Creating Animations with Vue

Introduction to Vue Animations

Eduardo Freitas

Software Developer & Automation Specialist

https://edfreitas.me

Overview

Understanding Vue Transitions

Ways to Implement Transitions

Transition Component

Transition Structure

Project Overview

Examples of Transitions

Custom Class Names

Understanding Vue Transitions

Most people think transitions are just decorations.

Well Designed Transitions

Highlight important information

Capture and direct your user's attention

Suggest a natural flow for your app

Provide guidance to users

Improve the quality of your app

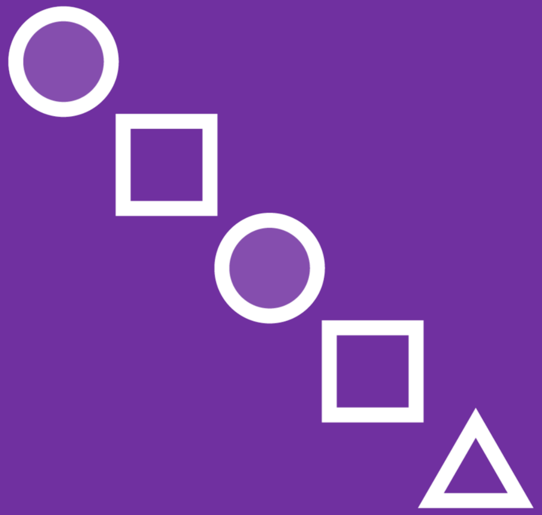
Strengthen your brand

UX Improvements

Transitions help improve the user experience, conversion, and retention rates.

Win-win for your organization,
and most importantly,
your users.

Ways to Implement Transitions



Vue offers

Different ways to implement transitions, to accommodate a broad range of developers.

Transition Implementation Options

CSS Transitions

Animation styling

JavaScript Hooks

Requires DOM editing

Third Party

External CSS/JS

Best Transition Implementation Fit

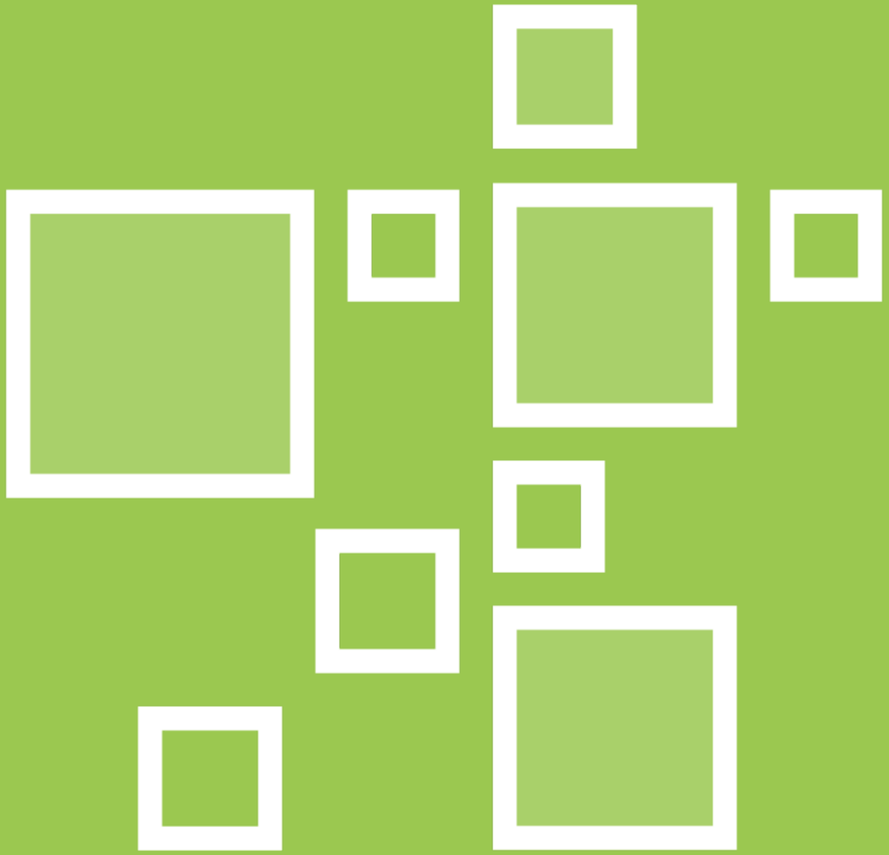
Animation styling - HTML & CSS experience

