25) React.memo VS useMemo   
React.memo - HOC. Запоминает компонент, перерендеревает компонент при изменении пропсов или стейта(зависимостей)   
useMemo - Хук. Запоминает результат выполнения функции, для предотвращения повторных вычислений

26) How do you memoize a component?   
React.memo

27) What is Virtual DOM?   
То как мы привыкли называть реконсилейшн. Ну или копия ДОМа.

28) How Virtual DOM works?   
Сравнивают ключи реакт элементов и перерисовуют изменившиеся

29) What is the difference between Shadow DOM and Virtual DOM?   
Virtual DOM один, а Shadow DOM прикрепляется к конкретному элементу.

30) What is React Fiber?   
Обновленная реализация ключевого алгоритма реакт. В файбере реализована линейная сложность алгоритма. Можно считать, это Стек для реакта)

31) What is the main goal of React Fiber?   
Упростить работу Реконсилейшн. Приоритезация работы. Планирование. Можно останавить работу и вернуться к ней позже, если что-то обрабатывается долго

32) What are controlled components?   
Компоненты, состояние которых мы меняем через Стейт

33) What are uncontrolled components?   
Когда реакт не знает про состояние какого-либо элемента

34) What is Lifting State Up in React?   
Context, state manager

35) What are the different phases of component lifecycle?   
Монтирование: constructor, getDrivedStatefromProps, render, componentDidMount   
Обновление: getDrivedStatefromProps, shouldComponentUpdate, render,getSnapshotBeforeUpdate, componentDidUpdate   
Размонтирование: componentWillUnmount

36)What are the lifecycle methods of React?   
Ответил выше

37) What are Higher-Order components?   
Компонент который принимает компонент в качестве аргумента и возвращает компонент

38) How to fetch data with React Hooks?   
Пишем хендлер, в котором обращаемся к серверу. После этого вызываем функцию в useEffect с теми условиями, которые нам нужны

39) What is context?   
Возможность пробрасывать данные вверх по дереву

40) What is children prop?   
Позволяет передать компоненту произвольные дочерние элементы

41) What is prop drilling?   
Передача пропсов вниз по дереву через большое количество компонентов

42) What is reconciliation?   
Ключевой алгоритм Реакт. С его помощью мы перерисовуем конкретное место где были изменения.

43) How do you conditionally render components?   
С помощью условных операторов или тернарки

44) What are error boundaries in React v16   
..

45) What are hooks?   
Возможность подключаться к методам жизненного цикла в функцианальных компонентов

46) React hooks rules --- в чем суть вопроса?

47) What are fragments?   
Позволяет создать обертку, не создавая оберток в ДОМ. Принимает в качестве атрибута key

48) Why fragments are better than container divs?    
Ответил выше   
   
**REDUX**

49) What is Flux?    
Архитектурный подход построения приложений, основанный на большом количестве источником истин.

50) What is Redux?   
Стейт менеджер. Основанный на одном источнике истины

51) What hooks does Redux have    
useDispatch(), useSelector()

52) What are the core principles of Redux?   
Один источник истины   
Редьюсеры - функции через которые мы делаем изменения - должны быть чистыми функциями

53) What are the downsides of Redux compared to Flux?   
В том что в редаксе один источник истины, а во флаксе - нет

54) Can I dispatch an action in reducer?   
Нет

55) How to access Redux store outside a component?   
через getState( ) в Thunk   
С помощью функции connect.   
Можно импортить стор в извращенных случаях)

56) What is the difference between React context and React Redux?   
Redux - это отдельная библиотека, со своими хуками, функциями и браузерными расширениями.    
React context такая себе обертка.   
Абсолютно разный подход

57) Should I keep all component's state in Redux store?    
Нет

58) What is the proper way to access Redux store?   
Создать Стор с помощью createStore(reducers). В компоненте можем получить через useSelector()

59) What is the purpose of the constants in Redux? --- непонятный вопрос

60) What is redux-saga? What is Redux Thunk?   
Промежуточная мидлвара для работы с запросами

61) What is an action in Redux?    
Объект с полем обязательным полем type и необязательным payload

**REACT ROUTER**

62) What is React Router?   
Библиотека для работы с роутами

63) What hooks does React Router have   
  useHistory - для управления маршрутом   
 useParam - вернет объект ключ/значение с параметрами урлу