Computed Properties and Watchers	The use of computed properties can usually be seen when we want to display data results that have a dependence on the reactive data that we have. watch property is used for several things, such as when doing a search based on the keywords typed by the user.
Class and Style Bindings	A common need for data binding is to manipulate the element's class list and inline styles. In addition to strings, expressions can also evaluate objects or arrays.
Conditional Rendering	The v-if instruction is used to conditionally render a block. The block will only render if the instruction expression returns true.
List Rendering	We can use the 'v-for' command to remove elements that are in an array. This v-for directive requires a special syntax that is item in items, where items means the initial data in the array and item
Event Handling	We can use the v-on directive to monitor DOM events and run some JavaScript when they are triggered.
Form Input Bindings	We can use v-model to create bidirectional data binding in form inputs, text areas, and select elements. It will automatically select the correct way to update the element based on the input type.
Components Basics	Components are reusable Vue instances with names
Component Registration	When registering a component, always give it a name. The name we give to the component may depend on where we are going to use it. When using components directly within the DOM, following the W3C Guidelines for creating custom tag names is highly recommended
Props	When we use templates in the DOM, the prop names in camelCase (Characteristics: all syllables are united and there are capital letters to separate them.) format need to use the kebab-case (The Kebab Case style uses a minus sign (-) to separate syllables. So that makes it look like a kebab) equivalent
Custom Events	Events do not provide automatic word transformation. The name of the event being emitted must be the same as the name of the event being listened to. Unlike components and props, event names are never used as variable or property names in JavaScript
Slots	Vue implements a content distribution API inspired by the Draft Web Part Specification, and the <slot> element can be used as a distribution outlet for our content.

Dynamic & Async Components	We can define an async component with Vue.component by passing a function that returns a promise. We can use keep-alive to maintain the current state of the component as it is being rendered dynamically with the component element.
Handling Edge Cases	In most cases, it is best to avoid reaching out to other component instances or manually manipulating DOM elements. However, there are cases where it has to be done.