Vue - The Complete Guide (incl. Router & Composition API)

-Maximilian Schwarzmüller

++

SECTION 3: RENDERING CONDITIONAL CONTENT & LISTS

39. Module Introduction   (1.35m)

40. Understanding the Problem   +   Resources   (3:16m)

41. Rendering Content Conditionally   (5:9m)
v-if: adds/removes element from DOM conditionally.
```
<p v-if="expression">Element added in DOM if expression output is truthy</p>
```

42. v-if, v-else and v-else-if   (4:18m)
Note: *v-if* and *v-else/v-else-if* must be adjacent elements.
```
<p v-if="expression_1">If expression 1 is truthy</p>
<p v-else-if="expression_2">Else if expression 2 is truthy</p>
<p v-else>Both above expressions are not truthy</p>
```

43. Using v-show Instead Of v-if   (2:25m)
v-show: hides/shows DOM elements using CSS classes. Used when element visibily changes more often.
```
<p v-show="expression">Show if expression output is truthy</p>
```

44. Rendering Lists of Data   (6:20m)
```
<ul>
   <li v-for="item in list">{{item}}</li> <!-- list = [1,2,3] -->
</ul>
```
Note: On dynamically adding new item, DOM doesn't re-render entire list, it just adds the new item element.

45. Diving Deeper Into v-for   (5:12m)
```
<p v-for="(element, index) in ['a', 'b']">{{index}}: {{element}}</p>
```
0: a

1: b
```
<p v-for="(value, key, index) in {a: 'Apple', b: 'Ball'}">{{index}} - {{key}}: {{value}}</p>
```
0 - a: Apple

1 - b: Ball
```
<span v-for="(number, index) in 5">{{index + ':' + number}}, </span>
```
0:1, 1:2, 2:3, 3:4, 4:5,

46. Removing List Items   (3:59m)

47. Lists & Keys   (8:26m)
key – special attribute to uniquely identify each element in a list, so that DOM updates element properly when it is modified.
```
<p v-for="item in [{id: 1, val: 'A'}, {id: 2, val: 'B'}]" :key="item.id">{{item.val}}</p>
```
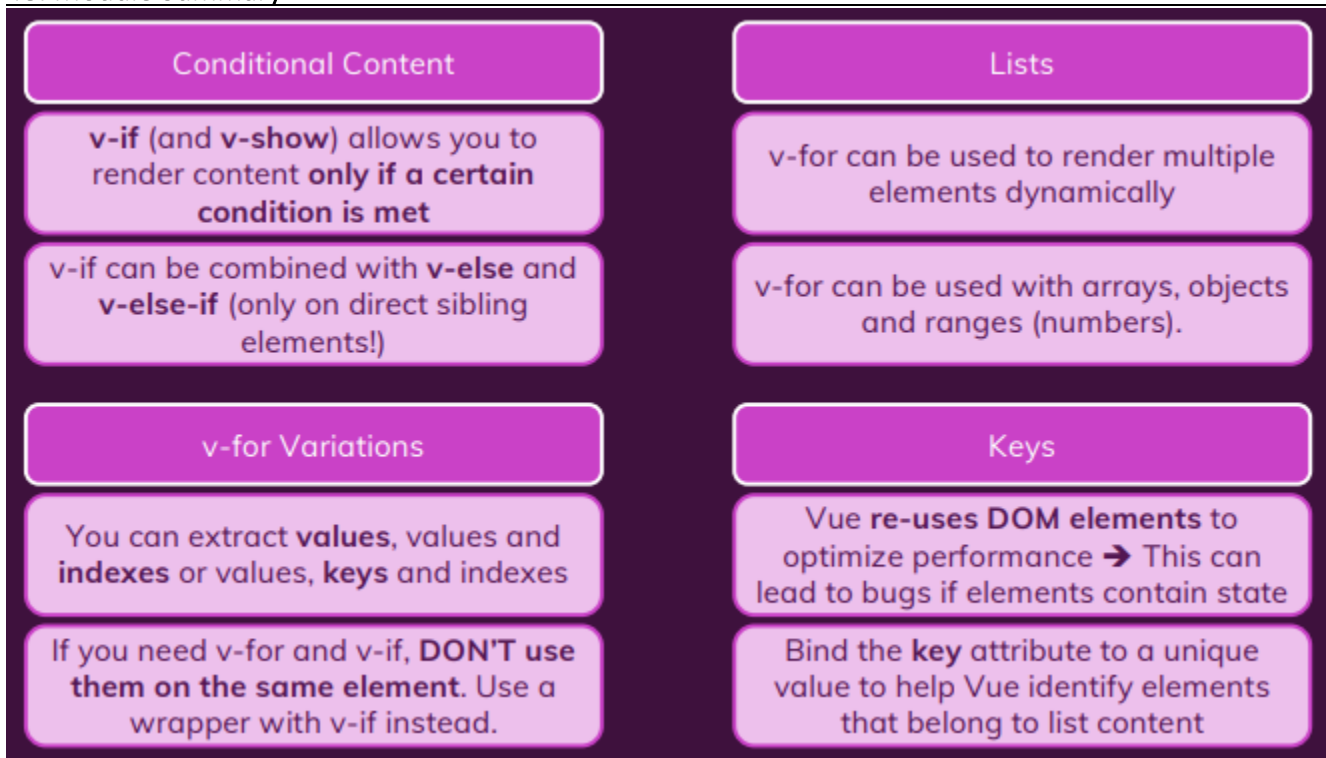
Vue - The Complete Guide (incl. Router & Composition API)

-Maximilian Schwarzmüller

+++

48. Module Summary   (2:15m)

Conditional Content	Lists
v-if (and **v-show**) allows you to render content **only if a certain condition is met**	v-for can be used to render multiple elements dynamically
v-if can be combined with **v-else** and **v-else-if** (only on direct sibling elements!)	v-for can be used with arrays, objects and ranges (numbers).

v-for Variations	Keys
You can extract **values**, values and **indexes** or values, **keys** and indexes	Vue **re-uses DOM elements** to optimize performance ➜ This can lead to bugs if elements contain state
If you need v-for and v-if, **DON'T use them on the same element**. Use a wrapper with v-if instead.	Bind the **key** attribute to a unique value to help Vue identify elements that belong to list content

Note: When both v-if and v-for are used to the same element, v-if is evaluated first. Hence, using v-if and v-for on the same element is not advised, owing to implicit precedence.

49. Module Resources

Resource: lists-cond-01-starting-setup