

Vue过渡动画的使用

在Vue的内部是可以直接使用过渡的效果，它内部封装了一个`<transition>`的组件，可以让元素在 进入/离开 的时候执行特定的过渡效果

```
<style>
  .box {
    width: 400px;
    height: 400px;
    border: 1px solid black;
    position: relative;
  }

  .login,
  .index {
    width: 100%;
    height: 100%;
    position: absolute;
    left: 0px;
    top: 0px;
    font-size: 32px;
    display: flex;
    justify-content: center;
    align-items: center;
  }

  .login {
    background-color: deeppink;
  }

  .index {
    background-color: lightseagreen;
  }

  /* 进入之前 */
  .fade-enter {
    opacity: 0;
  }

  /* 进入之后 */
  .fade-enter-to {
    opacity: 1;
  }

  /* 离开之前 */
  .fade-leave {
    transform: scale(1);
  }

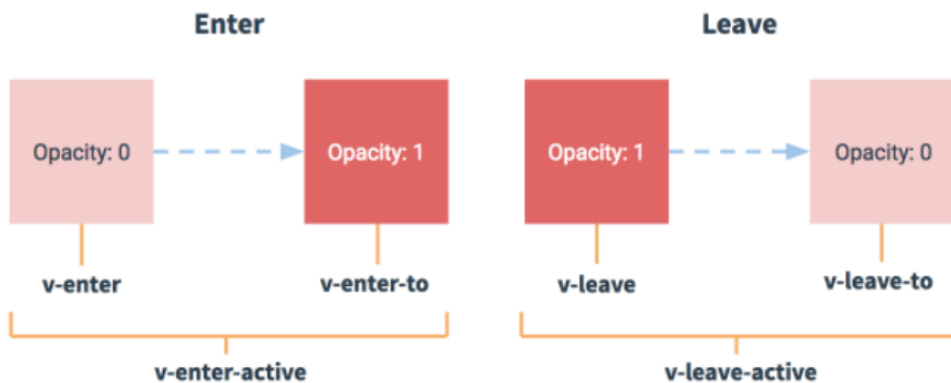
  /* 离开之后 */
  .fade-leave-to {
    transform: scale(0);
  }
}
```

```

/* 进入的变化过程 */
.fade-enter-active {
  transition: all 0.5s linear;
}

.fade-leave-active {
  transition: all 5s linear;
}
</style>
<body>
  <div id="app">
    <div class="box">
      <transition name="fade">
        <div class="login" v-show="flag">这是一个登陆页面</div>
      </transition>
    </div>
    <button type="button" @click="flag = !flag">切换</button>
  </div>
</body>
<script src="./js/vue.min.js"></script>
<script src="./js/vue-router.js"></script>
<script>
  new Vue({
    el: "#app",
    data: {
      flag: true
    }
  })
</script>

```



上面就是vue中 `<transition>` 的过渡图，然后我们根据图片上面的提示得到了上面的代码

1. `v-enter` 代表进入之前
2. `v-enter-to` 代表进入以后
3. `v-leave` 代表离开以前
4. `v-leave-to` 代表离开以后

而上面的4个状态会形成2个过程，分别是进入的过程与离开的过程

1. `v-enter-active` 进入的过程
2. `v-leave-active` 离开的过程

前面的 `v` 指的是在 `<transition>` 组件上面的 `name` 属性值

在这里还有一个小小的注意事项，如果有多个组件同时执行过的时候，则要使用 `<transition-group>` 去实现

```
<transition-group name="fade" tag="div">
  <div class="login" key="1" v-if="flag">这是一个登陆页面</div>
  <div class="index" key="2" v-if="!flag">这是一个登陆以后的页面</div>
</transition-group>
```

`<transition-group>` 默认情况下会帮我们生成一个 `span` 标签，但是我们仍然可以通过设置 `tag` 来生成指定的标签

案例

我们现在使用 `transition` 来完成一组左右切换的过程，这个过程后期会有项目当中的页面切换里面去使用

```
<!DOCTYPE html>
<html lang="zh">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>vue的transition使用</title>
  <style>
    * {
      margin: 0;
      padding: 0;
    }

    #app {
      width: 100vw;
      height: 100vh;
    }
    .box{
      width: 100%;
      height: 50%;
      position: relative;
      overflow: hidden;
    }
    .login,
    .index {
      width: 100%;
      height: 100%;
      display: flex;
      justify-content: center;
      align-items: center;
      position: absolute;
    }

    .login {
      background-color: deeppink;
    }
  </style>
</head>

<body>
  <div id="app">
    <div class="box">
      <div class="login">
        <h1>登陆页面</h1>
      </div>
      <div class="index">
        <h1>登陆后的页面</h1>
      </div>
    </div>
  </div>
</body>
</html>
```

```

.index {
    background-color: lightseagreen;
}
/* 进入之前 */
.slide-left-enter{
    transform: translateX(100%);
}
/* 进入之后 */
.slide-left-enter-to{
    transform: translateX(0);
}
/* 离开之前 */
.slide-left-leave{
    transform: translateX(0);
}
/* 离开之后 */
.slide-left-leave-to{
    transform: translateX(-100%);
}
/* 进入的过程 */
.slide-left-enter-active{
    transition: all 0.2s linear;
}
/* 离开的过程 */
.slide-left-leave-active{
    transition: all 0.2s linear;
}

/* -----反着走一次----- */
/* 进入之前 */
.slide-right-enter{
    transform: translateX(-100%);
}
/* 进入之后 */
.slide-right-enter-to{
    transform: translateX(0);
}
/* 离开之前 */
.slide-right-leave{
    transform: translateX(0);
}
/* 离开之后 */
.slide-right-leave-to{
    transform: translateX(100%);
}
/*向右进入的过程， */
.slide-right-enter-active{
    transition: all 0.2s linear;
}
/*向右离开的过程 */
.slide-right-leave-active{
    transition: all 0.2s linear;
}
}

</style>
</head>

<body>
    <div id="app">

