

# 一、v-if、v-else-if、v-else

## 1.v-if、v-else-if、v-else

- 这三个指令与JavaScript的条件语句if、else、else if类似。
- Vue的条件指令可以根据表达式的值在DOM中渲染或销毁元素或组件

## 2.简单的案例演示

```
<h2 v-if="isShow">
  <div>abc</div>
  <div>abc</div>
  <div>abc</div>
  {{message}}</h2>
<h1 v-else>isShow为false时，显示我</h1>

<h2 v-if="score>=90">优秀</h2>
<h2 v-else-if="score>=80">良好</h2>
<h2 v-else-if="score>=60">及格</h2>
<h2 v-else>不及格</h2>
```

## 3.v-if的原理：

- v-if后面的条件为false时，对应的元素以及其子元素不会渲染
- 也就是根本不会有对应的标签出现在DOM中

# 二、条件渲染案例

## 1.我们来做一个简单的小案例：

- 用户在登录时，可以切换使用用户账号登录还是邮箱地址登录
- 类型如下情景：

用户邮箱

```
<div id="app">
  <span v-if="isUser">
    <label for="username">用户账号</label>
    <input type="text" id="username" placeholder="用户账号登录" />
  </span>
  <span v-else>
    <label for="email">用户邮箱</label>
    <input type="text" id="email" placeholder="用户邮箱登录" />
  </span>
  <button @click="isUser=!isUser">切换类型</button>
</div>
<script>
  const app = new Vue({
    el: '#app',
    data: {
      message: '你好哈',
      isUser: true
    },
  })
</script>
```

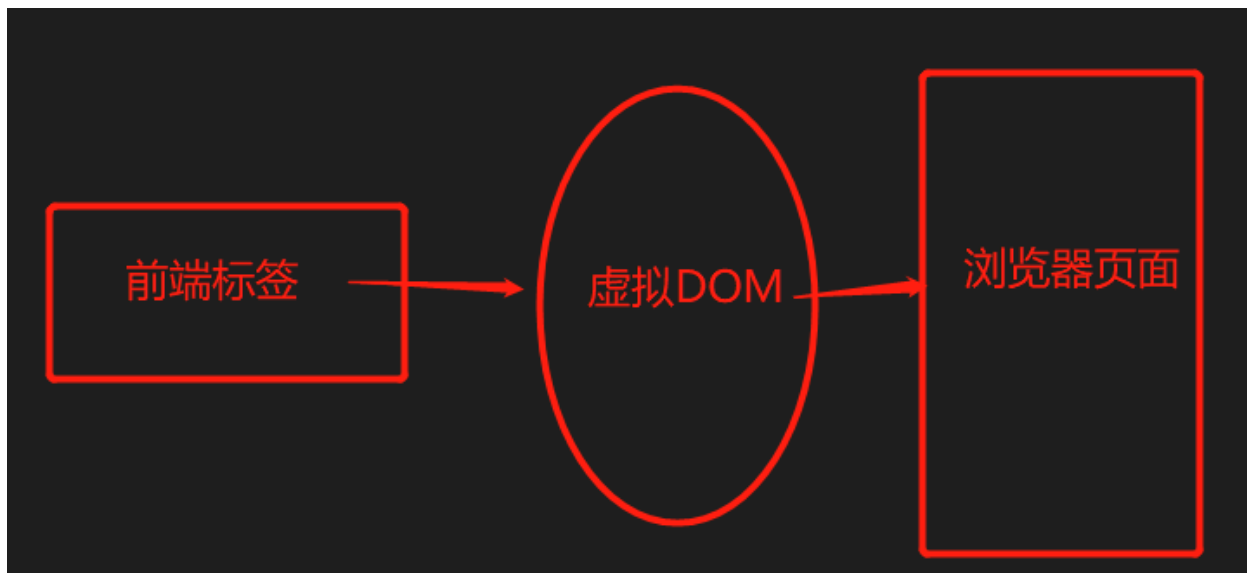
## 三、案例小问题

### 1.小问题:

- 如果我们在有输入内容的情况下，切换了类型，我们会发现文字依然显示之前输入的内容
- 但是按道理讲，我们应该切换到另外一个input元素中了。
- 在另一个input元素中，我们并没有输入内容。
- 为什么会出现这个问题呢？

