

## 07. 循环列表渲染

学习要点：

### 1.v-for 指令

本节课我们来开始学习 Vue 的 v-for 指令以及其它周边功能。

#### 一. v-for 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；

```
const dataObj = {  
  array : [  
    {  
      city : '北京'  
    },  
    {  
      city : '上海'  
    },  
    {  
      city : '深圳'  
    }  
  ],  
};  
  
<ul>  
  <li v-for="item in array">{{item.city}}</li>  
</ul>
```

PS: 插值使用 item.city 语法来获取每个元素；

2. 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;

3. v-for 指令也支持对象的遍历，并且支持(value, name, key)三组值；

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

```
const dataObj = {  
  object : {  
    name : 'Mr.Lee',  
    age : 100,  
    gender : '男'  }  
};
```

```
    }
  };
```

PS: 除了完全体，也可以使用: (value, name)或 value in object;

4. 支持使用<template>进行区块渲染;

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

5. v-for 也可以使用数值进行循环渲染;

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

6. 在循环中，我们可能要过滤掉某些数据，那就需要使用 v-if;

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

PS: 避免 v-for 和 v-if 在同一节点使用，由于 v-for 优先级比 v-if 高，意味着每次循环都要重复执行一遍;

7. 为了避免渲染本应该被隐藏的列表，我们把 v-if 应该放在 v-for 的外层;

```
<template v-if="flag">
  <li v-for="item in arrayCity">{{item.city}}</li>
</template>
```

8. 将属性更改为计算属性，过滤完毕之后再交给视图进行渲染输出;

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

```
computed : {
  arrayCity() {
    this.array.splice(1,1);
    return this.array;
  }
},
```