

项目八 过渡与动画

课程目标

- 本节将从 CSS 过渡 transition、CSS 动画 animation 及配合使用第三方 CSS 动画库(如 animate.css) 这三方面来详细介绍 Vue 过渡效果与 Vue 动画效果。

本章内容

- 8.1 transition组件
 - 8.2 单元素/组件的过渡
 - 8.3 多个元素的过渡
 - 8.4 多个组件的过渡
 - 8.5 综合案例
-

过渡与动画概述

- Vue实现了过渡与动画，Vue的过渡系统可以在元素从DOM中插入或移除时自动应用过渡效果。
- Vue.js会在适当的时机为触发CSS过渡或动画，也可以提供相应JavaScript钩子函数在过渡过程中执行自定义的DOM操作。

8.1 transition组件

- 在 Vue 中，提供了实现过渡动画的内置组件 transition。基本用法就是给我们需要动画的标签外面嵌套 transition 标签，并且给上属性。实现过渡动画一般格式：

使用过渡组件 transition。

```
<transition name="fade">
```

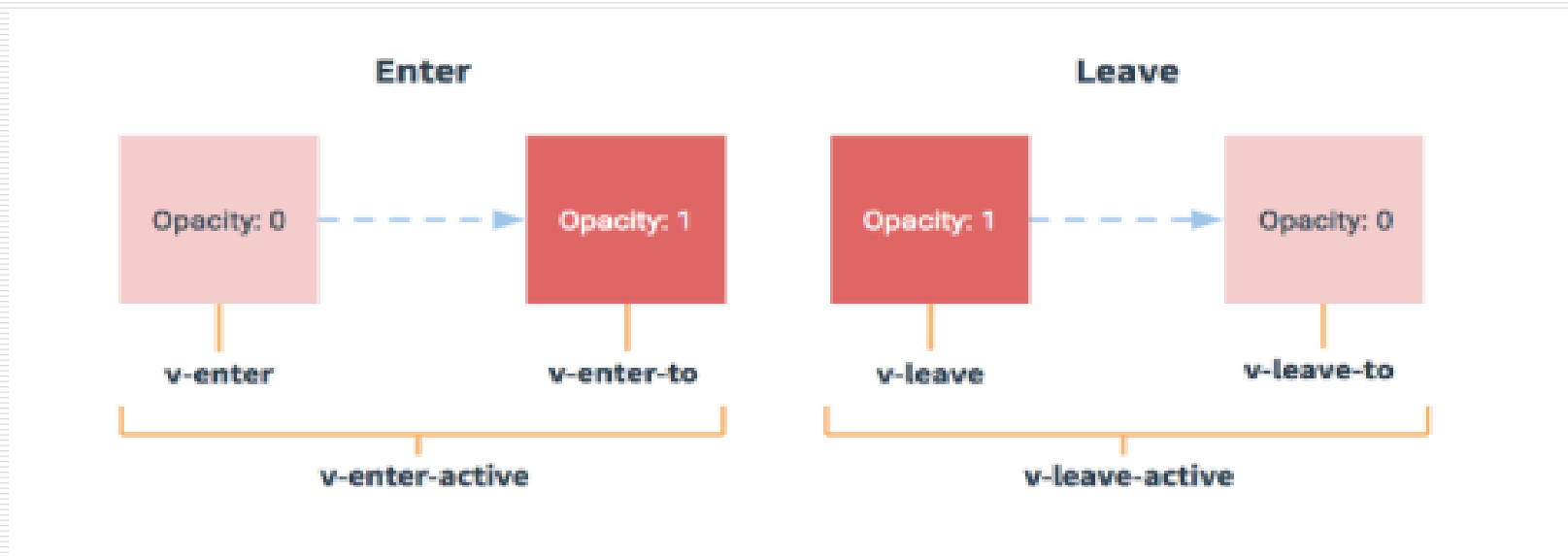
```
<!--需要动画的div标签-->
```

```
<div></div>
```

```
</transition>
```