React, Angular, JS – editing the DOM

Knowledge of a specific library - external

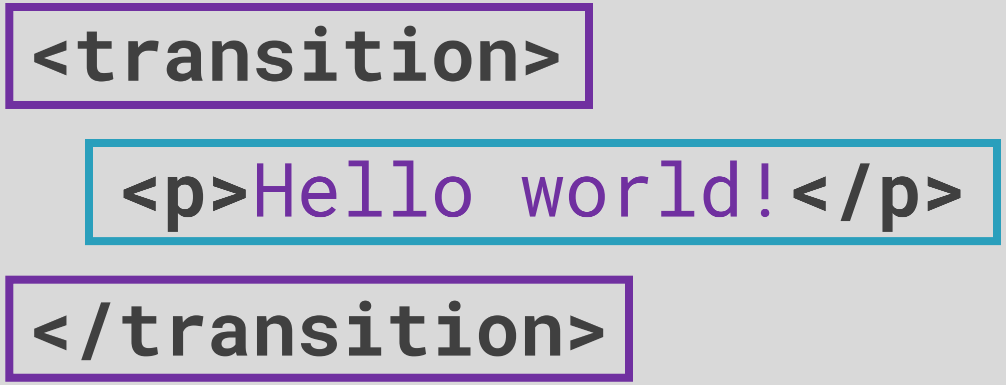
Choose the option you feel most comfortable with, and can most easily implement.

Transition Component



Sets Hooks & Classes

To your changing components so you can style them throughout different stages of the transition.



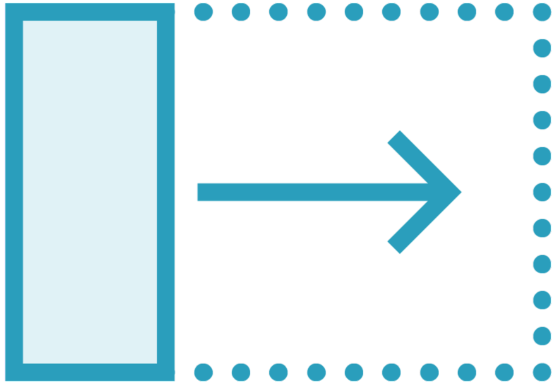
```
<transition>
  <p>Hello world!</p>
</transition>
```

Transition Component

The transition component wraps the element we want to animate.

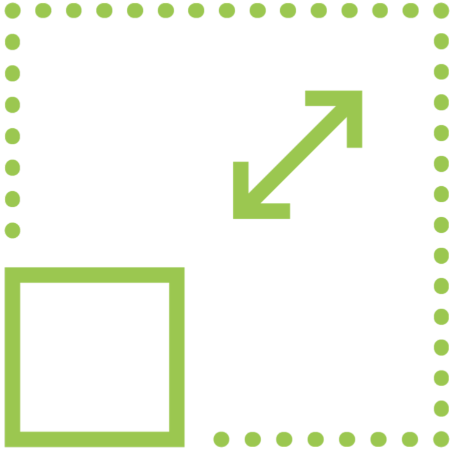
The transition element applies CSS classes to the element it wraps.

Types of Transition Classes



Entering

Transition classes for entering an animation



Leaving

Transition classes for leaving an animation

`v-enter-from`

◀ Starting state for enter. Added before the element is inserted.

`v-enter-active`

◀ Active state for enter. Applied during the entire entering phase. Added before the element is inserted.

`v-enter-to`

◀ Ending state for enter. Added one frame after the element is inserted.

`v-leave-from`

◀ Starting state for leave. Added immediately when a leaving transition is triggered.

`v-leave-active`

◀ Active state for leave. Applied during the entire leaving phase.

`v-leave-to`

◀ Ending state for leave. Added one frame after a leaving transition is triggered.

The v- Prefix

The v- prefix is the default when using a <transition> with no name

The v- prefix is not the default when using a <transition> with a name, such as <transition name=“ani">

Then the v-enter-from class would instead be ani-enter-from

Transition Structure

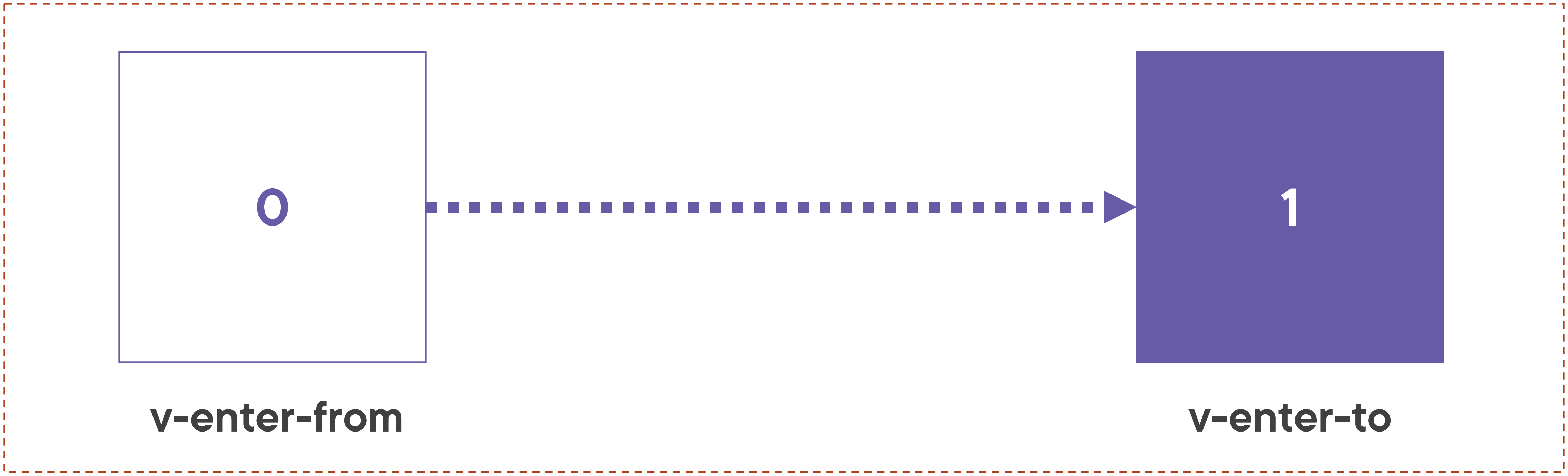
To understand transitions, we need to understand opacity.

