

JAVA

MODULE 4 - DAY 13

Event Handling with Vue.js

TODAY'S OBJECTIVES

- Adding Behavior to Components
 - “methods” section
- v-on
 - Keyboard Events
 - Mouse Events
- Passing arguments to event handlers
- v-on modifiers
 - Submit.prevent
 - Stop
 - Once
 - Passive

METHODS

```
export default {  
  name: 'order-form',  
  methods: {  
    displayFormData() {  
      // Method logic goes here  
    },  
    anotherMethod() {  
      // Method logic goes here  
    }  
  }  
};
```

V-ON

v-on is added directly to the HTML, similar to v-bind or v-model.

```
<a v-on:click="alertUser">DELETE</a>
```

```
<input type="text" name="firstName" v-on:change="updateName" />
```

KEYBOARD EVENTS

```
<textarea v-on:keyup.enter="processEnterKey"></textarea>
```

Special keys that you can listen for are:

- `.enter`
- `.tab`
- `.delete` (captures both "Delete" and "Backspace" keys)
- `.esc`
- `.space`
- `.up`
- `.down`
- `.left`
- `.right`

```
<input type="text" v-on:keyup.74="handleJ" />
```

MOUSE EVENTS

We can also listen for which mouse button was clicked using the same sort of trick. We can listen for different mouse buttons that triggered a `click` or `dblclick` event with the following flags:

- `.left`
- `.right`
- `.middle`

For example, we can attach these to a UI element with:

```
<li v-on:click.left="updateTotal" v-on:click.right="activateContextMenu">Add to Cart</li>
```

EVENT HANDLER ARGUMENTS

- If we don't specify arguments in our v-on value, the event object will be implicitly passed.
- If we pass arguments, we must explicitly pass the event object if we need access to it.

Implicit Event Object

```
<a href="#" id="increase" class="btn" v-on:click="updateCounter"
```

Argument Passed, event object not passed

```
<a href="#" id="increase" class="btn" v-on:click="updateCounter(1)"
```

Argument Passed, event object explicitly passed

```
<a href="#" id="increase" class="btn" v-on:click="updateCounter(2,$event)"
```

V-ON MODIFIERS

<https://vuejs.org/v2/guide/events.html#Event-Modifiers>

- `.stop`
- `.prevent`
- `.capture`
- `.self`
- `.once`
- `.passive`

```
<!-- the click event's propagation will be stopped -->
```

```
<a v-on:click.stop="doThis"></a>
```

```
<!-- the submit event will no longer reload the page -->
```

```
<form v-on:submit.prevent="onSubmit"></form>
```

```
<!-- modifiers can be chained -->
```

```
<a v-on:click.stop.prevent="doThat"></a>
```


LECTURE CODE: EVENT HANDLERS

INDIVIDUAL EXERCISE

Due Friday, 7/24, 9am