8.1 transition组件

- Vue提供了6个CSS类名在enter/leave的过渡中切换。



8.1 transition组件

过渡的类名

在进入/离开的过渡中，会有 6 个 class 切换。

1. `v-enter` : 定义进入过渡的开始状态。在元素被插入之前生效，在元素被插入之后的下一帧移除。
2. `v-enter-active` : 定义进入过渡生效时的状态。在整个进入过渡的阶段中应用，在元素被插入之前生效，在过渡/动画完成之后移除。这个类可以被用来定义进入过渡的过程时间，延迟和曲线函数。
3. `v-enter-to` : 2.1.8 版及以上定义进入过渡的结束状态。在元素被插入之后下一帧生效 (与此同时 `v-enter` 被移除)，在过渡/动画完成之后移除。
4. `v-leave` : 定义离开过渡的开始状态。在离开过渡被触发时立刻生效，下一帧被移除。
5. `v-leave-active` : 定义离开过渡生效时的状态。在整个离开过渡的阶段中应用，在离开过渡被触发时立刻生效，在过渡/动画完成之后移除。这个类可以被用来定义离开过渡的过程时间，延迟和曲线函数。
6. `v-leave-to` : 2.1.8 版及以上定义离开过渡的结束状态。在离开过渡被触发之后下一帧生效 (与此同时 `v-leave` 被删除)，在过渡/动画完成之后移除。

8.1 transition组件

- transition组件的具体使用，及CSS类名的使用。

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<style>
  .v-enter{
    opacity:0;
  }
  .v-enter-active{
    transition:opacity .5s;
  }
  .v-leave-active{
    transition:transform .5s; .
  }
  .v-leave-to{
    transform:translateX(10px);
  }
</style>
```


8.1 transition组件

```
</head>
<body>
<div id="demo">
<button v-on:click="show = ! show" >Toggle</button>
  <transition>
    <p v-if="show">我是精灵小天使</p>
  </transition>
</div>
<script type="text/javascript" src="../js/Vue.js"></script>
  <script>
    new Vue({
      el: '#demo',
      data: {
        show: true
      }
    })
  </script >
</body>
</html>
```

8.1 transition组件

如果 transition 组件定义了 name,这时所有 “v-” 开头的 class 类名更换为 “fade-” 开头。

```
<style>
.fade-enter{opacity:0;
}
.fade-enter-active{
transition:opacity .5s;
}
.fade-leave-active{
transition:transform .5s;
}
.fade-leave-to{
transform: translateX(10px);
}
</style>
```

8.2 单元素、组件的过渡

- 8.2.1 transition 标签结合 CSS 样式实现动画
 - 8.2.2 animate.css 结合 transition 实现动画
 - 8.2.3 钩子函数实现动画
-

8.2 单元素、组件的过渡

- Vue提供了transition的封装组件，在下列情形中，可以给任何元素和组件添加进入/离开过渡。

条件渲染 (使用 v-if)

条件展示 (使用 v-show)

动态组件

组件根节点

8.2 单元素、组件的过渡

实现过渡动画通常有如下几种方式:

使用 Vue 的 transition 标签结合 CSS 样式实现动画

利用 animate.css 结合 transition 实现动画。

利用 Vue 中的钩子函数实现动画。

8.2.1 transition 标签结合 CSS 样式实现动画

- 常用的过渡使用通常使用 CSS 过渡。在代码中设置 CSS 样式来实现过渡 `fade-enter-active`、`fade-leave-active`、`fade-enter`、`fade-leave-to`。

8.2.1 transition 标签结合 CSS 样式实现动画

□ 例 8.1 transition+css 实现过渡动画:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title> transition+css 实现过渡动画</title>
<style>
. button{
    width:400px;
    height: 40px;
    line -height:40px;
    background-color :#FFDAB9;
    text-align:center;
}
```

8.2.1 transition 标签结合 CSS 样式 实现动画

```
p{
  width:400px;
  margin:0;
  background-color: #FFFACD;
}
.fade-enter-active,
.fade-leave-active {
  transition: opacity .5s; }
.fade-enter,
.fade-leave-to
/* .fade-leave-active below version 2.1.8 */
{
  opacity: 0; }
</style>
</head>
<body>
  <div id="demo">
    <div class="button" v-on:click="show = ! show" >
      Toggle
    </div>
    <transition name="fade">
```

执行变换属性
为透明，变换
时间0.5毫秒。

8.2.1 transition 标签结合 CSS 样式 实现动画

`<p v-if= "show" >`慕课网是一款专业学习 IT开发知识技能的慕课平台，内容涵盖前端开发，JAVA，大数据，电子商务等。斤斗云学堂已累计培养数千学员，使学员在IT学习和就业方面得到了很大的帮助，也愿意在你学习和就业的路上祝你一臂之力，实现腾飞斤斗云学堂现在每天都有千余名学生在线学习，平台有完整的学习课件，题库，作业系统，学员可以在线学课程，交作业。学生通过按照学习大纲要求，视频学习，每节课测验，交作业，参加阶段考试，项目实训，掌握开发技能。`</p>`

```
</transition>
```

```
</div>
```

```
<script type=" text/javascript" src=" ../js/Vue.js"></script>
```

```
<script>
```

```
  new Vue({  
    el: '#demo',
```

```
    data:{  
      show: true  
    }  
  }
```

```
)
```

```
</script>
```

```
</body>
```

```
</html>
```

