

09. 循环列表渲染

学习要点：

1. 循环列表指令

本节课我们来开始学习 Vue3.x 的循环列表指令 `v-for` 的使用方式。

一. 循环列表指令

1. 使用 `v-for` 指令来渲染一个列表，其中 `array` 是数据源，`item` 是元素别名；

```
<ul>
  <li v-for="item in array">{{item.city}}</li>
</ul>
```

```
data() {
  return {
    array : [
      {
        city : '北京'
      },
      {
        city : '上海'
      },
      {
        city : '广州'
      },
      {
        city : '深圳'
      }
    ]
  }
},
```

PS: 也可以在遍历的元素项中设置 `index`，来获取有序编号，从 0 开始的；

```
<li v-for="(item, index) in array">{{index}}.{{item.city}}</li>
```

PS: 这里的 `in` 可以使用 JS 迭代器语法 `of`: `(item, index) of array`;

PS: `index` 参数在第二位置，`item` 在第一个参数位置；

2. 如果数据是一个对象，内部包含多个属性，那也可以使用 `v-for` 遍历；

```
<ul>
  <li v-for="item in object">{{item}}</li>
</ul>
```

```
data() {  
  return {  
    object : {  
      name : 'Mr.Lee',  
      gender : '男',  
      age : 100  
    }  
  }  
}
```

PS: 这里支持(value, name, key)的参数, 支持单独输出;

```
<li v-for="(value, name, key) in object">{{key}}.{{name}}.{{value}}</li>
```

3. 支持<template>进行多代码块的渲染方式, 比如:

```
<ul>  
  <template v-for="(item, index) in array">  
    <li>{{index}}</li>  
    <li>{{item.city}}</li>  
  </template>  
</ul>
```

4. v-for 支持数值循环渲染, 如下:

```
<span v-for="n in 10">{{n}}</span>
```

5. v-if 和 v-for 不推荐在同一元素上使用, 因为 if 比 for 权限高, 导致无法访问;
6. 那可以结合<template>来实现使用 if 来判断渲染, 具体如下:

```
<ul>  
  <template v-for="item in array">  
    <li v-if="item.city !== '上海'">{{item.city}}</li>  
  </template>  
</ul>
```