

Vue中的虚拟DOM

概念

虚拟DOM (Virtual DOM)是对DOM的js抽象，他们是js对象，能够描述DOM结构和关系。

应用在代码逻辑中的状态变化作用于虚拟DOM，最终映射到DOM上。



优点

- 轻量、快速
- 跨平台
- 兼容性处理

整体流程

1. 执行渲染函数获取VNode
2. 获取并执行patch
 - 初始化执行新增操作
 - 更新时执行patchVnode



执行渲染函数获取VNode

```
new Vue({
  render(h) {
    return h(tag, data, children)
  },
  components: {
    comp: Ctor
  }
})
```

mountComponent src\core\instance\lifecycle.js

定义updateComponent，挂载时调用_render获得VNode

`_render()` `src\core\instance\render.js`

生成VNode并返回

`vm.$createElement` `src\core\instance\render.js`

定义用户使用的createElement函数

`createElement` `src\core\vdom\create-element.js`

真正用来创建VNode的函数，如果遇上组件，则调用createComponent

`createComponent` `src\core\vdom\create-component.js`

自定义组件VNode创建函数

`VNode` `src\core\vdom\vnnode.js`

虚拟节点

获取并执行patch

获取patch：根据平台获取特定patch函数

执行patch：转换虚拟DOM真实DOM

`vm.update(vm.render(), hydrating)` `src\core\instance\lifecycle.js`

执行更新函数

`_patch_` `src\platforms\web\runtime\index.js`

指定patch函数

`patch` `src\platforms\web\runtime\patch.js`

获取平台特定的patch函数

传递dom节点操作和属性操作生成对应patch函数

patch

同层比较

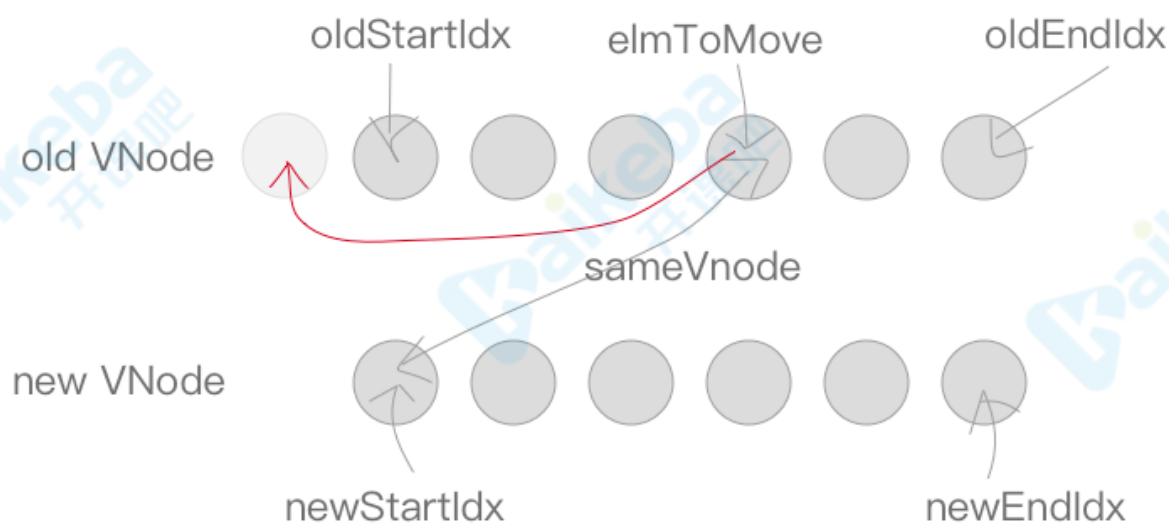
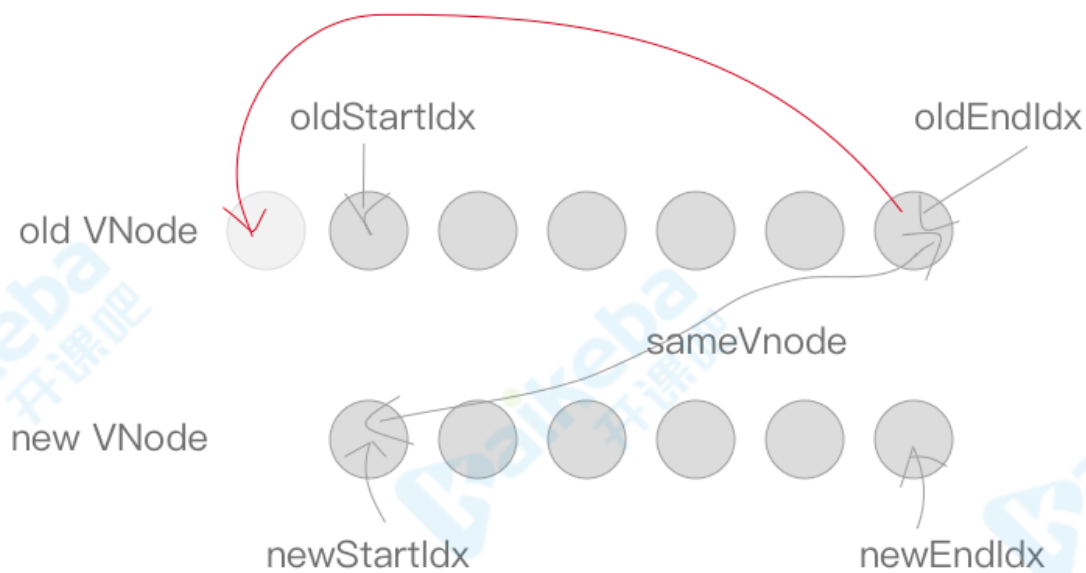


