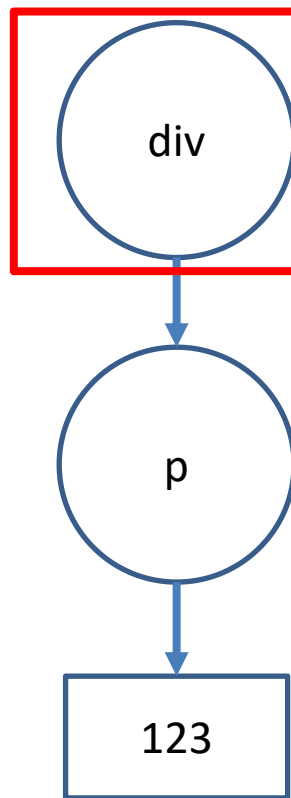


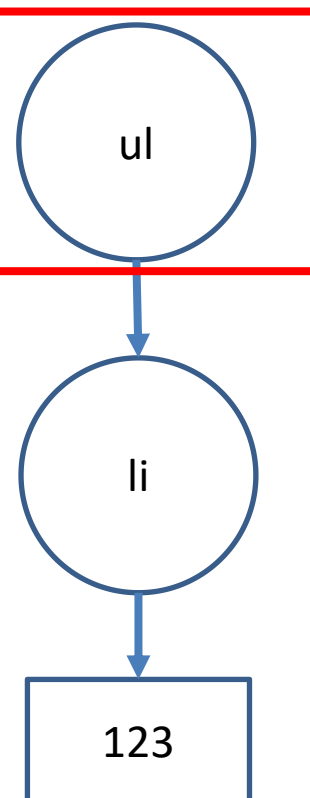
## diff算法

策略1：先同层级根元素比较。 => 如果根元素变化，那么不考虑复用，整个dom树删除重建

旧-虚拟DOM结构



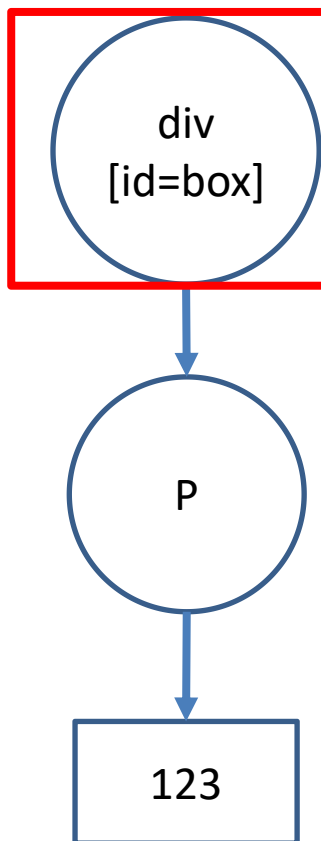
新-虚拟DOM结构



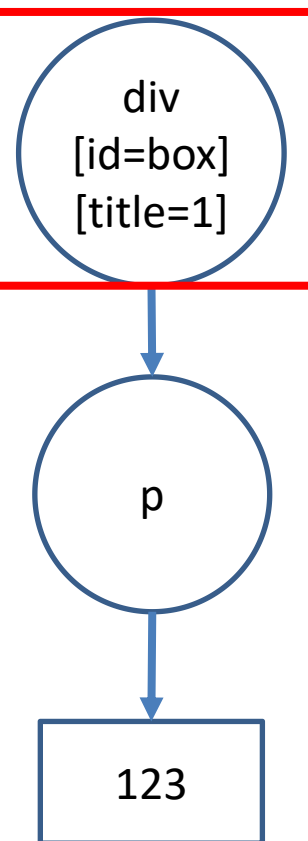
## diff算法

策略1：先同层级根元素比较。=> 如果根元素不变，对比出属性的变化更新，并考虑往下递归复用。

