

Transition e Transition Group

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA DA SEMANA

- **Transition**
- **Transition Group**
- VueRouter
- Vuex e gerenciamento de estado

Transition - Animação com Vue

Transition é um componente embutido, disponível em qualquer template de componente sem ter que registrá-lo. Ele pode ser usado para aplicar animações de entrada e saída em elementos ou componentes passados a ele por meio de seu slot padrão. A entrada ou saída pode ser acionada por um dos seguintes:

- Renderização condicional via v-if
- Exibição condicional via v-show
- Componentes dinâmicos alternando através do <component> elemento especial

Exemplo


```
<div style="margin:60px">
  <transition>
    <h1 v-show="show">Apresentar Header</h1>
  </transition>
  <button @click="show = !show">Alterar estado</button>
</div>
```

```
<style>
  .v-enter-active,
  .v-leave-active {
    transition: opacity 0.5s ease;
  }

  .v-enter-from,
  .v-leave-to {
    opacity: 0;
  }
</style>
```

Apresentar Header

Alterar estado



Classe de transição - Entrada

Existem três classes aplicadas para transições de entrada:

v-enter-from: Estado inicial para enter. Adicionado antes do elemento ser inserido, removido um quadro após o elemento ser inserido.

v-enter-active: Estado ativo para entrar. Aplicado durante toda a fase de entrada. Adicionado antes do elemento ser inserido, removido quando a transição/animação termina. Esta classe pode ser usada para definir a duração, atraso e curva de atenuação para a transição de entrada.

v-enter-to: Estado final para entrar. Adicionado um quadro após a inserção do elemento (ao mesmo tempo em que v-enter-from é removido), removido quando a transição/animação termina.

Classe de transição - Entrada

```
<template>
  <div>
    <transition>
      <h1 v-show="show">Mostrando conteúdo</h1>
    </transition>
    <button @click="show = !show">Alterar estado</button>
  </div>
</template>

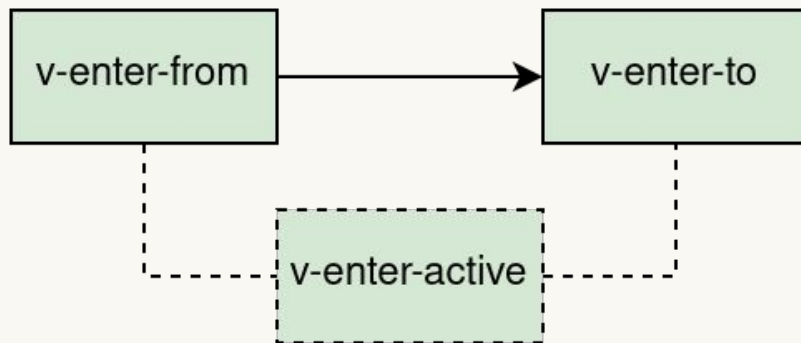
<script>
  export default {
    data() {
      return {
        show: false
      }
    }
  }
</script>

<style>
  .v-enter-from {
    opacity: 0;
  }

  .v-enter-active {
    transition: opacity 2s ease;
  }

  .v-enter-to {
    opacity: 1;
  }
</style>
```

Alterar estado



Classe de transição - Saída

Existem seis classes aplicadas para transições de entrada/saída:

v-leave-from: Estado inicial de saída. Adicionado imediatamente quando uma transição de saída é acionada, removido após um quadro.

v-leave-active: Estado ativo de saída. Aplicado durante toda a fase de saída. Adicionado imediatamente quando uma transição de saída é acionada, removido quando a transição/animação termina. Esta classe pode ser usada para definir a duração, atraso e curva de atenuação para a transição de saída.

v-leave-to: Estado final de saída. Adicionado um quadro depois que uma transição de saída é acionada (ao mesmo tempo v-leave-from é removida), removida quando a transição/animação termina.

