Vue.js template syntax

WASA: Web and Software Architecture

Enrico Bassetti

Text interpolation

Text interpolation is done via "Mustache" syntax (double curly braces).

```
<span>Message: {{ msg }}</span>
```

If msg contains Hello world!, the HTML output will be:

```
<span>Message: Hello world!</span>
```

Safe: HTML is automatically escaped. E.g., If msg is <b>Hello:

```
<span>Message: &lt;b&gt;Hello</span>
```

JavaScript expressions

You can use **JavaScript expression**:

```
{{ number + 1 }}

{{ ok ? 'YES' : 'NO' }}

{{ message.split('').reverse().join('') }}
```

JavaScript expressions

You **cannot use** flow control or statements:

```
<!-- won't work -->
{{ var a = 1 }}

<!-- won't work -->
{{ if (ok) { return message } }}
```

JavaScript expressions

Calling a function is permitted:

```
{{ formatDate(date) }}
```

Note: Vue.js will call these functions at each update.

Vue.js directives

In templates, Vue.js adds custom attributes to HTML tags.

These attributes are called **directives**.

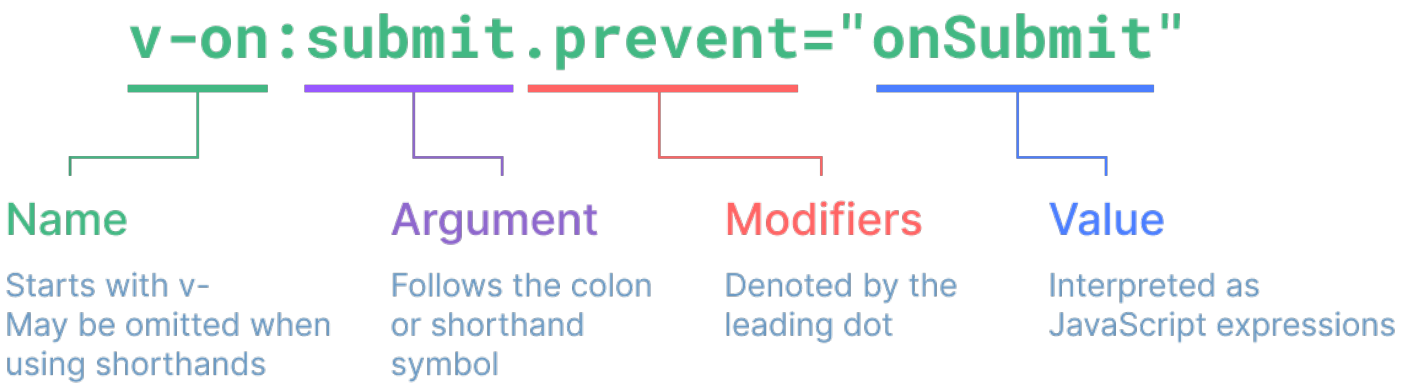

Image (C) by Vue.js documentation

Vue.js built-in directives

- v-text: set the text content (safe: HTML auto-escaped)
- v-html: set the HTML content (**unsafe**, use v-text instead)
- v-show: toggle element visibility
- v-if, v-else-if, and v-else: toggle element rendering
- v-for: repeat current element (and subtree)
- v-on: register event listener
- v-bind: bind attribute to a variable
- v-model: two way binding with a variable

Less common directive:
https://vuejs.org/api/built-in-directives.html

Element content

Set the element text content to the value of a variable.

```
<span v-text="msg"></span>
```

If msg is updated, Vue.js will update the content accordingly.

Events

Execute a function when something happens (e.g., a click).

```
<span v-on:click="alert('Hello!')">World</span>
```

On HTML elements, all standard events are supported: https://developer.mozilla.org/en-US/docs/Web/Events. When used on SFC, you can use SFC events (more on this later).

A shortcut:

```
<span @click="alert('Hello!')">World</span>
```

Attribute bindings

v-bind creates a binding from JavaScript to the element attribute.

```
<span v-bind:id="dynamicId"></span>
```

In this example, the id attribute for this will be set to the value of dynamicId JavaScript variable.

A shortcut:

```
<span :id="dynamicId"></span>
```

Attribute two-way bindings

v-model creates a two-way binding.

```
<input type="text" v-model="name" />
```

Only for: <input>, <select>, <textarea> and Vue.js components.

v-bind vs v-model

- v-bind is for displaying data from JavaScript to the user
- v-model is for displaying data and letting the user edit data

v-model is similar to v-bind + v-on and some glue code.

Conditional rendering

To conditionally render a part of the template, use v-if:

```
<p v-if="showMessage">
  Vue.js will render this part only when showMessage is true
</p>
```

Conditional visibility

To show/hide a part of the template, use v-show:

```
<p v-show="showMessage">
  This part is visible only when showMessage is true
</p>
```

Note: the element is still rendered, but it's invisible!

Looping

v-for will repeat the element (and its content) for each iteration.

```
<div v-for="item in items">{{ item }}</div>
```

In this example, if items is the array ['a', 'b'], the resulting HTML is:

```
<div>a</div>
<div>b</div>
```

Links

- https://vuejs.org/guide/essentials/template-syntax.html
- https://vuejs.org/api/built-in-directives.html
- https://developer.mozilla.org/en-US/docs/Web/Events