

Vue预习课

Vue预习课

拓展知识——过度&动画

css过度动画

小试牛刀

过度类名

使用CSS动画库

JavaScript 钩子

列表过度

拓展知识——过度&动画

Vue 在插入、更新或者移除 DOM 时，提供多种不同方式的应用过渡效果。包括以下工具：

- 在 CSS 过渡和动画中自动应用 class
- 可以配合使用第三方 CSS 动画库，如 Animate.css
- 在过渡钩子函数中使用 JavaScript 直接操作 DOM
- 可以配合使用第三方 JavaScript 动画库，如 Velocity.js

[过度&动画](#)

css过度动画

小试牛刀

transition组件会为嵌套元素自动添加class，可用于做css过度动画

范例：消息弹框动画

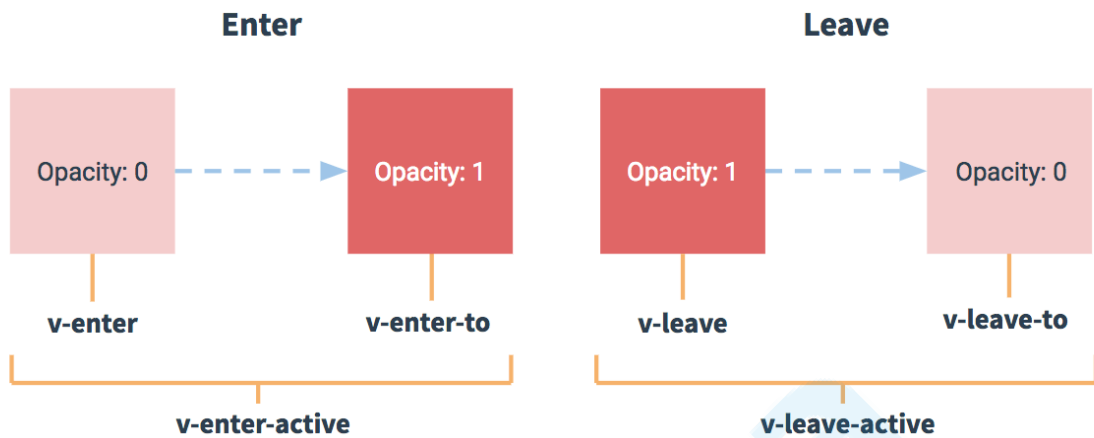
```
<style>
  /* 定义过度动画 */
  .fade-enter-active,
  .fade-leave-active {
    transition: opacity .5s;
  }

  .fade-enter,
  .fade-leave-to {
    opacity: 0;
  }
</style>

<script>
  vue.component('message', {
    // 使用transition组件应用过度动画
    template: `
      <transition name="fade">
        ...
      </transition>
    `,
  })
</script>
```

```
</script>
```

过度类名



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。

```
.fade-enter { opacity: 0; }
```

2. **v-enter-active**: 定义**进入过渡生效时的状态**。在元素被插入之前生效，在过渡/动画完成之后移除。

```
.fade-enter-active { transition: opacity .5s; }
```

3. **v-enter-to**: 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。

```
.fade-enter-to { opacity: 1; }
```

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。

```
.fade-leave { opacity: 1; }
```

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。

```
.fade-leave-active { transition: opacity .5s; }
```

6. **v-leave-to**: 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。

```
.fade-leave-to { opacity: 0; }
```

使用CSS动画库

通过[自定义过度类名](#)可以有效结合[Animate.css](#)这类动画库制作更精美的动画效果。

引入animate.css

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

transition设置

```
<transition enter-active-class="animated bounceIn"
  leave-active-class="animated bounceOut">
```

JavaScript 钩子

可以在<transition>属性中声明 JavaScript 钩子，使用JS实现动画。

```
<transition
  v-on:before-enter="beforeEnter" // 动画开始前，设置初始状态
  v-on:enter="enter" // 执行动画
  v-on:after-enter="afterEnter" // 动画结束，清理工作
  v-on:enter-cancelled="enterCancelled" // 取消动画
  v-on:before-leave="beforeLeave"
  v-on:leave="leave"
  v-on:after-leave="afterLeave"
  v-on:leave-cancelled="leaveCancelled"
></transition>
```

范例：用JS实现消息动画

opacity修改不用css做

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

加上js钩子

```
Vue.component('message', {
  template: `
    <transition ... @before-enter="beforeEnter" @enter="enter">
      ...
    </transition>
  `,
  props: {
    message: String
  }
})
```

```

methods: {
  beforeEnter(e1) {
    e1.style.opacity = 0 // 设置初始状态
  },
  enter(e1, done) {
    document.body.offsetHeight; // 触发回流激活动画
    e1.style.opacity = 1 // 设置结束状态
  }
},
})

```

纯js方案:

```

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

```

```

Vue.component('message', {
  template: `
    <transition name="fade"
      :css="false" // 禁用css
      @before-enter="beforeEnter"
      @enter="enter"
      @before-leave="beforeLeave"
      @leave="leave">
    </transition>
  `,
  methods: {
    beforeEnter(e1) {
      e1.style.opacity = 0
    },
    enter(e1, done) {
      velocity(e1, { opacity: 1 }, { duration: 500, complete: done })
    },
    beforeLeave(e1) {
      e1.style.opacity = 1
    },
    leave(e1, done) {
      velocity(e1, { opacity: 0 }, { duration: 500, complete: done })
    }
  },
})

```

[JavaScript-钩子](#)

列表过度

利用transition-group可以对v-for渲染的每个元素应用过度

范例: 给列表添加增加过度

```
<transition-group name="fade">
  <div v-for="c in courses" :key="c.name">
    {{ c.name }} - ¥{{c.price}}
    <button @click="addToCart(c)">加购</button>
  </div>
</transition-group>
```

[列表过度](#)

Kaikeba
开课吧