**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>

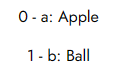
Note: On dynamically adding new item, DOM doesn’t re-render entire list, it just adds the new item element.

**45. Diving Deeper Into v-for (5:12m)**

  <p v-for="(element, index) in ['a', 'b']">{{index}}: {{element}}</p>



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



<span v-for="(number, index) in 5">{{index + ':' + number}}, </span>



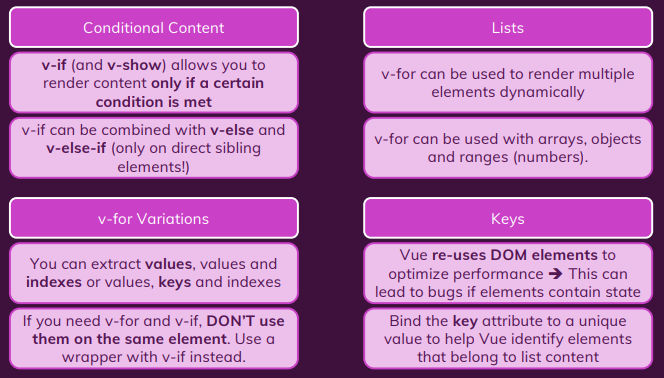
**46. Removing List Items (3:59m)**

**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>

**48. Module Summary (2:15m)**



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