TransitionGroup

`<TransitionGroup>` 는 리스트로 렌더링되는 요소 또는 컴포넌트의 삽입, 제거, 순서 변경을 애니메이션화하기 위해 설계된 내장 컴포넌트입니다.

`<Transition>` 과의 차이점

`<TransitionGroup>` 는 `<Transition>` 과 동일한 props, CSS 트랜지션 클래스, JavaScript 훅 리스너를 지원하지만, 다음과 같은 차이점이 있습니다:

- 기본적으로 래퍼 요소를 렌더링하지 않습니다. 하지만 `tag` prop을 사용하여 렌더링할 요소를 지정할 수 있습니다.
- 트랜지션 모드는 사용할 수 없습니다. 더 이상 상호 배타적인 요소 간에 전환하지 않기 때문입니다.
- 내부의 요소들은 **항상** 고유한 `key` 속성을 가져야 합니다.
- CSS 트랜지션 클래스는 그룹/컨테이너 자체가 아니라 리스트의 개별 요소에 적용됩니다.

> ⓘ TIP
>
> DOM 내 템플릿에서 사용할 때는 `<transition-group>` 으로 참조해야 합니다.

입장 / 퇴장 트랜지션

다음은 `<TransitionGroup>` 을 사용하여 `v-for` 리스트에 입장/퇴장 트랜지션을 적용하는 예시입니다:

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

CSS

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

Add at random index Remove at random index

1
2
3
4
5

이동 트랜지션

위의 데모에는 몇 가지 명백한 결함이 있습니다: 항목이 삽입되거나 제거될 때, 주변 항목들이 부드럽게 이동하지 않고 즉시 "점프"합니다. 몇 가지 추가 CSS 규칙을 추가하여 이를 개선할 수 있습니다:

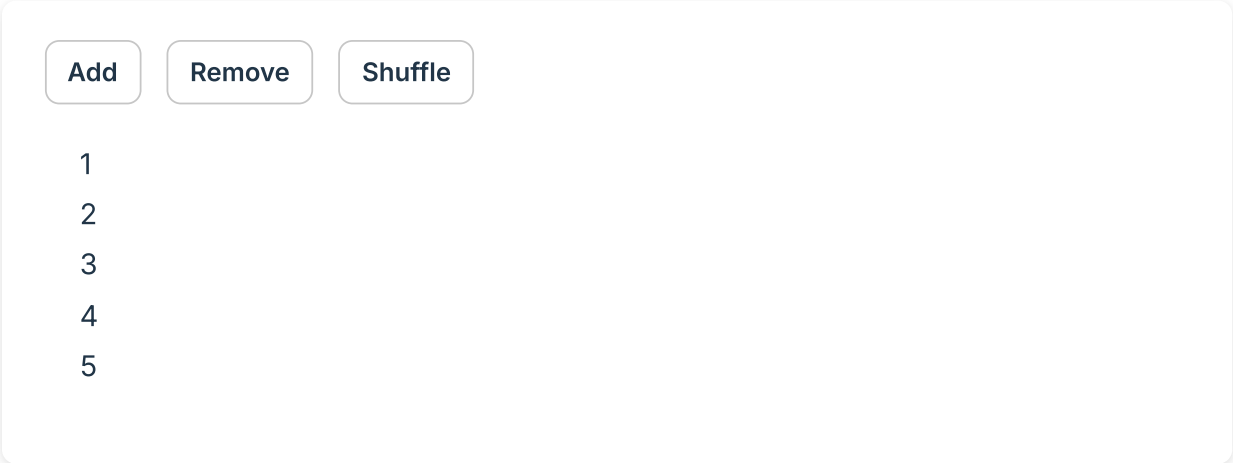
CSS

```
.list-move, /* 이동하는 요소에 트랜지션 적용 */
.list-enter-active,
.list-leave-active {
  transition: all 0.5s ease;
}

.list-enter-from,
.list-leave-to {
  opacity: 0;
  transform: translateX(30px);
}
```

```
/* 확장하는 항목이 레이아웃 흐름에서 제거되어
   이동 애니메이션이 올바르게 계산될 수 있도록 합니다. */
.list-leave-active {
  position: absolute;
}
```

