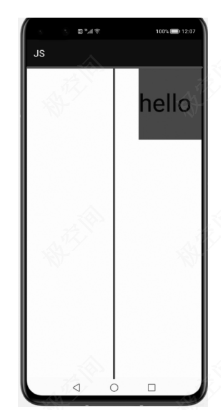
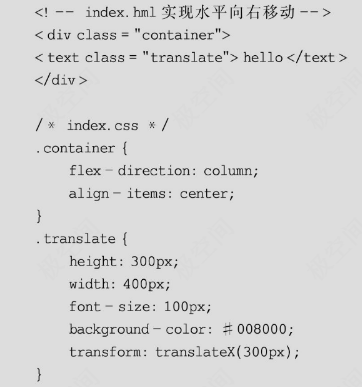
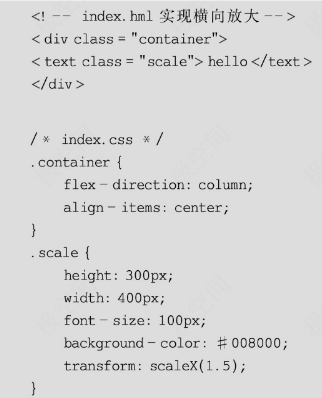
# transform静态动画

* 静态动画的核心是transform样式，其中包含3种变换类型，且每一次的样式设置只能支持一种类型的变化
* 静态动画只有开始状态和结束状态，而不能设置中间状态
* 1、**translate变换类型**
  + translate变换类型可以将组件沿水平或垂直方向移动一定的距离
  + 创建JS项目工程，修改index.hml和index.css文件，示例是一个水平向右移动的示例代码，代码如下（左侧图）
  + 上述代码实现了一个text组件水平向右移动的效果 ，translateX（300px）将text组件水平从基准线向右移动300px，其中右为正值，左为负值
  + 同理，若修改为translateY（），则是以Y轴基准线为标准，下为正值，上为负值
  + 运行结果如图（右侧图），灰线为基准线

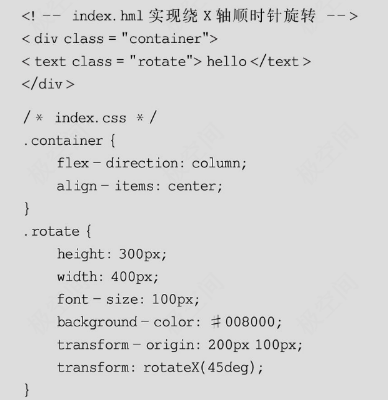
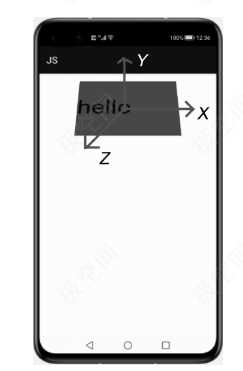




* 2、**scale变换类型**
  + scale样式可将组件沿横向或纵向，缩小或放大一定比例
  + 下面对text组件进行横向放大，代码如下（左侧图）
  + scaleX（1.5）表示将text文本横向放大1.5倍
  + 运行上述代码可以直观看到效果（右侧图）



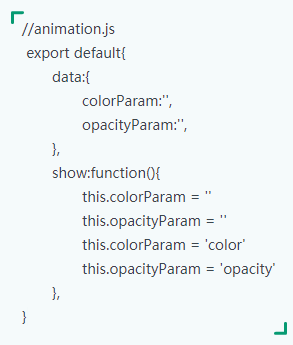
* 3、**rotate变换类型**
  + rotate样式可以将组件沿横轴或纵轴或中心点，旋转一定的角度
  + 如下示例将text文本绕X轴顺时针旋转45°，代码如下（左侧图）
  + 运行结果如图（右侧图）



# animation连续动画

* 前面讲过的静态动画只有开始状态和结束状态，而没有中间状态，因此静态动画看起来更像是图片之间的切换而不是真正意义上的动画，如果需要设置中间的过渡状态和转换效果，则只能由连续动画实现
* 为了使动画变得连贯，HarmonyOS中的JS引入了连续动画
* 连续动画最主要的功能贡献者就是animation样式，通过它可以定义动画的开始、结束状态及期间变化的速度
* 在引入连续动画之后，可以定义组件的宽、高、颜色和透明度等的变化速度和程度，利用这一点可以实现一些有趣的功能，例如进度条、渐变色块等
* 在本节中利用animation实现颜色、透明度和宽度变化的功能
* 首先新建JS工程，修改代码如下（图1）
* 在页面中定义了两个文本组件，分别用来演示颜色变化，以及透明度和宽度变化
* input组件的单击事件show（）在文件中的实现代码如下（图2）
* 其中，colorParam和opacity Param分别与hml文件中的样式进行数据绑定，在show（）方法中，首先将这两个参数设置为默认格式，并在css文件中进行动画实现，css文件代码如下（图3）
* 在.color和.opacity样式中，使用animation-name属性定义了各自的动画样式（@keyframes）
* 其中@keyframes可以自定义动画样式，例如代码中定义的Color-frames和Opacity-frames, from代表动画的开始状态，to代表结束状态，期间的过渡动画由系统自动计算完成，当然也可以使用animation-timing-function属性来描述动画执行的速度曲线，使动画更加平滑
* 运行上述代码，初始页面如图所示（图4）
* 单击“开始”按钮后，会播放定义的动画（按时间顺序排列），如图所示（图5）
* 可以看到“颜色”块的背景色从蓝色最终变为绿色，“透明度”块的透明度从不透明变为全透明，并且宽度从最开始一直变为0，类似一个反向的进度条







* 通过 animation 样式可以实现的效果有：
  + animation-name：设置动画执行后应用到组件上的背景颜色、透明度、宽高和变换类型
  + animation-delay 和 animation-duration：分别设置动画执行后元素延迟和持续的时间
  + animation-timing-function：描述动画执行的速度曲线，使动画更加平滑
  + animation-iteration-count：定义动画播放的次数
  + animation-fill-mode：指定动画执行结束后是否恢复初始状态
* animation 样式需要在 css 文件中先定义 keyframe，在 keyframe 中设置动画的过渡效果，并通过一个样式类型在 hml 文件中调用