Buse Çarık 27/06/18

INTERRA R&D Material Design

Material Design

Button

**INTRODUCTION**

**PURPOSE OF THIS DOCUMENT**

The aim of this project is analyzing material design button’s styles and attributes.

**OVERVIEW**

The project that is described in this document, includes material button’s types and styles. The types are filled elevated button, filled unelevated button and floating action button and the styles are

Widget.MaterialComponents.Button, Widget.MaterialComponents.Button.UnelevatedButton, Widget.MaterialComponents.Button.TextButton, TextAppearance.MaterialComponents.MaterialButton, Widget.MaterialComponents.Button.TextButton, Widget.MaterialComponents.Button.UnelevatedButton, TextAppearance.MaterialComponents.MaterialButton.Secondary, Widget.MaterialComponents.FloatingActionButton

To use the material components, add implementation 'com.android.support:design:28.0.0-alpha1' to the dependencies in build.gradle file.

Also, use android.support.design.button.MaterialButton in XML file.

There are different styles for material button:

🡪 Filled, elevated button: default (When you add Material.Components.Button, this is called. Background color is primary color by default.)

🡪Filled, unelevated button: unelevated button with colored background (primary color).

🡪Text button: transparent background, colored text. Use for less important button.

🡪Floating action button: provides a quick access to important or most used actions

In style resource file, there are several existing styles for material button. While a custom style was creating, you must extend one of those existing styles as parent.

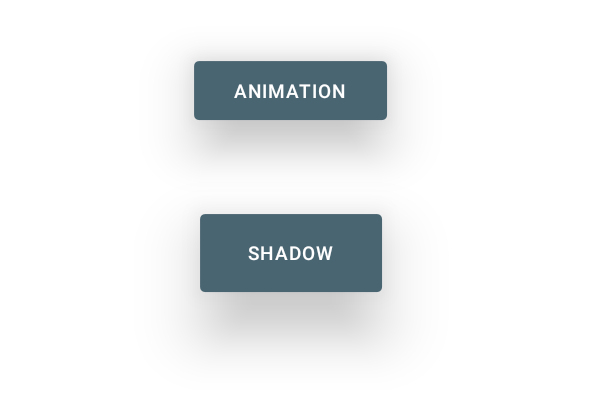
These styles are:

* Widget.MaterialComponents.Button
* TextAppearance.MaterialComponents.MaterialButton
* Widget.MaterialComponents.Button.TextButton
* Widget.MaterialComponents.Button.UnelevatedButton
* TextAppearance.MaterialComponents.MaterialButton.Secondary
* Widget.MaterialComponents.FloatingActionButton

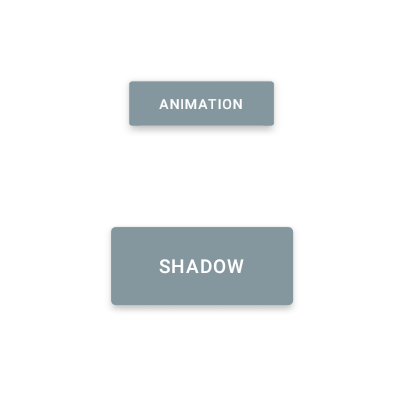
**Animation**

With Animator resource file, animations can be set to the view. To set, stateListAnimator feature is used in XML file. In the selector option, different states can be found. With objectAnimator subclass, objects’ property’s can be changed in different states.

For example, their scale can be increased, when the button is pressed or with translationZ attribute, depth sensation can be created.



In default, animation button’s translationZ value is 30dp. The elevation is in default mode. The shadow button’s translationZ value is also 30dp and the elevation value is 6dp.

