Accessibility,
Inputs &
Vue

Isn’t the Web Accessible?

«The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect.»

Sir Tim Berners-Lee, 1997

https://www.w3.org/Press/IPO-announce

It is.

In theory.

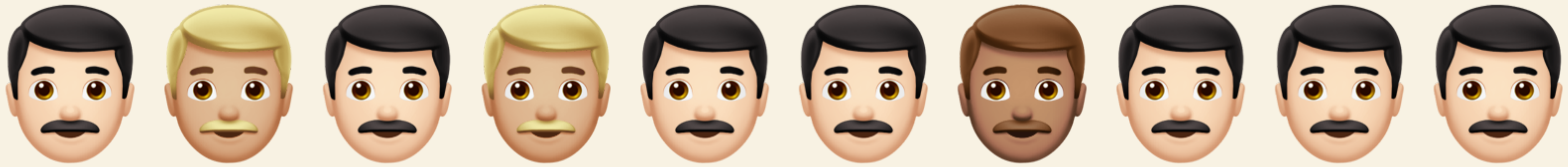
«59,653,607 distinct accessibility errors were detected across the 1 million home pages—an average of 59.6 errors per page.»

The WebAIM Million – Errors and Error Density, 2019
https://webaim.org/projects/million/

Neglect

WCAG Failure Type	# of home pages	% of home pages
Low contrast text	852,868	85.3%
Missing alternative text for images	679,964	68%
Empty links	581,408	58.1%
Missing form input labels	528,482	52.8%
Missing document language	329,612	33.1%
Empty buttons	250,367	25%

Framework	# (and %) of home pages	Avg. # of errors	% decrease/increase of errors
MooTools	27,033 (2.7%)	52.3	−12.6%
TweenMax	25,802 (2.6%)	58.3	−2.3%
Mustache	11,803 (1.2%)	65.5	9.9%
Knockout.js	7,046 (0.7%)	65.6	10.0%
React	31,960 (3.2%)	65.5	10.1%
Backbone.js	22,678 (2.3%)	69.3	16.6%
Angular	14,939 (1.5%)	70.9	19.2%
RequireJS	21,168 (2.1%)	71.1	19.7%
Handlebars	15,307 (1.5%)	73.5	23.6%
Vue.js	6,639 (0.7%)	73.7	23.7%
Prototype	14,376 (1.4%)	74.2	24.8%



It’s not us. It’s society.

But also us.

#PerfMatters CONFERENCE

April 2 & 3, 2019, Redwood City, CA

www.PerfMattersConf.com

How Privilege Defines Performance

Tatiana Mac

Web Accessibility In Context

https://www.smashingmagazine.com/2019/06/web-accessibility-context/

We need to act.

If we don’t want to look like a bunch of overpaid bollocks.

Complexity

🚀

Accessibility is no rocket science.

Babelfish

WAI

W3C Web Accessibility Initiative

https://www.w3.org/WAI/

ARIA

Accessible Rich Internet Applications

https://www.w3.org/WAI/standards-guidelines/aria/

WCAG

Web Content Accessibility Guidelines

https://www.w3.org/WAI/standards-guidelines/wcag/

a11y

Shorthand for accessibility

Making the Web Accessible

> «The only way this work gets done is if we start small, and if we work together.»

Ethan Marcotte – The web we broke
https://ethanmarcotte.com/wrote/the-web-we-broke/

Collaboration

The A11Y Project

https://a11yproject.com/

A11Y

STYLE GUIDE

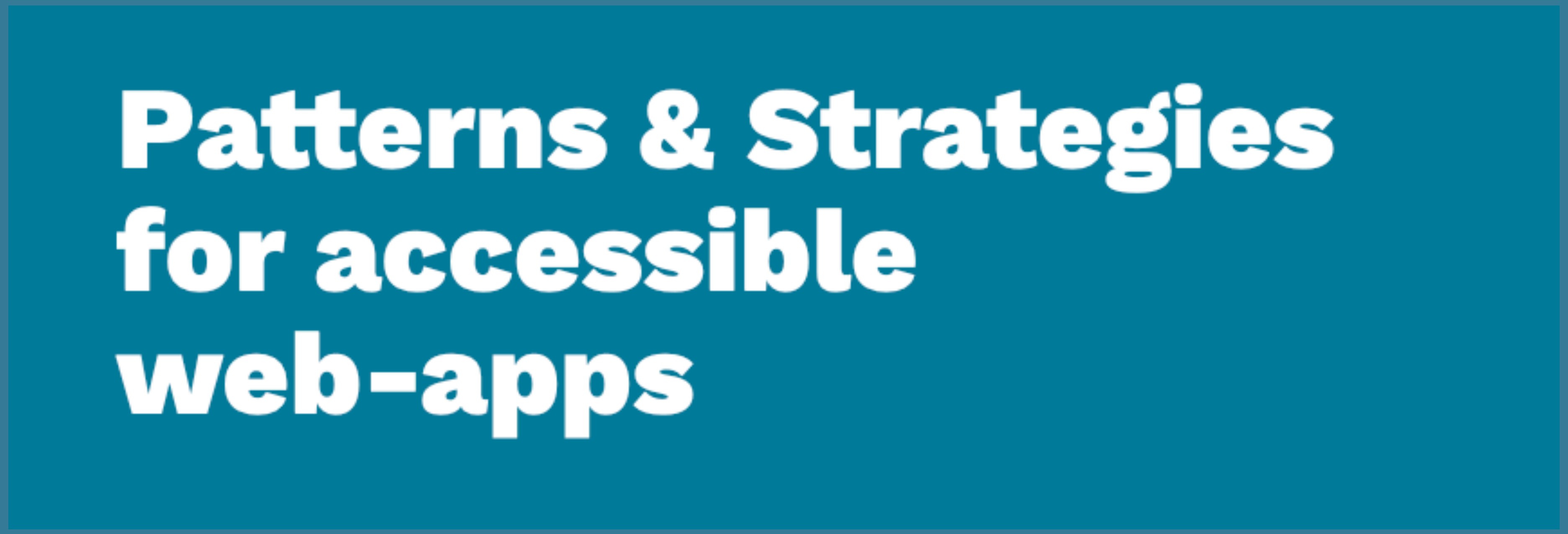
https://a11y-style-guide.com/

https://inclusive-components.design/

Inclusive
Components

Reach UI

https://ui.reach.tech/



Patterns & Strategies
for accessible
web-apps

https://accessible-app.com/

du Deque University

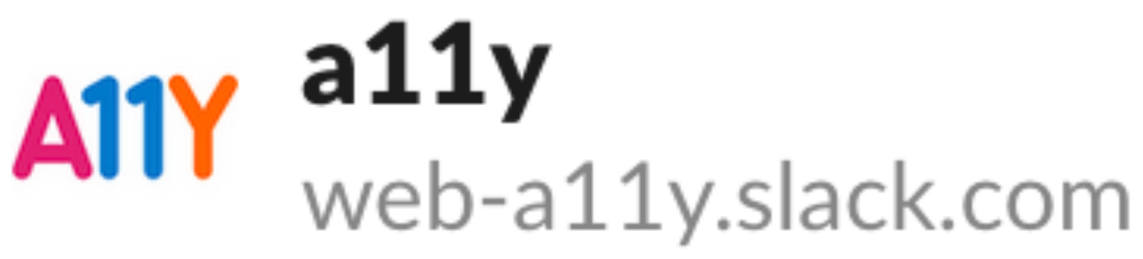
https://dequeuniversity.com/

axess lab

https://axesslab.com/

A11yWeekly

https://a11yweekly.com/



a11y

web-a11y.slack.com

Use semantic markup.

And ARIA where applicable.

Use Assistive Technology.

Ask users.

They are the experts.

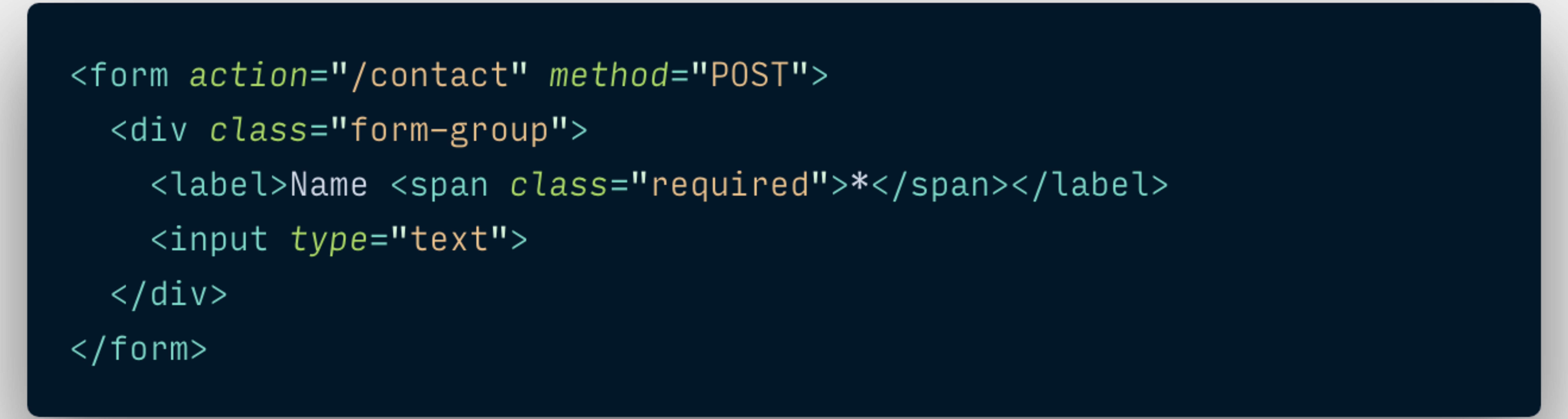
Put Accessibility in the Definition of Done

… and be it your own.

Accessible Inputs

> «59% of the 3.4 million form inputs identified were unlabeled (either via `<label>`, `aria-label`, or `aria-labelledby`).»

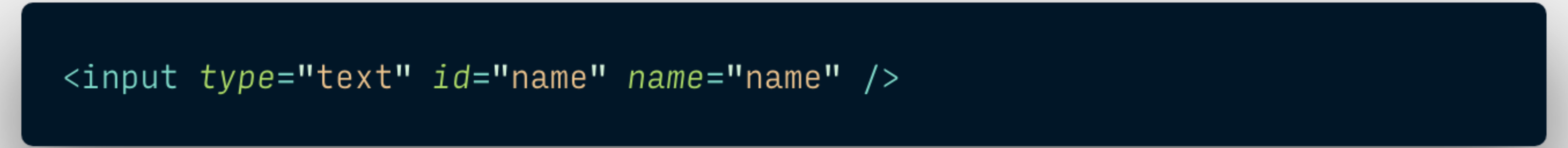
The WebAIM Million – Form Labeling



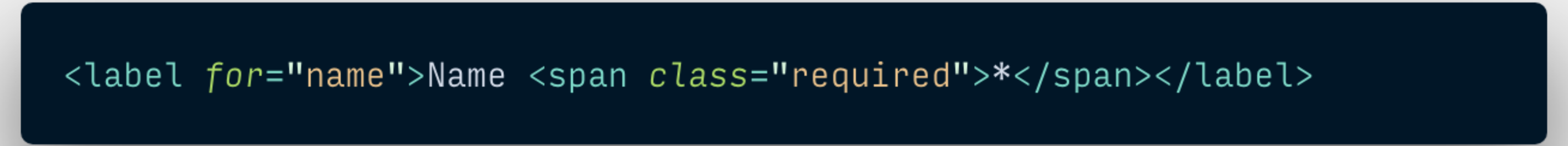
```html
<form action="/contact" method="POST">
  <div class="form-group">
    <label>Name <span class="required">*</span></label>
    <input type="text">
  </div>
</form>
```