旧-虚拟DOM结构



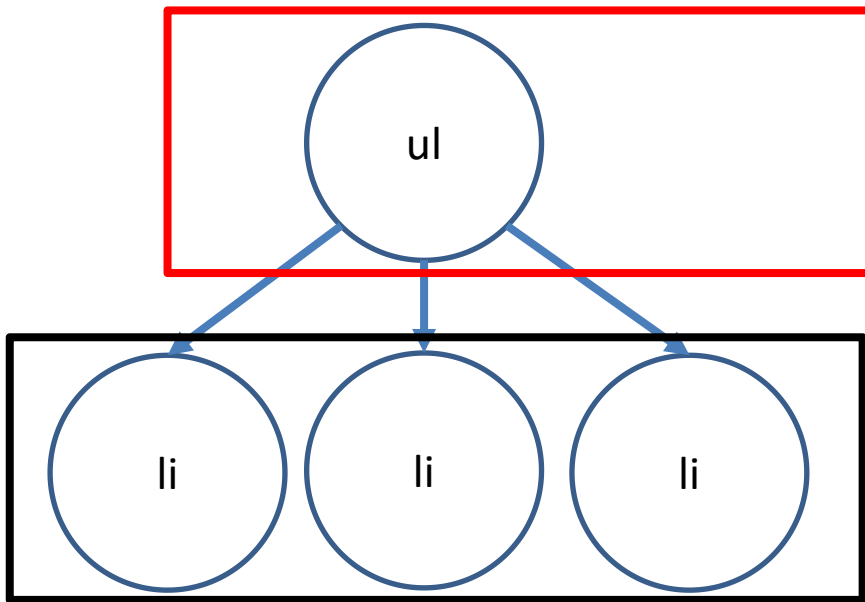
新-虚拟DOM结构



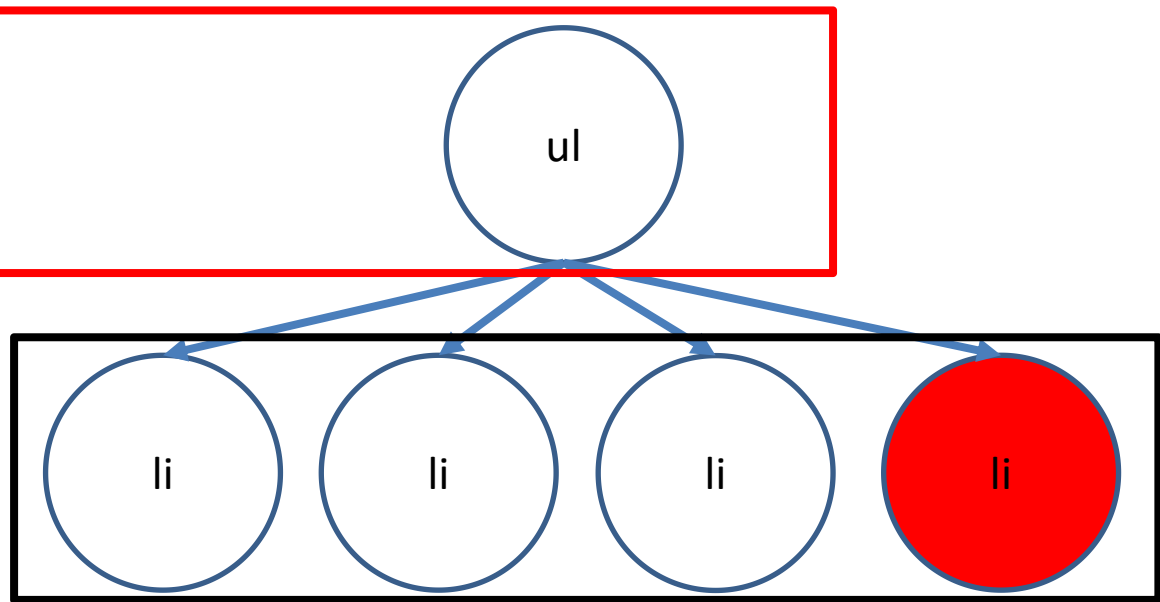
## diff算法

策略2：对比同级兄弟元素时，默认按照下标进行对比复用。

旧-虚拟DOM结构



新-虚拟DOM结构



对比同级兄弟元素时，如果指定了 key，就会按照相同 key 的元素来进行对比复用。

diff算法如何比较新旧虚拟DOM的呢？



## 小结

### 1. 同层级根元素先比较

- (1) 如果根元素变了，删除重建dom树
