In Vue.js, event modifiers are special directives that you can use to modify the behavior of DOM events when they are triggered. There are four different event modifiers available in Vue.js:

stop: Stops the propagation of the event so that it does not bubble up to parent elements.

Example: <a @click.stop="doSomething">Click me</a>

prevent: Prevents the default action associated with an event from occurring (e.g., submitting a form or following a link).

Example: <form @submit.prevent="onSubmit">...</form>

capture: Listens for an event during the capture phase instead of during bubbling (i.e., before it reaches its target element). This modifier is rarely used in practice.

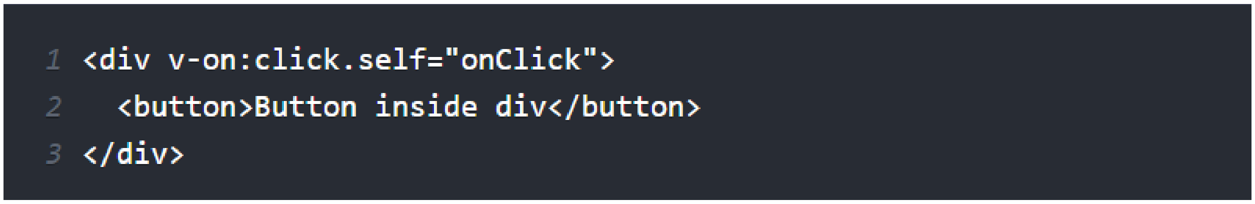
Example: <div @click.capture="doSomething">...</div>

passive: Improves scroll performance by indicating to the browser that no preventDefault() call will be made on this listener, which allows browsers to optimize scrolling and avoid potential stutters.

Example: <div @touchmove.passive="onTouchMoveHandler" />

Additionally, there is another modifier called **self**, which only triggers an event if it originated from the same element where its listener was attached.

Example:



```
<div v-on:click.self="onClick">
  <button>Button inside div</button>
</div>
```

In this example, clicking on anywhere other than button tag inside div won't trigger onClick method as we have added .self.