**Enter/Leave & List Transitions**

## [Overview](https://vuejs.org/v2/guide/transitions.html#Overview)

* автоматически применять CSS-классы для анимаций и переходов
* интегрировать сторонние библиотеки CSS-анимаций, такие как Animate.css
* использовать JavaScript для работы с DOM напрямую в хуках переходов
* интегрировать сторонние библиотеки JavaScript-анимаций, такие как Velocity.js

## [Transitioning Single Elements/Components](https://vuejs.org/v2/guide/transitions.html#Transitioning-Single-Elements-Components)

<transition name="fade">

<p v-if="show">привет</p>

</transition>

.fade-enter-active, .fade-leave-active {

transition: opacity .5s;

}

.fade-enter, .fade-leave-to /\* .fade-leave-active до версии 2.1.8 \*/ {

opacity: 0;

}

### # [Transition Classes](https://vuejs.org/v2/guide/transitions.html#Transition-Classes)

1. v-enter:
2. v-enter-active:
3. v-enter-to:
4. v-leave:
5. v-leave-active:
6. v-leave-to:

Для каждого из этих классов указывается префикс, соответствующий имени перехода. Префикс v- применяется по умолчанию, если элемент <transition> используется без указания параметра name . Например, для <transition name="my-transition"> вместо класса v-enter будет применяться my-transition-enter.

### # [CSS Transitions](https://vuejs.org/v2/guide/transitions.html#CSS-Transitions)

<button @click="show = !show">

Переключить отрисовку

</button>

<transition name="slide-fade">

<p v-if="show">привет</p>

</transition>

.slide-fade-enter-active {

transition: all .3s ease;

}

.slide-fade-leave-active {

transition: all .8s cubic-bezier(1.0, 0.5, 0.8, 1.0);

}

.slide-fade-enter, .slide-fade-leave-to

/\* .slide-fade-leave-active до версии 2.1.8 \*/ {

transform: translateX(10px);

opacity: 0;

}

### # [CSS Animations](https://vuejs.org/v2/guide/transitions.html#CSS-Animations)

<button @click="show = !show">Переключить отображение</button>

<transition name="bounce">

<p v-if="show">

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris facilisis enim libero, at lacinia diam fermentum id. Pellentesque habitant morbi tristique senectus et netus.</p>

</transition>

.bounce-enter-active {

animation: bounce-in .5s;

}

.bounce-leave-active {

animation: bounce-in .5s reverse;

}

@keyframes bounce-in {

0% {

transform: scale(0);

}

50% {

transform: scale(1.5);

}

100% {

transform: scale(1);

}

}

### #[Custom Transition Classes](https://vuejs.org/v2/guide/transitions.html#Custom-Transition-Classes)

* enter-class
* enter-active-class
* enter-to-class (в версии 2.1.8+)
* leave-class
* leave-active-class
* leave-to-class (в версии 2.1.8+)

осы дайын класстарды cdn-байланыс арқылы пайдалануға болады

<link href="https://cdn.jsdelivr.net/npm/animate.css@3.5.1" rel="stylesheet" type="text/css">

<div id="example-3">

<button @click="show = !show">

Переключить отрисовку

</button>

<transition

name="custom-classes-transition"

enter-active-class="animated tada"

leave-active-class="animated bounceOutRight"

>

<p v-if="show">привет</p>

</transition>

</div>

### #[Using Transitions and Animations Together](https://vuejs.org/v2/guide/transitions.html#Using-Transitions-and-Animations-Together)

анимация аяқталғанын білу үшін transitionend, либо animationend комондаларын қолдануға болады

### #[Explicit Transition Durations](https://vuejs.org/v2/guide/transitions.html#Explicit-Transition-Durations)

анимацияның аяқталуын анықтау, басқару үшін duration осы команданы қолдану арқылы болады,

<transition :duration="1000">...</transition>

<transition :duration="{ enter: 500, leave: 800 }">...</transition>

### #[JavaScript Hooks](https://vuejs.org/v2/guide/transitions.html#JavaScript-Hooks)

Анимацияны осылай функция қылып команда беруге болады

<button @click="show = !show">

Переключить

</button>

<transition

v-on:before-enter="**beforeEnter**"

v-on:enter="**enter**"

v-on:leave="**leave**"

v-bind:css="false"

