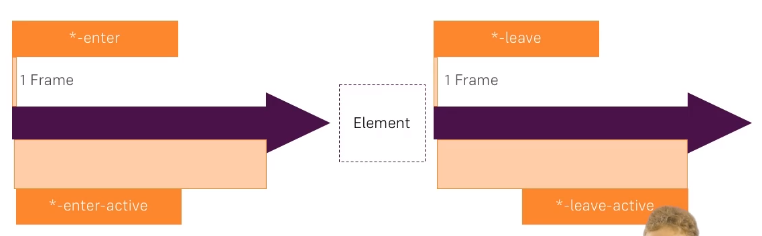
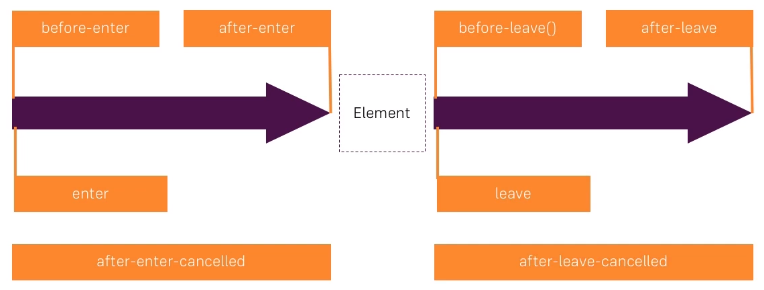
**Section 36 – Adding Animations And Transitions**

Chapter 515 – Chapter 539

1. We can use <transition></transition> to animate element. We can use this tag only for one element. For more than one element, we can use <transition-group></transition-group>.
2. Process Animation



1. Transition : Fade, Animation : Slide
2. Transition JS Hooks



1. :css=”false” means for that element we don’t need css style.
2. JavaScript Animation

beforeEnter(el) {

console.log('beforeEnter');

this.elementWidth = 100;

el.style.width = this.elementWidth + 'px';

},

enter(el, done) {

console.log('enter');

let round = 1;

const interval = setInterval(() => {

el.style.width = (this.elementWidth + round \* 10) + 'px';

round++;

if (round > 20) {

clearInterval(interval);

done();

}

}, 20);

},

afterEnter(el) {

console.log('afterEnter');

},

enterCancelled(el) {

console.log('enterCancelled');

},

beforeLeave(el) {

console.log('beforeLeave');

this.elementWidth = 300;

el.style.width = this.elementWidth + 'px';

},

leave(el, done) {

console.log('leave');

let round = 1;

const interval = setInterval(() => {

el.style.width = (this.elementWidth - round \* 10) + 'px';

round++;

if (round > 20) {

clearInterval(interval);

done();

}

}, 20);

},

afterLeave(el) {

console.log('afterLeave');

},

leaveCancelled(el) {

console.log('leaveCancelled');

},

1. Final Result

App.vue

<template>

    <div class="container">

        <div class="row">

            <div class="col-xs-12 col-sm-8 col-sm-offset-2 col-md-6 col-md-offset-3">

                <h1>Animations</h1>

                <hr>

                <select v-model="alertAnimation" class="form-control">

                    <option value="fade">Fade</option>

                    <option value="slide">Slide</option>

                </select>

                <br><br>

                <button class="btn btn-primary" @click="show = !show">Show Alert</button>

                <br><br>

                <transition :name="alertAnimation">

                    <div class="alert alert-info" v-show="show">This is some Info</div>

                </transition>

                <transition name="slide" type="animation" appear>

                    <div class="alert alert-info" v-if="show">This is some Info</div>

                </transition>

                <transition

                        enter-active-class="animated bounce"

                        leave-active-class="animated shake"

                >

                    <div class="alert alert-info" v-if="show">This is some Info</div>

                </transition>

                <transition :name="alertAnimation" mode="out-in">

                    <div class="alert alert-info" v-if="show" key="info">This is some Info</div>

                    <div class="alert alert-warning" v-else key="warning">This is some Warning</div>

                </transition>

                <hr>

                <button class="btn btn-primary" @click="load = !load">Load / Remove Element</button>

                <br><br>

                <transition

                        @before-enter="beforeEnter"

                        @enter="enter"

                        @after-enter="afterEnter"

                        @enter-cancelled="enterCancelled"

