

Name: Vanessa Josephine Mattea

ID: 221458997

#### Task 4) New JavaScript Concepts / Components

These concepts / components will be taught in detail in the upcoming weeks, thus this task only required me to try and understand their definitions and why we need these new concepts. I went to Vue.js' website to read about these concepts and components and then write a short reflection.

Components	Brief explanation
Computer properties	Computer properties allow us to compute a much more complex logic and prevent templates becoming bloated and hard to maintain, meanwhile watchers is useful to compute operations that would react to data changes.
Class and style bindings	Class and style bindings such as v-bind, can be used to manipulate element's attributes (i.e class and style) easily.
Conditional rendering	Conditional rendering such as v-if, is used like any other conditional statement in other programming languages. It will render a block of code, only if the expression's value is a truthy.
List rendering	List rendering such as v-for is used to render a list of items within an array.
Event handling	Event handling such as v-on is used to listen to DOM events, then when triggered, it can run JavaScript code.
Form input bindings	Form input bindings such as v-model is used to automatically chooses the correct way to update data on user input events.
Components basics	Components are Vue instances that are named and reusable (e.g <button-counter>. Components accept the same options as new Vue, i.e data, computed, watch, etc, except a few root-specific options such as el.
Component registration	To register a component, we must assign a concise yet meaningful name, and we can use different name casings, such as kebab-case or PascalCase. We can also register the component globally, which means that it can be used in the template of any 'new Vue' instance, or locally.
Props	Props are custom attributes that can be registered on a component, which is used to pass data from parent components to child components.
Custom events	In Vue.js, custom events enable custom inputs which then can work with v-model (v-model is basically a two-way binding). So instead of using v-bind to the value and v-on to the input, I can just use v-model. This saves a lot of time and also easier to implement on the web app.
Slots	Slots enable us to pass content to components. Slots can also be utilized as reusable templates which can render different content that is based on input props. We can bind slot with a template with the aid of v-slot. We can also give a meaningful name to the slot.

Name: Vanessa Josephine Mattea

ID: 221458997

Dynamic and async components	Dynamic components allow us to switch between components seamlessly. We can use a special attribute 'is' or an element <keep-alive> for dynamic components. Async components allow us to render certain component when needed and will cache the result for future re-renders.
Handling edge cases	Unusual situations can happen which requires us to bend Vue's rules a bit. In most scenarios, it's best practice to not mess with other component instances, however there are some cases when it is appropriate to do so, such as accessing the root instance and accessing the parent component instance of a web app.