8.2.2 animate.css 结合transition 实现动画

- 首先来介绍下自定义过渡类名，在实现动画中会用到，可以通过以下特性来自定义过渡类名：

enter-class

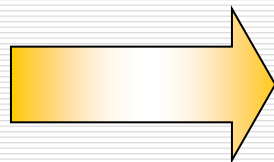
leave-class

enter-to-class

leave-to-class

leave-active-class

enter-active-class



它们的优先级高于普通的类名，对于 Vue 的过渡系统和其他第三方 CSS 动画库，如 animate.css 相结合使用非常有用。

8.2.2 animate.css 结合 transition 实现动画

例 8.2 animate.css+transition 实现动画

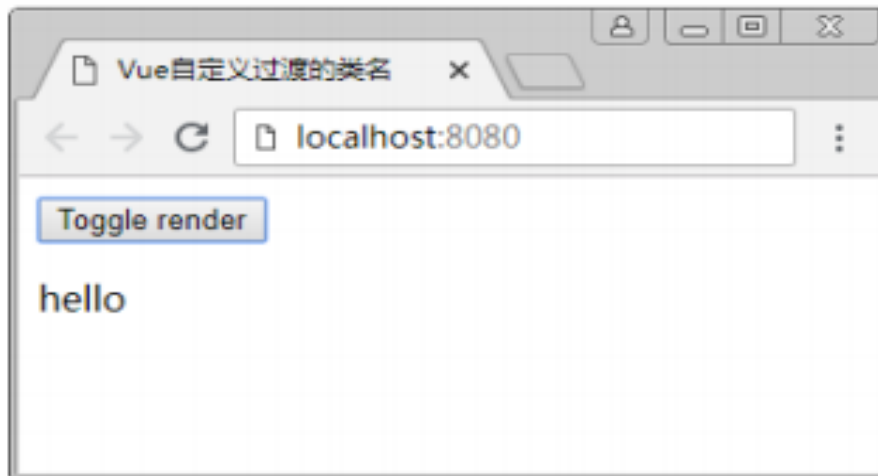
```
<!DOCTYPE html>
<html>
<head>
  <meta charset= "utf-8" />
  <title>Vue自定义过渡类名</title>
  <link href= " https://cdn. jsdelivr.net/ npm/animate.css @3.5.1" rel="stylesheet" type=
"text/css">
</head>
<body>
  <div id="demo" >
    <button @click="show = ! show">
      Toggle render
    < /button>
```

8.2.2 animate.css 结合 transition 实现动画

```
<transition name=" custom-classes -transition" enter-active-  
class="animated tada"leave-active-class=" animated bounceOutRight">  
  <p v-if="show" >hello</p>  
  </transition>  
</div>  
  
  <script src=" ../js/Vue.js"></script>  
<script>  
new Vue({  
  
  el: '#demo',  
  data: {  
show: true  
}  
  })  
</script>  
</body>  
</html>
```

8.2.2 animate.css 结合 transition 实现动画

- 引入第三方 animate.css 动画文件，设定动画效果 `enter-active-class="animated tada"` `leave-active-class="animated bounceOutRight"`。运行结果如下图所示。



8.2.3 钩子函数实现动画

例 8.3 钩子函数实现动画

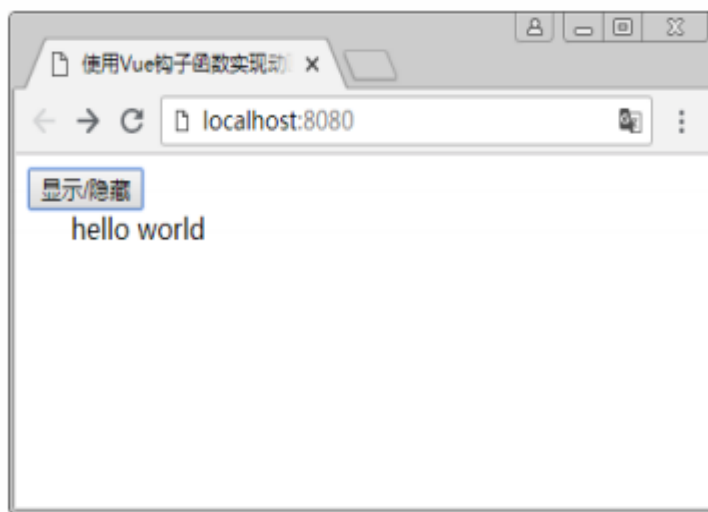
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name "viewport" content= "width=device-width, initial-scale=1.0">
<title> 使用Vue钩子函数实现动画</title>
<style>
  .show {
    transition: all 0.5s;}
  </style>
</head>
<body>
  <div id="app">
    <button @click="loggle"> 显示/隐藏</button><br>
    <transition @before-enter="beforeEnter" @enter="enter" @after-enter="afterEnter" v-bind:css="false">
      <div class="show" v-show-"isshow">hello world</div>
    </transition>
  </div>
<script src='. ./js/Vue.js"></script>
```

8.2.3 钩子函数实现动画

