

JS高级 -- MVVM

题目

1. 说一下 对组件化的理解

- 组件的封装
 - 视图
 - 数据
 - 变化逻辑（数据驱动视图变化）
- 组件的复用
 - props 传递
 - 复用

2. JSX本质是什么

- JSX 语法
- JSX 解析成JS
 - JSX其实是语法糖
 - 开发环境会将JSX编译成JS代码
 - JSX的写法大大降低了学习成本和编码工作量
 - 同时，JSX增加了debug的成本
- 独立的标准
 - JSX是React引入的，但不是React独有的
 - React已经将它作为一个独立标准开放，其他项目也可用
 - `React.createElement`是可以自定义修改的
 - 说明：本身功能已经完备，和其他标准兼容和扩展性没问题

3. JSX和vdom的关系

- 为何需要 vdom ： JSX 需要渲染成 html ,数据驱动视图
- `React.createElement` 和 `h` ,都生成 vnode
- 何时 patch ： `ReactDOM.render()` 和 `setState`
- 自定义组件的解析 ： 初始化实例，然后执行 `render`

4. React setState 过程

- `setState` 的异步
 - 可能会一次执行多次 `setState`
 - 你无法规定、限制用户如何使用 `setState`
 - 没必要每次`setState`都重新渲染，考虑性能
 - 即便是每次重新渲染，用户也看不到中间的效果
- vue 修改属性也是异步
- `setState` 的过程
 - 每个组件实例，都有 `renderComponent` 方法
 - 执行 `renderComponent` 会重新执行实例的 `render`
 - `render` 函数返回 `newVnode` ,然后拿到 `preVnode`
 - 执行 `patch(preVnode,newVnode)`

5.React vs Vue

- 两者的本质区别
 - Vue 本质是 MVVM 框架，由 MVC 发展而来
 - React 本质是前端组件化框架，由后端组件化发展而来
 - 但是并不妨碍两者都能实现相同的功能
- 模板和组件化的区别
 - Vue 使用模板（最初由 angular 提出）
 - React 使用 JSX
 - 模板语法上，更加倾向于 JSX
 - 模板分离上，更加倾向于 Vue （模板和JS混在一起，未分离，虽然数据和模板分离了）
 - 组件化的区别
 - React本身就是组件化，没有组件化就不是 React
 - Vue 也支持组件化，不过是 MVVM 上的扩展
 - React 的组件化做的更彻底，更清晰
- 两者的共同点
 - 都支持组件化
 - 数据驱动视图，数据和视图分离
- 总结
 - 国内使用，首推vue，文档易读、易学、社区够大
 - 团队水平较高，推荐使用React。组件化和JSX（JSX标准化）