× edit text blank

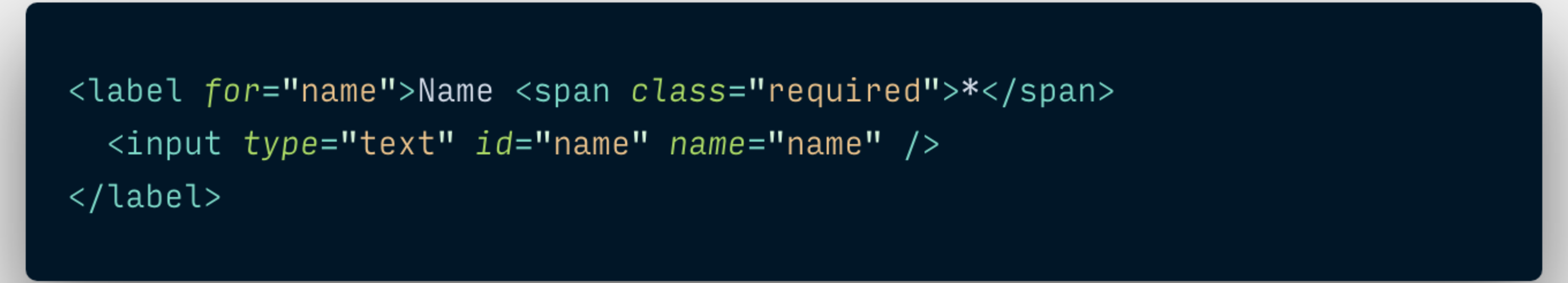
Labeling



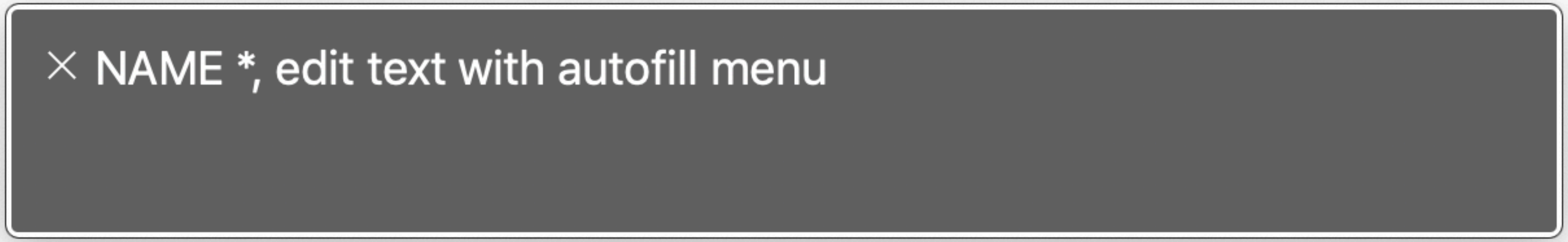
```html
<input type="text" id="name" name="name" />
```



```html
<label for="name">Name <span class="required">*</span></label>
```

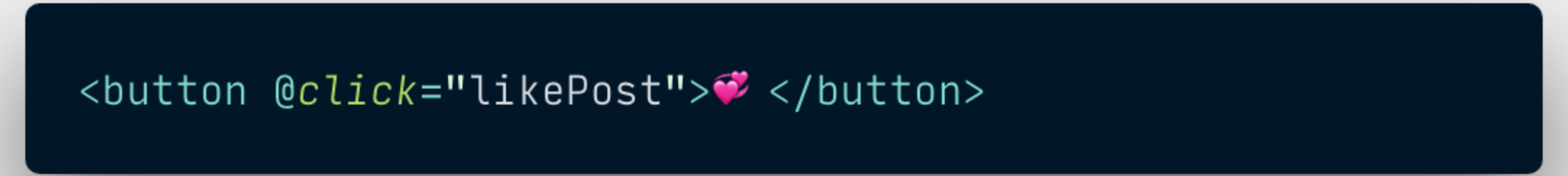


```
<label for="name">Name <span class="required">*</span>
  <input type="text" id="name" name="name" />
</label>
```

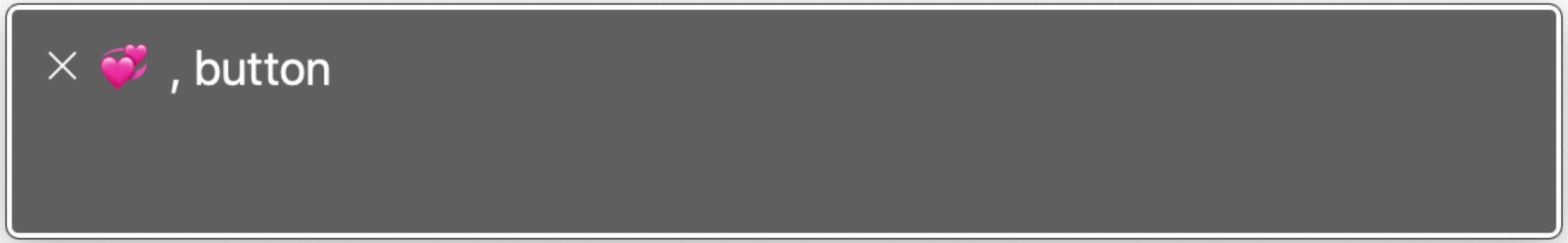


NAME *, edit text with autofill menu

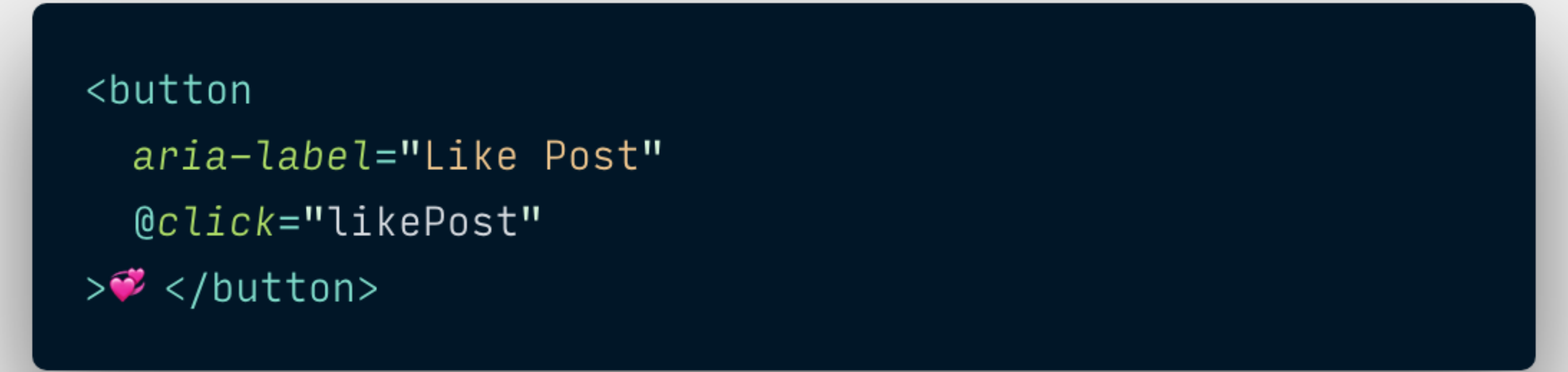
`aria-label`



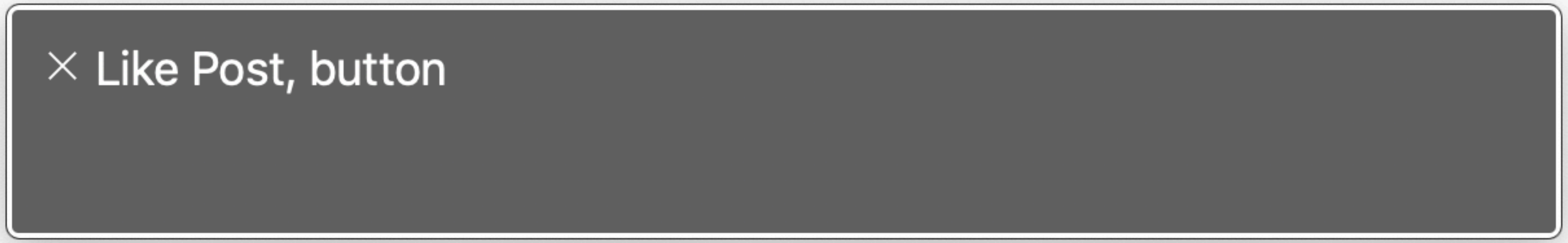
```
<button @click="likePost">💞 </button>
```



× 💞 , button



```
<button
  aria-label="Like Post"
  @click="likePost"
>💕 </button>
```

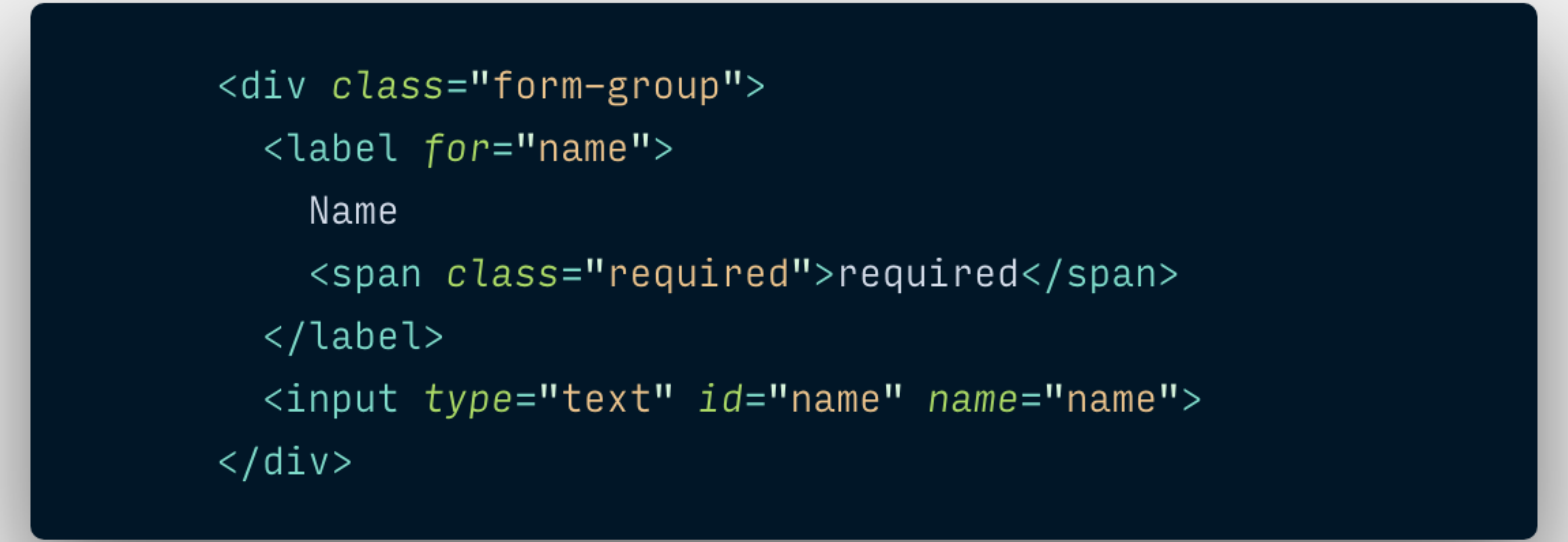


× Like Post, button

Accessible Icon Buttons

https://www.sarasoueidan.com/blog/accessible-icon-buttons/

Marking required fields



```
<div class="form-group">
  <label for="name">
    Name
    <span class="required">required</span>
  </label>
  <input type="text" id="name" name="name">
</div>
```

NAME REQUIRED, edit text

« WAI-ARIA provides a collection of accessibility states and properties […].»

