

列表渲染

v-for 与对象

使用 `v-for` 来遍历一个对象的所有属性。遍历的顺序会基于对该对象调用 `Object.keys()` 的返回值来决定。

```
const myObject = reactive({
  title: 'How to do lists in Vue',
  author: 'Jane Doe',
  publishedAt: '2016-04-10'
})
```

```
<ul>
  <li v-for="value in myObject">
    {{ value }}
  </li>
</ul>
// How to do lists in Vue
// Jane Doe
// 2016-04-10
```

可以通过提供第二个参数表示属性名 (例如 `key`):

```
<li v-for="(value, key) in myObject">
  {{ key }}: {{ value }}
</li>
// title: How to do lists in Vue
// author: Jane Doe
// publishedAt: 2016-04-10
```

第三个参数表示位置索引:

```
<li v-for="(value, key, index) in myObject">
  {{ index }}. {{ key }}: {{ value }}
</li>
// 0. title: How to do lists in Vue
// 1. author: Jane Doe
// 2. publishedAt: 2016-04-10
```

在 `v-for` 里使用范围值

`v-for` 可以直接接受一个整数值。在这种用例中，会将该模板基于 `1...n` 的取值范围重复多次。`n` 的初值是从 `1` 开始而非 `0`。

```
<span v-for="n in 10">{{ n }}</span>
// 12345678910
```

<template> 上的 v-for

与模板上的 `v-if` 类似，你也可以在 `<template>` 标签上使用 `v-for` 来渲染一个包含多个元素的块。例如：

```
<ul>
  <template v-for="item in items">
    <li>{{ item.msg }}</li>
    <li class="divider" role="presentation"></li>
  </template>
</ul>
```