# 9.4 Hero动画

Hero指的是可以在路由(页面)之间“飞行”的widget，简单来说Hero动画就是在路由切换时，有一个共享的widget可以在新旧路由间切换。由于共享的widget在新旧路由页面上的位置、外观可能有所差异，所以在路由切换时会从旧路逐渐过渡到新路由中的指定位置，这样就会产生一个Hero动画。

你可能多次看到过 hero 动画。例如，一个路由中显示待售商品的缩略图列表，选择一个条目会将其跳转到一个新路由，新路由中包含该商品的详细信息和“购买”按钮。 在Flutter中将图片从一个路由“飞”到另一个路由称为**hero动画**，尽管相同的动作有时也称为 **共享元素转换**。下面我们通过一个示例来体验一下hero 动画。

为什么要将这种可飞行的共享组件称为hero（英雄），有一种说法是说美国文化中的超人是可以飞的，那是美国人心中的大英雄，还有漫威中的超级英雄基本上都是会飞的，所以Flutter开发人员就对这种“会飞的widget”就起了一个富有浪漫主义的名字hero。当然这种说法并非官方解释，但却很有意思。

#### 示例

假设有两个路由A和B，他们的内容交互如下：

A：包含一个用户头像，圆形，点击后跳到B路由，可以查看大图。

B：显示用户头像原图，矩形；

在AB两个路由之间跳转的时候，用户头像会逐渐过渡到目标路由页的头像上，接下来我们先看看代码，然后再解析：

// 路由A  
class HeroAnimationRoute extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return Container(  
 alignment: Alignment.topCenter,  
 child: InkWell(  
 child: Hero(  
 tag: "avatar", //唯一标记，前后两个路由页Hero的tag必须相同  
 child: ClipOval(  
 child: Image.asset("images/avatar.png",  
 width: 50.0,  
 ),  
 ),  
 ),  
 onTap: () {  
 //打开B路由   
 Navigator.push(context, PageRouteBuilder(  
 pageBuilder: (BuildContext context, Animation animation,  
 Animation secondaryAnimation) {  
 return new FadeTransition(  
 opacity: animation,  
 child: Scaffold(  
 appBar: AppBar(  
 title: Text("原图"),  
 ),  
 body: HeroAnimationRouteB(),  
 ),  
 );  
 })  
 );  
 },  
 ),  
 );  
 }  
}

路由B:

class HeroAnimationRouteB extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return Center(  
 child: Hero(  
 tag: "avatar", //唯一标记，前后两个路由页Hero的tag必须相同  
 child: Image.asset("images/avatar.png"),  
 ),  
 );  
 }  
}

我们可以看到，实现Hero动画只需要用Hero组件将要共享的widget包装起来，并提供一个相同的tag即可，中间的过渡帧都是Flutter Framework自动完成的。必须要注意， 前后路由页的共享Hero的tag必须是相同的，Flutter Framework内部正是通过tag来确定新旧路由页widget的对应关系的。

Hero动画的原理比较简单，Flutter Framework知道新旧路由页中共享元素的位置和大小，所以根据这两个端点，在动画执行过程中求出过渡时的插值（中间态）即可，而感到幸运的是，这些事情不需要我们自己动手，Flutter已经帮我们做了！