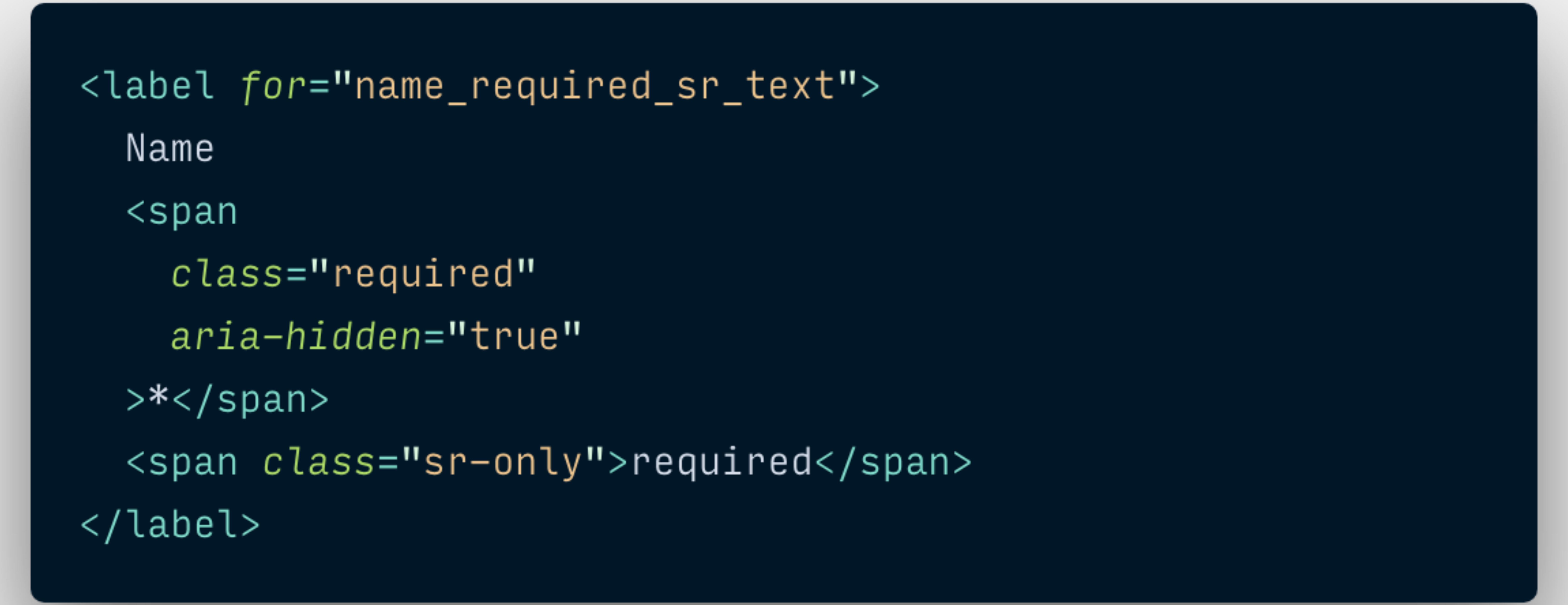
https://www.w3.org/TR/wai-aria/#usage

`aria-hidden`

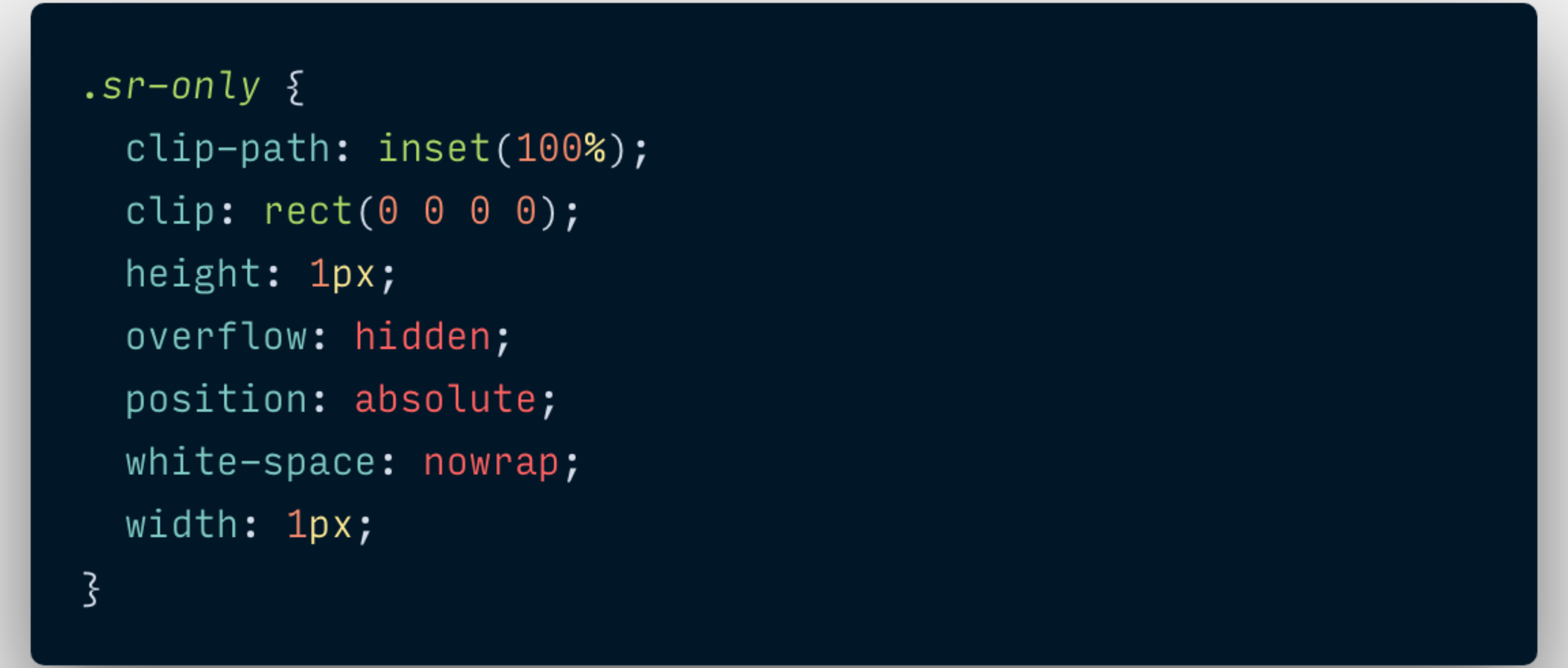
Hide an element from Assistive Technology

Hiding elements from screen readers using `aria-hidden`

https://www.accessibility-developer-guide.com/examples/sensible-aria-usage/hidden/

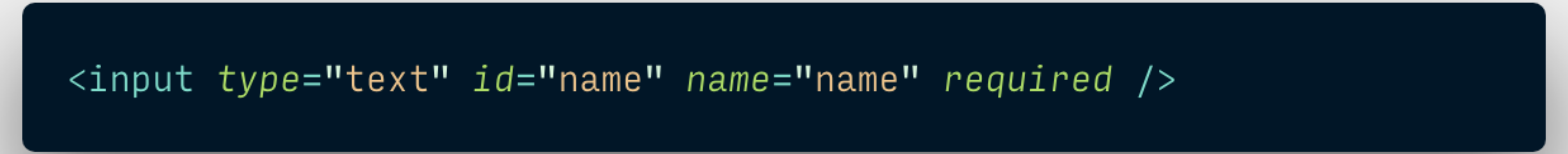


```html
<label for="name_required_sr_text">
  Name
  <span
    class="required"
    aria-hidden="true"
  >*</span>
  <span class="sr-only">required</span>
</label>
```



```css
.sr-only {
  clip-path: inset(100%);
  clip: rect(0 0 0 0);
  height: 1px;
  overflow: hidden;
  position: absolute;
  white-space: nowrap;
  width: 1px;
}
```

× NAME REQUIRED, edit text

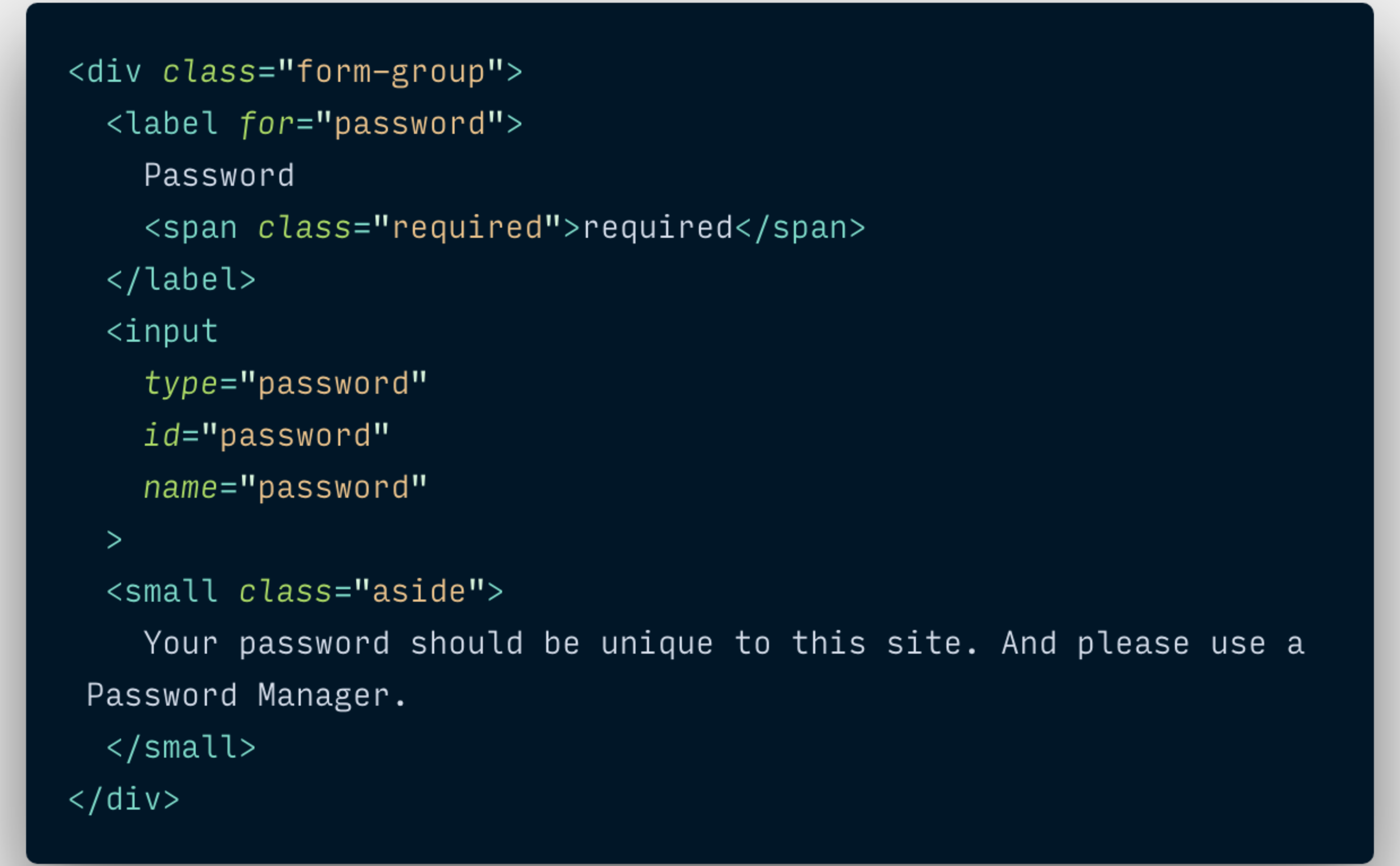


```html
<input type="text" id="name" name="name" required />
```

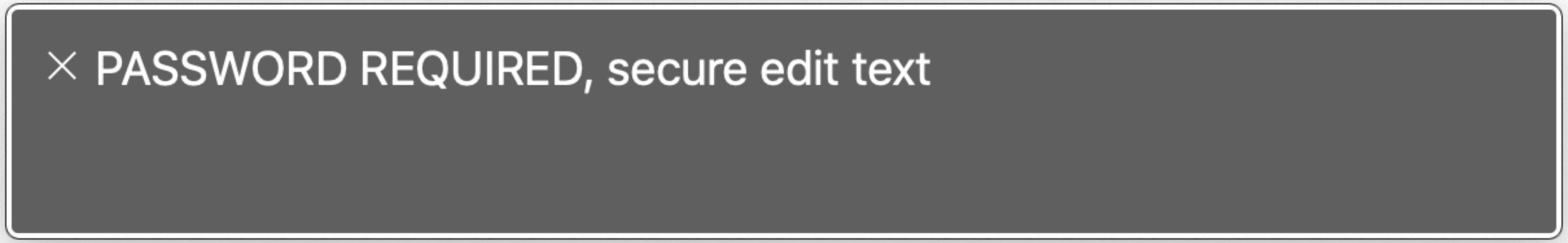
Required Attribute Requirements

https://developer.paciellogroup.com/blog/2019/02/required-attribute-requirements/

Describing fields



```html
<div class="form-group">
  <label for="password">
    Password
    <span class="required">required</span>
  </label>
  <input
    type="password"
    id="password"
    name="password"
  >
  <small class="aside">
    Your password should be unique to this site. And please use a Password Manager.
  </small>
</div>
```