>

<p v-if="show">

Демо

</p>

</transition>

methods: {

**beforeEnter**: function (el) {

el.style.opacity = 0

el.style.transformOrigin = 'left'

},

**enter**: function (el, done) {

Velocity(el, { opacity: 1, fontSize: '1.4em' }, { duration: 300 })

Velocity(el, { fontSize: '1em' }, { complete: done })

},

**leave**: function (el, done) {

Velocity(el, { translateX: '15px', rotateZ: '50deg' }, { duration: 600 })

Velocity(el, { rotateZ: '100deg' }, { loop: 2 })

Velocity(el, {

rotateZ: '45deg',

translateY: '30px',

translateX: '30px',

opacity: 0

}, { complete: done })

}

}

## [Transitions on Initial Render](https://vuejs.org/v2/guide/transitions.html#Transitions-on-Initial-Render)

Если вы хотите, чтобы пользователь увидел анимацию перехода и при изначальной отрисовке, добавьте атрибут appear:

<transition appear>

<!-- ... -->

</transition>

<transition

appear

appear-class="custom-appear-class"

appear-to-class="custom-appear-to-class" (в версии 2.1.8+)

appear-active-class="custom-appear-active-class"

>

<!-- ... -->

</transition>

<transition

appear

v-on:before-appear="customBeforeAppearHook"

v-on:appear="customAppearHook"

v-on:after-appear="customAfterAppearHook"

v-on:appear-cancelled="customAppearCancelledHook"

>

<!-- ... -->

</transition>

## [Transitioning Between Elements](https://vuejs.org/v2/guide/transitions.html#Transitioning-Between-Elements)

Бірдей элементтерді шатасулардан сақтау үшін әртүрлі key="exampleName" қолдануға болады

<transition>

<button v-if="isEditing" key="save">

Сохранить

</button>

<button v-else key="edit">

Редактировать

</button>

</transition>

Жоғарыдағы кодты қысқа түрде осылай жазуға болады

<transition>

<button v-bind:key="isEditing">

{{ isEditing ? 'Сохранить' : 'Редактировать' }}

</button>

</transition>

### #[Transition Modes](https://vuejs.org/v2/guide/transitions.html#Transition-Modes)

Осы командаларды қолданып автоматты дайын анимациялра орындауға болады

* in-out: Сначала появляется новый элемент, и только после этого исчезает старый.
* out-in: Сначала исчезает старый элемент, и только после этого появляется новый.

<transition name="fade" mode="out-in">

<!-- ... кнопки ... -->

</transition>

  <transition name="component-fade" mode="in-out">

      <button v-bind:key="isEditing">

        {{ isEditing ? "Сохранить" : "Редактировать" }}

      </button>

  </transition>

    <button @click="isEditing = !isEditing">on</button>

.component-fade-enter-active,

.component-fade-leave-active {

  transition: opacity 0.3s ease;

}

.component-fade-enter, .component-fade-leave-to

/\* .component-fade-leave-active до версии 2.1.8 \*/ {

  opacity: 0;

}

## [Transitioning Between Components](https://vuejs.org/v2/guide/transitions.html#Transitioning-Between-Components)

Төмендегі мысал арқылы компоненттерге де анимация переход жасаса болады

<transition name="component-fade" mode="out-in">

<component v-bind:is="view"></component>

</transition>

new Vue({

el: '#transition-components-demo',

data: {

view: 'v-a'

},

components: {

'v-a': {

template: '<div>Компонент А</div>'

},

'v-b': {

template: '<div>Компонент Б</div>'

}

}

})

.component-fade-enter-active, .component-fade-leave-active {

transition: opacity .3s ease;

}

.component-fade-enter, .component-fade-leave-to

/\* .component-fade-leave-active до версии 2.1.8 \*/ {

opacity: 0;

