大丈夫。

したがって、スライダーは美しく見え、同様に美しく機能しています。

しかし、特にデザイナーである人々の間で注目されている人にとっては、スライダーとスライダーの間に大きな違いがあることに気付くでしょう。

そのため、1つは明らかに、アクティブな白い面と非アクティブな灰色の面があることです。これは、プロパティ、acticeColorをタップして色に変更できるため、変更が非常に簡単です。

。白い。

しかし、すぐに親指の色、つまりメイントグルの色もアクティブトラックと同じ色に変化することに気付くでしょう。

また、これを見ると、そのホルダーまたはハンドルは、ここにある小さなものよりもはるかに大きいです。

それでは、どのようにしてデザインをこのように見せるのでしょうか？

スライダークラスのドキュメントに進むと、activeColorまたはinactiveColorなどの通常のプロパティが変更されていることがわかります。

そして、リストの一番下に到達すると、このアクティブなトラックに影響を与えずに、たとえばそのハンドルのサイズやハンドルの色を変更することしかできないことに気付くでしょう。

それでは、スライダーのプロパティを変更する以外に何ができるでしょうか？

それでは、Flutterウィジェットをカスタマイズして、希望するデザインに正確に一致させる1つの方法を紹介します。

これを読み通すと、スライダーの表示方法を決定するために、スライダーはSliderThemeウィジェットまたはThemeData（ウィジェットツリーの上のウィジェット）から利用可能なSliderThemeDataを使用することがわかります。または、いくつかの色をオーバーライドできますが、きめ細かい制御はSliderThemeDataから行われます。

これは私たちができることの一つであり、以前にテーマについて話しました。

そのため、すでにアプリテーマにプライマリカラーを追加し、このThemeDataを使用してアプリ全体でscaffoldBackgroundColorを変更しました。しかし、個々のコンポーネントだけに特定のテーマを設定できるとも言っています。

別のウィジェット内にウィジェットを埋め込むことでそれを行うことができます。

したがって、この場合、SliderTheme内に埋め込むことができ、SliderThemeにはdataと呼ばれるプロパティがあり、SliderThemeDataと呼ばれるものが必要です。

ENTERキーを押すと、たとえばトラックの高さ、activeTrackColor、inactiveTrackColor、thumbColorなど、これらすべての設定ができることがわかります。

ただし、このようにSlideThemeDataを使用するには、すべてに値を指定する必要があります。そうしないと、デフォルトではnullになり、色やtrackHeightはありません。

そして、今保存していた場合、これらのプロパティはどれもnullにできないため、スライダーが完全に壊れていることがわかります。値を指定する必要がありました。しかし、デフォルトのテーマに満足しているのに、20個の値を指定してこのコードをすべて入力することなく、いくつかのことを微調整したい場合はどうしますか？

Flutterが持っているすべてのウィジェットで本当に役立つものがあります。それは.ofと呼ばれるものです。

したがって、デフォルトのSliderThemeがすでにあることがわかります。そして、それを利用して、.ofというメソッドを使用します。このメソッドは、指定されたコンテキストで最も近いSliderThemeからそのデータプロパティを返すデータを返します。だからSliderTheme.ofを書くことができます。

そして、ここから来るビルドコンテキストを提供できます。

そして、このコンテキストは基本的にアプリの現在の状態です。

アプリがそのまま表示される方法です。

そして、これには現在のSliderThemeが含まれます。

そのため、.ofと言うと、デフォルトでそのままのコピーを取得します。そして、以前に見た.copyWithを使用して、暗いテーマをコピーして、いくつかの変更を追加しただけで、再び書き込むことができます。

このcopyWith内では、SliderThemeについて変更したいものだけを提供できます。

そして、すべてをゼロから作成する必要はありません。

したがって、変更したいものは、たとえばthumbShapeです。

したがって、デフォルトのサイズとしてではなく、通常の半径が6の丸いスライダーのつまみの形が必要であるが、より大きな半径が必要であると言って、サイズを変更できます。

15に変更します。

そして今、あなたが私がセーブを打ったとき、あなたが見ることができる、私の親指はずっと大きい。しかし、通常クリックすると、少し透明な外側のリングの上に小さなオーバーレイが表示されることに注意してください。