- (2) 如果根元素没变，对比属性。并考虑往下递归复用。

### 2. 兄弟元素比较

- (1) 默认按照下标，进行对比复用
- (2) 如果设置了key，就会按照相同key的元素进行复用



思考

同层级兄弟元素比较新旧变化，默认按下标比较，

如果设置了key，则优先相同key的兄弟元素比较。

那么设置 key 和 不设置key 有什么区别呢？ 设置key有什么用呢？

## key的作用 - 无 key 的情况

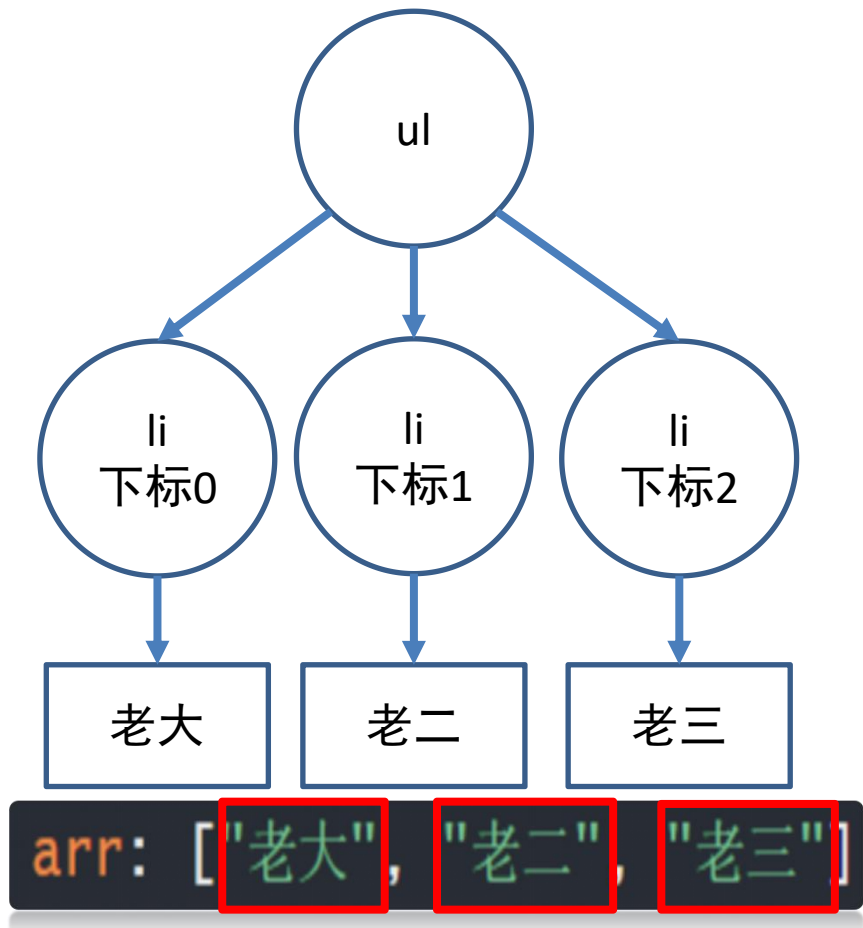
**无key的情况：**默认diff更新算法，是同级兄弟，按照 **下标** 对比新旧dom的差异。

