

Vue3 对比 Vue2 的变化

- 在 Vue2 的时候使用 `defineProperty` 来进行数据的劫持, 需要对属性进行重写添加 `getter` 及 `setter` 性能差。
- 当新增属性和删除属性时无法监控变化。需要通过 `$set`、`$delete` 实现
- 数组不采用 `defineProperty` 来进行劫持 (浪费性能, 对所有索引进行劫持会造成性能浪费) 需要对数组单独进行处理

Vue3 中使用 `Proxy` 来实现响应式数据变化。从而解决了上述问题。

CompositionAPI

- 在 Vue2 中采用的是 `OptionsAPI`, 用户提供的 `data`, `props`, `methods`, `computed`, `watch` 等属性 (用户编写复杂业务逻辑会出现反复横跳问题)
- Vue2 中所有的属性都是通过 `this` 访问, `this` 存在指向明确问题
- Vue2 中很多未使用的方法或属性依旧会被打包, 并且所有全局 API 都在 `Vue` 对象上公开。Composition API 对 `tree-shaking` 更加友好, 代码也更容易压缩。
- 组件逻辑共享问题, Vue2 采用 `mixins` 实现组件之间的逻辑共享; 但是会有数据来源不明确, 命名冲突等问题。Vue3 采用 `CompositionAPI` 提取公共逻辑非常方便

简单的组件仍然可以采用 `OptionsAPI` 进行编写, `compositionAPI` 在复杂的逻辑中有着明显的优势~。 `reactivity` 模块中就包含了很多我们经常使用到的 API 例如: `computed`、`reactive`、`ref`、`effect` 等

珠峰前端架构直播课(每周)

- 晋级高级前端-对标阿里 P6+
- 前端技术专家联合研发

JS高级 / VUE / REACT / NodeJS / 前端工程化 / 项目实战 / 测试 / 运维



长按识别二维码加微信
获取直播地址和历史精彩视频

珠峰架构