Opacity

Opacity is the degree to which content behind an element is hidden, and is the opposite of transparency.

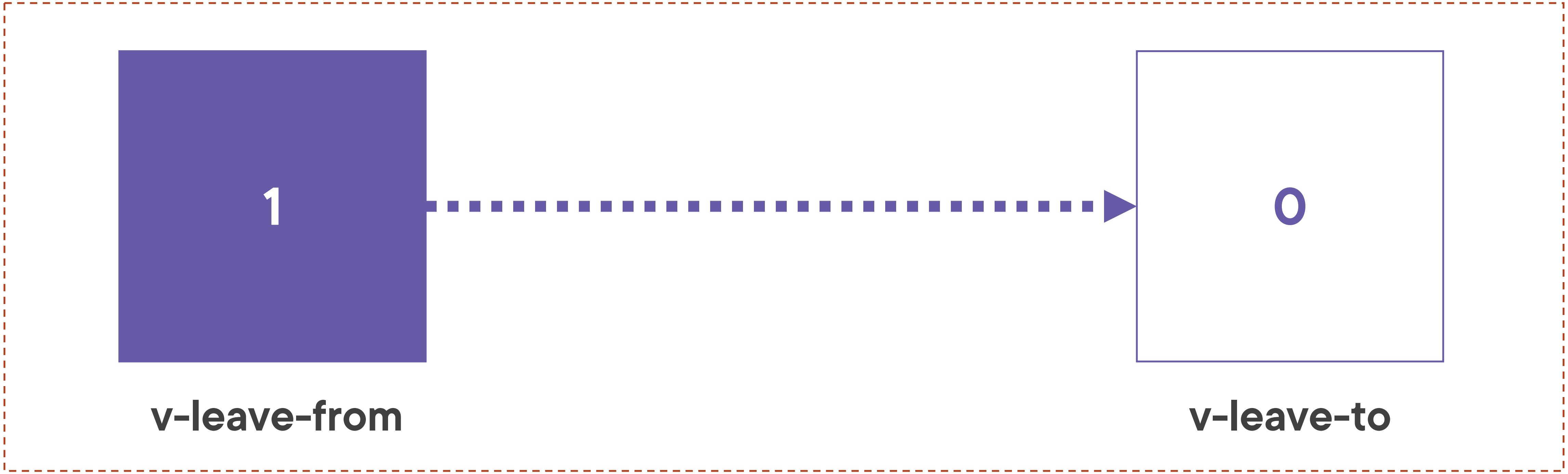
https://developer.mozilla.org/en-US/docs/Web/CSS/opacity

Opacity - Entering a Transition



v-enter-active

Opacity - Leaving a Transition



v-leave-active

CSS Transitions Types

CSS Transitions

The most common transition types uses CSS transitions

CSS Animations

The v-enter-from is removed on an animationend event

Demo

Demo – Project Overview

Demo

Demo – Transitions

Custom Class Names

Custom Transitions Classes

enter-from-class

enter-active-class

enter-to-class

leave-from-class

leave-active-class

leave-to-class

Alternatively, there are also JavaScript hooks emitted by the transition element.

Available JavaScript Hooks

before-enter / before-leave

enter / leave

after-enter / after-leave

enter-cancelled / leave-cancelled

```
<transition @before-enter='beforeEnter'>
  <p>Hello world!</p>
</transition>
```

Declaring Event Handlers

This is how we can declare a JavaScript event handler.

Then, we can handle the event in the JavaScript code.

```
// done is an optional callback method
beforeEnter(el, done) {
    done()
}
```

Transition Event Handler

The first parameter is the element that is being animated.

The second parameter is an optional callback method.

Summary

Understanding Vue Transitions

Ways to Implement Transitions

Transition Component

Transition Structure

Project Overview & Examples

Custom Class Names