Notes on applying keyframes to an existing component  
  
(Way that will have the least hiccups on implementing the transition animation)  
To have a certain div undergo a transition under some user action, it’s easiest to wrap the div in a <transition> element. This element should be v-bound to enter-active-class, which checks whether the transition element should be active or not. For example, to allow the Pane to slide in and out,

:enter-active-class="(position =='left')? 'slide-to-the-left-enter' :'slide-to-the-right-enter'"

Was used, where position where should be 'left' if the element should be active (which in this case, means open). Note, the function which is bound to enter-active-class should **not** specify any details of the desired animation, only the starting and end points if necessary. Specifying animation properties in the function leads to bugs!  
  
For completeness, to have the pane open and close, the full element

<transition name="slide" :enter-active-class="(position =='left')? 'slide-to-the-left-enter' :'slide-to-the-right-enter'"

:leave-active-class="(position =='left')? 'slide-to-the-left-leave' :'slide-to-the-right-leave'" appear>

Was used.

To set an animation, the details of the animation are set as a CSS class. For example here, slide-to-the-left is called when position is 'left'.

.slide-to-the-left-enter{

animation: slideleft 1s;

}

Here, slideleft is a keyframes function which is called later. Although this might seem redundant, it’s useful as it means code can be kept tidy, as similar animations can be used the same way without needing to write extra code. For example, for the pane, a reverse of slide to the left was needed as a CSS class, and so we could write.slide-to-the-left-leave{

animation: slideleft 1s;

animation-direction: reverse;

}

Which is easy to read, and well organised. This attaches the element to the animation.   
A keyframes function needs a name, and some actions. For example here, the name of the keyframes function is slideleft, and the actions are to translate the element to the left.

@keyframes slideleft {

0%{transform: translateX(-100%);}

100%{transform:translateX(0);}

}

Take some time to check out what you can do using keyframes!