**COMPONENTES DINÂMICOS**

Crie um novo projeto: npm init vue@latest.

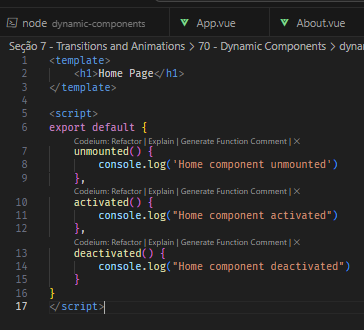
Vá na pasta do novo projeto: cd dynamic-project

Instale as dependências npm: npm install

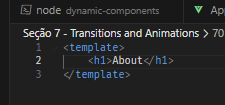
Inicie o server: npm run dev

Vamos utilizar como objetivo criar um select que tenha duas opções, a home o About. Quando selecionar uma ou outra irá mostrar o componente escolhido:

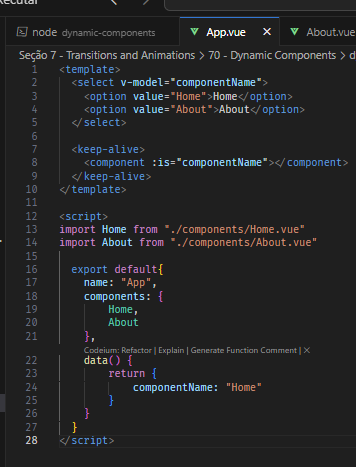
O componente de Home será este:



O componente de about este:



O código no app.vue:



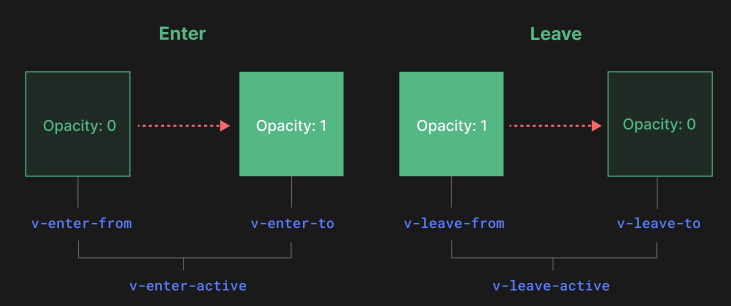
Vamos analisar por partes:

Primeiramente, os componentes são importados no app.vue e colocados dentro da propriedade components no export default. Em seguida, montamos o seletor dentro do template, possuindo duas opções. A tag keep-alive serve para o componente não ser destruído, e a tag component serve para exibir um componente. O bind :is vai informar ao vue qual componente ele deve mostrar. No seletor, colocamos um v-model para que de acordo com a option escolhida, seu valor vá para o select e este altere dentro do data o valor. Alterando dentro do data o valor, irá alterar o valor do componente dentro do bind e isso fará com que o componente seja exibido conforme a escolha no select.

**ANIMANDO COM TRANSIÇÕES CSS**

O componente de transição é built-in, isso quer dizer que ele está disponível para qualquer componente de template sem precisar registrá-lo, ele já vem no Vue. Ele pode ser usado para aplicar nas animações de entradas e saídas de elementos ou componentes.

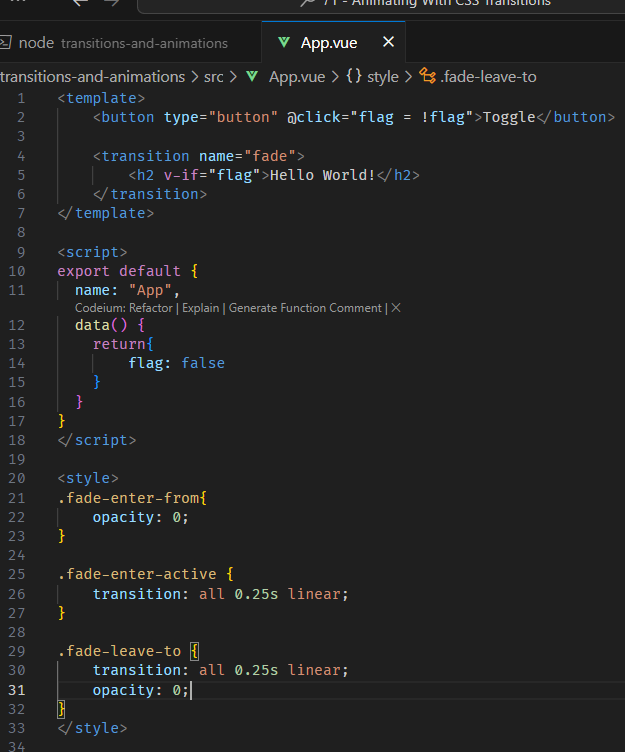
Funcionamento das transições:



