

Vue3 设计思想

在 Vue3 中依然保留了 Vue2 的特色

1 声明式框架

Vue3 依旧是声明式的框架，用起来简单。

命令式和声明式区别

- 早在 JQ 的时代编写的代码都是命令式的，命令式框架重要特点就是关注过程
- 声明式框架更加关注结果。命令式的代码封装到了 Vuejs 中，过程靠 vuejs 来实现

声明式代码更加简单，不需要关注实现，按照要求填代码就可以（给上原材料就出结果）

- 命令式编程：

```
let numbers = [1,2,3,4,5]
let total = 0
for(let i = 0; i < numbers.length; i++) {
  total += numbers[i] - 关注了过程
}
console.log(total)
```

- 声明式编程：

```
let total2 = numbers.reduce(function (memo, current) {
  return memo + current
}, 0)
console.log(total2)
```

2 采用虚拟 DOM

传统更新页面，拼接一个完整的字符串 innerHTML 全部重新渲染，添加虚拟 DOM 后，可以比较新旧虚拟节点，找到变化在进行更新。虚拟 DOM 就是一个对象，用来描述真实 DOM 的

```
const vnode = {
  __v_isVNode: true,
  __v_skip: true,
  type,
  props,
```

```
key: props && normalizeKey(props),
ref: props && normalizeRef(props),
children,
component: null,
el: null,
patchFlag,
dynamicProps,
dynamicChildren: null,
appContext: null,
};
```

3 区分编译时和运行时

- 我们需要有一个虚拟 DOM，调用渲染方法将虚拟 DOM 渲染成真实 DOM（缺点就是虚拟 DOM 编写麻烦）
- 专门写个编译时可以将模板编译成虚拟 DOM（在构建的时候进行编译性能更高，不需要再运行的时候进行编译，而且 vue3 在编译中做了很多优化）

4 Vue3 新增设计

- Vue3.0 更注重模块上的拆分，在 2.0 中无法单独使用部分模块。需要引入完整的 Vuejs(例如只想使用使用响应式部分，但是需要引入完整的 Vuejs)，Vue3 中的模块之间耦合度低，模块可以独立使用。 **拆分模块**
- Vue2 中很多方法挂载到了实例中导致没有使用也会被打包（还有很多组件也是一样）。通过构建工具 Tree-shaking 机制实现按需引入，减少用户打包后体积。同时也移除了一些不需要的功能（filter、inline-template） **重写 API**
- Vue3 允许自定义渲染器，扩展能力强。不会发生以前的事情，改写 Vue 源码改造渲染方式。 **扩展更方便**
- 使用 RFC 来确保改动和设计都是经过 Vuejs 核心团队探讨并得到确认的。也让用户可以了解每一个功能采用或废弃的前因后果。 **采用RFC**

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