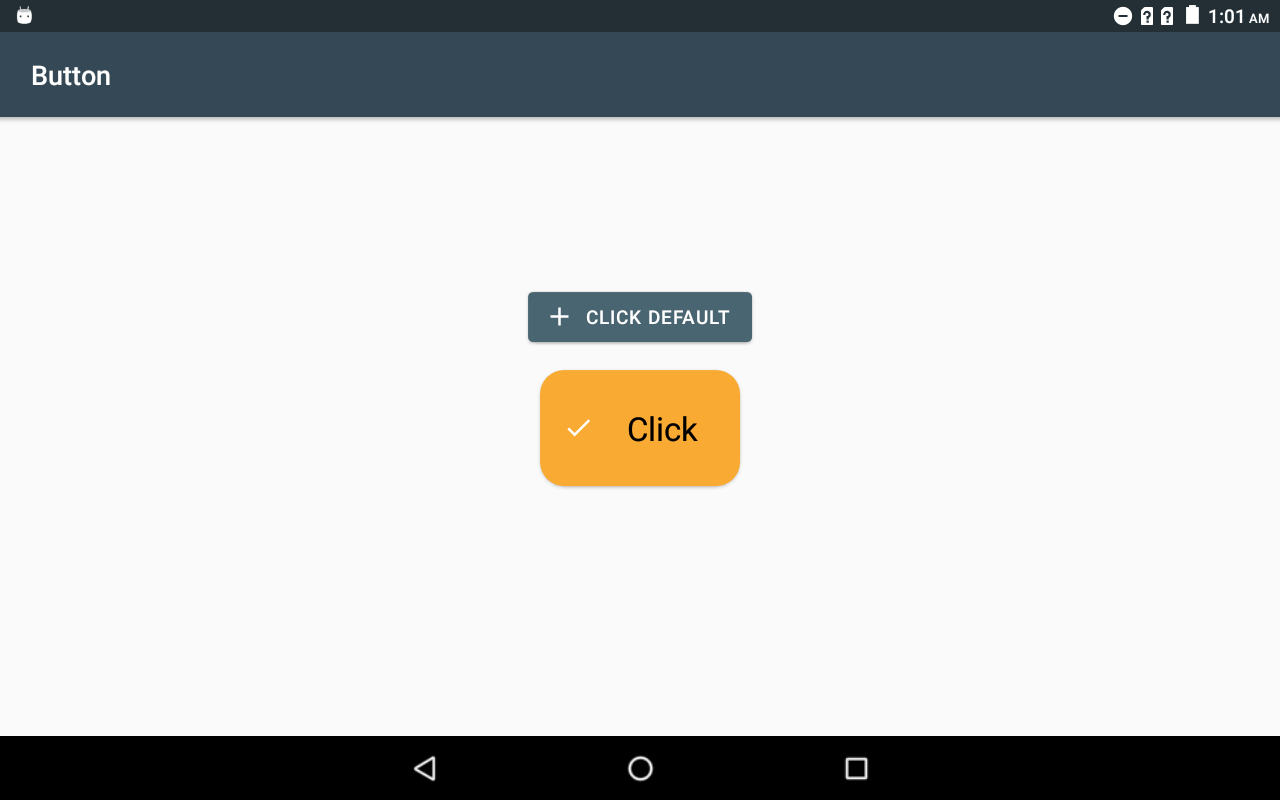
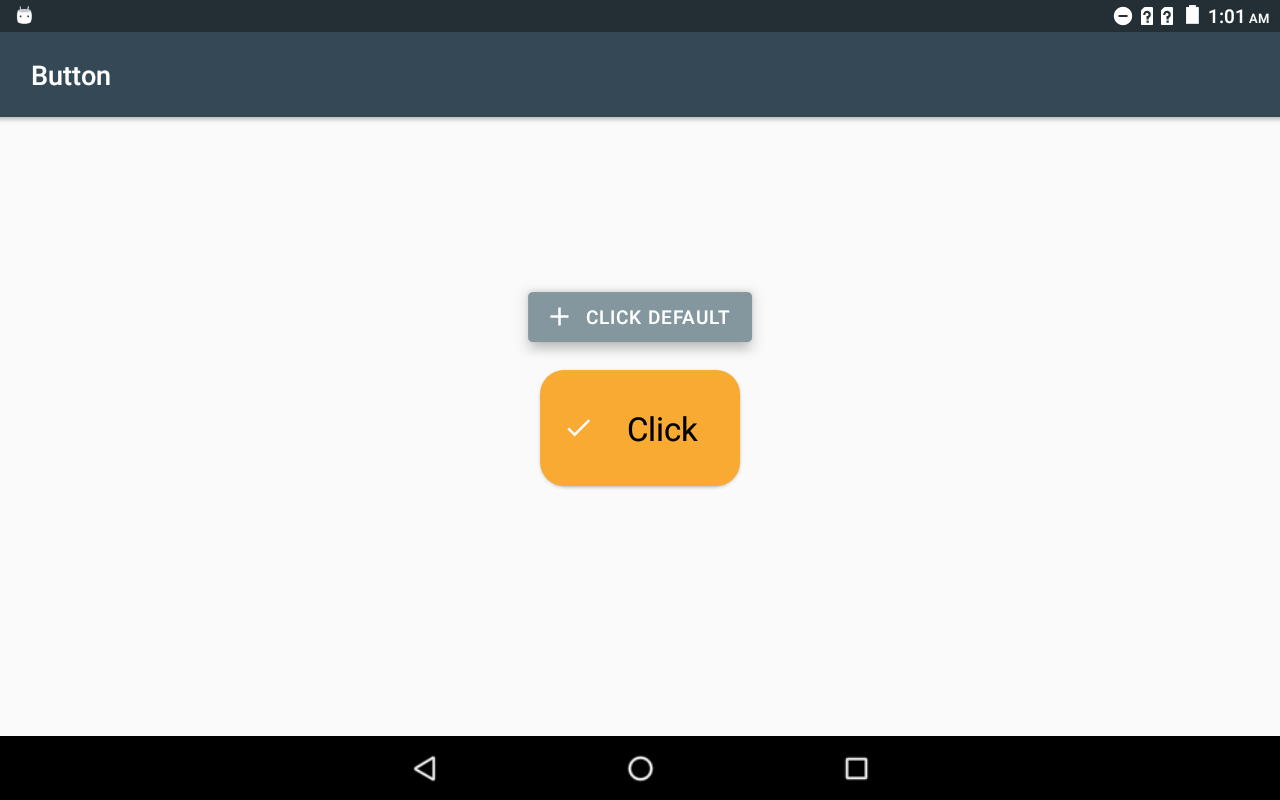


When the buttons are pressed, animation button’s scale is reduced to 0.75 and its translationZ value is also decreased to 5dp. In the shadow button, the scale is not changed, the elevation and translationZ value are decreased to 2dp.

**Widget.MaterialComponents.Button**

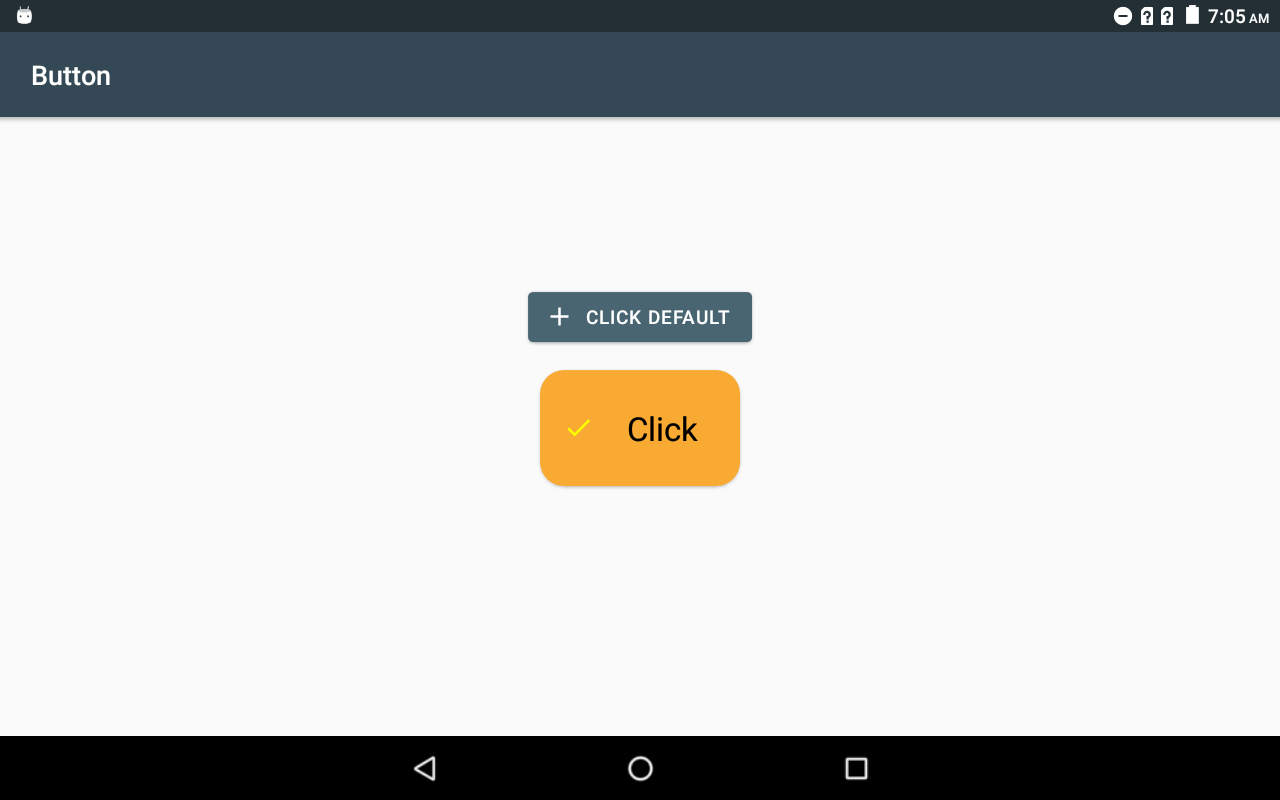
Its parent is Widget.AppCompat.Button. Its customized attributes are backgroundTint(set a tint to the background of button(background attribute not change the background color)), rippleColor(change the background color of button when the button is pressed), cornerRadius(set the corner of button’s roundness), insetLeft, insetRight, insetTop, insetBottom(set the border of the button), textAppearance (same with the view), iconTint(set the icon’s tint), iconPadding(arrange space between icon and and its surrondings), stateListAnimator(set an animation when the button’s situation is changed),

additionalPaddingLeftForIcon, additionalPaddingRightForIcon(add extra space between border of button and left and right of icon).