이제 훨씬 더 부드럽게 보입니다. 전체 리스트가 섞일 때도 자연스럽게 애니메이션이 적용됩니다:



전체 예제

커스텀 TransitionGroup 클래스

`<TransitionGroup>` 에 `moveClass` prop을 전달하여 이동하는 요소에 대한 커스텀 트랜지션 클래스를 지정할 수도 있습니다. 이는 `<Transition>` 에서의 커스텀 트랜지션 클래스와 동일한 방식입니다.

리스트 트랜지션의 스태거링

데이터 속성을 통해 JavaScript 트랜지션과 통신함으로써, 리스트 내 트랜지션을 스태거(순차적으로 지연)하는 것도 가능합니다. 먼저, 항목의 인덱스를 DOM 요소의 데이터 속성으로 렌더링 합니다:

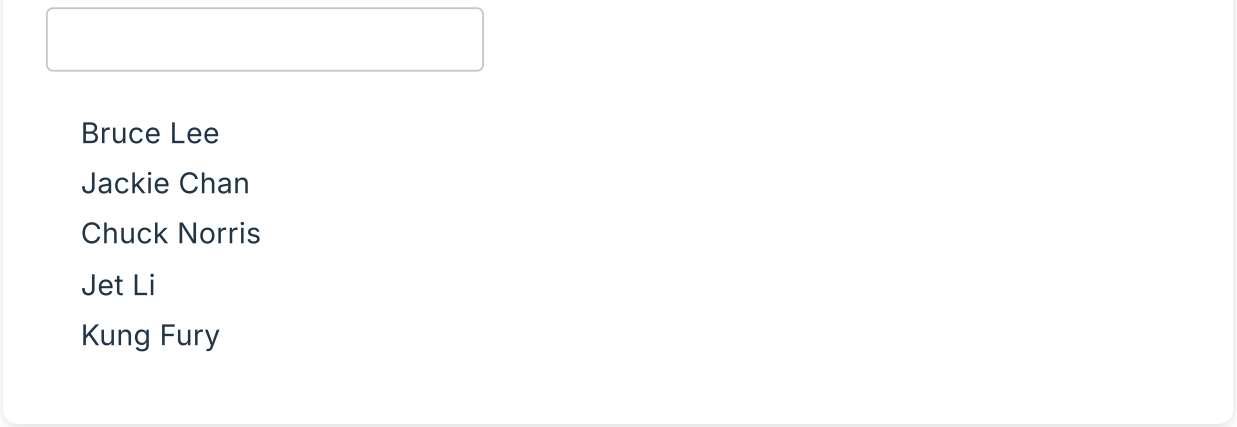
template

```
<TransitionGroup
  tag="ul"
  :css="false"
  @before-enter="onBeforeEnter"
  @enter="onEnter"
  @leave="onLeave"
>
  <li
```

```
      :key="item.msg"
      :data-index="index"
    >
      {{ item.msg }}
    </li>
  </TransitionGroup>
```

그런 다음, JavaScript 훅에서 데이터 속성을 기반으로 지연을 주어 요소를 애니메이션화합니다. 이 예제에서는 GSAP 라이브러리를 사용하여 애니메이션을 수행합니다:

```js
function onEnter(el, done) {
  gsap.to(el, {
    opacity: 1,
    height: '1.6em',
    delay: el.dataset.index * 0.15,
    onComplete: done
  })
}
```



플레이그라운드에서 전체 예제 보기

관련 문서

`<TransitionGroup>` API 레퍼런스

GitHub에서 이 페이지 편집

Previous
Transition

Next
KeepAlive