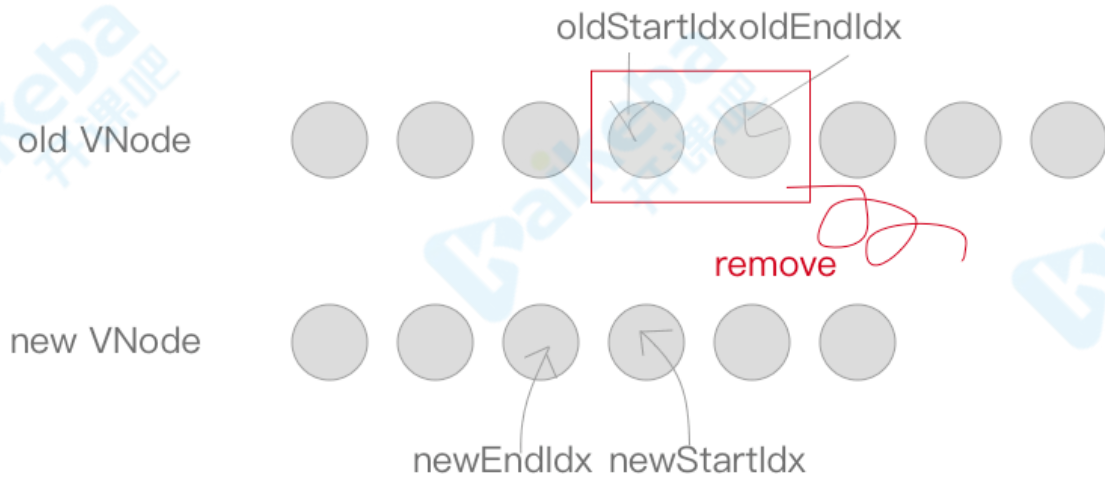
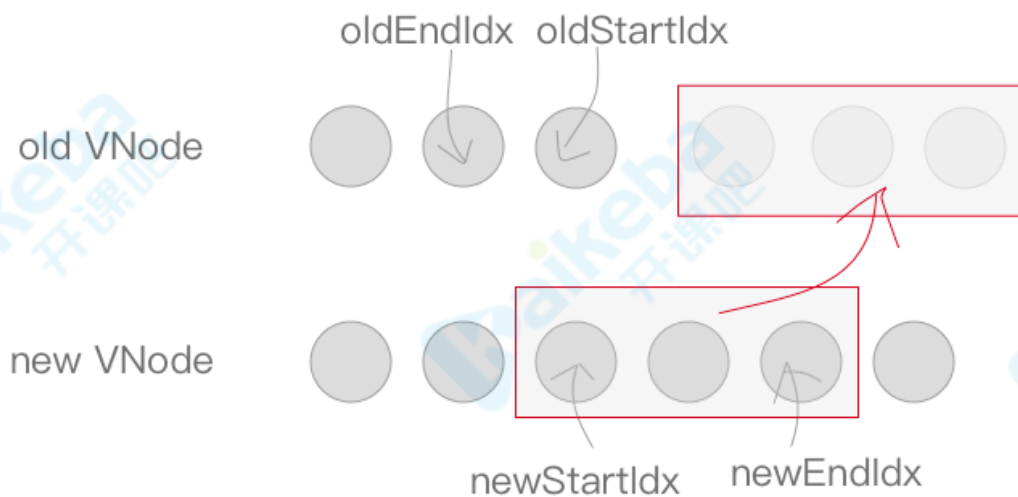
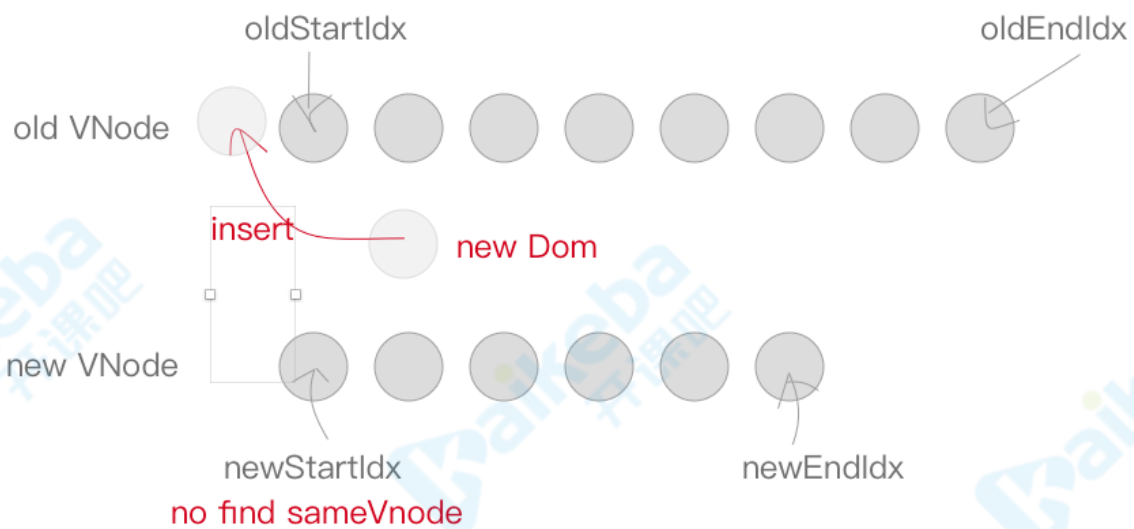
patchVnode

包括三种操作：属性或文本更新，子节点更新

updateChildren







属性操作