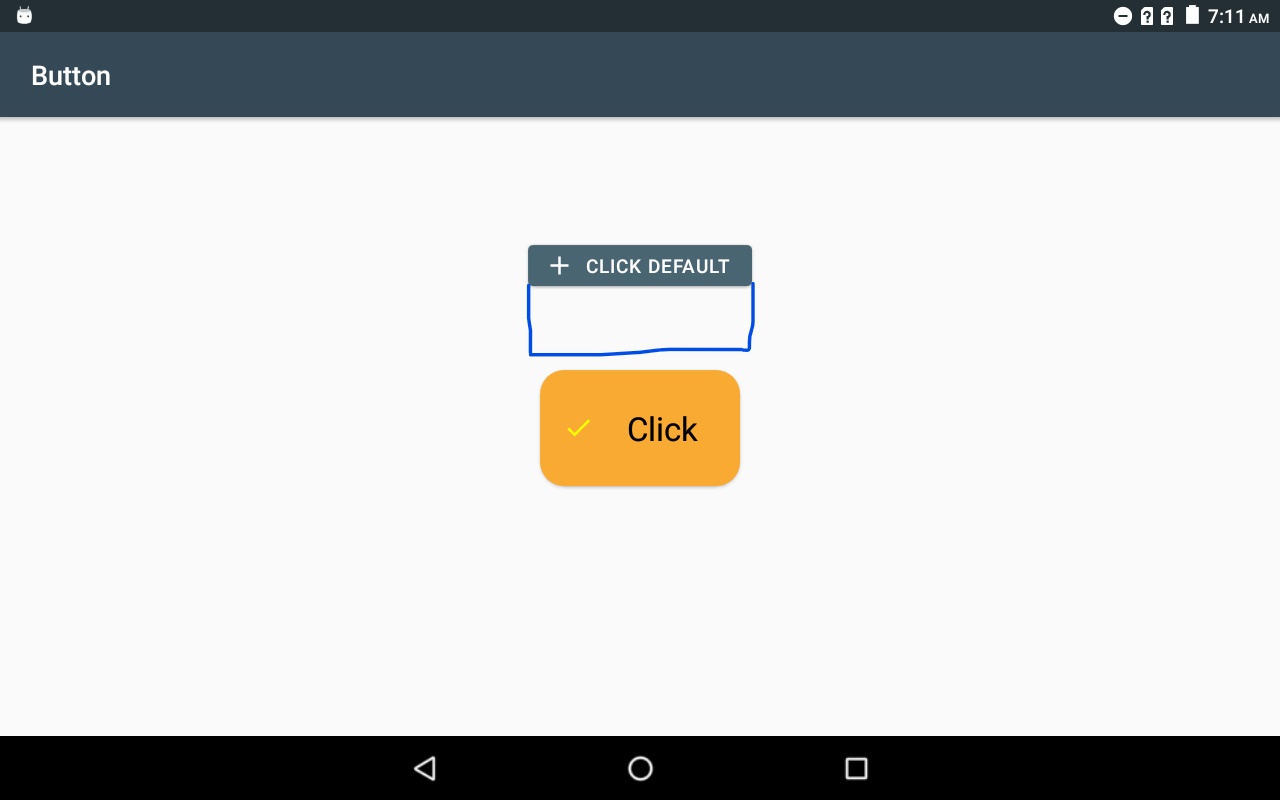
 

The click default is default settings of material button. The backgroundTint set a colorAccent, ripple color is white, corner radius is 4dp and state list animator attribute has a default object animator class.

In the second button which is customized, backgroundTint is set as colorSecondary, corner radius is set as 18dp, a check icon is added, adding extra 2dp left padding.