```
<script
new Vue({
el: '#app' ,
data: {
  isshow: false },
  methods: {
    toggle: function (){
      this-isshow = !this. isshow;},
    beforeEnter: function (el) {
      console.log("beforeEnter");
      el.style="padding-left:100px"; },
    enter: function(el, done){
      el. offsetHeight;
      el.style="padding-left:0px";
      done(); },
    afterEnter: function (el) {
      console.log("afterEnter"); }
  }
})
</script>
</body>
</html>
```

8.2.3 钩子函数实现动画

例 8.3 运行后单击显示/隐藏，helloworld 从右向左过渡，再次单击隐藏运行结果如下图所示



8.3 多个元素的过渡

- 最常见的多元素过渡是一个列表或者表格和描述这个列表表格为空消息的元素，如下代码片段。使用 CSS 过渡实现列表和标签的过渡。

```
<transition>
  <table v-if="items.length > 0">
    <!-- ... -->
  </table>
  <p v-else>没有发现数据</p>
</transition>
```

8.3 多个元素的过渡

例 8.4 多元素的过渡

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>多元素的过渡</title>
  <style>
    .fade-enter,
    .fade-leave-to {
      opacity: 0;
    }

    .fade-enter-active,
    .fade-leave-active {
      transition: opacity .5s;
    }
  </style>
  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
</head>
```

```
<body>
  <div id="app">
    <button @click="clear">清空数据</button>
    <button @click="reset">重置</button>
    <transition name="fade">
      <ul v-if="items.length > 0">
        <li v-for="item in items">{{item}}</li>
      </ul>
      <p v-else>没有显示的数据.</p>
    </transition>
  </div>
  <script>
    new Vue({
      el: '#app',
      data: {
        items: ['JavaScript高级', 'Vue', 'CSS3']
      },
      methods: {
        clear() {
          this.items.splice(0);
        },
        reset() {
          history.go();
        }
      }
    })
  </script>
</body>
</html>
```

8.4 多个组件的过渡

- 多个组件的过渡简单很多，不需要使用 key 特性。只需要使用动态组件。
- 多组件的过渡使用动态组件
:is="currentView"只需要设置切换的动画，使用 CSS 样式来设置 name="fade"为过渡命名， mode="out-in"为过渡的模式。

8.4 多个组件的过渡

例 8.5 多组件的过渡

```
<style>

    .fade-enter,
    .fade-leave-to {
        opacity: 0;
    }

    .fade-enter-active,
    .fade-leave-active {
        transition: .5s;
    }

</style>
```

```
<body>
<div id="app">
<button @click="change">切换页面</button>
<transition name="fade" mode="out-in">
  <component :is="currentView"></component>
</transition>
</div>
<script>
new Vue({
  el: '#app',
  data: {
    index: 0,
    arr: [
      { template: `<div>组件A</div>` },
      { template: `<div>组件B</div>` },
      { template: `<div>组件C</div>` }
    ],
  },
  computed: {
    currentView() {
      return this.arr[this.index];
    }
  },
  methods: {
    change() {
      this.index = (++this.index) % 3;
    }
  }
})
</script>
</body>
```

8.5 综合案例

□ 例 8.6 表格中的过渡动画

编号: 品牌名称:

编号	品牌名称	创立时间	操作
1	超星在线学习平台	2020/5/7	删除
2	在线慕课平台	2020/5/7	删除

您确定要删除数据吗？

- 这个案例中定义过滤器dataFrm，自定义focus、color、v-if、v-for指令以及事件处理，并使用钩子函数beforeEnter、enter、beforeLeave、leave等实现过渡与动画。

本章小结

- 本章主要讲了 Vue 过渡与动画如何实现，实现过渡动画的内置组件 transition，过渡的 CSS 类名，自定义过渡的类名，钩子函数实现过渡与动画。
- 从单元素/组件过渡到多个元素的 过渡，多个组件的过渡，使用过渡与动画可以在视觉上有一个很好的体验。

习题

- 8.1 简述过渡与动画的含义及语句。
- 8.2 简述实现过渡与动画的三种方式。
- 8.3 编写一个过渡与动画的案例。