

# 《Jetpack Compose 系列课》

## Compose 手势

让人人都能享受到高品质的教育服务

# Compose 手势

## 目 录

1

点击

2

滚动

3

拖动

4

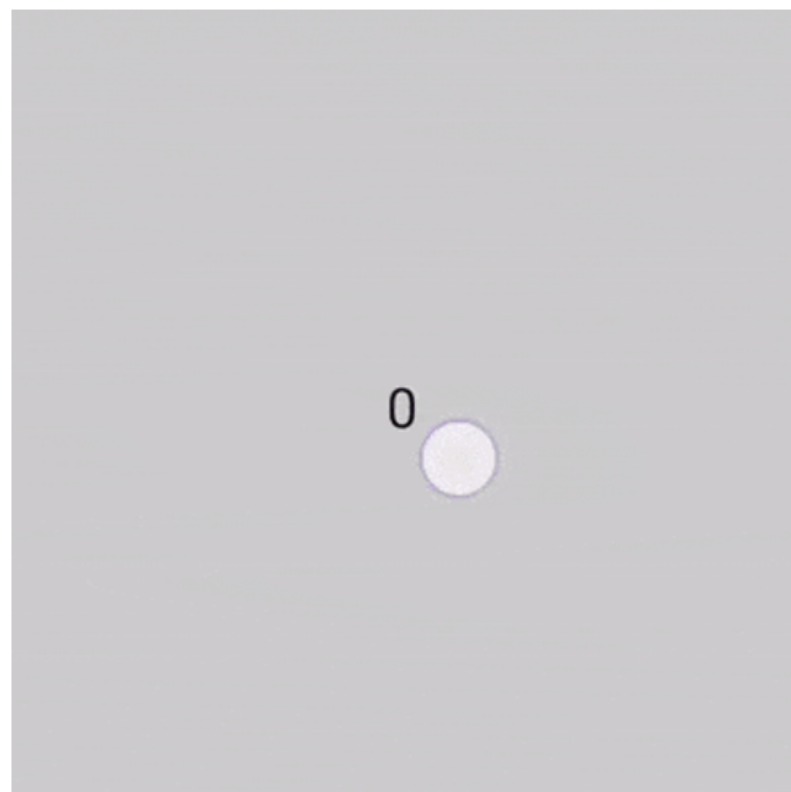
滑动

5

多点触控

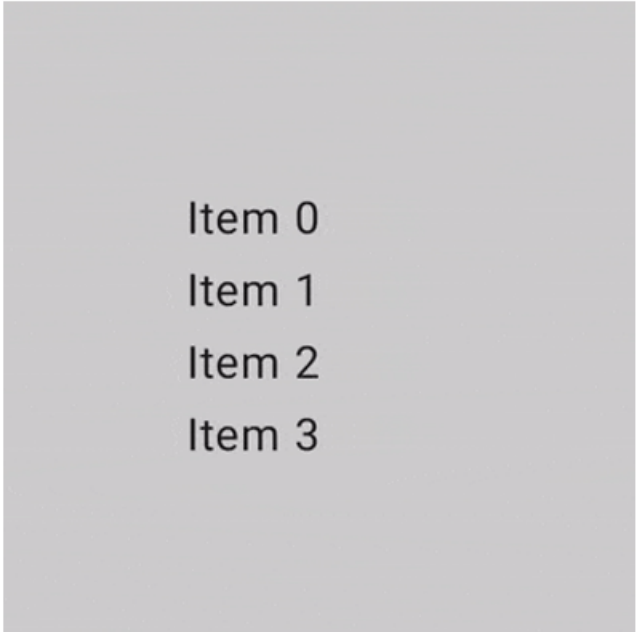
# 点击

- clickable 修饰符允许应用检测对已应用该修饰符的元素的点击。
- 当需要更大灵活性时，您可以通过 pointerInput 修饰符提供点按手势检测器。



# 滚动修饰符

- verticalScroll 和 horizontalScroll 修饰符提供一种最简单的方法，可让用户在元素内容边界大于最大尺寸约束时滚动元素。
- 借助 ScrollState，您可以更改滚动位置或获取当前状态。如需使用默认参数创建此列表，请使用 rememberScrollState()。

