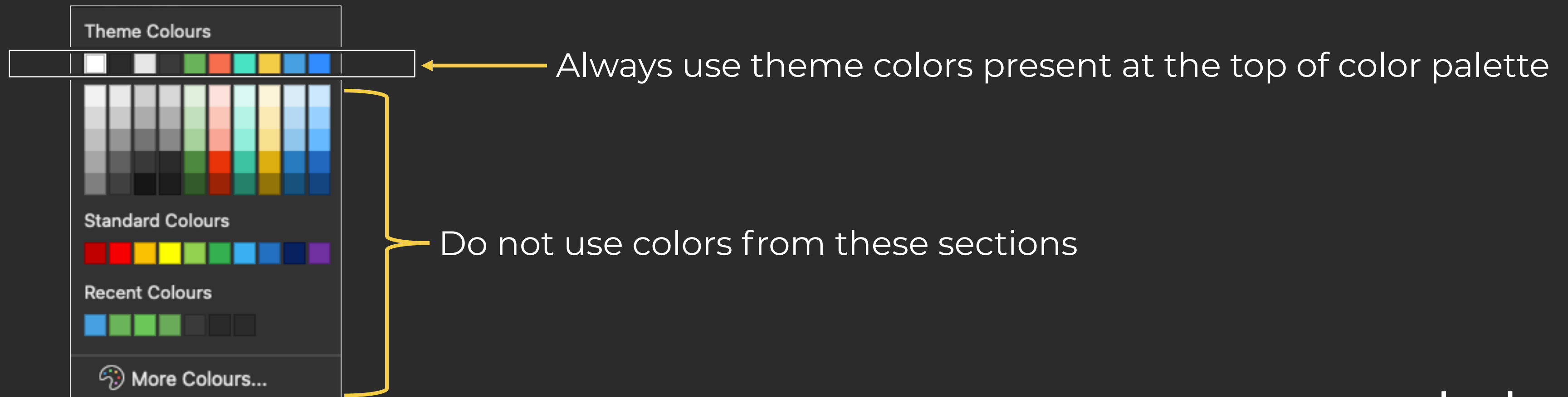


Guidelines to Use Only Brand Colors in Slides

- The **RGB** and **Hexadecimal** values of theme colors are as follows:

Hex #FFFFFF RGB 255, 255, 255	Hex #2B2B2B RGB 43, 43, 43	Hex #E6E6E6 RGB 230, 230, 230	Hex #3A3A3A RGB 58, 58, 58		
Hex #69B359 RGB 105, 179, 89	Hex #F76E4E RGB 247, 110, 78	Hex #48E3C3 RGB 72, 227, 195	Hex #F3CD46 RGB 243, 205, 70	Hex #46A0E0 RGB 70, 160, 224	Hex #008CFF RGB 0, 140, 255

- Use only Theme colors provided in the color palette dropdown





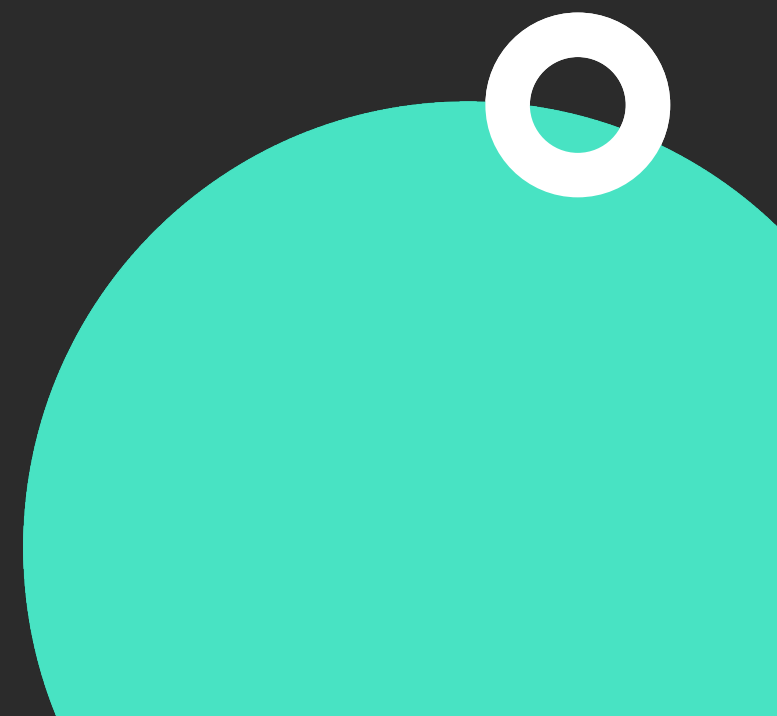
Android Animations and Transitions with Jetpack Compose



