Vue Definitions

* **Computed Properties and Watchers**-A Vue template can handle simple logic like {{message}}. However, if too many complex operators and logic is added to the template, then it becomes harder to understand what is happening in the template. Computed properties solve this problem by storing all the complex logic away from templates. Watch does the same thing, but it is more repetitive than a computed property.
* **Class and Style Bindings**-The v-bind directive dynamically binds 1 or more attributes to expressions. Data binding helps us manipulate an element’s class list and other expressions. When v-bind is used with class and style, Vue provides special enhancements where the expressions can evaluate to strings, objects, or arrays.
* **Conditional Rendering**-A certain code block will only be rendered if the v-if directive’s expression is a true value. If it is false, then the code block will be rendered according to the v-else directive’s value.
* **List Rendering**-The v-for directive can be used to iterate over an array and perform actions on each array item or just to simply display the array to the browser. It has a special syntax of “item in items” where items is the entire array while item is 1 element from the array.
* **Event Handling**-The v-on directive can be is used to listen to DOM events and runs JavaScript functions when certain actions are triggered. For example, the v-on directive can be used to perform a certain JavaScript function when the user clicks a button.
* **Form Input Bindings**-The v-model directive can bind data in a 2-way manner on form input or text areas. It automatically updates an element based on the correct input type specified in the Vue instance.
* **Components Basics**-Components are reusable Vue instances with unique names.They are created just like new Vue instances but can be called on by their name in other code blocks without repeating the code each time.
* **Component Registration**-Components can be registered or created either globally or locally. Globally registered components can be included in any new Vue instance while locally registered components are only available when they are called upon in a code block. Locally registered components avoid the unnecessary JavaScript occupation when a component is not being used.
* **Props**-Props or properties are used to pass data from a parent component to its child components where components are arranged in a tree hierarchical manner.
* **Custom Events**-Vue can listen to events on elements and trigger certain functions to run when the events are called. Events are used to emit some events between parent and child components. They are case-sensitive so exact syntax must be always maintained.
* **Slots**-Slots can pass data outside of the regular parent-child interactions. Any type of content like HTML can be placed in new places.
* **Dynamic and Async Components**-Dynamic components can be used to cache user action or data so that when the user changes the state of a component, he or she does not lose their previous activity. Asynchronous components allows a browser to only load a component from the server when it’s necessary in order to display the webpage as quickly as possible.
* **Handling Edge Cases**-These are unique Vue instances where regular Vue rules do not apply, and out-of-the-box approaches can be implemented.