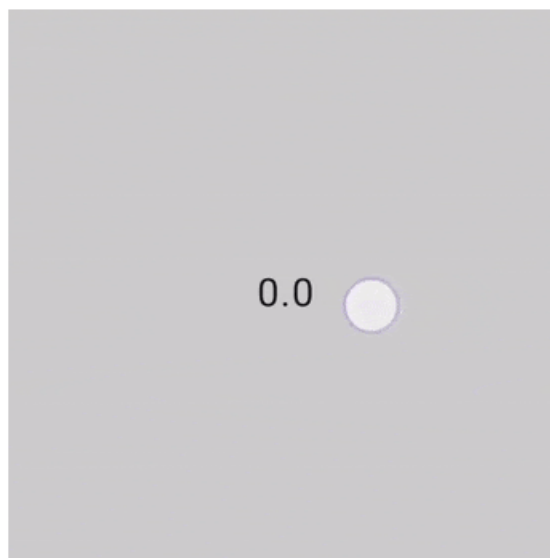


Item 0  
Item 1  
Item 2  
Item 3

# 可滚动的修饰符

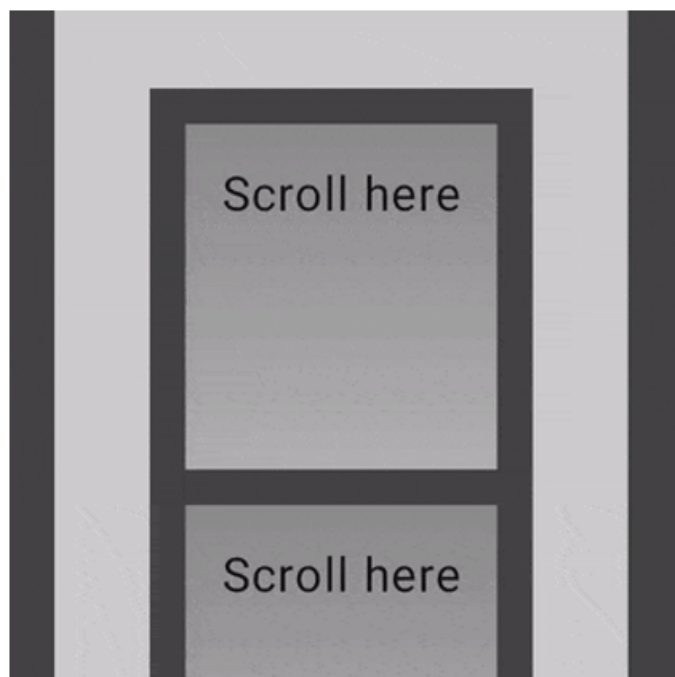
- scrollable 修饰符与滚动修饰符不同，区别在于 scrollable 可检测滚动手势，但不会偏移其内容。

此修饰符需要 ScrollableController 才能正常运行。构造 ScrollableController 时，您必须提供一个 consumeScrollDelta 函数，该函数将在每个滚动步骤（通过手势输入、平滑滚动或投掷）调用，并且增量以像素为单位。为了确保正确传播事件，必须从此函数返回使用的滚动距离量。



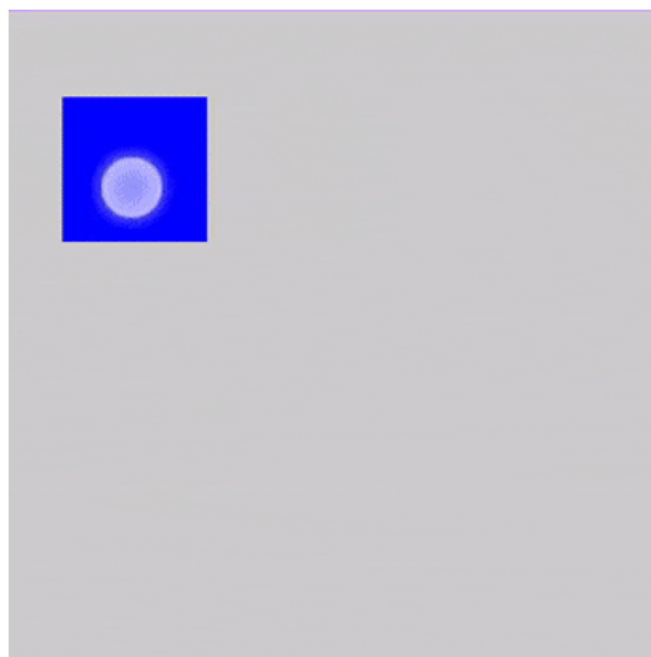
# 自动嵌套滚动

- 简单的嵌套滚动无需您执行任何操作。启动滚动操作的手势会自动从子级传播到父级，这样一来，当子级无法进一步滚动时，手势就会由其父元素处理。



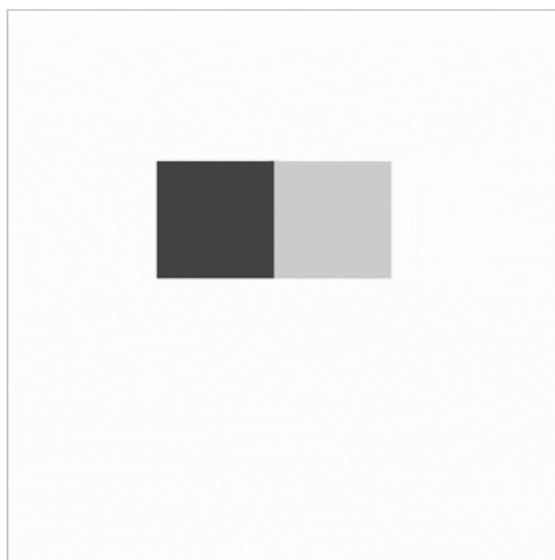
# 拖动

- draggable 修饰符是向单一方向拖动手势的高级入口点，并且会报告拖动距离（以像素为单位）。
- 请务必注意，此修饰符与 scrollable 类似，仅检测手势。您需要保存状态并在屏幕上表示，例如通过 offset 修饰符移动元素：



# 滑动

- 使用 `swipeable` 修饰符，您可以拖动元素，释放后，这些元素通常朝一个方向定义的两个或多个锚点呈现动画效果。其常见用途是实现“滑动关闭”模式。





# 多点触控： 平移、缩放、旋转

- 如需检测用于平移、缩放和旋转的多点触控手势，您可以使用 transformable 修饰符。

此修饰符本身不会转换元素，只会检测手势。