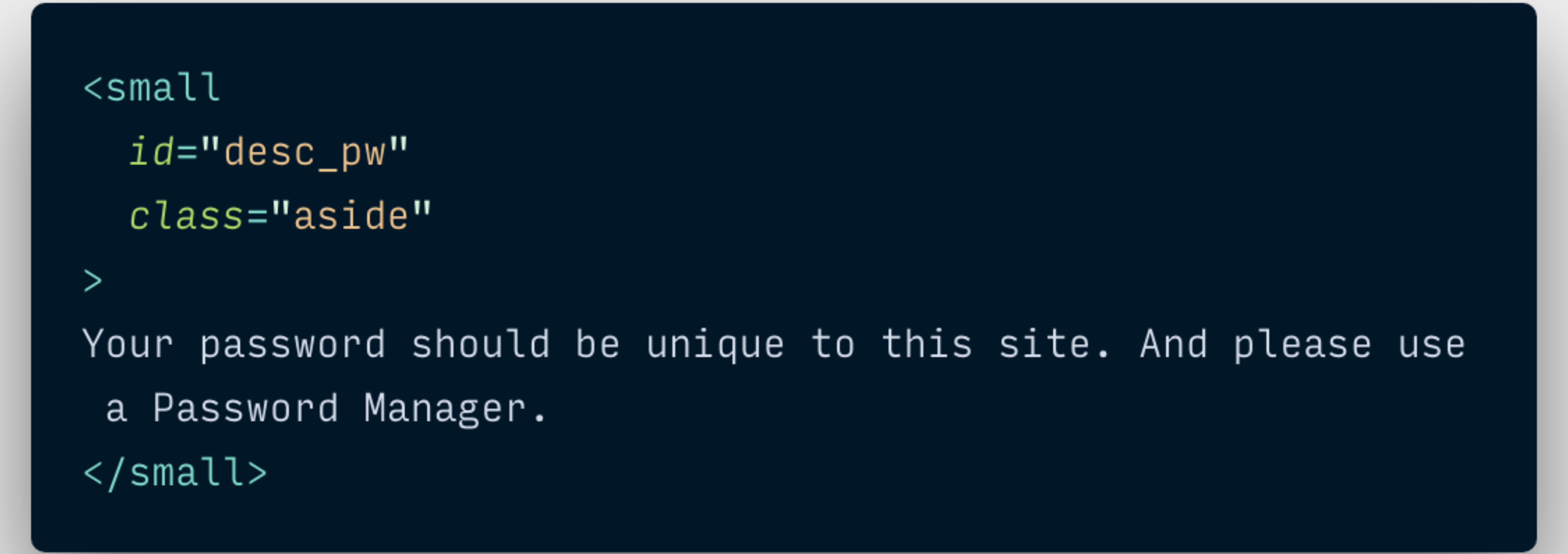
PASSWORD REQUIRED, secure edit text

`aria-describedby`

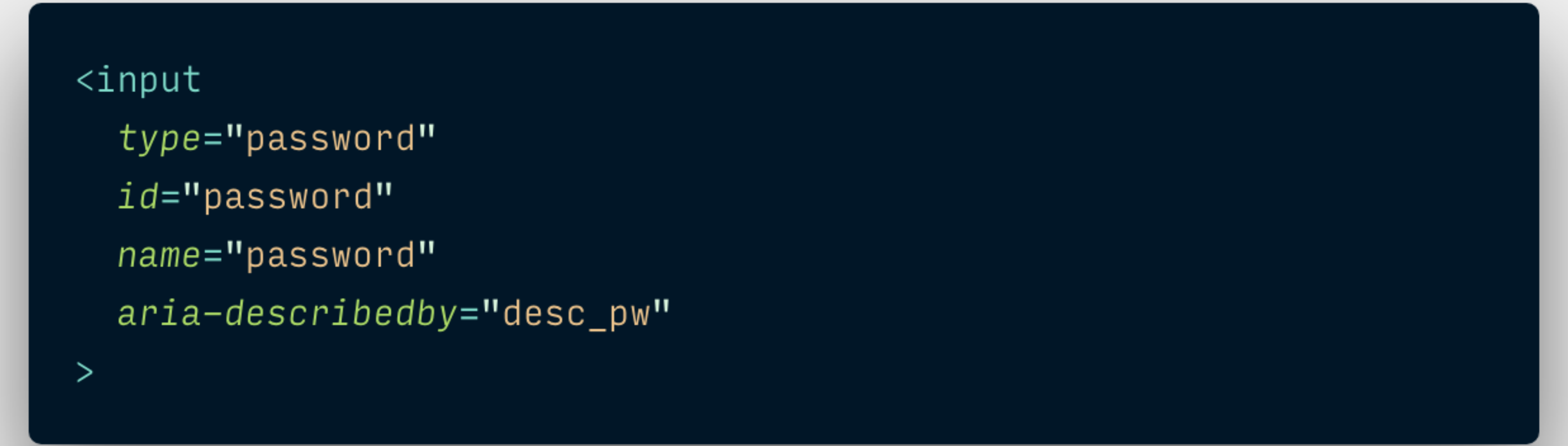
Describe an element using the text content of another

Describing `aria-describedby`

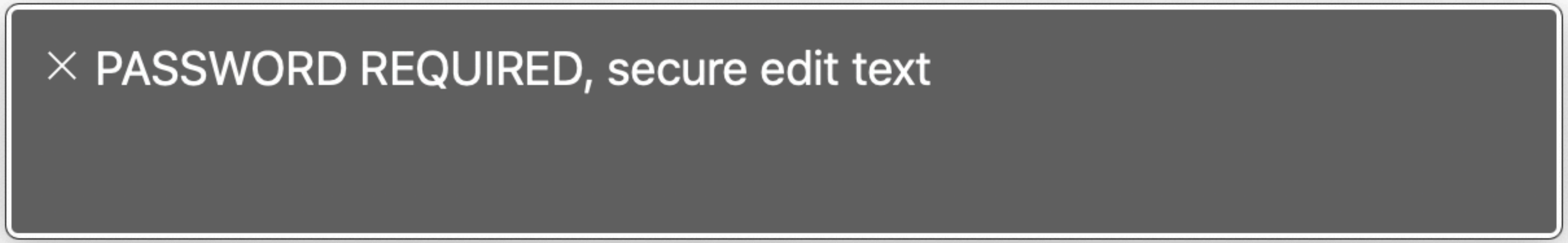
https://developer.paciellogroup.com/blog/2018/09/describing-aria-describedby/



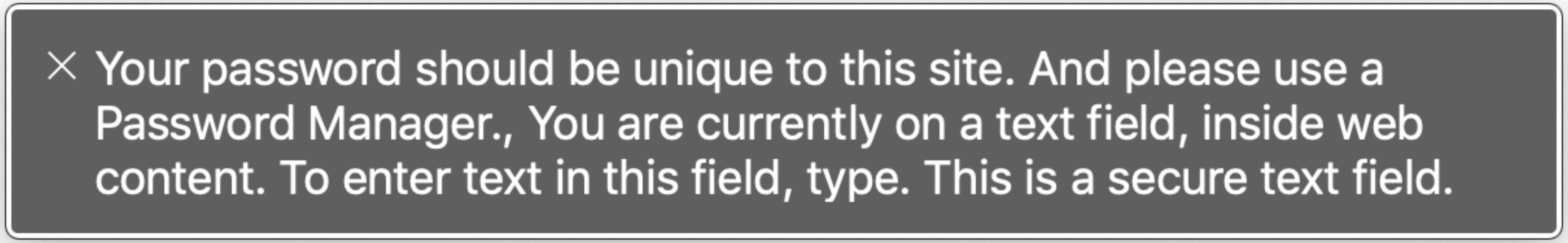
```html
<small
  id="desc_pw"
  class="aside"
>
Your password should be unique to this site. And please use
 a Password Manager.
</small>
```



```
<input
  type="password"
  id="password"
  name="password"
  aria-describedby="desc_pw"
>
```



× PASSWORD REQUIRED, secure edit text

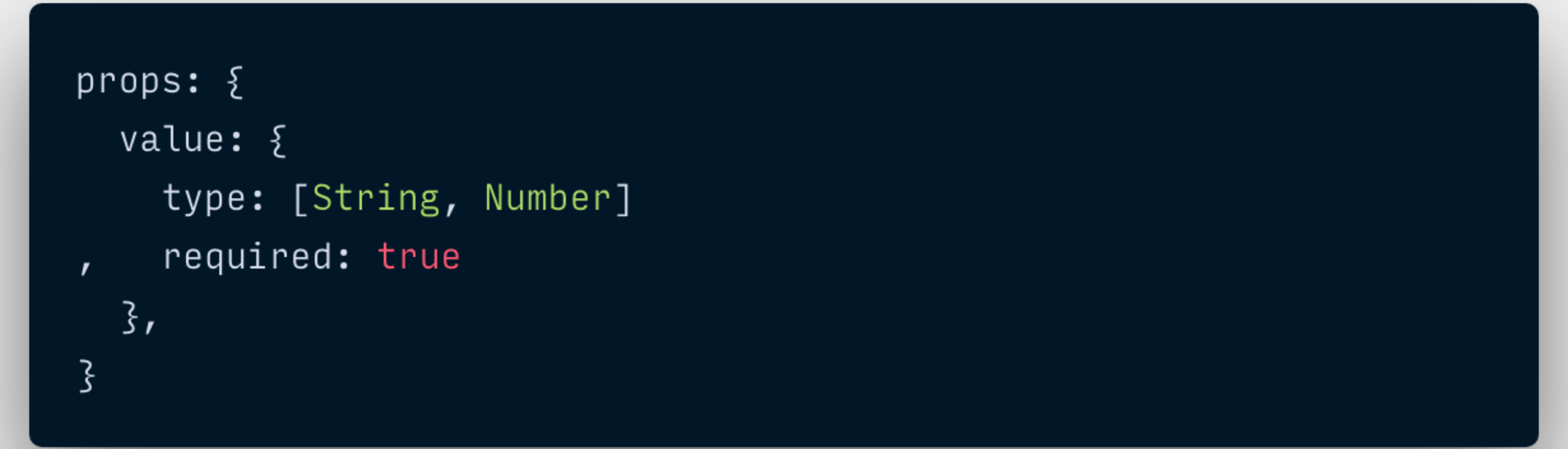


Your password should be unique to this site. And please use a Password Manager., You are currently on a text field, inside web content. To enter text in this field, type. This is a secure text field.

