

Module 3 Day 12

Vue Methods &
Vue Event Handling



Module 3 Day 12

Can you ?

- Implement event handling using Vue using the `v-on` directive
- Use "event modifiers", like propagation and default, in Vue
- Use the `methods` : property of the Vue object to define methods for a component
- Utilize component methods inside of event handlers

Vue Methods

Before tackling handlers we will introduce one more piece of Vue.JS, the Vue method.

- A Vue method is similar to a function or method in other languages - they are called when needed, optionally taking in parameters, and providing some kind of output.
- Just like the computed section, the ***methods*** section is JavaScript object of the export default object in the script section of a Vue component.

Vue Methods vs Computed Properties

Methods and Computed properties were designed for different purposes.

- A ***computed*** property is used to return derived data based on the data in your JSON model or other source. These are used in the same way as derived properties of a class.
- A ***method*** is used to define a process that manipulates the state of the Model or the View itself. Methods may also return values and accept parameters. Vue methods resemble functions in other languages and may be invoked within the component.

Defining Vue Methods

Vue methods go into their own section, they are a peer of the data and computed sections.

```
<script>
export default {
  name: "product-review",
  data() {
    ...
  },
  computed: {
    ...
  },
  methods: {
    //your methods go here
  }
}
</script>
```

Defining Vue Methods

Vue methods are defined in a similar fashion as computed properties, with successive methods split by a comma:

```
methods: {  
  numberOfReviews(reviews, starType) {  
    return reviews.reduce( (currentCount, review) => {  
      return currentCount + ( review.rating === starType ? 1 : 0 );  
    }, 0 );  
  },  
  
  addNewReview() {  
    this.reviews.unshift(this.newReview);  
    this.resetForm();  
  },  
  
  resetForm() {  
    this.showForm = false;  
    this.newReview = {};  
  }  
}
```

- Here we have three distinct methods being defined.
- The first method shows that a method can take on parameters and return a value.

Calling Vue Methods

Vue methods are flexible and can be called in the following ways:

- Within a v-on directive in the template section (more on this later)
- By a computed property: When we do this, the computed property needs to take a parameter called “vm” which stands for the current Vue instance: i.e. **vm.myMethod()**;
- By another function.

Event Handling Review

- Last week, in JavaScript Event Handling, we added event listeners to DOM elements so that certain actions might be taken in response to events that take place on the web page.
- The Vue framework provides a directive to do this within components.

The v-on directive

- The v-on directive is used with the following pattern:

v-on: **event**= **'action to take'**

- Here are some examples:

Here we saying: when the user clicks on the span, set the component 'filter' property to 0.

```
<span class="amount" v-on:click="filter = 0">{{ averageRating }}</span>
```

Here we saying: when the user submits the form, call the method **addNewReivew *without posting back to the server***

```
<form v-if="showForm === true" v-on:submit.prevent="addNewReview">
```

Event modifiers: prevent

- The v-on directive can be modified with a prevent keyword, which prevents the default behavior of a HTML element from executing:

```
<form v-if="showForm === true"  
v-on:submit.prevent="addNewReview">
```

Event Modifiers: stop

- The v-on directive can also be modified with a stop keyword, disabling event bubbling up the DOM.
- For a full overview of Vue events, please see the official documentation: <https://vuejs.org/v2/guide/events.html>