虚拟 DOM

虚拟 DOM 又称 `VDOM` 或 `virtual DOM`。

- DOM 操作非常耗费性能；
- 以前用 JQuery，可以自行控制 DOM 操作的时机，手动调整。

1. 有何有效控制 DOM 操作

Vue 和 React 是数据驱动视图，如何有效控制 DOM 操作，是提升性能的关键。JS 计算很快，比 DOM 渲染快很多，利用 VDOM 实现用 JS 模拟 DOM 结构，计算出最小的变更，操作 DOM。

- 用 JS 模拟 DOM 结构（VNode）；
- 新旧 VNode 对比，得出最小的更新范围，最后更新 DOM。

```
<div id="container" class="container">
    <p>VDOM</p>
    <ul style="font-size: 20px">
        <li>a</li>
    </ul>
</div>
```

使用 JS 模拟上面的 DOM 结构，实现 VNode：

```
{
    tag: 'div',
    props: {
        id: 'container',
        className: 'container'
    },
    children: [
        {
            tag: 'p',
            children: 'VDOM'
        },
        {
            tag: 'ul',
            props: {
                style: 'font-size: 20px',
            },
            children: [
                {
                    tag: 'li',
                    children: 'a'
                }
            ]
        }
```

```
    ]
}
```

以上 JS 写法（VNode 结构，虚拟节点树描述）并非标准，可能不同的库中使用的 key 不同，比如：tag 换为 element。但一般都会包括：标签、属性、子节点（`tag`、`props`、`children`）等三个基本的 key。

2. 通过 snabbdom 学习 VDOM

vue 参考的 snabbdom 实现的 VDOM 和 diff，简洁强大的 VDOM 库，易学易用。

vue3 重写了 VDOM 的代码，优化了性能。React VDOM 具体实现和 Vue 也不同。

snabbdom 重点：

- h 函数
- VNode 数据结构
- patch 函数（`patch(element, VNode)` 和 `patch(VNode, newVNode)`）