

Vue 3 的 Composition API 和 React 的 Hooks 的区别

- Composition API 只会被调用一次，而 React Hooks 函数会被调用多次；
- Composition API 无需使用 `useMemo`, `useCallback`，因为 `setup` 只调用一次；而 React Hooks 需要；
- Composition API 无需考虑调用顺序，而 React 需要保证 Hooks 顺序一致；

Vue 3 的 Composition API 和 React 的 Hooks 都是用于在函数组件中管理状态和副作用的 API，但它们有一些不同之处。

1. 语法差异

- Composition API: Vue 3 的 Composition API 使用 `setup` 函数来定义和组合逻辑，通过 `ref` 和 `reactive` 等 Vue 提供的 API 来创建响应式数据。它还提供了 `toRefs` 和 `toRaw` 等辅助函数来更好地管理响应式数据。
- Hooks: React 的 Hooks 使用 `useState` 和 `useEffect` 等 Hook 来管理状态和副作用。在最新的 React 版本中，Hooks API 还增加了 `useContext` 和 `useReducer` 等 Hook，以更好地管理复杂的状态和副作用。

2. 响应式系统

- Composition API: Vue 3 的 Composition API 使用了 Vue 3 的响应式系统，通过 `ref` 和 `reactive` 等 API 来创建响应式数据，可以更好地利用 Vue 3 的响应式系统提供的功能，例如 `v-model` 指令和 `computed` 属性等。
- Hooks: React 的 Hooks 并没有内置的响应式系统，而是通过调用其他 React API（例如 `useState` 和 `useContext`）来管理状态和副作用。因此，它需要手动处理响应式更新和副作用。

3. 社区支持和应用广泛性

- Hooks: Hooks 是 React 的官方推荐状态管理 API，得到了广泛的应用和支持。大多数 React 库和框架都支持 Hooks，并且新的 React 功能通常都会优先支持 Hooks。
- Composition API: Vue 3 的 Composition API 解决了一些 Hooks 无法解决的问题，社区支持和广泛应用仍然不如 Hooks。