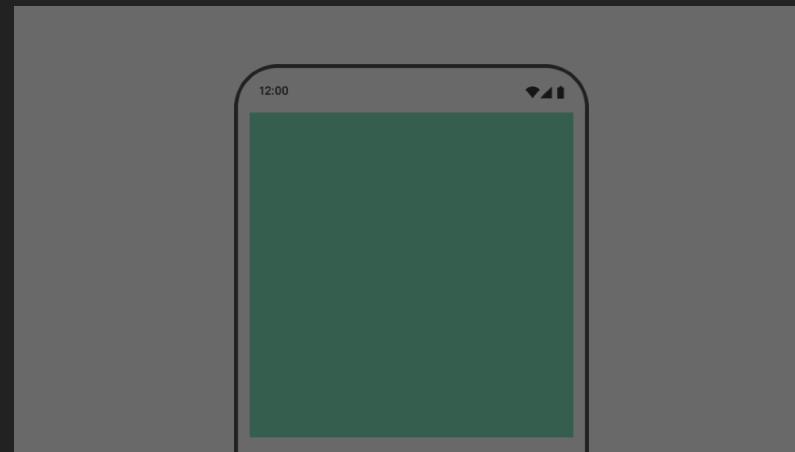


# updateTransition()

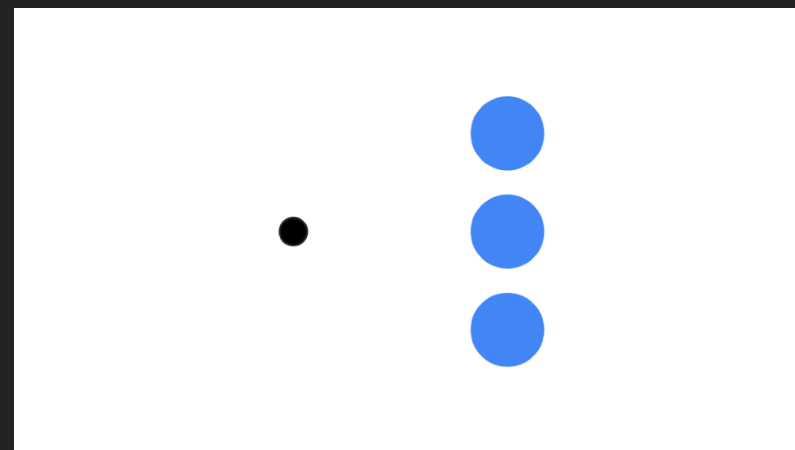
```
transition.an
```

```
val bgCo f animateColor(...) {...} State<Color>
val bgCo v animations List<Transition<MutableState<Boolean>>.Trans...
val bgCo f animateFloat(...) {...} State<Float>
    if f animateDp(...) {...} State<Dp>
    f animateInt(...) {...} State<Int>
    f animateIntOffset(...) {...} State<IntOffset>
val sca f animateIntSize(...) {...} State<IntSize>
    if f animateOffset(...) {...} State<Offset>
    f animateRect(...) {...} State<Rect>
return f animateSize(...) {...} State<Size>
    f animateValue(typeConverter: TwoWayConverter<T,... State<T>
```

# Value change APIs



Single: `animate*AsState`



Multiple: `Transition` or `InfiniteTransition`