}

## [List Transitions](https://vuejs.org/v2/guide/transitions.html#List-Transitions)

v-for цикл болған жағдайда <transition> орнына

<transition-group> қолдану керек

### #[List Entering/Leaving Transitions](https://vuejs.org/v2/guide/transitions.html#List-Entering-Leaving-Transitions)

<transition-group> - қа байланысты Мысал

<button v-on:click="add()">Добавить</button>

    <button v-on:click="remove()">Удалить</button>

    <transition-group name="list" tag="div">

      <span v-for="item in items" v-bind:key="item" class="list-item">

        {{ item }}

      </span>

    </transition-group>

  data() {

    return {

      items: [1, 2, 3, 4, 5, 6, 7, 8, 9],

      nextNum: 10,

    };

  },

  methods: {

    randomIndex() {

      return Math.floor(Math.random() \* this.items.length);

    },

    add() {

      this.items.splice(this.randomIndex(), 0, this.nextNum++);

    },

    remove() {

      this.items.splice(this.randomIndex(), 1);

    },

  },

.list-item {

  display: inline-block;

  margin-right: 10px;

  border: 1px solid;

}

.list-enter-active,

.list-leave-active {

  transition: all 1s;

}

.list-enter, .list-leave-to /\* .list-leave-active до версии 2.1.8 \*/ {

  opacity: 0;

  transform: translateY(30px);

}

### #[List Move Transitions](https://vuejs.org/v2/guide/transitions.html#List-Move-Transitions)

**v-move тәсілі, данныйларды анимациялық переместит жасауға көмектеседі**

<script src="https://cdnjs.cloudflare.com/ajax/libs/lodash.js/4.14.1/lodash.min.js"></script>

осы ссылканы қолдану арқылы [**FLIP**](https://aerotwist.com/blog/flip-your-animations/) тәсіліне доступ ашылады

Мысал:

<script src="https://cdnjs.cloudflare.com/ajax/libs/lodash.js/4.14.1/lodash.min.js"></script>

<div id="list-complete-demo" class="demo">

<button v-on:click="shuffle">Перемешать</button>

<button v-on:click="add">Добавить</button>

<button v-on:click="remove">Удалить</button>

<transition-group name="list-complete" tag="p">

<span

v-for="item in items"

v-bind:key="item"

class="list-complete-item"

>

{{ item }}

</span>

</transition-group>

</div>

new Vue({

el: '#list-complete-demo',

data: {

items: [1,2,3,4,5,6,7,8,9],

nextNum: 10

},

methods: {

randomIndex: function () {

return Math.floor(Math.random() \* this.items.length)

},

add: function () {

this.items.splice(this.randomIndex(), 0, this.nextNum++)

},

remove: function () {

this.items.splice(this.randomIndex(), 1)

},

shuffle: function () {

this.items = \_.shuffle(this.items)

}

}

})

.list-complete-item {

transition: all 1s;

display: inline-block;

margin-right: 10px;

}

.list-complete-enter, .list-complete-leave-to

/\* .list-complete-leave-active до версии 2.1.8 \*/ {

opacity: 0;

transform: translateY(30px);

}

.list-complete-leave-active {

position: absolute;

}

### #[Staggering List Transitions](https://vuejs.org/v2/guide/transitions.html#Staggering-List-Transitions)

Серпімді анимация

input - қа жазылған кезде сразу рякция (функция) орындалады

Мысал:

<script src="https://cdnjs.cloudflare.com/ajax/libs/velocity/1.2.3/velocity.min.js"></script>

<div id="staggered-list-demo">

<input v-model="query">

<transition-group

name="staggered-fade"

tag="ul"

v-bind:css="false"

v-on:before-enter="beforeEnter"

v-on:enter="enter"

v-on:leave="leave"

>

<li

v-for="(item, index) in computedList"

v-bind:key="item.msg"

v-bind:data-index="index"

>{{ item.msg }}</li>

</transition-group>

</div>

new Vue({

el: '#staggered-list-demo',

data: {

query: '',

list: [

{ msg: 'Брюс Ли' },

{ msg: 'Джеки Чан' },

{ msg: 'Чак Норрис' },

{ msg: 'Джет Ли' },

{ msg: 'Кунг Фьюри' }

]