- 老大
- 老二
- 老三

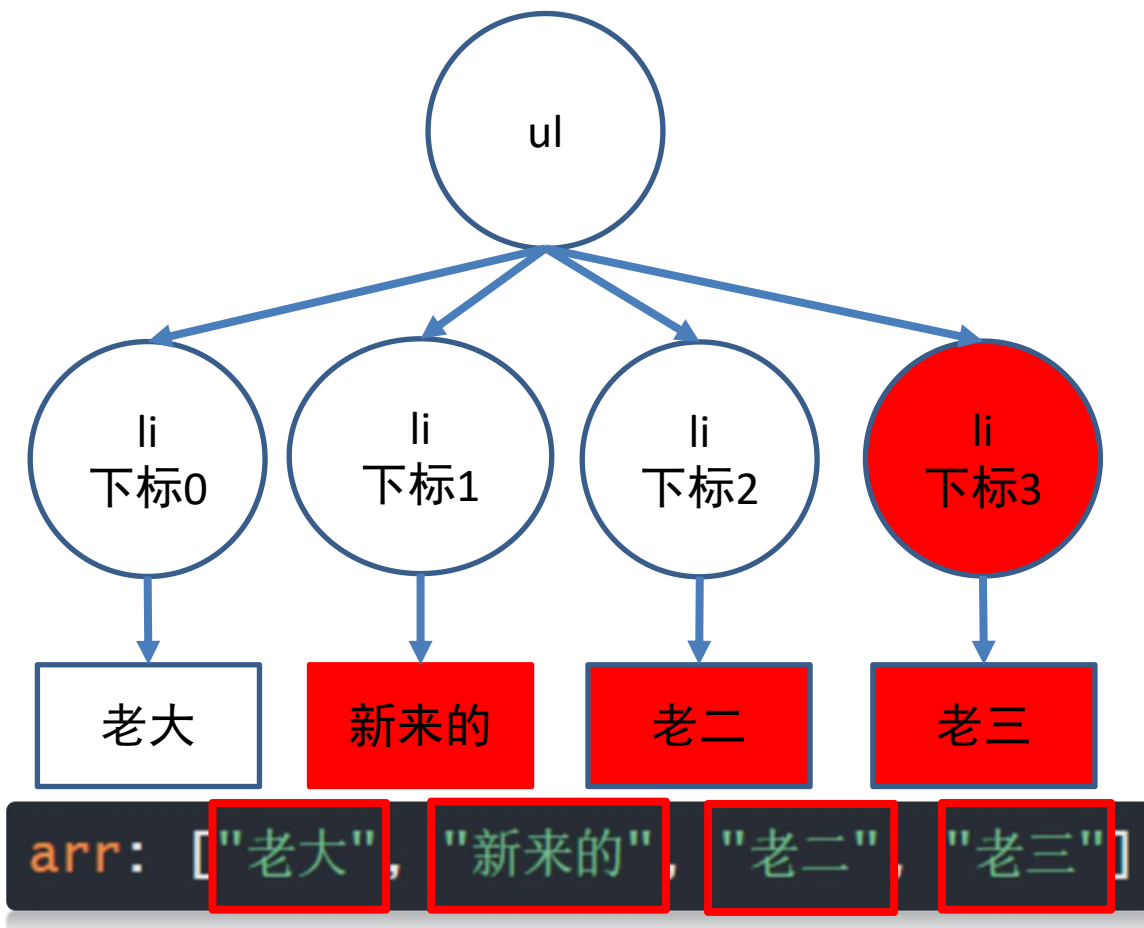
下标为1的位置新增一个

## key的作用 - 无 key 的情况（按照下标对比）

旧-虚拟DOM结构



新-虚拟DOM结构

