

1: 初步使用动画

```
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
<script src="../lib/vue-2.4.0.js"></script>
<!-- 2. 自定义两组样式, 来控制 transition 内部的元素实现动画 -->
<style>
  /* v-enter 【这是一个时间点】 是进入之前, 元素的起始状态, 此时还没有开始进入 */
  /* v-leave-to 【这是一个时间点】 是动画离开之后, 离开的终止状态, 此时, 元素 动画已经结束了 */
  .v-enter,
  .v-leave-to {
    opacity: 0;
    transform: translateX(150px);
  }

  /* v-enter-active 【入场动画的时间段】 */
  /* v-leave-active 【离场动画的时间段】 */
  .v-enter-active,
  .v-leave-active {
```

```
</head>
<style>
  .v-enter, .v-leave-to {
    opacity: 0;
    transform: translateX(30px);
  }
  .v-enter-active,
  .v-leave-active {
    transition: all 5s ease-in-out;
  }
</style>

<body>
  <div id="app">
    <input type="button" value="切换" @click = "flag=!flag">
    <transition> <h3 v-if="flag">出现/不出现</h3> </transition>
  </div>
  <script>
    var vm = new Vue({
      el: "#app",
      data: {
        flag: true
      },
      methods: {}
    });
```

2: 修改v-前缀

```

54 <hr>
55
56 <input type="button" value="toggle2" @click="flag2=!flag2">
57 <transition name="my">
58   <h6 v-if="flag2">这是一个H6</h6>
59 </transition>
60 </div>
61
62 <script>
63   // 创建 Vue 实例，得到 ViewModel

```

3: 动画的钩子函数

JavaScript 钩子

可以在属性中声明 JavaScript 钩子

```

<transition
  v-on:before-enter="beforeEnter"
  v-on:enter="enter"
  v-on:after-enter="afterEnter"
  v-on:enter-cancelled="enterCancelled"
  v-on:before-leave="beforeLeave"
  v-on:leave="leave"
  v-on:after-leave="afterLeave"
  v-on:leave-cancelled="leaveCancelled"
>
</transition>

```

4: 半场动画

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <script src="../lib/vue-2.4.0.js"></script>
  <link rel="stylesheet" href="../lib/bootstrap-3.3.7.css">
</head>
<style>
  #ball {

```

```
        height: 15px;
        width: 15px;
        border-radius: 50%;
        background: red;
    }
</style>
```

```
<body>
  <div id="app">

    <input type="button" value="移动小球" @click="flag=!flag">
      <transition @before-enter="beforEnter" @enter="enter"
@after-enter="afterEnter">
        <div id="ball" v-show="flag"></div>
      </transition>

  </div>
</script>
```

```
    var vm = new Vue({
      el: "#app",
      data: {
        flag: false
      },
      methods: {
        // 注意： 动画钩子函数的第一个参数：el，表示 要执行动画的那
        个DOM元素，是个原生的 JS DOM对象
        // 大家可以认为， el 是通过
        document.getElementById('') 方式获取到的原生JS DOM对象
```

```
        beforEnter(el) {
          // beforeEnter 表示动画入场之前，此时，动画尚未开始，
          可以在 beforeEnter 中，设置元素开始动画之前的起始样式
          // 设置小球开始动画之前的，起始位置
          el.style.transform = "translate(0, 0)"
```

```
        },
        enter(el, done) {
          // 这句话，没有实际的作用，但是，如果不写，出不来动画效
          果；
          // 可以认为 el.offsetWidth 会强制动画刷新
          el.offsetWidth
          // enter 表示动画 开始之后的样式，这里，可以设置小球完
          成动画之后的，结束状态
          el.style.transform = "translate(150px, 450px)"
          el.style.transition = 'all 1s ease'
```

```

// 这里的 done, 其实就是 afterEnter 这个函数, 也就
是说: done 是 afterEnter 函数的引用
done();

},
afterEnter(el) {
  console.log("ok");
  // 动画完成之后, 会调用 afterEnter
  // console.log('ok')
  this.flag = !this.flag
}
}
});

</script>

</body>

</html>

```

6: transition-group标签中的appear tag 属性

```

</div>

<!-- <ul> -->
<!-- 在实现列表过渡的时候, 如果需要过渡的元素, 是通过 v-for 循环渲染出来的, 不能使用
transition 包裹, 需要使用 transitionGroup -->
<!-- 如果要为 v-for 循环创建的元素设置动画, 必须为每一个 元素 设置 :key 属性 -->
<!-- 给 transition-group 添加 appear 属性, 实现页面刚展示出来时候, 入场时候的效果 -->
<!-- 通过 为 transition-group 元素, 设置 tag 属性, 指定 transition-group 渲染为指定的元
素, 如果不指定 tag 属性, 默认, 渲染为 span 标签 -->
<transition-group appear tag="ul">
  <li v-for="(item, i) in list" :key="item.id" @click="del(i)">
    {{item.id}} --- {{item.name}}
  </li>
</transition-group>
<!-- </ul> -->

</div>

```