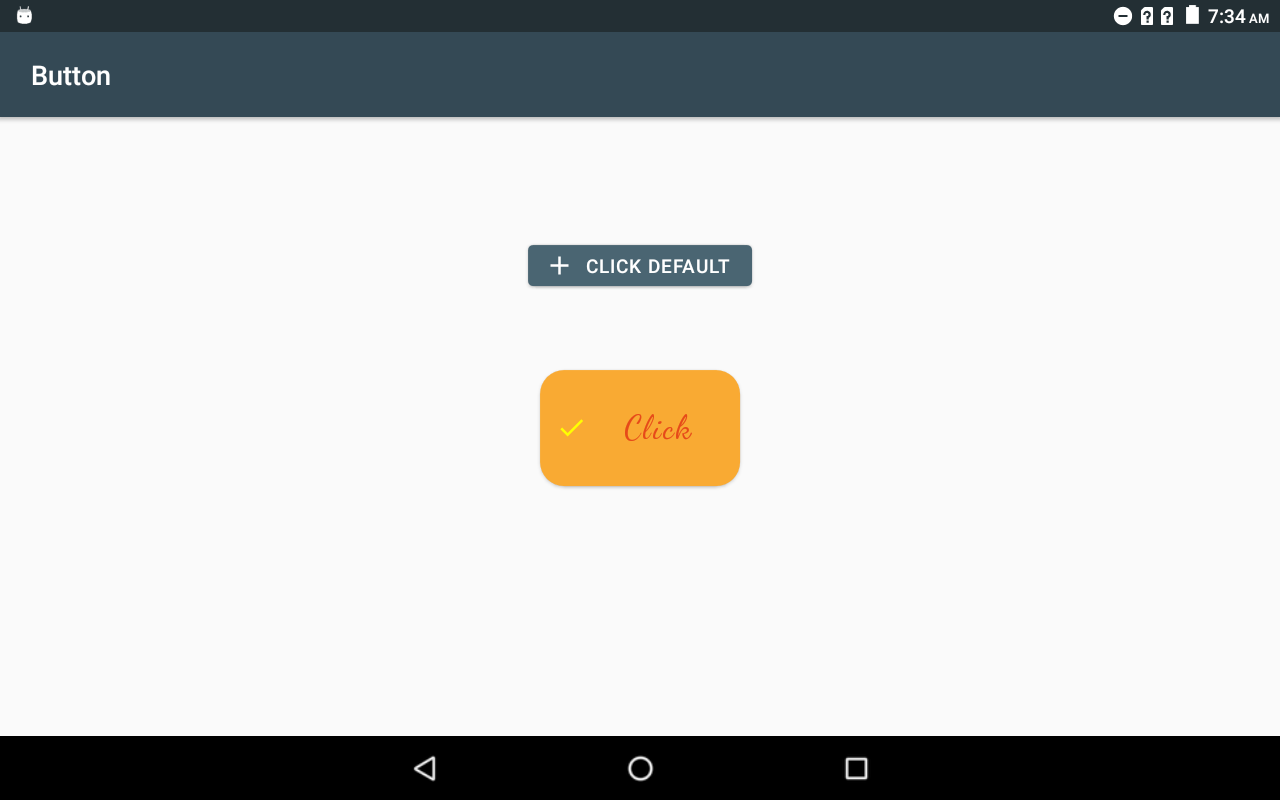
android:iconTint



android:insetBottom



android:rippleColor



android:textAppearance (custom style)

**TextAppearance.MaterialComponents.MaterialButton.Secondary**

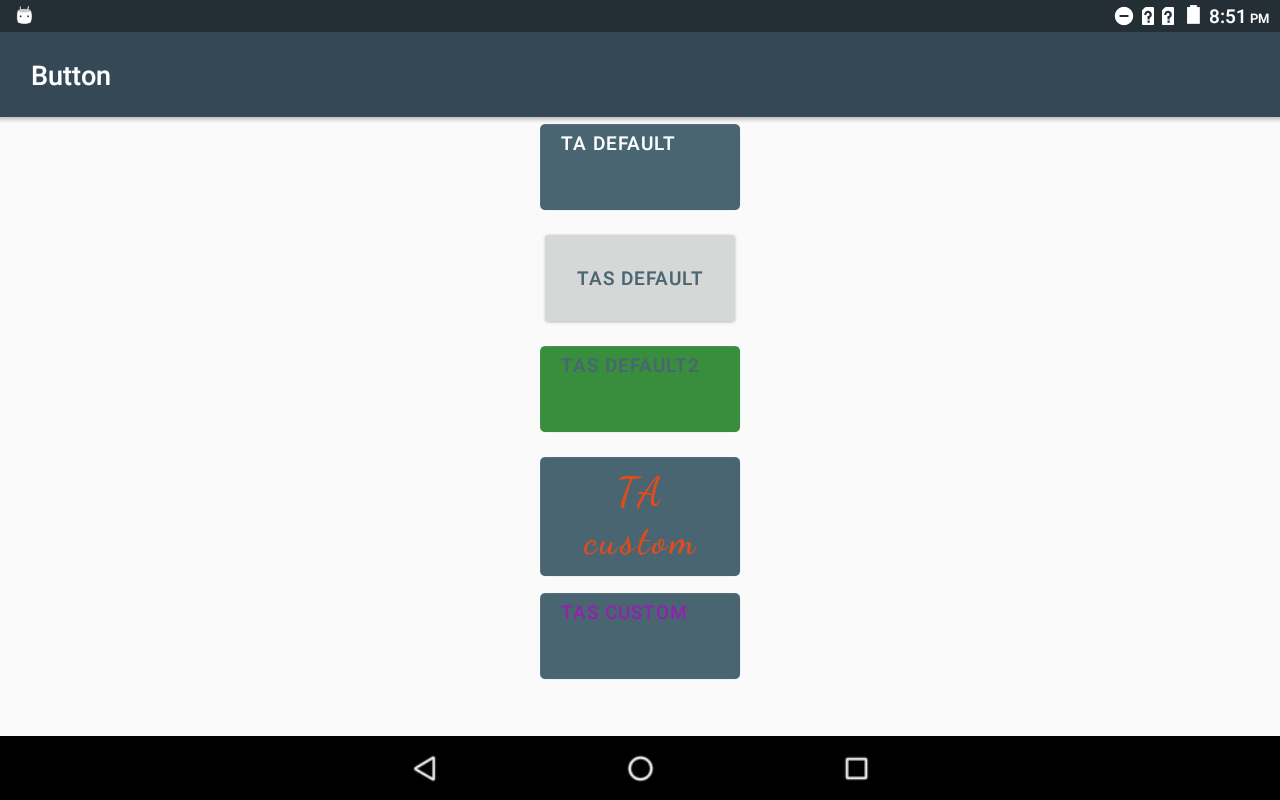
This style only changes the text color attribute as a difference. Other attributes are the same with view attributes. This style does not contain material design button attributes.

**TextAppearance.MaterialComponents.MaterialButton**

This style’s parent is android:textAppearance. textAppearance attributes are textColor, textColorHighlight, textColorHint, textColorLink, textSize, textStyle(bold, italic, normal). Material button text appearance customize textColor, textAllCaps, textSize, fontFamily(monospace, casual, cursive, sans-serif, sans-serif-condensed, serif, serif-monospace, sans-serif-smallcaps) and letterSpacing. This style does not have material design button attributes.

In fontFamily, this style creates a new element which is sans-serif-medium(default).

In default, clickable attribute is false.



(TA = textAppearance)

(TAS =textAppearance Secondary)

TA default’s style is TextAppearance.MaterialComponents.MaterialButton. In default, inAllCaps is false, textColor is white and fontFamily is sans-serif-medium. And the button is created as android.support.design.button.MaterialButton in XML file.

TA custom has a custom style. Its parent is TextAppearance.MaterialComponents.MaterialButton. And again the button is created as android.support.design.button.MaterialButton in XML file. Its textAppearance has a custom style.

android.support.design.button.MaterialButton’s default background color is colorAccent.