Classe de transição - Saída

```
<template>
  <div>
    <transition>
      <h1 v-show="show">Mostrando conteúdo</h1>
    </transition>
    <button @click="show = !show">Alterar estado</button>
  </div>
</template>

<script>
  export default {
    data() {
      return {
        show: false
      }
    }
  }
</script>

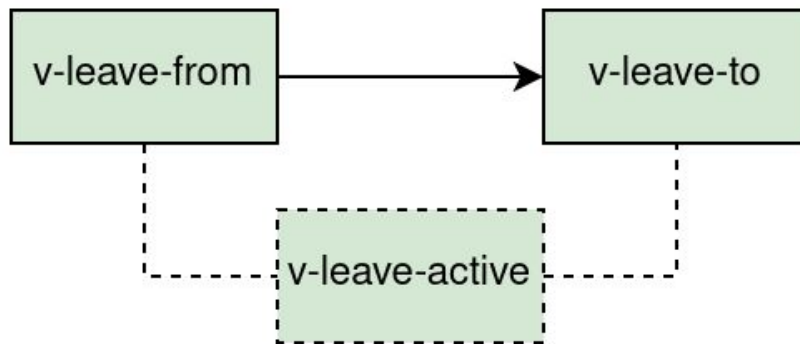
<style>
  .v-leave-from {
    opacity: 1;
  }

  .v-leave-active {
    transition: all 2s ease;
  }

  .v-leave-to {
    opacity: 0;
  }
</style>
```

Mostrando conteúdo

Alterar estado



Nomeando transições

Também é possível fazer a identificação da transição através da propriedade “name” do elemento “transition”. Neste caso, a classe nomeada de transição iniciará com o nome (substituindo o “v-”) seguido do evento.

```
<transition name="header">
  <h1 v-show="show">Mostrando conteúdo</h1>
</transition>
```

```
<style>
  .header-enter-from {
    opacity: 0;
  }

  .header-enter-active {
    transition: opacity 1s;
  }

  .header-enter-to {
    opacity: 1;
  }
</style>
```

Modo de transições

Por padrão as transições de entrada e saída ocorrem de forma simultânea, ou seja, se houver a apresentação de um conteúdo enquanto ocorre a saída de outro, ambas acontecem ao mesmo tempo. Exemplo:

```
<transition name="btn">  
  <button @click="show = false" v-if="show">Não mostrar</button>  
  <button @click="show = true" v-else>Mostrar</button>  
</transition>
```

Não mostrar

Modo de transições

Nestes casos o Vue oferece modos de transições para controlarmos os eventos de transições como nossa necessidade. Esses modos são atribuídos na propriedade "mode" do elemento "transition" e podem receber os valores: "in-out" e "out-in". onde:

in-out: Executa primeiro a transição de entrada e em seguida o de saída:

```
<transition name="btn" mode="in-out">
```

Não mostrar

out-in: Executa primeiro a transição de saída e em seguida o de entrada:

```
<transition name="btn" mode="out-in">
```

Mostrar

Transition Group

O Transition Group é um componente interno projetado para animar a inserção, remoção e alteração de ordem de elementos ou componentes que são renderizados em uma lista.

Número

0

⬆

Adicionar Item

- 1
- 4
- 6
- 7

O Transition Group suporta os mesmos adereços, classes de transição CSS e listeners de gancho JavaScript que <Transition>, com as diferenças:

- Por padrão, ele não renderiza um elemento wrapper. Mas você pode especificar um elemento a ser renderizado com o tag prop.
- Os modos de transição não estão disponíveis, porque não estamos mais alternando entre elementos mutuamente exclusivos.
- Os elementos internos **sempre precisam** ter um key atributo exclusivo.
- As classes de transição CSS serão aplicadas a elementos individuais na lista, **não** ao próprio grupo/contêiner.

Transition Group

```
<template>
  <div>
    <div>
      <label>Número</label>
      <input type="number" v-model="novoItem" />
      <button @click="adicionar">Adicionar Item</button>
    </div>
    <transition-group name="list" tag="ul">
      <li v-for="item in itens" :key="item">
        {{ item }}
      </li>
    </transition-group>
  </div>
</template>

<script>
export default {
  data() {
    return {
      novoItem: 0,
      itens: [1, 4, 6, 7]
    }
  },
  methods: {
    adicionar() {
      this.itens.push(this.novoItem);
      this.itens.sort();
    }
  }
}
</script>

<style>
.list-enter-active,
.list-leave-active {
  transition: all 0.5s ease;
}
.list-enter-from,
.list-leave-to {
  opacity: 0;
  transform: translateX(30px);
}
</style>
```

Número 0



Adicionar Item

- 1
- 4
- 6
- 7

Transition Group - Classe “move”

No Transition Group também disponibiliza uma nova classe chamada: “move”, que adiciona um estilo “position” para “absolute” padrão na execução do evento, criando um efeito de inclusão e exclusão mais natural

```
.list-move,  
.list-enter-active,  
.list-leave-active {  
  transition: all 0.5s ease;  
}
```

Número Adicionar Item

- 1
- 4
- 6
- 7



DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>