

VUE METHODS & EVENT HANDLERS

VUE METHODS

- Vue methods are similar to a function or method in other languages - they are called when needed, optionally taking in parameters and possibly providing some kind of output.
- Just like the **computed** section, the **methods** section is JavaScript and goes inside the **<script>** section.

VUE METHODS VS. COMPUTED PROPERTIES

- Computed properties:
 - They return values based on your component's internal data (they are derived).
 - They are cached. If the value isn't different every time computed properties are a better choice.
- Methods are executed only when called and can execute code based on input parameters

EXAMPLE OF ADDING THE METHODS SECTION

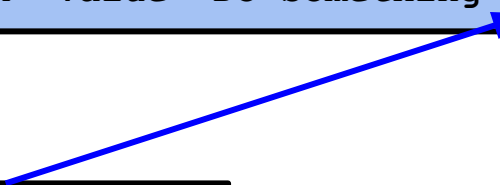
```
<script>
export default {
  data() {
    // data
  },
  computed: {
    // computed properties
  },
  methods: {
    // methods
  }
}
</script>
```

THE V-ON DIRECTIVE

- The **v-on:** directive acts similarly to an eventListener in vanilla JavaScript
- You specify the event that you want an element to respond to using the **v-on:** syntax:

```
<input type="button" value="Do Something" v-on:click="handleButtonClick" />
```

v-on: for click event



handleButtonClick() method will be called when a click event occurs.



THE V-ON DIRECTIVE

- You can pass the actual event to a method called by `v-on:` by passing `$event` as a parameter to the method.

```
<input type="button" value="Do Something"  
  v-on:click="handleButtonClick($event)" />
```

- You can use the `@` symbol as shorthand for `v-on:`:

```
<input type="button" value="Do Something"  
  @click="handleButtonClick($event)" />
```

THE V-ON:KEYUP DIRECTIVE

- **v-on:keyup** can be used to listen for a key
 - `v-on:keyup.enter="processEnterKey"`
- Can listen for:
 - `.enter`
 - `.tab`
 - `.delete` (captures both "Delete" and "Backspace" keys)
 - `.esc`
 - `.space`
 - `.up`
 - `.down`
 - `.left`
 - `.right`
- Can listen for a specific key.
 - `v-on:keyup.j="handleJ"`

PREVENT AND STOP

- The `v-on:` directive can be modified to add the `preventDefault` and `stopPropagation` behaviors:
 - `v-on:submit.prevent="doFormSubmit"`
 - `v-on:submit.stop="doFormSubmit"`
 - Can be chained:
 - `v-on:submit.stop.prevent="doFormSubmit"`

V-SHOW VS. V-IF

- **v-show** can be used to show/hide element
 - **v-show="isSelected"** will show element if **isSelected** is **true**, hide otherwise
 - choose unless there is reason to use **v-if**
- **v-if** can be used to include/remove element in/from DOM
 - **v-if="isSelected"** will include element in DOM if **isSelected** is **true**, will remove otherwise.
 - preferred for clearing out forms

EXERCISE NOTES

- You can use the `find` function of an array the same way you use `filter` but it will return one item (if found) rather than an array like `filter`.
- If you connect multiple checkboxes to the same array using `v-model`, the checkbox values will be added or removed from the array based on whether the checkbox is checked or not.