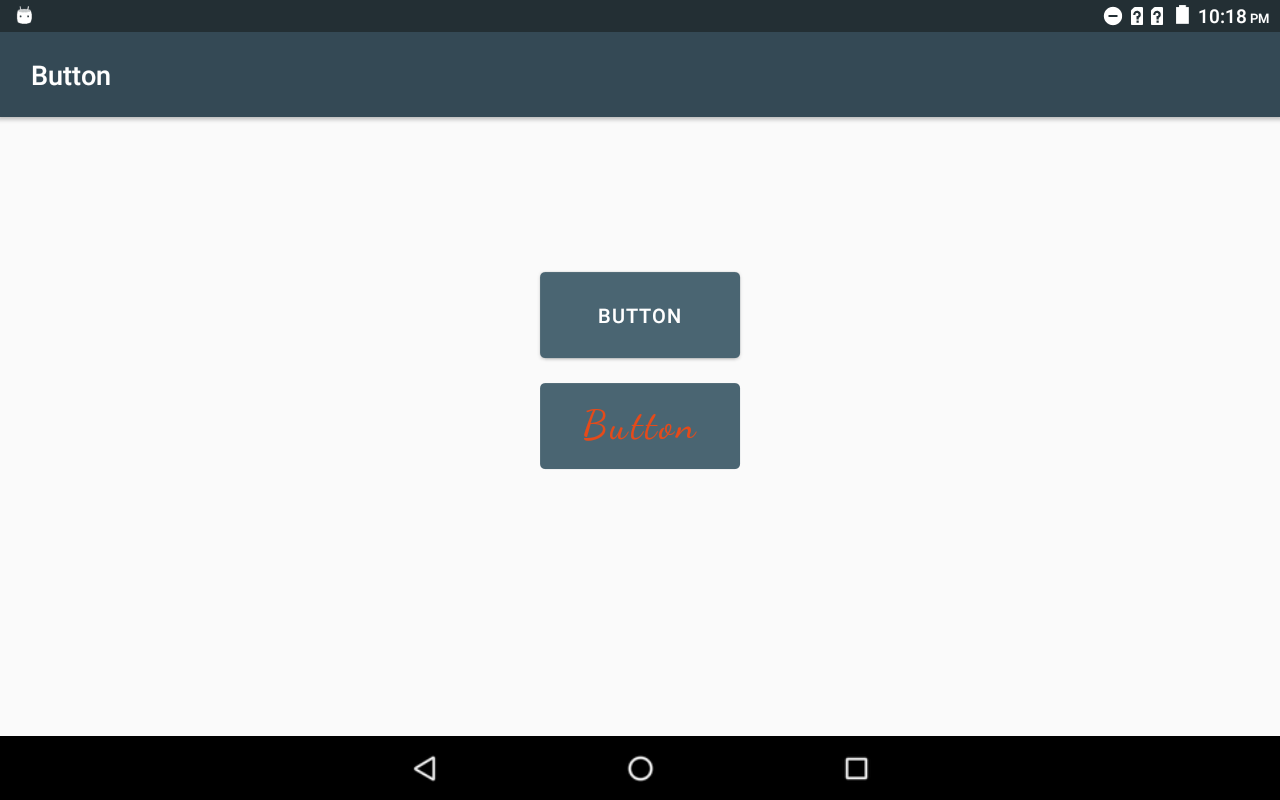
TAS default’s style is TextAppearance.MaterialComponents.MaterialButton.Secondary. Secondary text appearance has only one custom attributes which is textColor. And the default text color is colorAccent. So, when we use android.support.design.button.MaterialButton, the

background color is also become colorAccent, as a result, the text is not visible. To understand that, first create TAS default that is used <Buttom/> in XML file which makes default background color (grey). And TAS default 2 is created as <android.support.design.button.MaterialButton />. The backgroundTint color is set to green to observe secondary style in that type button.

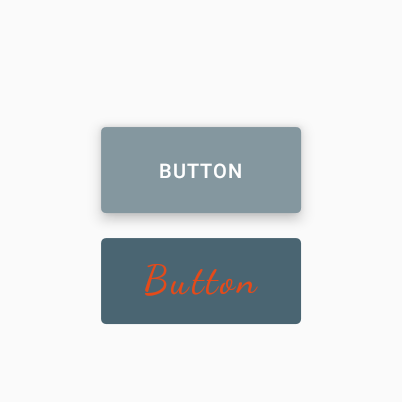
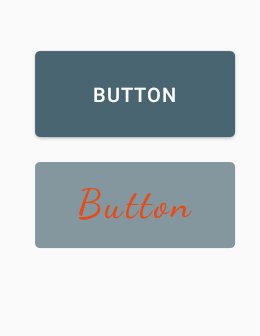
TAS custom has a custom style that extends from TextAppearance.MaterialComponents.MaterialButton.Secondary. The modified attribute is only textColor so the text color is set as purple in that example.

**Widget.MaterialComponents.Button.UnelevatedButton**

This style customizes only stateListAnimator attribute.



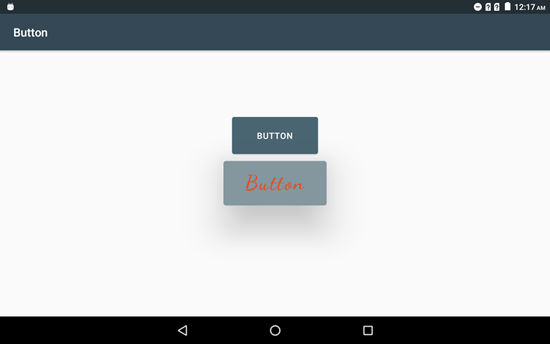
The button with white text is default material button and the button with red text’s style extends from Widget.MaterialComponents.Button.UnelevatedButton.

(material button is pressed) (material unelevated button is pressed)

In material button, when the button is pressed, its elevation increases and it is felt that the button become closer. But in default unelevated button, when the button is pressed, the elevation does not change.

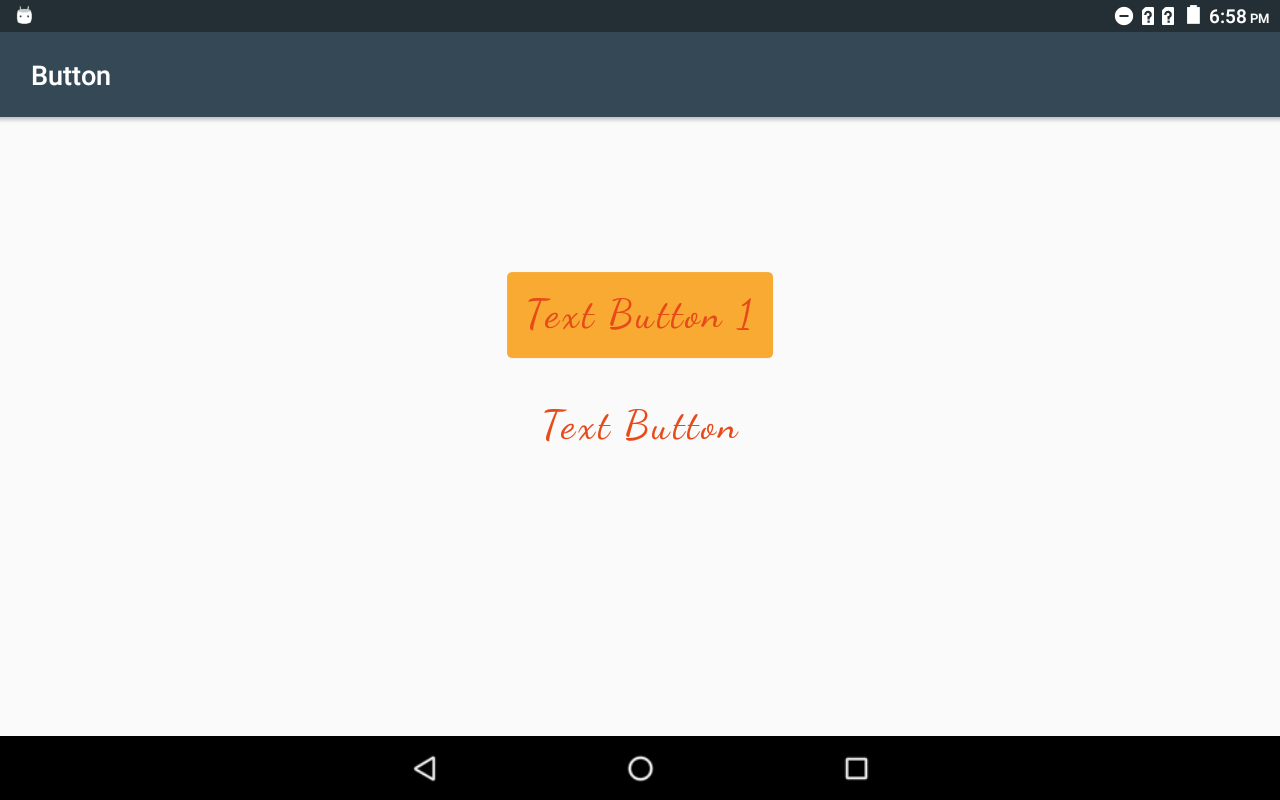
The stateListAnimator attribute can be customized also in unelevated button with animator resource file:



**Widget.MaterialComponents.Button.TextButton**

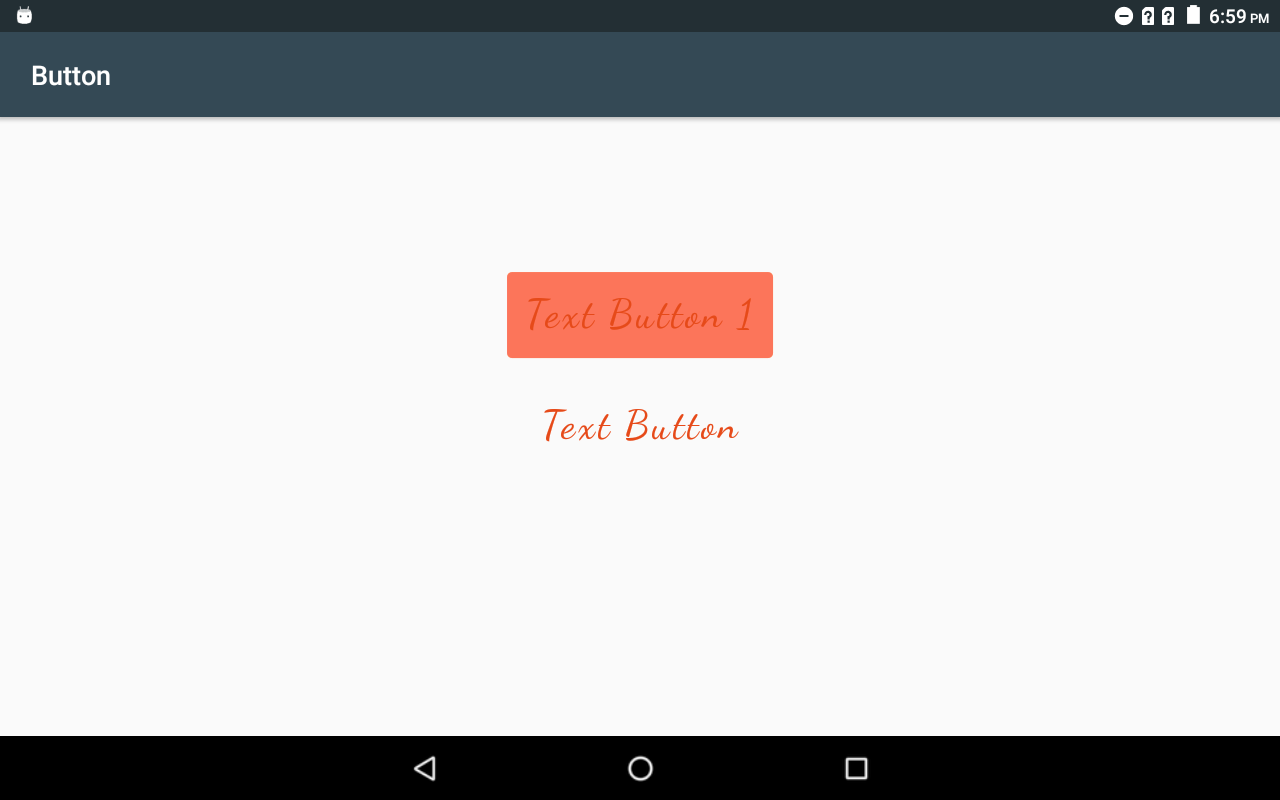
This style’s parent is Widget.MaterialComponents.Button.UnelevatedButton. Its default background color is transparent therefore you cannot observe the clickable and focusable attributes. Background color cannot be changed but you can set backgroundTint color which make enable to observe clickable and focusable attributes. Also, because of the absence of background color, you cannot observe ripple color. Besides, with textAppearance you can add a style to change text’s features.

Its customized attributes: backgroundTint, iconPadding, iconTint, rippleColor, paddingRight, paddingLeft, textAppearance, stateListAnimator(unelevatedButton’s attribute)