Em que,

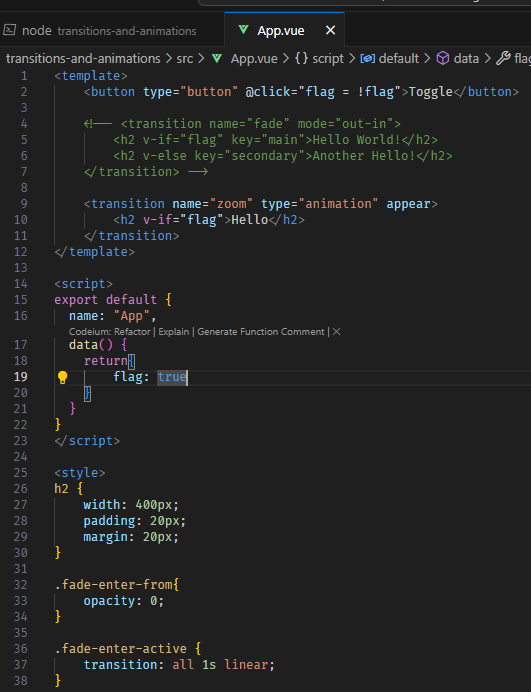
1. v-enter-from: elemento começando a surgir na tela.
2. v-enter-active: aplicado durante a fase de surgimento.
3. v-enter-to: fase final de entrada.
4. v-leave-from: iniciando fase de saída.
5. v-leave-active: aplicado durante a fase de saída.
6. v-leave-to: fase final de saída.

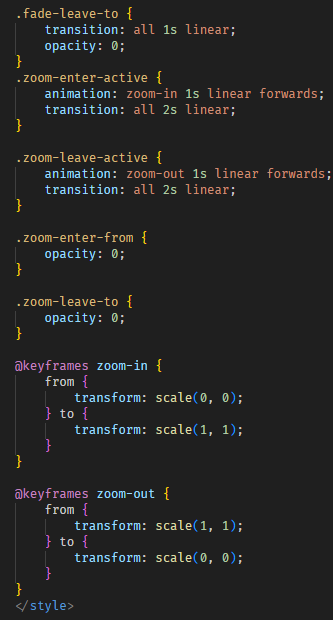
Veja um exemplo:



**TRANSIÇÃO DO TIPO ANIMAÇÃO**

Considere:



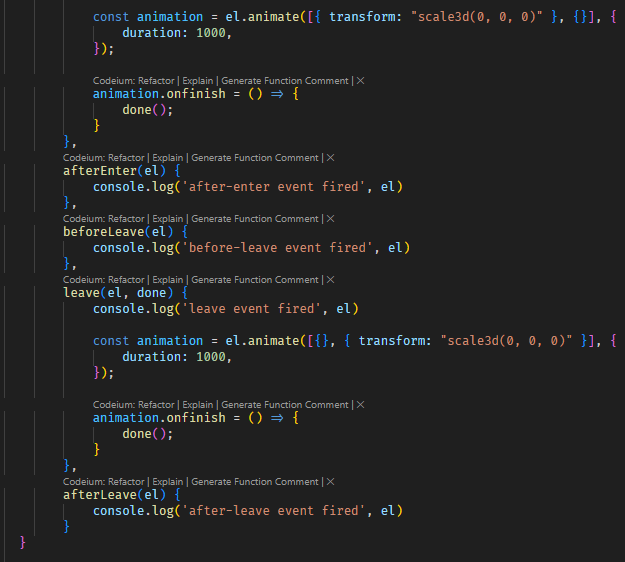
****

Perceba que dei ao componente de transition um name chamado zoom. Isso vai fazer com que ele vá buscar nos styles aquelas classes que comecem com .zoom . Além disso, o type foi definido como animation, fazendo com que as propriedades de animation dentro da classe de zoom do styles sejam compreendidas.

**ANIMAÇÃO COM JAVASCRIPT**

Considere:



****

Vou explicar como essa animação foi construída:

Primeiramente, criamos o componente de transition e como propriedade dele adicionei algumas funções com o @., além disso, setei um bind :css como falso, afinal a animação será com JS. Em seguida, no export default, dentro dos methods declarei as funções e o que cada uma faz. Para esta animação, usei somente a função enter e a leave, as demais são só para poder ver que são chamadas no console.log. O truque foi aplicar um transform em scale3d partindo do absoluto 0 e aumentando por 1 segundo para a animação de entrada. Na animação de saída, exatamente o contrário.

**\*Para animar uma lista, o componente transition não funciona, e deve ser utilizado o TransitionGroup.**