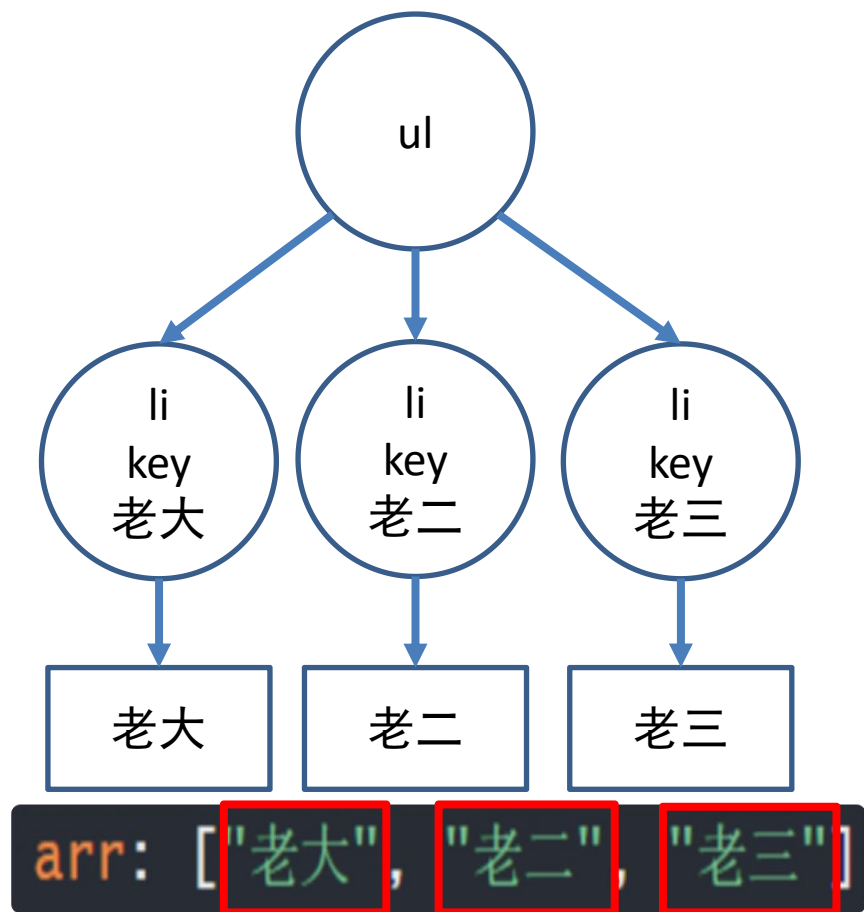


## key的作用 - 有 key 的情况

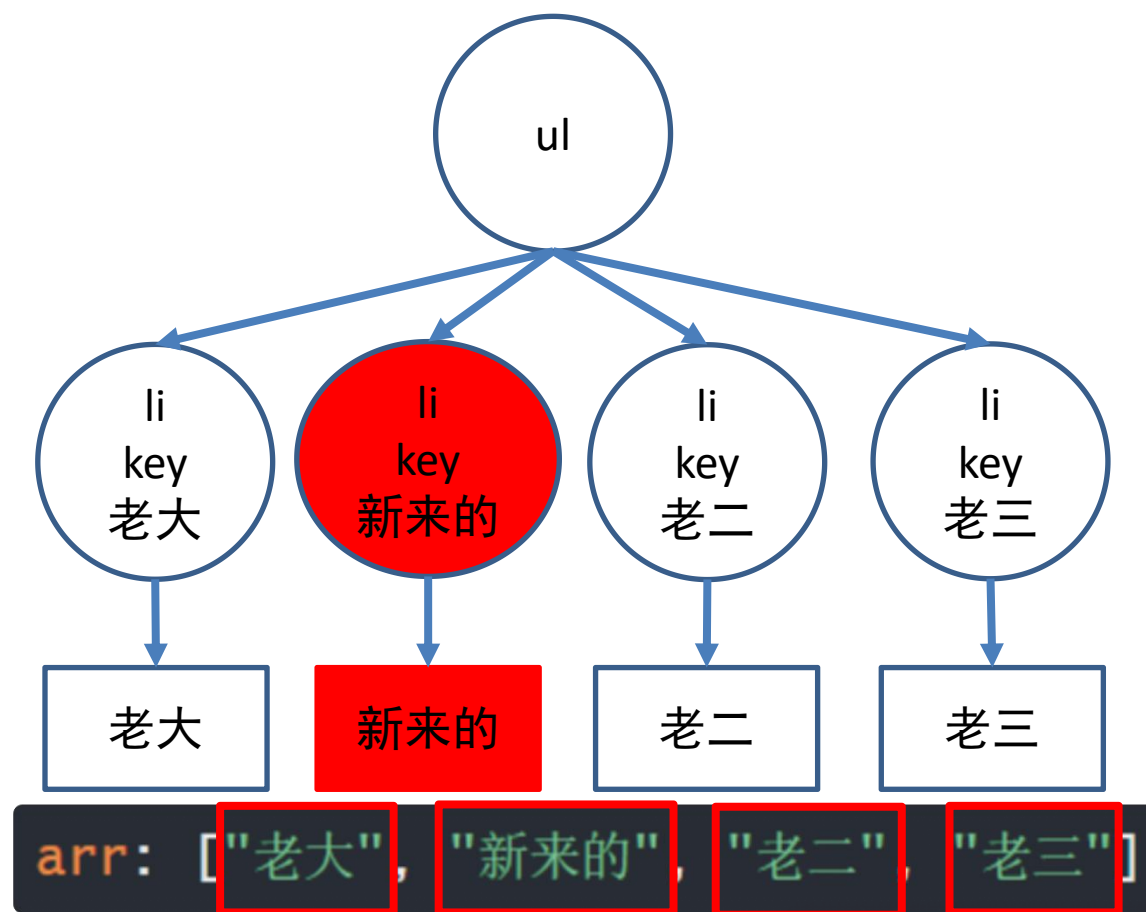
有key的情况：根据diff更新算法，同级兄弟元素，在设置了key后，会让相同key的元素进行对比。