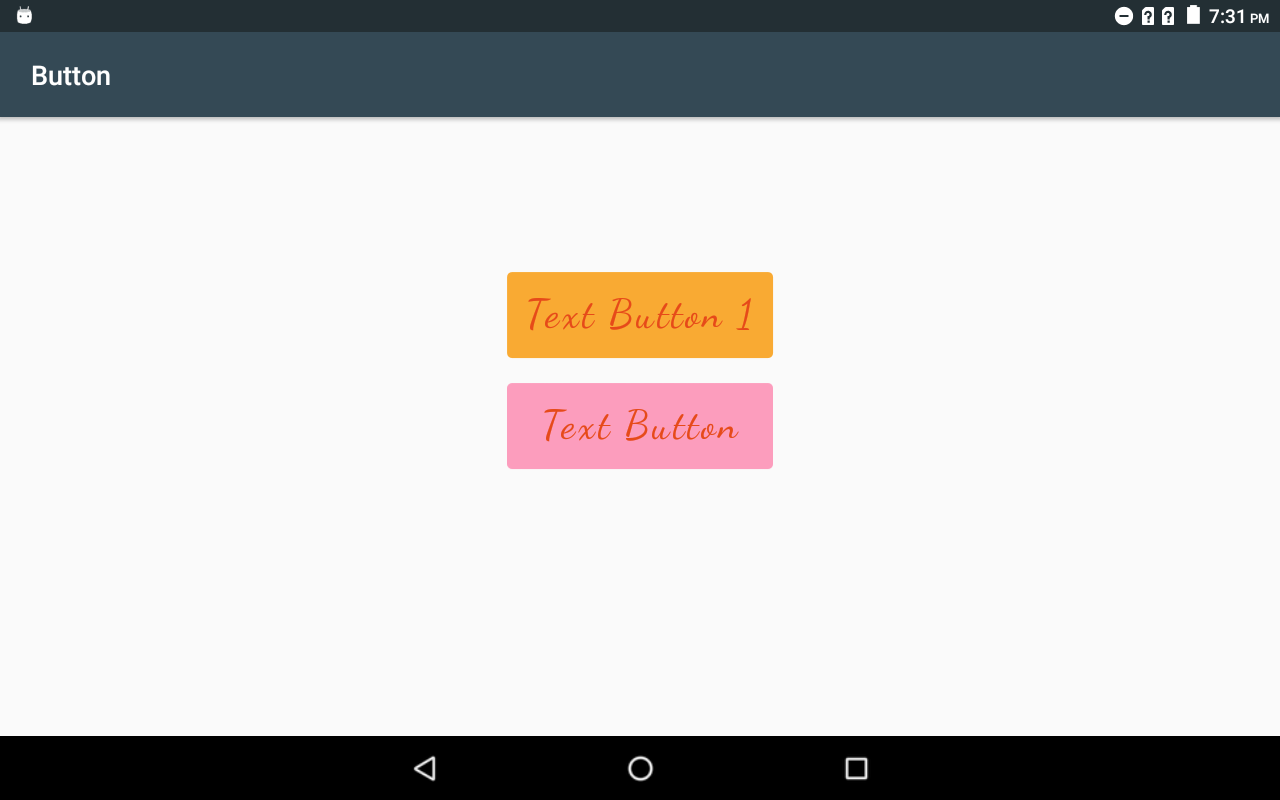


Text Button 1 is applied a custom style that parent is Widget.MaterialComponents.Button.TextButton. Its text appearance is set to another custom textAppearance style. Its backgroundTint attribute is set to secondary color of app. IconPadding, paddingLeft and right are the same with the other material button style. Also, rippleColor is set to the pink.

The only difference between Text Button 1 and Text Button is backgroundTint. backgroundTint is not set in Text Button. Therefore, its background is transparent.



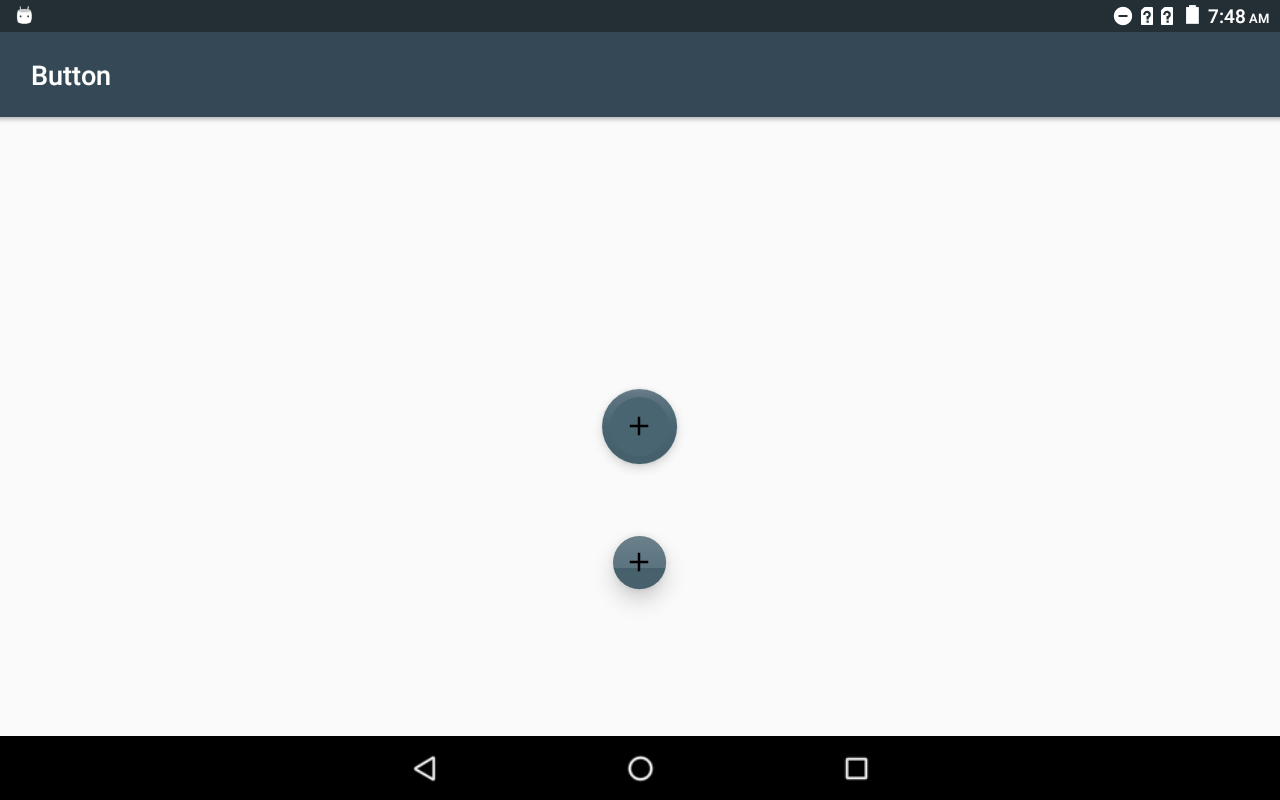
Text Button 1 is pressed.



Text Button is pressed.