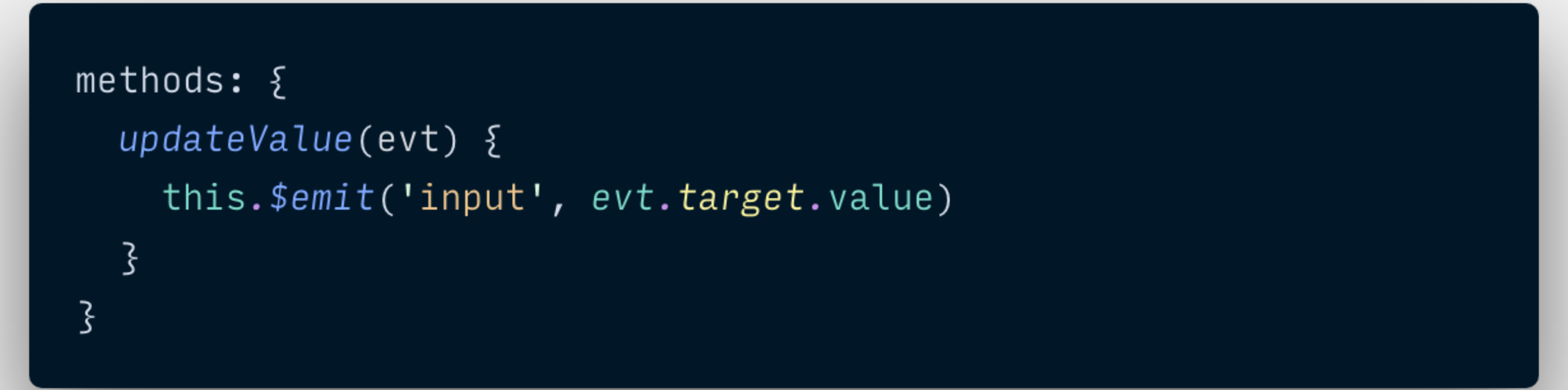
Expanding the Toolbox

Validation & Vue

`v-model` compatibility

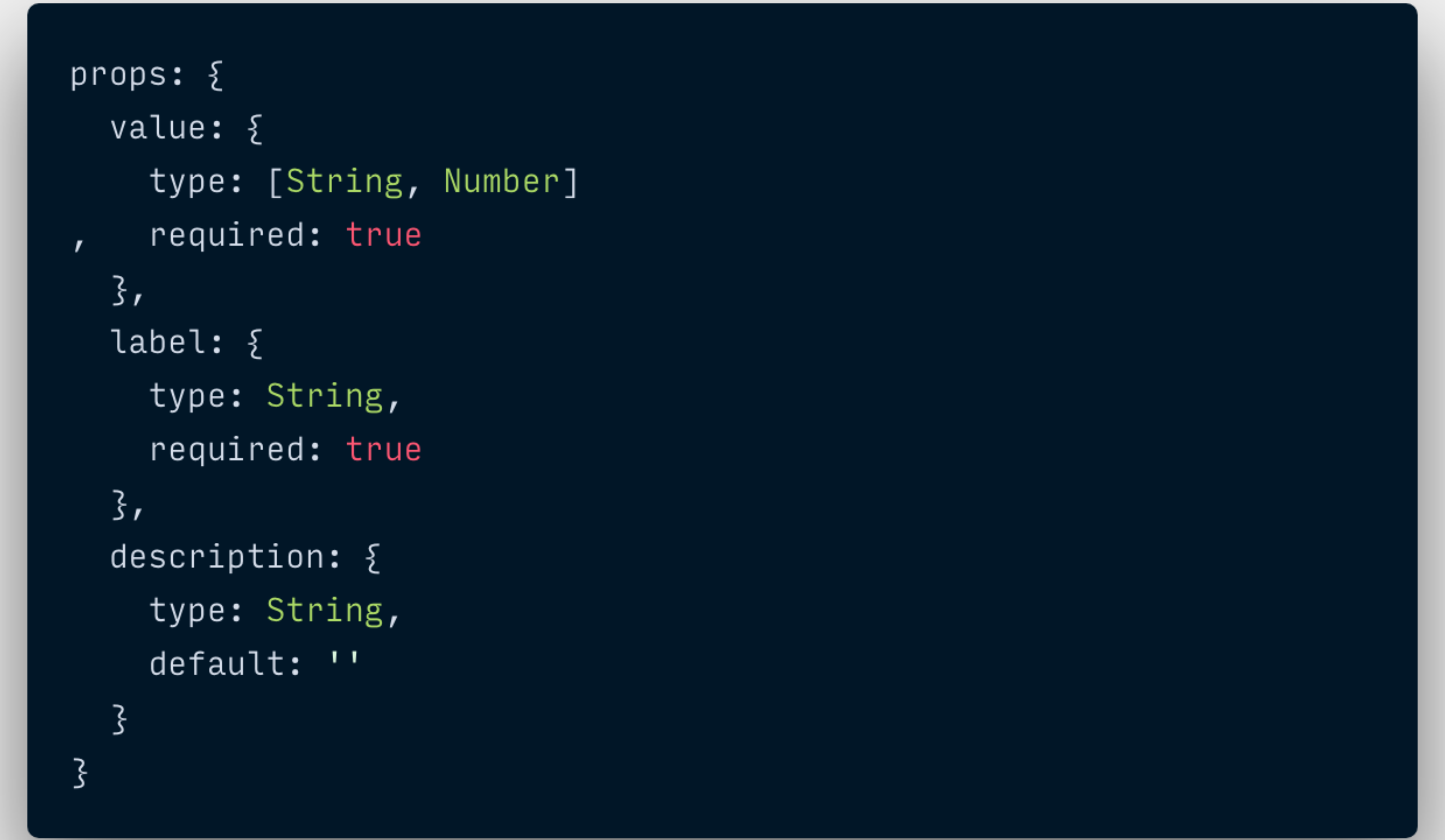


```
props: {
  value: {
    type: [String, Number]
,   required: true
  },
}
```

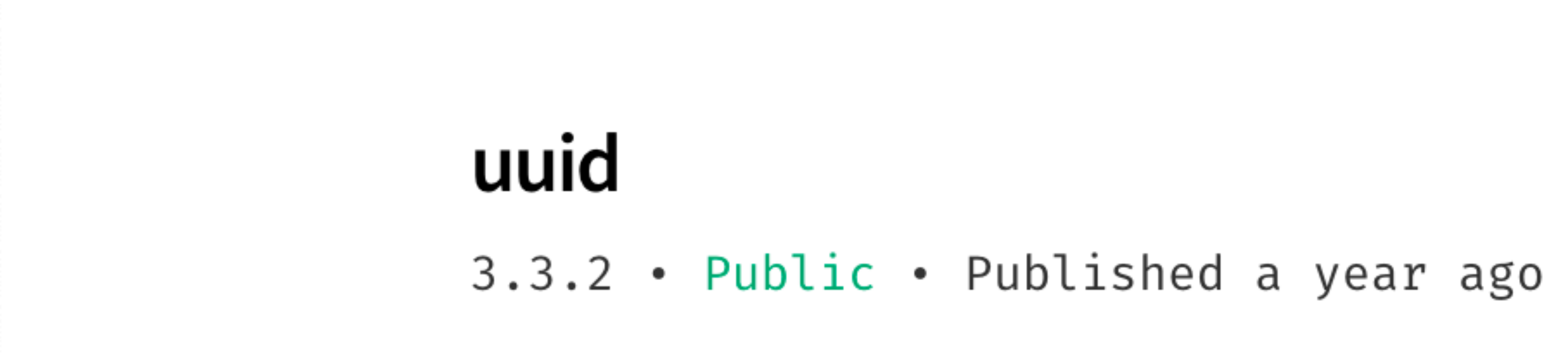


```
methods: {
  updateValue(evt) {
    this.$emit('input', evt.target.value)
  }
}
```

Basic Accessibility



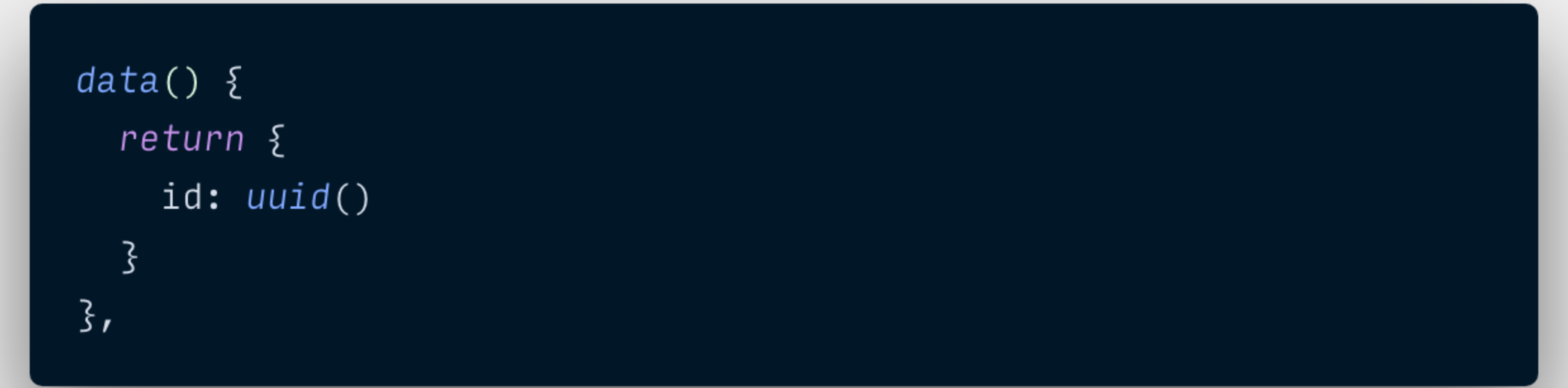
```
props: {
  value: {
    type: [String, Number]
,   required: true
  },
  label: {
    type: String,
    required: true
  },
  description: {
    type: String,
    default: ''
  }
}
```



