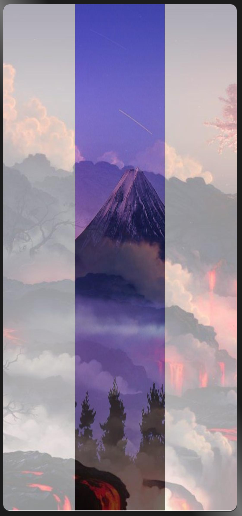
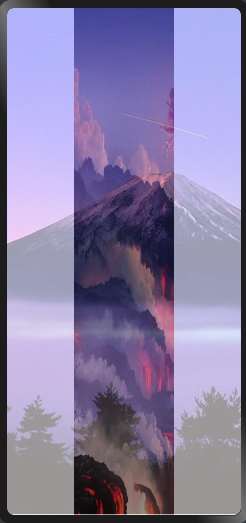
# HarmonyOS应用开发-ets页面间转场



|  |
| --- |
| 说明：该组件从API version 7开始支持。后续版本如有新增内容，则采用上角标单独标记该内容的起始版本。 |

## 一、页面转场通过在全局pageTransition方法内配置页面入场组件和页面退场组件来自定义页面转场动效。

|  |  |  |
| --- | --- | --- |
| 组件名称 | 参数 | 参数描述 |
| PageTransitionEnter | Object | 页面入场组件，用于自定义当前页面的入场效果 |
| PageTransitionExit | Object | 页面退场组件，用于自定义当前页面的退场效果 |

PageTransitionEnter和PageTransitionExit组件支持的属性：

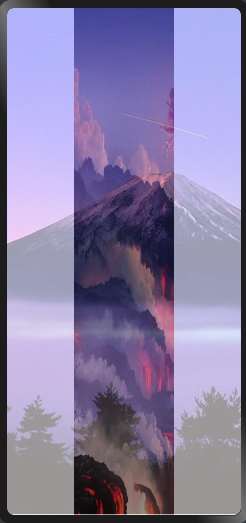
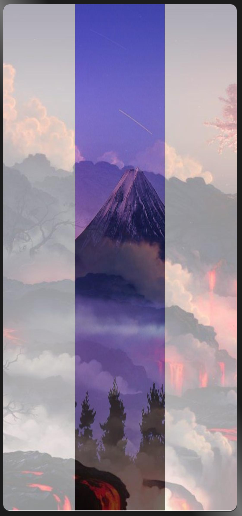
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数类型 | 默认值 | 必填 | 参数描述 |
| slide | SlideEffect | Right | 否 | 设置转场的滑入效果 |
| translate | {  x? : number,  y? : number,  z? : number  } | - | 否 | 设置页面转场时的平移效果，为入场时起点和退场时终点的值，和slide同时设置时默认生效slide。 |
| scale | {  x? : number,  y? : number,  z? : number,  centerX? : number,  centerY? : number  } | - | 否 | 设置页面转场时的缩放效果，为入场时起点和退场时终点的值。 |
| opacity | number | 1 | 否 | 设置入场的起点透明度值或者退场的终点透明度值。 |

PageTransitionEnter和PageTransitionExit组件支持的事件：

|  |  |
| --- | --- |
| 事件 | 功能描述 |
| onEnter(type: RouteType, progress: number) => void | 回调入参为当前入场动画的归一化进度[0 - 1]。 |
| onExit(type: RouteType, progress: number) => void | 回调入参为当前退场动画的归一化进度[0 - 1]。 |

## 二、示例

效果展示：

示例代码：

index.ets

|  |
| --- |
| @Entry @Component struct PageTransitionExample1 {  @State scale: number = 1  @State opacity: number = 1  @State active: boolean = false  build() {  Column() {  Navigator({ target: 'pages/page1', type: NavigationType.Push }) {  Image($rawfile("fss.jpg")).width("100%").height("100%")  }  .onClick(() => {  this.active = true  })  }.scale({ x: this.scale }).opacity(this.opacity)  } // 自定义方式1：完全自定义转场过程的效果  pageTransition() {  PageTransitionEnter({ duration: 1200, curve: Curve.Linear })  .onEnter((type: RouteType, progress: number) => {  this.scale = 1  this.opacity = progress  }) // 进场过程中会逐帧触发onEnter回调，入参为动效的归一化进度(0% -- 100%)  PageTransitionExit({ duration: 1500, curve: Curve.Ease })  .onExit((type: RouteType, progress: number) => {  this.scale = 1 - progress  this.opacity = 1  }) // 退场过程中会逐帧触发onExit回调，入参为动效的归一化进度(0% -- 100%)  } } |

page1.ets

|  |
| --- |
| // page1.ets @Entry @Component struct AExample {  @State scale: number = 1  @State opacity: number = 1  @State active: boolean = false  build() {  Column() {  Navigator({ target: 'pages/index' ,type: NavigationType.Push}) {  Image($rawfile("gz.jpg")).width("100%").height("100%")  }  }.height("100%").width("100%").scale({ x: this.scale }).opacity(this.opacity)  } // 自定义方式1：完全自定义转场过程的效果  pageTransition() {  PageTransitionEnter({ duration: 1200, curve: Curve.Linear })  .onEnter((type: RouteType, progress: number) => {  this.scale = 1  this.opacity = progress  }) // 进场过程中会逐帧触发onEnter回调，入参为动效的归一化进度(0% -- 100%)  PageTransitionExit({ duration: 1500, curve: Curve.Ease })  .onExit((type: RouteType, progress: number) => {  this.scale = 1 - progress  this.opacity = 1  }) // 退场过程中会逐帧触发onExit回调，入参为动效的归一化进度(0% -- 100%)  } } |

完整代码地址：

<https://gitee.com/jltfcloudcn/jump_to/tree/master/PageTransitionAnimation>