### 2.问题解答

- 这是因为Vue在进行DOM渲染时，出于性能考虑，会尽可能的复用已经存在的元素，而不是重新创建新的元素。



- 在上面的案例中，Vue内部会发现原来的input元素不再使用，直接作为else中的input来使用了。

### 3.解决方案:

- 如果我们不希望Vue出现类似重复利用的问题，可以给对应的input添加key
- 并且我们需要保证key的不同

```
<div id="app">
  <span v-if="isUser">
    <label for="username">用户账号</label>
    <input type="text" id="username" placeholder="用户账号登录" key="username" >
  </span>
  <span v-else>
    <label for="email">用户邮箱</label>
    <input type="text" id="email" placeholder="用户邮箱登录" key="email" >
  </span>
  <button @click="isUser=!isUser">切换类型</button>
</div>
```

## 四、v-show

v-show的用法和v-if非常相似，也用于决定一个元素是否渲染

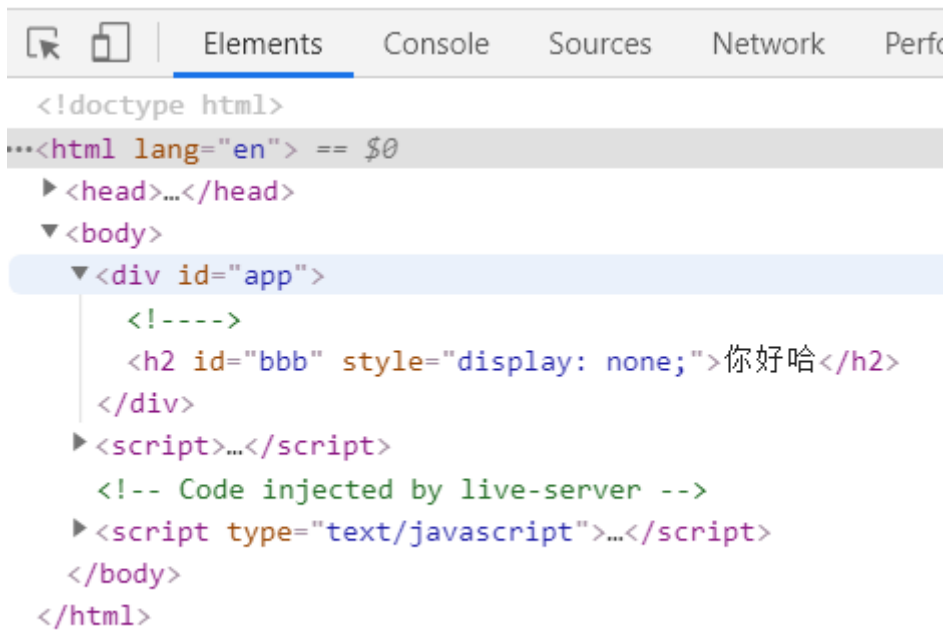
### v-if和v-show对比

1.v-if和v-show都可以决定一个元素是否渲染，那两者有什么区别？

- v-if 当条件为false时，压根不会有对应的元素在DOM中。
- v-show当条件为false时，仅仅将元素的display属性设置为none而已。

```
<!-- v-if:当条件为false时，包含v-if指令的元素，根本不会存在DOM中 -->
<h2 v-if="isShow" id="aaa">{{message}}</h2>

<!-- v-show:当条件为false时，v-show只是给我们的元素添加一个行内样式: display -->
<h2 v-show="isShow" id="bbb">{{message}}</h2>
```



```
<!doctype html>
...<html lang="en"> == $0
  <head>...</head>
  <body>
    <div id="app">
      <!-->
      <h2 id="bbb" style="display: none;">你好哈</h2>
    </div>
    <script>...</script>
    <!-- Code injected by live-server -->
    <script type="text/javascript">...</script>
  </body>
</html>
```

## 2.开发中如何选择?

- 当需要在显示和隐藏之间切换很频繁时, 使用v-show
- 当只有一次切换时, 通过使用v-if