

Module 3-12

VUE Methods
VUE Event Handling

VUE Methods

Before tackling handlers we will introduce one more tool to our repertoire, 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 with the computed section, the methods section is comprised of JavaScript, thus should be part of the script section in a VUE component.

VUE Methods vs Computed Properties

— — —
Methods and Computed properties were designed for different purposes.

- You use a computed property, to generate “derived data” in which your output is based on the data in your JSON data model.
 - Computed values are cached.
- You use a method when you want a tool that resembles a functions in other languages.
 - Methods are executed only when called.

Defining VUE Methods

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

```
<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 work flexibly and can be called in the following contexts:

- Within a v-on directive in the template section (more on this later)
- By another function.

Let's Create Some Methods

Event Handling Review

— — —

- Recall that a few lectures ago 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 facilitate this.

The v-on directive

- The v-on directive takes on 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 JSON data 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**

```
<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">
```

Note that we are overriding the default behavior of the form submission, and instead choosing to handle the scenario ourselves with our own method.

Event Modifiers: stop

— — —

- The v-on directive can be modified with a stop keyword, disabling event bubbling up the DOM.

Let's Implement Some Event Handlers