},

computed: {

computedList: function () {

var vm = this

return this.list.filter(function (item) {

return item.msg.toLowerCase().indexOf(vm.query.toLowerCase()) !== -1

})

}

},

methods: {

beforeEnter: function (el) {

el.style.opacity = 0

el.style.height = 0

},

enter: function (el, done) {

var delay = el.dataset.index \* 150

setTimeout(function () {

Velocity(

el,

{ opacity: 1, height: '1.6em' },

{ complete: done }

)

}, delay)

},

leave: function (el, done) { сарымен боялған мына переменныйды қолдануды үйрену керек

var delay = el.dataset.index \* 150

setTimeout(function () {

Velocity(

el,

{ opacity: 0, height: 0 },

{ complete: done }

)

}, delay)

}

}

})

.list-complete-item {

transition: all 1s;

display: inline-block;

margin-right: 10px;

}

.list-complete-enter, .list-complete-leave-to

/\* .list-complete-leave-active до версии 2.1.8 \*/ {

opacity: 0;

transform: translateY(30px);

}

.list-complete-leave-active {

position: absolute;

}

## [Reusable Transitions](https://vuejs.org/v2/guide/transitions.html#Reusable-Transitions)

Анимациялық переходты компанент арқылы , қайта-қайта орындауға болады

## [Dynamic Transitions](https://vuejs.org/v2/guide/transitions.html" \l "Dynamic-Transitions" \o "Dynamic Transitions) ???

<transition v-bind:name="transitionName">

<!-- ... -->

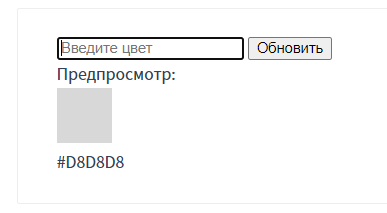
</transition>

# State Transitions *Примерлер*

## [Animating State with Watchers](https://vuejs.org/v2/guide/transitioning-state.html#Animating-State-with-Watchers)

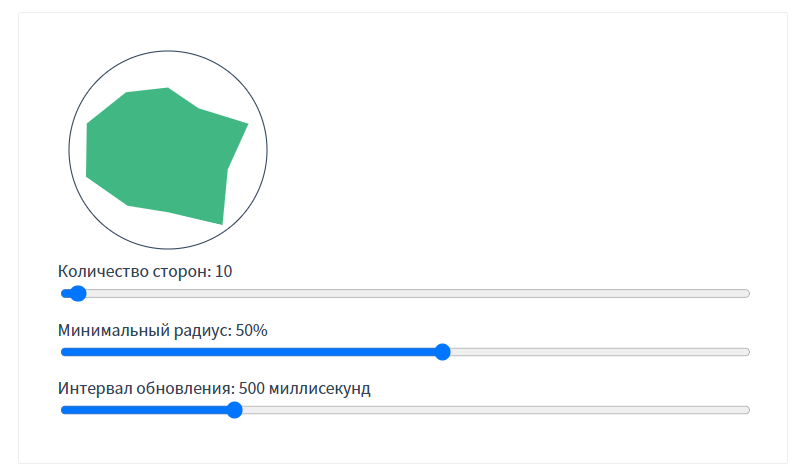
input - қа жазылған нәрсе анимация түрінде жәйлап өзгереді

бағыттау өзгерту нәтижесінде жағдай анимацияланады

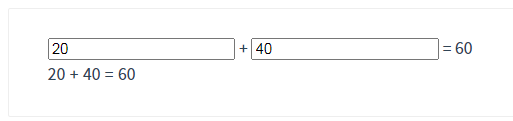
## [Dynamic State Transitions](https://vuejs.org/v2/guide/transitioning-state.html#Dynamic-State-Transitions)

Күйдің динамикалық ауысуының анимациялық тәсілі



## [Organizing Transitions into Components](https://vuejs.org/v2/guide/transitioning-state.html#Organizing-Transitions-into-Components)

Компонент ретінде ауысуларды көрсету , компоненттерді қолданып анимацияларды орынау



## [Bringing Designs to Life](https://vuejs.org/v2/guide/transitioning-state.html#Bringing-Designs-to-Life)

живой, тірі сияқты анимациялар істеу