**Widget.MaterialComponents.FloatingActionButton**

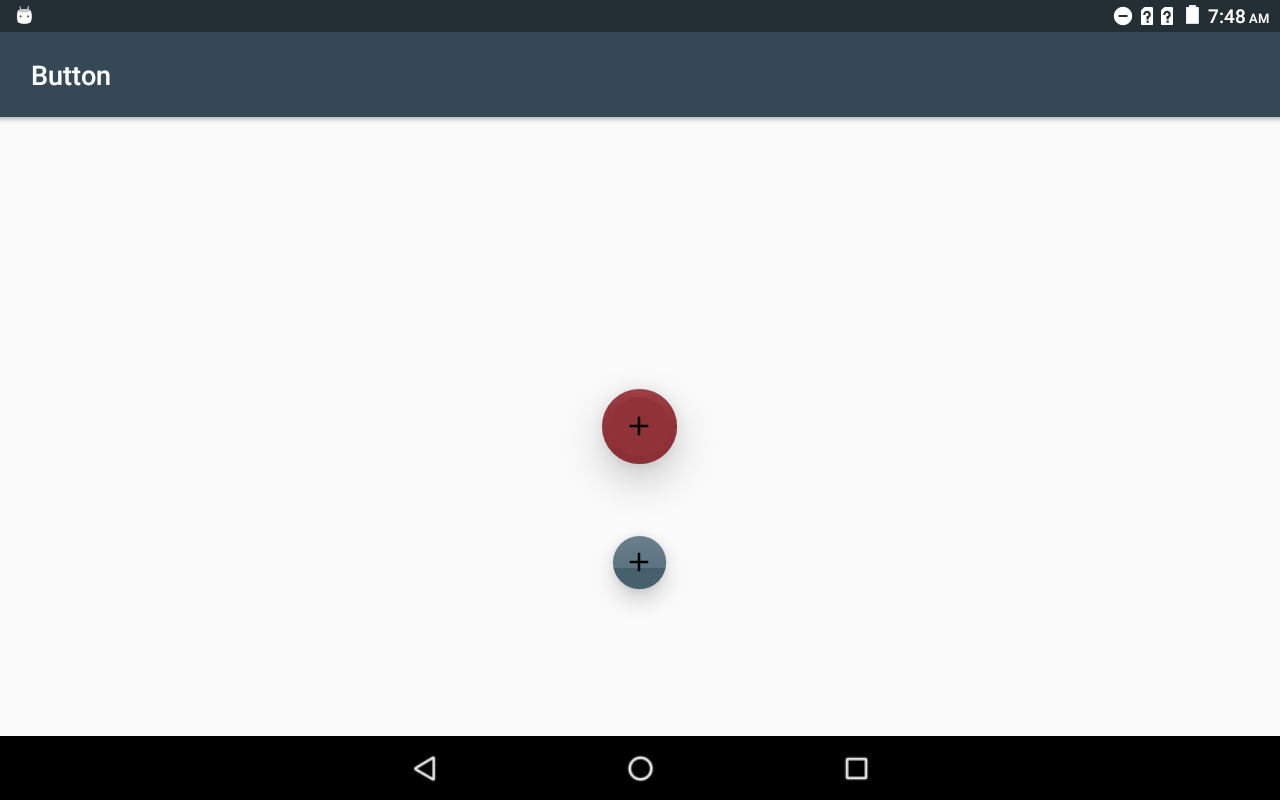
Its parent is Widget.Design.FloatingActionButton. It is a circular button that usually contains an icon in the middle. Its custom attributes are elevation, hoveredFocusedTranslationZ(in hovered state, it set the z axis as 2dp), pressedTranslationZ(in pressed state, it set the z axis as 6dp), rippleColor(same as the material button), showMotionSpec(there is a custom object animator class for that attribute) and hideMotionSpec(there is a custom object animator class for that attribute), fabSize(set the size of button, three options: mini, normal, auto) and fabCustomSize(custom your fab size besides the mini and normal).



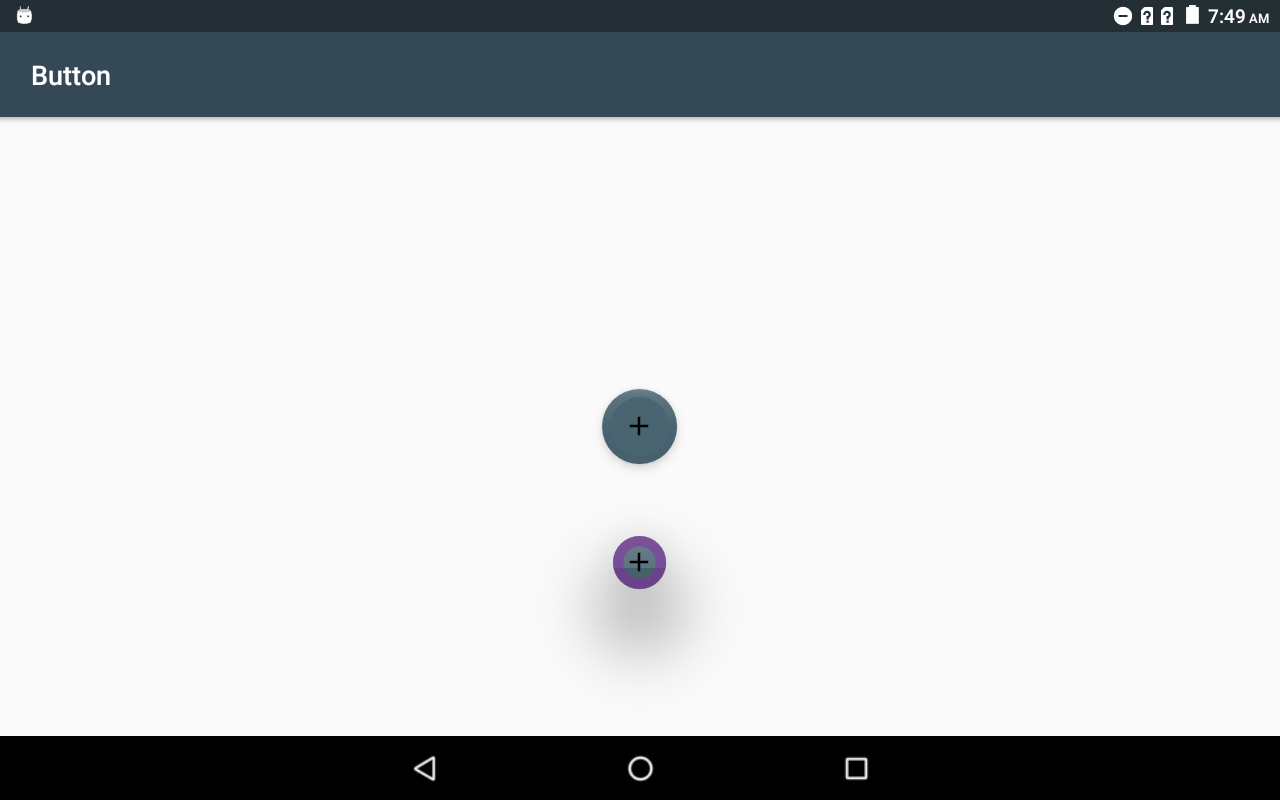
fabSize: The upper one is normal, bottom one is mini.

borderWidth: 6dp in upper one, 24dp in bottom one (default 0.5dp)

elevation is 12 dp in mini fab.



Upper fab is pressed. Ripple color is set as red. PressedTranslationZ is 16dp.



Mini fab is pressed. Ripple color is set as purple. PressedTranslationZ is 32dp. Middle does not turn purple because of the borderWidth value.

**References**

Homepage. (n.d.). Retrieved June 27, 2018, from <https://material.io/>

Android Developers. (n.d.). Retrieved June 27, 2018, from https://developer.android.com/