Mehdi Haghgoo

 @IAmMehdiHaghgoo

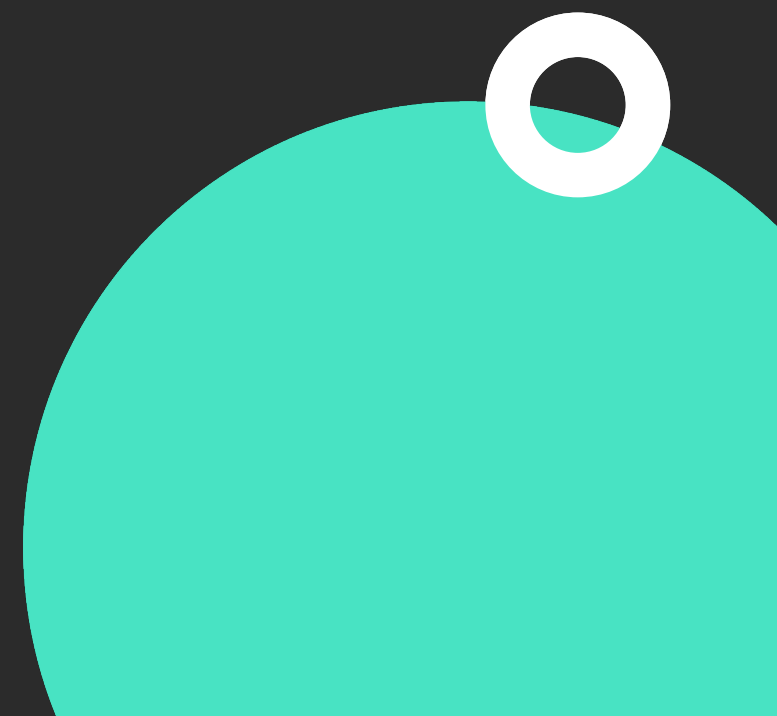
 in/mehdihaghgoo





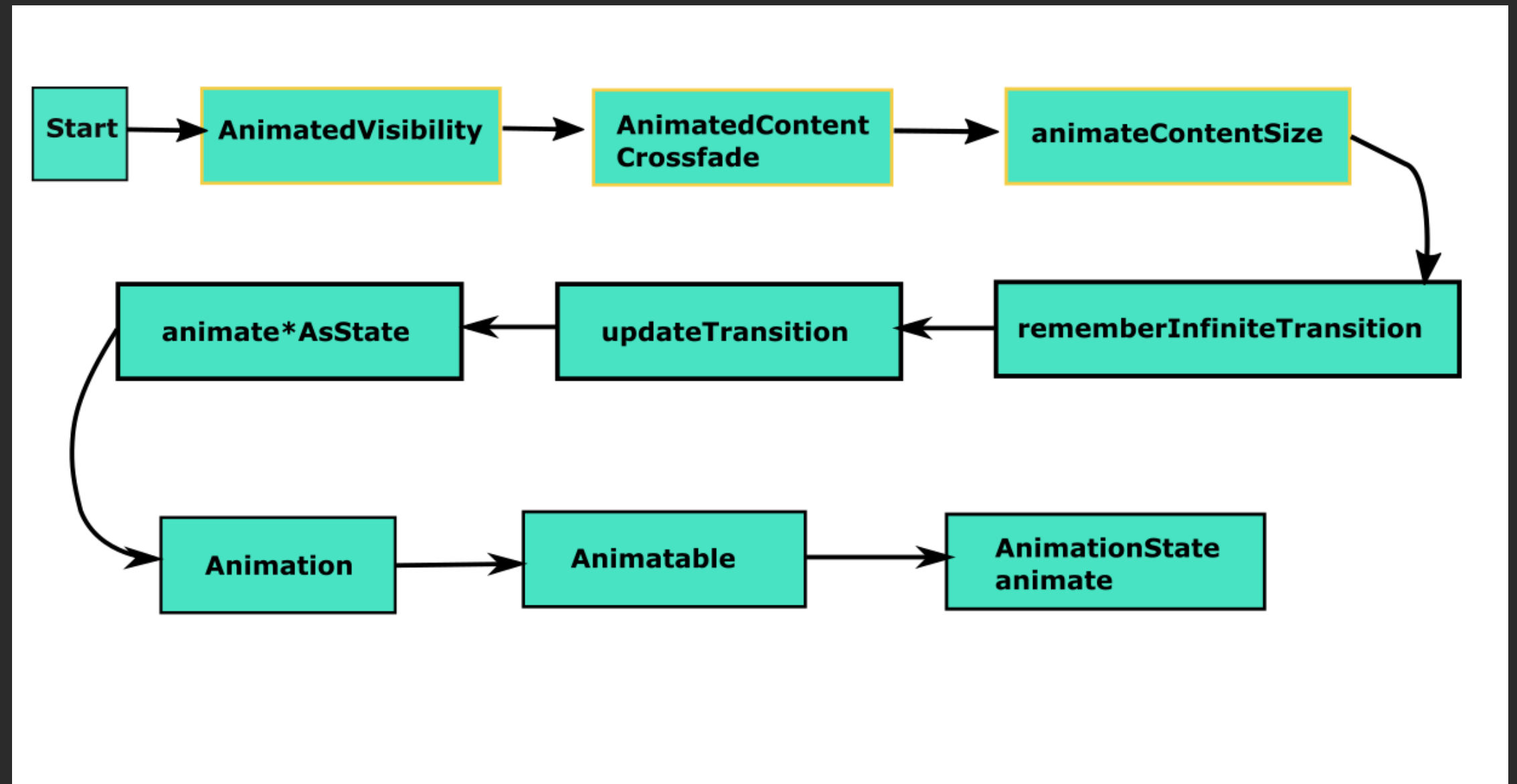
Section 2

Animating Layout Change : High-Level Animations



Section Overview

- Learn animation APIs :
 - AnimatedVisibility
 - AnimatedContent
 - Crossfade
 - animateContentSize
- Code
- Recap



AnimatedVisibility

- Animate appearance or disappearance of content
 - Can wrap any content that is only visible when certain condition is true
 - By default a fade-through animation is performed
 - Custom appearance / disappearance animation can be used

```
AnimatedVisibility(  
    visible = booleanState,  
    enter = ...,  
    exit = ...,  
    label = "My Label"  
) {  
    GreetingBox()  
}
```



AnimatedContent

- Animates its content based on a target state
 - Supports custom ContentTransform via transitionSpec parameter
 - You should use the lambda parameter passed to the AnimatedContent body

```
AnimatedContent(targetState = someState,  
label = "MyLabel", transitionSpec = {  
    fadeIn() with fadeOut() using  
    SizeTransform()  
}) { targetState ->  
    SomeContent(targetState)  
}
```



Crossfade

- Toggles between two layouts with a crossfade animation
- State parameter passed to body lambda can be used to decide which content to show

```
Crossfade (targetState = someState,  
          animationSpec = ...,  
  label = "My Crossfade"  
) { state->  
    when(state) {  
        A -> BoxA( )  
        B -> BoxB( )  
    }  
}
```



animateContentSize

- Modifier that animates a component's size change

```
Box(modifier  
    .animateContentSize() //Be sure to put  
    this before size modifiers  
    .size(  
        if (someCondition) 200.dp else  
        150.dp  
    )  
)
```



Tips

- Many high-level animation APIs take a label parameter to allow easier preview and debugging in Android Studio
- `AnimatedVisibility` and `AnimatedContent` let their children use `animateEnterExit` for custom animation



Code Time

- Folder : s2-animating-layout-changes/starter-project
- Files:
 - AnimatedContentCrossfadeSheet.kt
 - AnimatedContent
 - Crossfade
 - AnimatedVisibilitySheet.kt
 - AnimatedVisibility
 - AnimateContentSizeSheet.kt
 - animateContentSize



Section Summary

- Animate layout change using :
 - AnimatedVisibility
 - AnimatedContent
 - Crossfade
 - animateContentSize
- What's next?
 - High-level animations continued...

