react 面试题集锦

React 面试题集锦

基础概念

1. **什么是React？它的核心特点是什么？**
   - React是一个用于构建用户界面的JavaScript库
   - 核心特点：组件化、声明式编程、虚拟DOM、单向数据流
2. **解释虚拟DOM的工作原理及其优势**
   - 虚拟DOM是真实DOM的轻量级JavaScript表示
   - 优势：提高性能（最小化DOM操作）、跨平台能力、更简单的编程模型
3. **JSX是什么？它和HTML有什么区别？**
   - JSX是JavaScript的语法扩展，允许在JavaScript中写类似HTML的代码
   - 区别：JSX使用className而不是class，style接收对象而非字符串，事件命名使用驼峰式

组件相关

4. **类组件和函数组件有什么区别？**
   - 类组件：使用ES6 class，有生命周期方法，有this和state
   - 函数组件：无状态，更简洁，React 16.8后可用Hooks添加状态和生命周期功能
5. **React组件的生命周期方法有哪些？**
   - 挂载阶段：constructor, render, componentDidMount
   - 更新阶段：shouldComponentUpdate, render, componentDidUpdate
   - 卸载阶段：componentWillUnmount
6. **什么是受控组件和非受控组件？**
   - 受控组件：表单数据由React组件管理（value+onChange）
   - 非受控组件：表单数据由DOM自身管理（refs获取值）

Hooks

7. **React Hooks是什么？常用的Hooks有哪些？**
   - Hooks是React 16.8引入的特性，允许函数组件使用状态和其他React特性
   - 常用Hooks：useState, useEffect, useContext, useReducer, useMemo, useCallback, useRef
8. **useEffect和useLayoutEffect有什么区别？**
   - useEffect：异步执行，不会阻塞浏览器绘制
   - useLayoutEffect：同步执行，在浏览器绘制前完成，适合需要同步更新DOM的场景
9. **如何自定义Hook？**
   - 自定义Hook是一个以"use"开头的函数，可以调用其他Hook
   - 示例：自定义useFetch Hook用于API调用

状态管理

10. **React中的状态提升(Lifting State Up)是什么？**
    - 将共享状态移动到最近的共同祖先组件中
    - 通过props传递状态和回调函数给子组件
11. **React Context是什么？它的使用场景？**
    - Context提供了一种在组件树中共享值的方式，而不必显式地通过props传递
    - 使用场景：主题、用户认证、多语言等全局数据
12. **Redux和React Context的区别？**
    - Redux：单一状态树、严格的更新逻辑、中间件支持、时间旅行调试
    - Context：简单的状态共享解决方案，不适合高频更新

性能优化

13. **React的性能优化手段有哪些？**
    - 使用React.memo避免不必要的重新渲染
    - 使用useMemo和useCallback避免重复计算和函数重建
    - 代码分割和懒加载组件
    - 虚拟化长列表（react-window或react-virtualized）
14. **React中的key属性有什么作用？**
    - 帮助React识别哪些元素发生了变化，提高diff算法效率

- 应该使用稳定的、唯一的标识作为key，避免使用数组索引

15. **React的diffing算法是如何工作的?**
    - 基于两个假设：不同类型的元素产生不同的树，开发者可以使用key暗示稳定元素
    - 比较策略：逐层比较，列表比较使用key

进阶概念

16. **什么是错误边界(Error Boundaries)?**
    - 捕获子组件树中JavaScript错误的React组件
    - 使用componentDidCatch和getDerivedStateFromError生命周期方法

17. **React Portal是什么？使用场景?**
    - 提供了一种将子节点渲染到DOM节点之外的方式
    - 使用场景：模态框、工具提示、弹出菜单等需要脱离正常文档流的组件

18. **解释React的Fiber架构**
    - React 16引入的新的协调引擎
    - 特性：增量渲染、任务分片、优先级调度、恢复中止任务的能力

实战问题

19. **如何在React中处理表单?**
    - 受控组件方式（推荐）
    - 非受控组件方式（使用ref）
    - 表单库如Formik或React Hook Form

20. **如何在React中实现代码分割?**
    - React.lazy动态导入组件
    - Suspense组件提供加载状态
    - 路由级别的代码分割（与React Router配合）

21. **如何优化React应用的加载性能?**
    - 代码分割和懒加载
    - 服务端渲染或静态生成
    - 预加载关键资源
    - 使用CDN和缓存策略

22. **如何处理React中的内存泄漏?**

- 清除定时器和事件监听器
- 取消未完成的API请求（AbortController）
- 在useEffect中返回清理函数

23. **React中如何实现动画?**

- CSS过渡和动画
- React Transition Group
- Framer Motion等专用动画库

24. **如何在React中测试组件?**

- Jest作为测试框架
- React Testing Library或Enzyme用于组件测试
- 集成测试和端到端测试（Cypress等）

25. **React 18有哪些新特性?**

- 并发渲染（Concurrent Rendering）
- 自动批处理（Automatic Batching）
- 新的Suspense特性
- 新的Hooks（useId, useTransition等）