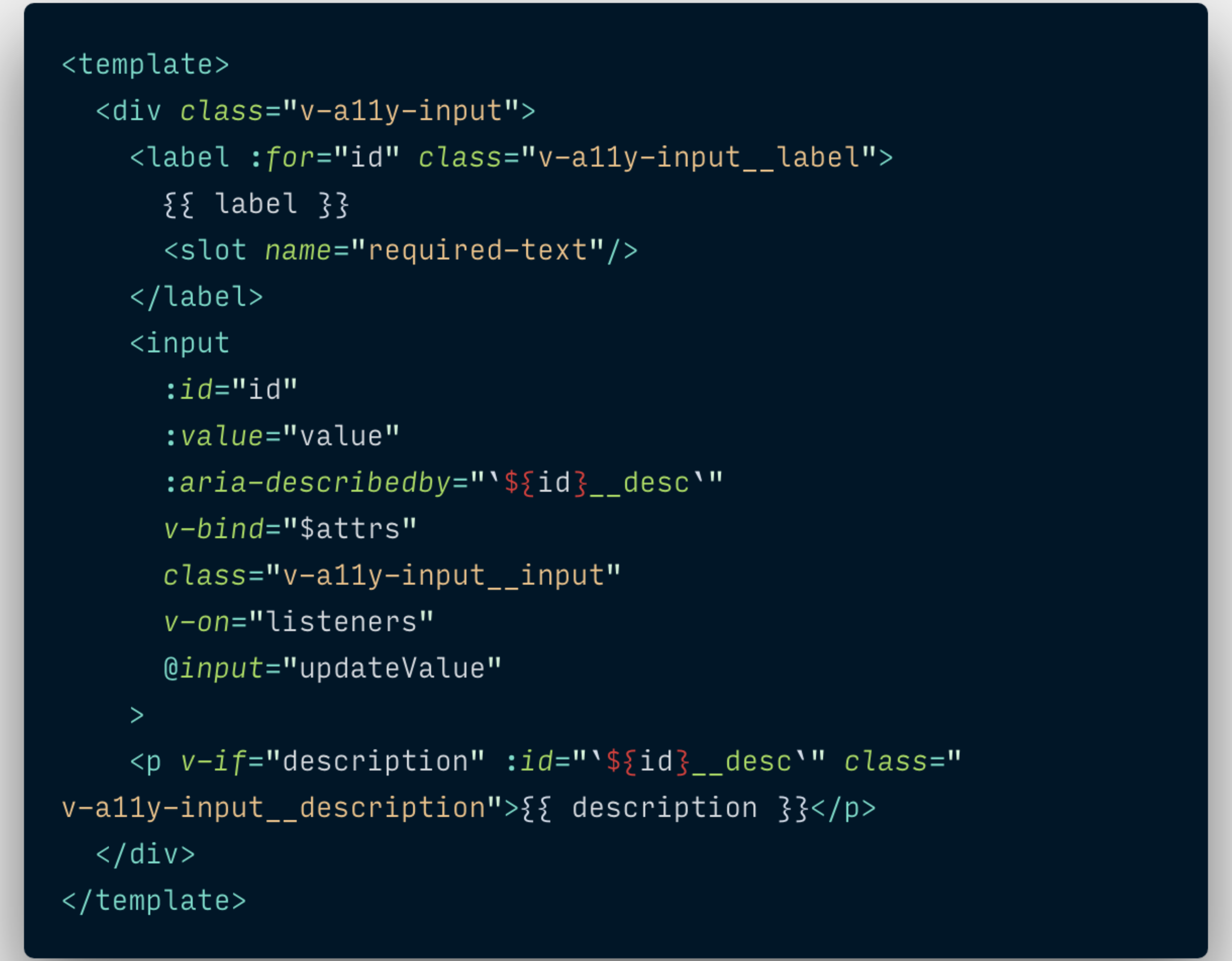
uuid

3.3.2 • Public • Published a year ago

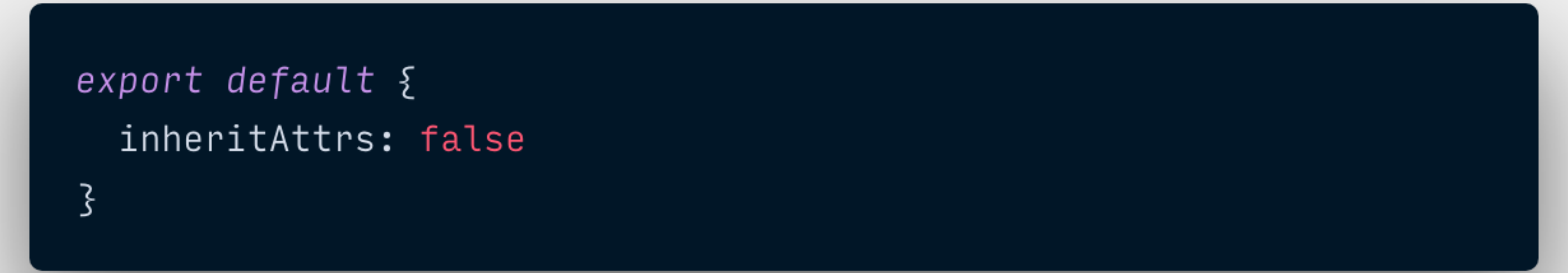
https://www.npmjs.com/package/uuid



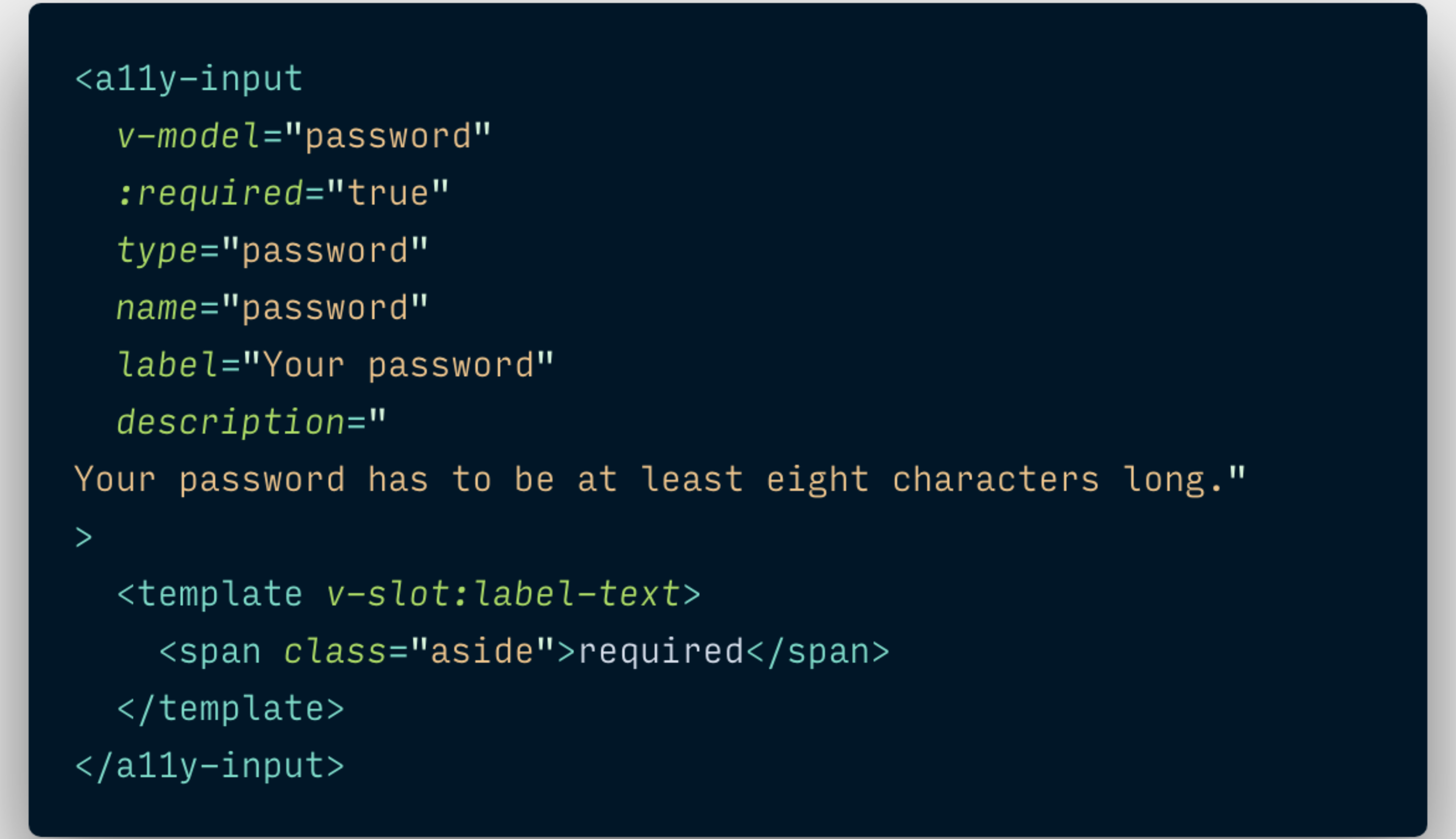
```
data() {
  return {
    id: uuid()
  }
},
```



```
<template>
  <div class="v-a11y-input">
    <label :for="id" class="v-a11y-input__label">
      {{ label }}
      <slot name="required-text"/>
    </label>
    <input
      :id="id"
      :value="value"
      :aria-describedby="`${id}__desc`"
      v-bind="$attrs"
      class="v-a11y-input__input"
      v-on="listeners"
      @input="updateValue"
    >
    <p v-if="description" :id="`${id}__desc`" class="
v-a11y-input__description">{{ description }}</p>
  </div>
</template>
```

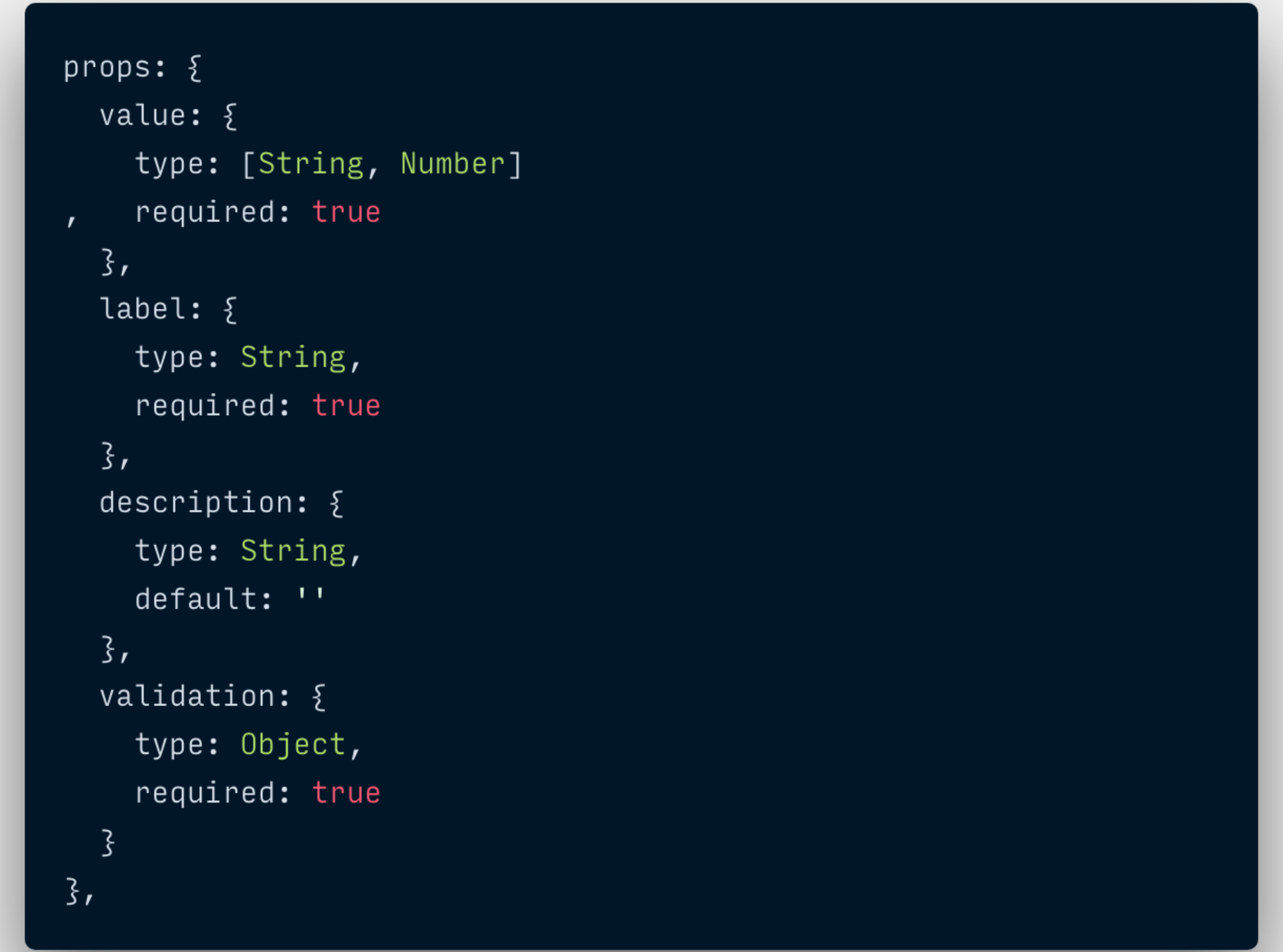


```
export default {
  inheritAttrs: false
}
```

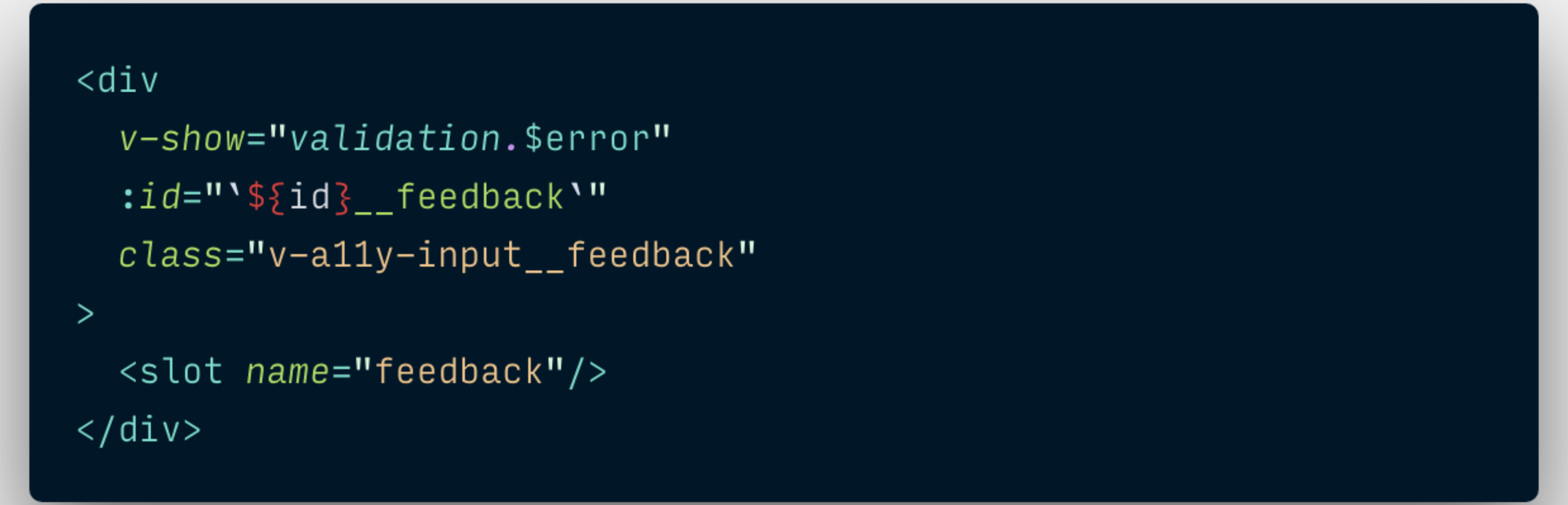


```
<a11y-input
  v-model="password"
  :required="true"
  type="password"
  name="password"
  label="Your password"
  description="
Your password has to be at least eight characters long."
>
  <template v-slot:label-text>
    <span class="aside">required</span>
  </template>
</a11y-input>
```

Validation



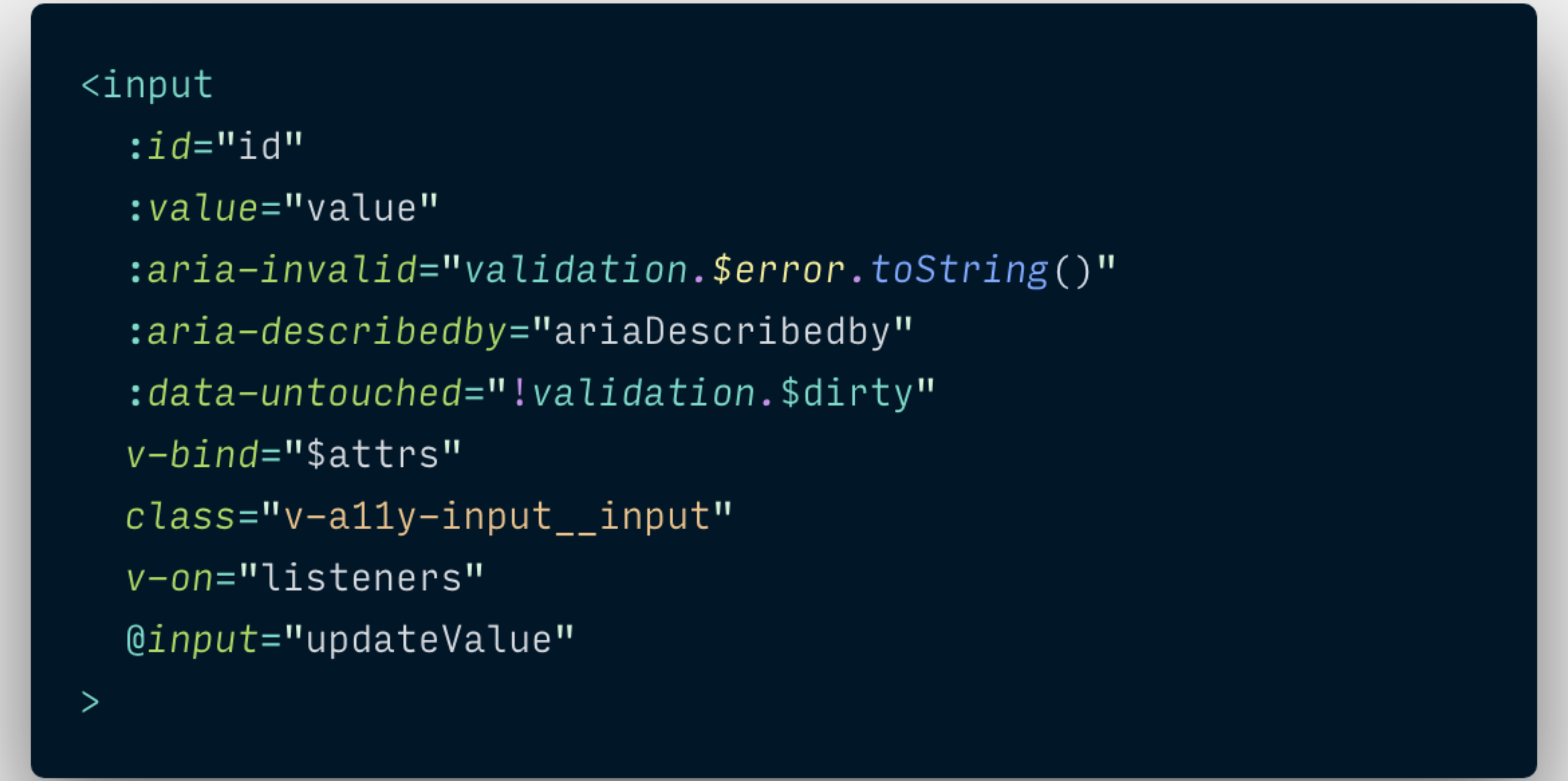
```
props: {
  value: {
    type: [String, Number]
,   required: true
  },
  label: {
    type: String,
    required: true
  },
  description: {
    type: String,
    default: ''
  },
  validation: {
    type: Object,
    required: true
  }
},
```



