

## 第五课 基础指令

### 学习目录

#### ➤ 基础指令

### 一. 基础指令

#### 1.v-show

根据指令属性值表达式的 true 与 false 值，切换元素的 display 显示与隐藏，v-show 的元素不管显示还是隐藏，元素本身始终会被渲染并保留在 DOM 中，v-show 不支持 <template> 元素。

```
<div v-show="showFlag">xingxingclassroom</div>
```

#### 2.v-if

根据指令属性值表达式的值来有条件的渲染元素。在表达式的值为 false 时，v-if 的元素及它的数据绑定与组件会被销毁，在表达式的值为 true 时，v-if 的元素会被重新建立。v-if 可以支持 <template> 元素。

```
<div v-if="num">xingxingclassroom</div>
```

#### ps: v-show 与 v-if 区别

1.v-if 是条件渲染，因为它会把符合条件要求的元素和组件进行销毁和重建，v-show 则不同，不管条件是什么，元素总是会被渲染，只是根据表达式值的不同简单的进行显示与隐藏。

2.v-if 如果条件表达式刚开始为 false, 则什么也不做, 直到条件表达式第一次变为 true 时, 才会渲染元素。

3.由于元素需要销毁与重建, 因此 v-if 有更多的切换开销, 而 v-show 有更多的初始渲染开销。所以, 如果条件表达式需要频繁切换, 那么可以使用 v-show 更好, 如果在运行时条件表达式很少改变, 那么使用 v-if 更好。

### 3.v-if、v-else、v-else-if

```
<div v-if="Math.random() * 10 > 3">随机数大于 3</div>
```

```
<div v-else-if="Math.random() * 10 > 5">随机数大于 5</div>
```

```
<div v-else>随机数小于 3</div>
```

v-else 与 v-else-if 前一个兄弟元素必须有 v-if 或 v-else-if。

### 4.v-for

把一个数组或者一个对象数据多次渲染一个元素或者一个模板块, 从而在视图上循环构建多个元素或者模板块。

#### v-for 循环数组

```
<div v-for="num in [100,200,300]">
```

```
  {{ num }}
```

```
</div>
```

```
var objArr = [
```

```
  {title:'标题 1',content:'内容 1'},
```

```
  {title:'标题 2',content:'内容 2'},
```

```
    {title:'标题 3',content:'内容 3'},  
  ];  
  
<div v-for="item in objArr">  
  
  <div>{{ item.title }}</div>  
  
  <div>{{ item.content }}</div>  
  
</div>
```

这里的 num、item 相当于当前数组元素的别名，v-for 也可以为在渲染的时候使用数组索引。

```
<div v-for="(item, index) in [100,200,300]"> </div>
```

在 v-for 中也可以使用其他定义好的数据，也可以使用 of 替代 in 来作为分隔符进行列表渲染。

```
<div v-for="item of [100,200,300]">  
  
  <div>{{num + item}}</div>  
  
</div>
```

### v-for 循环对象

```
var object = {  
  
  title: '标题 1',  
  
  content: '内容 1',  

```

num: 100

}

<div v-for="value in object"></div>

<div v-for="(value, key) in object"></div>

<div v-for="(value, key, index) in object"></div>

v-for 在遍历对象时，遍历的顺序不会像数组那样可以预期，不一样 js 的引擎可能会有不一样的遍历顺序。

因为 vue 是数据驱动视图改变，因此为了给 Vue 一个标记提示，方便它能跟踪每个节点的标记身份，在需要对视图元素进行重用和重新的时候能够快速的排序。因此你需要为每一个遍历元素提供一个唯一的 key 属性，这个 key 属性的属性值必须是唯一的，不能相同，重复的 key 属性值会造成渲染错误等问题。

<div v-for="item in items" :key="item.id"></div>

强烈建议大家养成习惯，每次使用 v-for 要加上 key 属性，这个 key 不仅仅是给 v-for 来使用，它是 vue 识别节点的通用机制。它主要用在 vue 的虚拟 dom 算法中，用来对比新旧节点从而识别出来 VNodes。

如果不使用 key，vue 会使用一种性能消耗比较多的方法就地修改或者复用相同类型元素的算法。而使用 key 时，vue 会基于根据 key 的标记变化重新排列元素对应的顺序，同时会移除没有 key 属性值的元素，这样在性能上消耗相对更小，更有利于视图更新。

## 谢谢观看！

我是星星课堂老师：周小周