                        @before-leave="beforeLeave"

                        @leave="leave"

                        @after-leave="afterLeave"

                        @leave-cancelled="leaveCancelled"

                        :css="false">

                    <div style="width: 300px; height: 100px; background-color: lightgreen" v-if="load"></div>

                </transition>

                <hr>

                <button class="btn btn-primary"

                        @click="selectedComponent == 'app-success-alert' ? selectedComponent = 'app-danger-alert' : selectedComponent = 'app-success-alert'">

                    Toogle Components

                </button>

                <br><br>

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

                    <component :is="selectedComponent"></component>

                </transition>

                <hr>

                <button class="btn btn-primary" @click="addItem">Add Item</button>

                <br><br>

                <ul class="list-group">

                    <transition-group name="slide">

                        <li

                                class="list-group-item"

                                v-for="(number, index) in numbers"

                                @click="removeItem(index)"

                                style="cursor: pointer"

                                :key="number">{{ number }}

                        </li>

                    </transition-group>

                </ul>

            </div>

        </div>

    </div>

</template>

<script>

    import DangerAlert from './DangerAlert.vue';

    import SuccessAlert from './SuccessAlert.vue';

    export default {

        data() {

            return {

                show: false,

                load: true,

                alertAnimation: 'fade',

                elementWidth: 100,

                selectedComponent: 'app-success-alert',

                numbers: [1, 2, 3, 4, 5]

            }

        },

        methods: {

            beforeEnter(el) {

                console.log('beforeEnter');

                this.elementWidth = 100;

                el.style.width = this.elementWidth + 'px';

            },

            enter(el, done) {

                console.log('enter');

                let round = 1;

                const interval = setInterval(() => {

                    el.style.width = (this.elementWidth + round \* 10) + 'px';

                    round++;

                    if (round > 20) {

                        clearInterval(interval);

                        done();

                    }

                }, 20);

            },

            afterEnter(el) {

                console.log('afterEnter');

            },

            enterCancelled(el) {

                console.log('enterCancelled');

            },

            beforeLeave(el) {

                console.log('beforeLeave');

                this.elementWidth = 300;

                el.style.width = this.elementWidth + 'px';

            },

            leave(el, done) {

                console.log('leave');

                let round = 1;

                const interval = setInterval(() => {

                    el.style.width = (this.elementWidth - round \* 10) + 'px';

                    round++;

                    if (round > 20) {

                        clearInterval(interval);

                        done();

                    }

                }, 20);

            },

            afterLeave(el) {

                console.log('afterLeave');

            },

            leaveCancelled(el) {

                console.log('leaveCancelled');

            },

            addItem() {

                const pos = Math.floor(Math.random() \* this.numbers.length);

                this.numbers.splice(pos, 0, this.numbers.length + 1);

            },

            removeItem(index) {

                this.numbers.splice(index, 1);

            }

        },

        components: {

            appDangerAlert: DangerAlert,

            appSuccessAlert: SuccessAlert

        }

    }

</script>

<style>

    .fade-enter {

        opacity: 0;

    }

    .fade-enter-active {

        transition: opacity 1s;

    }

    .fade-leave {

        /\*opacity: 1;\*/

    }

    .fade-leave-active {

        transition: opacity 1s;

        opacity: 0;

    }

    .slide-enter {

        opacity: 0;

        /\*transform: translateY(20px);\*/

    }

    .slide-enter-active {

        animation: slide-in 1s ease-out forwards;

        transition: opacity .5s;

    }

    .slide-leave {

    }

    .slide-leave-active {

        animation: slide-out 1s ease-out forwards;

        transition: opacity 1s;

        opacity: 0;

        position: absolute;

    }

    .slide-move {

        transition: transform 1s;

    }

    @keyframes slide-in {

        from {

            transform: translateY(20px);

        }

        to {

            transform: translateY(0);

        }

    }

    @keyframes slide-out {

        from {

            transform: translateY(0);

        }

        to {

            transform: translateY(20px);

        }

    }

</style>

DangerAlert.vue

<template>

    <div class="alert alert-danger">This is dangerous!</div>

</template>

SuccessAlert.vue

<template>

    <div class="alert alert-success">This is successful!</div>

</template>

Main.js

import Vue from 'vue'

import App from './App.vue'

new Vue({

  el: '#app',

  render: h => h(App)

})