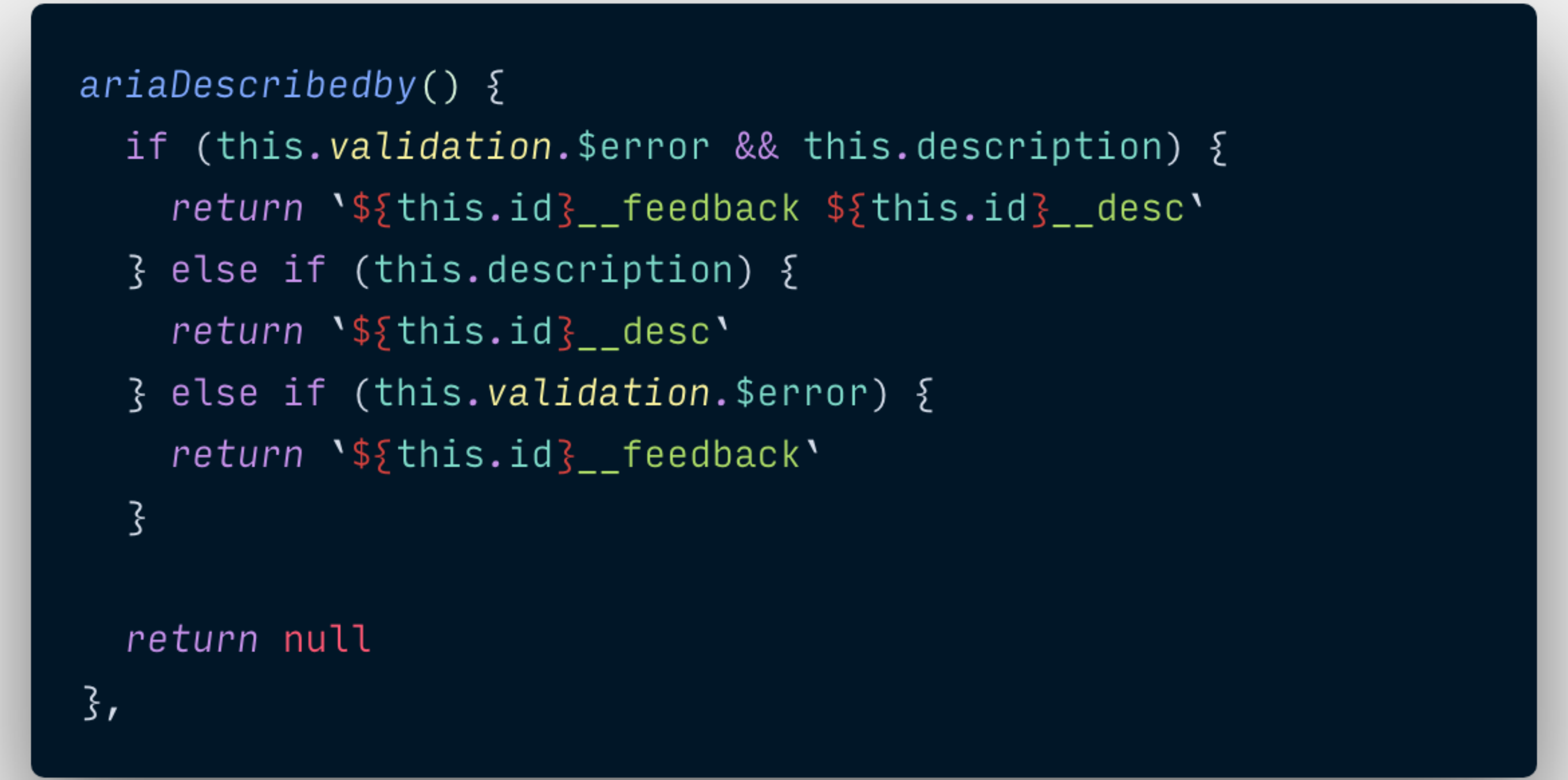
```
<div
  v-show="validation.$error"
  :id="`${id}__feedback`"
  class="v-a11y-input__feedback"
>
  <slot name="feedback"/>
</div>
```

`aria-invalid`

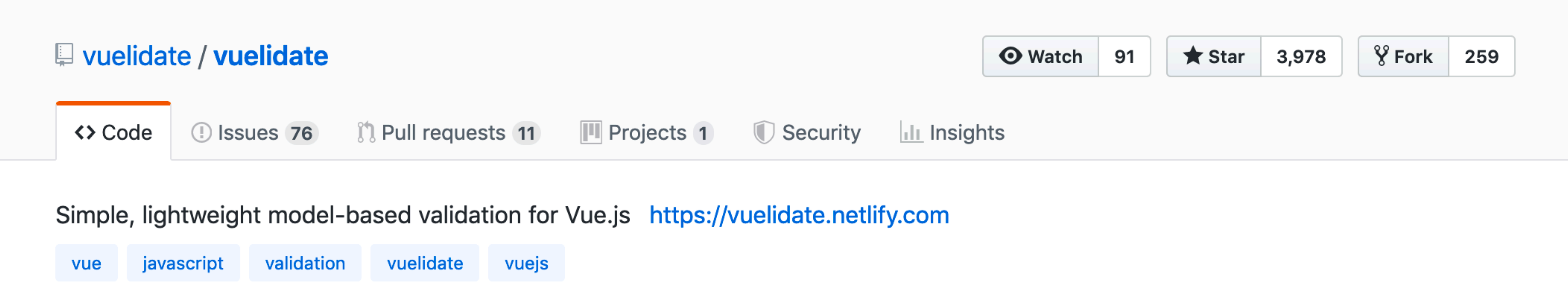
Communicate the validity of a value



```
<input
  :id="id"
  :value="value"
  :aria-invalid="validation.$error.toString()"
  :aria-describedby="ariaDescribedby"
  :data-untouched="!validation.$dirty"
  v-bind="$attrs"
  class="v-a11y-input__input"
  v-on="listeners"
  @input="updateValue"
>
```

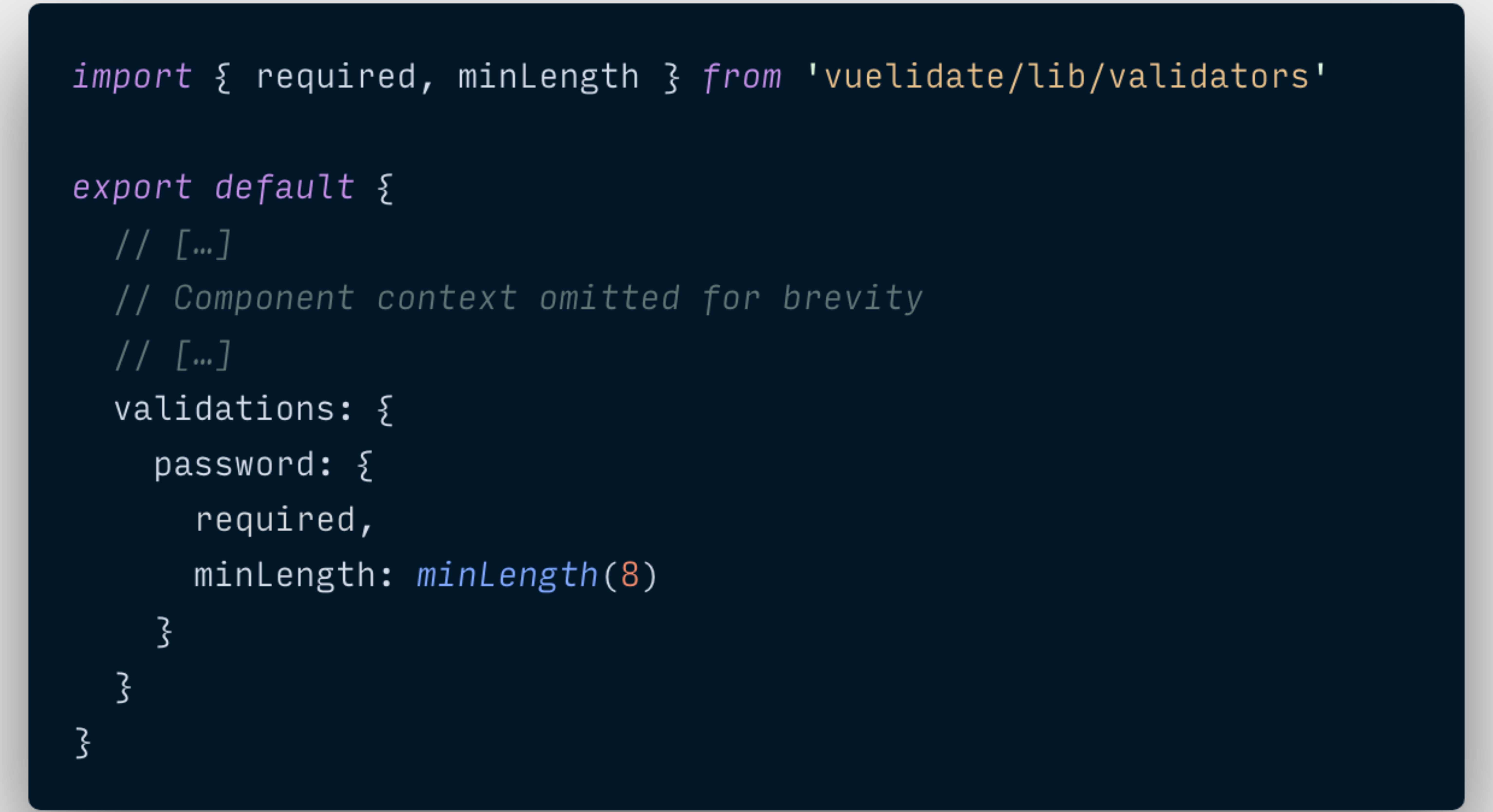


```
ariaDescribedby() {
  if (this.validation.$error && this.description) {
    return `${this.id}__feedback ${this.id}__desc`
  } else if (this.description) {
    return `${this.id}__desc`
  } else if (this.validation.$error) {
    return `${this.id}__feedback`
  }

  return null
},
```