key的要求：必须是字符串 或者 数字，且要保证唯一性！（标准的key需要指定成 id）

旧-虚拟DOM结构



新-虚拟DOM结构

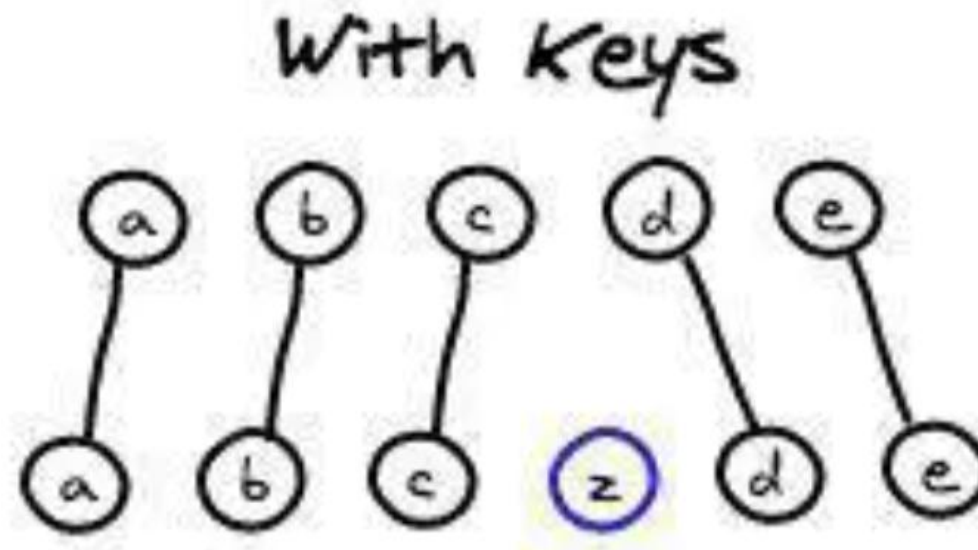
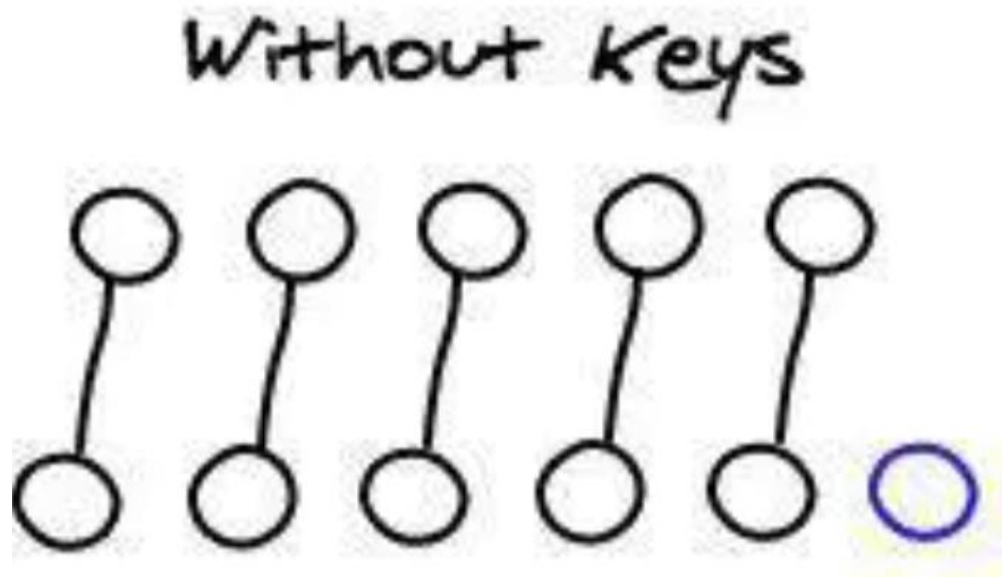




## key的作用

列表循环加:key="唯一标识", 可以标识元素的唯一性, 可以更好地区别各个元素。

key的作用: 提高虚拟DOM的对比复用性能





## 小结

### 1. 设置 和 不设置 key 有什么区别？

不设置 key，默认同级兄弟元素 **按照下标** 进行比较。

设置了key，按照 **相同key** 的新旧元素比较。

### 2. key值要求是？

**字符串或者数值，唯一不重复**

有 id 用 id，有唯一值用唯一值，实在都没有，才用索引

### 4. key的好处？

**key的作用：提高虚拟DOM的对比复用性能**



## 总结

vue就地复用策略？

对比虚拟dom的差异，就地（同层级，同位置）复用结构

为什么要对比虚拟dom呢？什么是虚拟dom呢？

真实dom太复杂。虚拟dom就是一个描述真实dom的对象。

diff算法如何比较新旧虚拟DOM？

1 先对比根元素

根元素改变 - 删除当前DOM树重新建

根元素未变 - 对比属性 - 更新属性，并考虑向下递归对比复用

2 同级兄弟元素，对比更新：

无key - 就地按下标更新 / 有key - 按key比较