

VUE ESSENTIALS CHEAT SHEET

(BetterForms v1)



Vue Mastery

EXPRESSIONS

```
<div>
  <p>I have a {{ model.product }}</p>
  <p>{{ model.product + 's' }}</p>
  <p>{{ model.isWorking ? 'YES' : 'NO' }}</p>
  <p>{{ product.getSalePrice() }}</p>
</div>
```

DIRECTIVES

Element inserted/removed based on truthiness:

```
<p v-if="model.inStock">{{ model.product }}</p>
```

```
<p v-else-
  if="model.onSale">...</p> <p
  v-else>...</p>
```

Toggles the display: none CSS property:

```
<p v-show="model.showProductDetails">...</p>
```

Two-way data binding:

```
<input v-model="model.firstName" >
```

v-model.lazy="..." Syncs input after change event

v-model.number="..." Always returns a number

v-model.trim="..." Strips whitespace

LIST RENDERING

```
<li v-for="item in model.items" :key="item.id">
  {{ item }}
</li>
```

key always recommended

To access the position in the array:

```
<li v-for="(item, index) in model.items">...
```

To iterate through objects:

```
<li v-for="(value, key) in model.object">...
```

Using v-for with a component:

```
<cart-product v-for="item in model.products"
  :product="item" :key="item.id">
```

BINDING

```
<a v-bind:href="model.url">...</a>
```

shorthand

```
<a :href="model.url">...</a>
```

True or false will add or remove attribute:

```
<button :disabled="model.isButtonDisabled">...
```

If isActive is truthy, the class 'active' will appear:

```
<div :class="{ active: model.isActive }">...
```

Style color set to value of activeColor:

```
<div :style="{ color: model.activeColor }">
```

ACTIONS / EVENTS

Calls addToCart method on component:

```
<button v-on:click="namedAction('myNamedAct',{})">...
```

shorthand

```
<button @click="namedAction( ...
```

Arguments can be passed:

```
<button @click="namedAction('myNamedAct',{key:123})">...
```

To prevent default behavior (e.g. page reload):

```
<form @submit.prevent="namedAction('myNamedAct',{})">
```

Only trigger once:

```
<img @mouseover.once="namedAction('myNamedAct',
  {})">...
```

.stop

Stop all event propagation

.self

Only trigger if event.target is element itself

Keyboard entry example:

```
<input @keyup.enter="namedAction('myNamedAct',{})">
```

Call onCopy when control-c is pressed:

```
<input @keyup.ctrl.c="namedAction('myNamedAct',{})">
```

Key modifiers:

.tab	.up	.ctrl
.delete	.down	.alt
.esc	.left	.shift
.space	.right	.meta

Mouse modifiers:

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



Need help on your path to Vue Mastery?
Checkout our tutorials on **VueMastery.com**