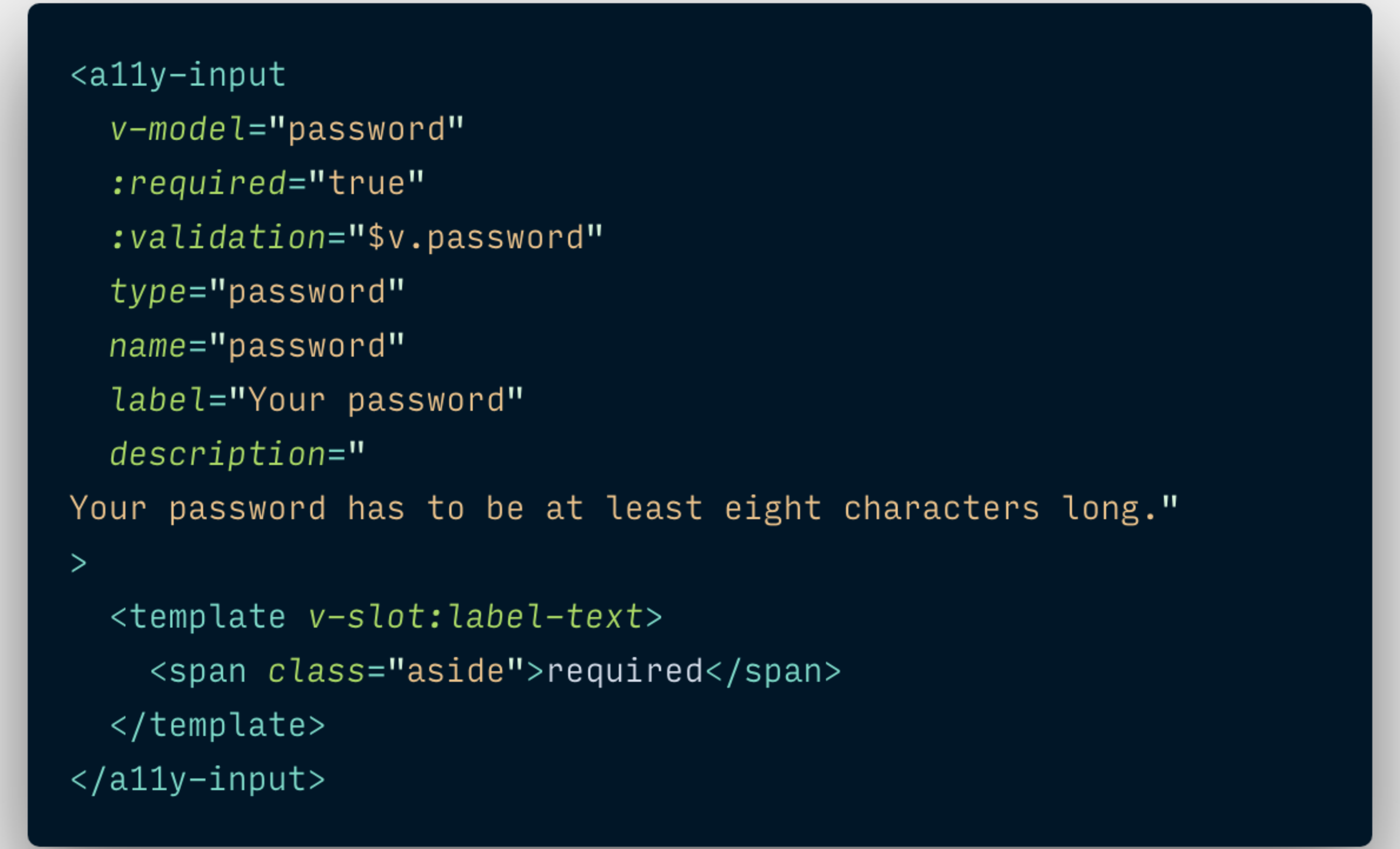


Simple, lightweight model-based validation for Vue.js https://vuelidate.netlify.com

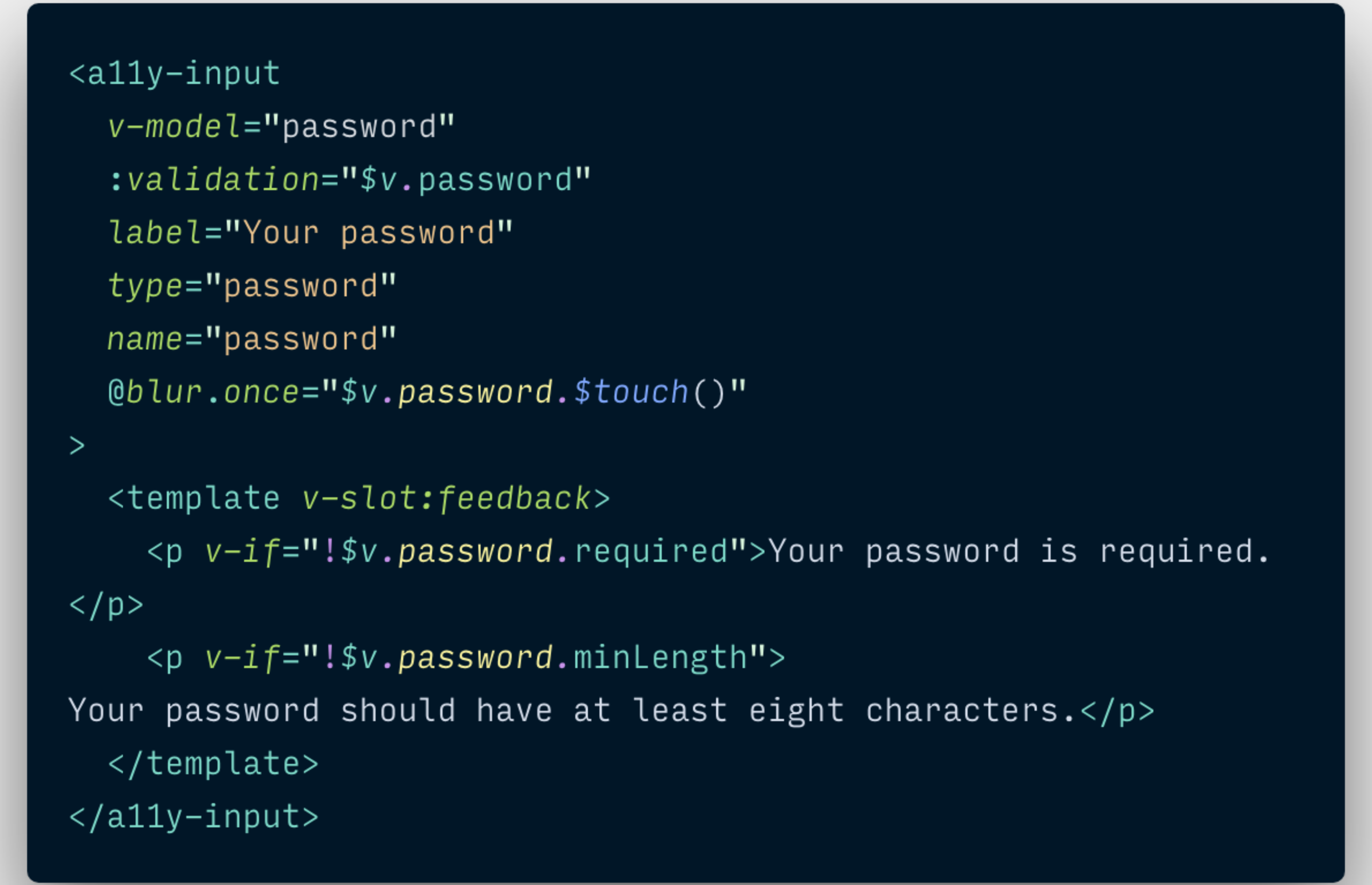
https://github.com/vuelidate/vuelidate



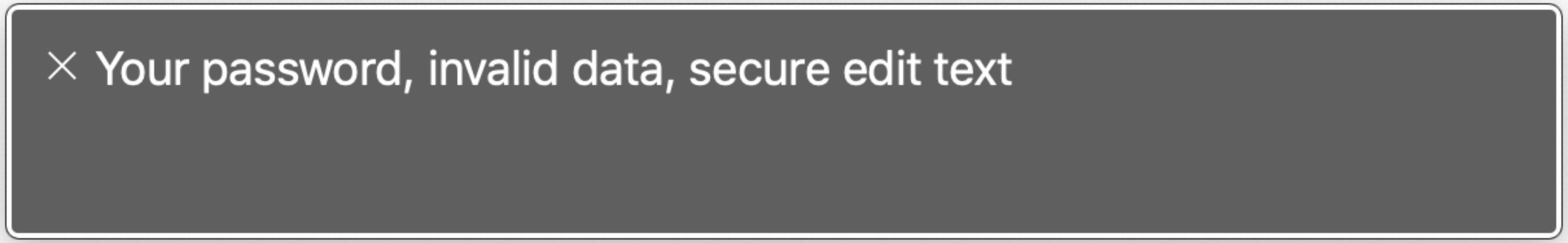
```
import { required, minLength } from 'vuelidate/lib/validators'

export default {
  // […]
  // Component context omitted for brevity
  // […]
  validations: {
    password: {
      required,
      minLength: minLength(8)
    }
  }
}
```



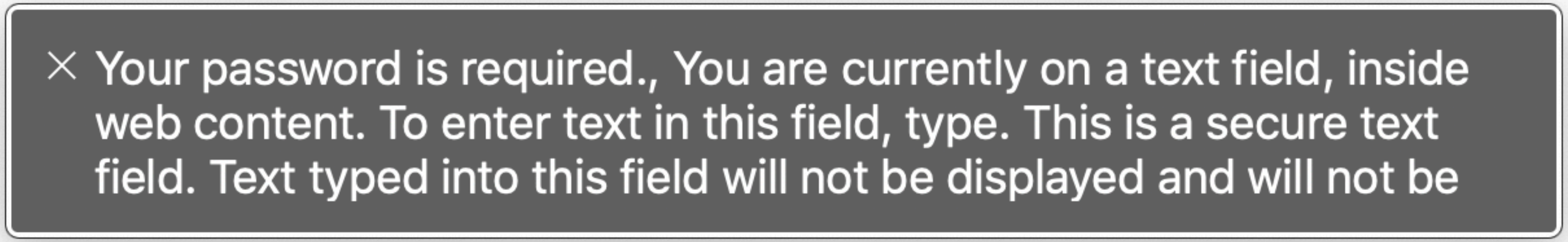
```
<a11y-input
  v-model="password"
  :required="true"
  :validation="$v.password"
  type="password"
  name="password"
  label="Your password"
  description="
Your password has to be at least eight characters long."
>
  <template v-slot:label-text>
    <span class="aside">required</span>
  </template>
</a11y-input>
```



```
<a11y-input
  v-model="password"
  :validation="$v.password"
  label="Your password"
  type="password"
  name="password"
  @blur.once="$v.password.$touch()"
>
  <template v-slot:feedback>
    <p v-if="!$v.password.required">Your password is required.
</p>
    <p v-if="!$v.password.minLength">
Your password should have at least eight characters.</p>
  </template>
</a11y-input>
```



Your password, invalid data, secure edit text

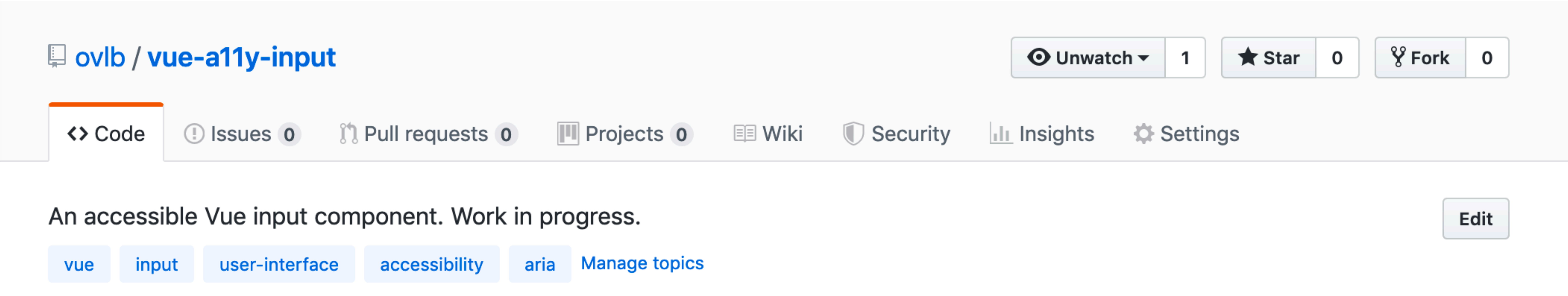


Your password is required., You are currently on a text field, inside web content. To enter text in this field, type. This is a secure text field. Text typed into this field will not be displayed and will not be

> «The only way this work gets done is if we start small, and if we work together.»

Ethan Marcotte – The web we broke

https://ethanmarcotte.com/wrote/the-web-we-broke/



An accessible Vue input component. Work in progress.

https://github.com/ovlb/vue-a11y-input

Danke.

Thank you.

Merci.

спасибо.

Oscar Braunert code & design
www.ovl.design o@ovl.design