```

```

<button type="button" @click="toLeft">&lt;----向左切换</button>
<button type="button" @click="toRight">向右切换-----&gt;</button>
<transition-group tag="div" class="box" :name="slideDir">
  <login v-if="flag" :key="1"></login>
  <index v-else :key="2"></index>
</transition-group>
</div>
<template id="temp1">
  <div class="login">
    <h2>这是一个登陆的页面</h2>
  </div>
</template>
<template id="temp2">
  <div class="index">
    <h2>这是一个主页</h2>
  </div>
</template>
</body>
<script src="./js/vue.js"></script>
<script>
  let login = {
    template: "#temp1"
  };
  let index = {
    template: "#temp2"
  }
  new Vue({
    el: "#app",
    data:{
      flag:true,
      slideDir:"slide-right"
    },
    methods:{
      toLeft(){
        this.slideDir = "slide-left";
        this.flag = !this.flag;
      },
      toRight(){
        this.slideDir = "slide-right"
        this.flag = !this.flag;
      }
    },
    components: {
      login,
      index
    }
  })
</script>
</html>

```

transition与路由的配合

在我们开发项目的时候，如果页面通过路由跳转也是可以通过 transition 来实现动画效果 的

```

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

```

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>路由实现过渡效果</title>
  <style>
    * {
      margin: 0;
      padding: 0;
      list-style-type: none;
    }

    #app {
      width: 100vw;
      height: 100vh;
      display: flex;
      flex-direction: column;
    }

    .content-box{
      flex: 1;
      overflow: auto;
    }

    .tab-bar {
      display: flex;
      flex-direction: row;
      justify-content: space-around;
      height: 55px;
      border-top: 2px solid red;
    }

    .tab-bar li {
      width: 55px;
      height: 55px;
      display: flex;
      justify-content: center;
      align-items: center;
      border: 1px solid red;
    }

    .page-view{
      width: 100%;
      height: 100%;
      overflow: auto;
    }

    .page-view img{
      width: 100%;
    }

    /* ----- */
    /* 进入之前 */
    .slide-left-enter{
      transform: translateX(100%);
    }

    /* 进入以后，离开以前 */
    .slide-left-enter-to,.slide-left-leave{
      transform: translateX(0);
    }

    /* 离开以后 */
    .slide-left-leave-to{
```

```

        transform: translateX(-100%);
    }
    /* 变化的过程 */
    .slide-left-enter-active, .slide-left-leave-active{
        transition: all 1s linear;
        /* 我要让它不占位置 */
        position: fixed;

    }
    /* ----- */
    /* 进入之前 */
    .slide-right-enter{
        transform: translateX(-100%);
    }
    .slide-right-enter-to, .slide-right-leave{
        transform: translateX(0);
    }
    .slide-right-leave-to{
        transform: translateX(100%);
    }
    .slide-right-enter-active, .slide-right-leave-to{
        transition: all 1s linear;
        /* 我要让它不占位置 */
        position: fixed;
    }
</style>
</head>

<body>
    <div id="app">
        <div class="content-box">
            <transition :name="slideDir">
                <router-view></router-view>
            </transition>

        </div>
        <ul class="tab-bar">
            <li @click="$router.push({name: 'one'})">第一个</li>
            <li @click="$router.push({name: 'two'})">第二个</li>
            <li @click="$router.push({name: 'three'})">第三个</li>
            <li @click="$router.push({name: 'four'})">第四个</li>
        </ul>
    </div>
    <template id="temp1">
        <div class="page-view">
            
            <h2>第一个页面</h2>
        </div>
    </template>
    <template id="temp2">
        <div class="page-view">
            
            <h2>第二个页面</h2>
        </div>
    </template>
    <template id="temp3">
        <div class="page-view">
            

```

```

        <h2>第三个页面</h2>

    </div>
</template>
<template id="temp4">
    <div class="page-view">
        
        <h2>第四个页面</h2>
    </div>
</template>
</body>
<script src="./js/vue.js"></script>
<script src="./js/vue-router.js"></script>
<script>
    let one = {
        template:"#temp1"
    };
    let two = {
        template:"#temp2"
    };
    let three = {
        template:"#temp3"
    };
    let four = {
        template:"#temp4"
    };

    // 新建了一个路由管理对象
    const router = new VueRouter({
        mode:"hash",
        routes:[
            {
                path:"/one",
                name:"one",
                component:one,
                meta:{
                    index:1
                }
            },
            {
                path:"/two",
                name:"two",
                component:two,
                meta:{
                    index:2
                }
            },
            {
                path:"/three",
                name:"three",
                component:three,
                meta:{
                    index:3
                }
            },
            {
                path:"/four",
                name:"four",

```



```
        component:four,
        meta:{
            index:4
        }
    },
]
})
new Vue({
  el: "#app",
  router,
  data:{
    slideDir:"slide-left"
  },
  watch:{
    $route(to,from){
      if(from.meta.index<to.meta.index){
        this.slideDir = "slide-left";
      }
      else{
        this.slideDir = "slide-right";
      }
    }
  }
})
</script>

</html>
```



第一个页面

第一个		第二个		第三个		第四个	
-----	--	-----	--	-----	--	-----	--