そして、通常の完全に不透明な形状よりも小さいため、これはなくなりました。

先に進んでそれを追加し、さらに大きくします。したがって、オーバーレイシェイプを追加して、丸いスライダーオーバーレイシェイプにすることができます。

デフォルトでは、16であることに注意してください。

したがって、親指の形状より1ピクセルだけ大きくなります。

それで、もっと大きくしましょう。

30にしましょう。

これで、これをクリックすると、オーバーレイがアニメーション表示されます。今はかなり見栄えがいい。

thumbColorはどうですか？

thumbColorプロパティを使用してこれを変更できます。また、0xFFEB1555になるカスタムカラーである以前のピンク色に変更します。

そして、それはアプリ全体でほとんど持っているピンク色です。

保存してみましょう。ただし、スライダーは変更されていません。これは、activeColorがそのthumbColorをオーバーライドしているためです。

それを取り除こう。

そこで、代わりに、activeTrackColorを追加します。これは、colors.whiteになります。

保存すると、左側が白、右側がまだ灰色、親指がピンクになっていることがわかります。ただし、クリックしても、実際にはオーバーレイの色はありません。

オーバーレイカラーも追加しましょう。

そして、thumbColorと同じ色になりますが、少し透明にする必要があります。

これが私の色の最初の部分の出番です。

このコードには4つのコンポーネントがあり、常に0xで始まり、アルファまたは透明度の量が2文字で、次に赤、緑、青だと言ったことを覚えていますか？透明性を高めたい場合はどうでしょうか？

さて、これらの最初の2つの数値を変更します。

デフォルトの量を有効にしたとき、最初にどれだけの透明度を得たかがとても気に入りました。

では、デフォルトの金額がどうなったのかを知るにはどうすればよいですか？

さて、Macの場合はCOMMAND、Windowsの場合はCONTROLを押すと、スライダーテーマに移動できます。SliderThemeをクリックすると、SliderThemeウィジェットに移動できます。

そして、ここでoverlayColorプロパティを検索します。すべてを調べてみると、色がoverlayLightAlphaと呼ばれるアルファ量を持つテーマのプライマリカラーとして設定されていることがわかります。

それでは、このoverlayLightAlphaを見つけてみましょう。

そのため、それを強調表示して、COMMAND + FまたはCONTROL + Fを押します。このドキュメントを読み進めていくと、ここに定数として表示されます。

そのため、彼らはoverlayLightAlphaを0x29に設定しました。

したがって、彼らは言う約16％の不透明度を与えます。

そのため、これを色にコピーし、最初の4文字を0x29に置き換えて、保存をクリックすると、親指が、ピンク色の不透明度の16％のオーバーレイ色になります。使用しています。これは、ここで見たデザインに完全に似ています。前に述べたように、スライダーテーマをここに保持するか、アプリに複数のスライダーがあり、アプリ全体のスライダーテーマが必要な場合は、スライダーテーマをコピーしてここに貼り付けることができます。そして、アプリ全体のスライダーはすべて同じテーマを持つことになります。

ただし、この場合、スライダーは1つしかないため、実際にどこにあるかは関係ありません。

そして、同じコンポーネントに関連するすべてのものを同じ場所にまとめておくのが好きです。一貫性を保つために、おそらくinactiveTrackColorもここに配置する必要があります。

したがって、非アクティブなトラックの色を追加し、この色をここに貼り付けます。ここから削除できます。

つまり、すべてのスタイリングはテーマ内で行われ、すべてのプロパティまたは機能は実際のスライダーコンポーネント内で行われます。

複数の場所で使用したい場合、もちろんこのウィジェットを抽出できますが、この例では実際には1つしかありません。

だから、私はそれをそのままにしておきます。また、必要に応じて、これらすべての色とテーマを定数に移動することができ、必要に応じてこのメインファイルを少しきれいに保つことができます。このレッスンでは、私があなたに見せようとしているのは、何でもできるということです。さまざまなFlutterウィジェットを組み合わせた独自のウィジェットを作成するか、素材コンポーネントなどのFlutterウィジェットを使用するときはいつでも、テーマをタップしていつでもテーマをタップすることができます。通常付属するプロパティだけでなく、さらにカスタマイズできます。

たとえば、マテリアルコンポーネントからのアラートダイアログを使用している場合、テーマクラスを見て、変更可能なダイアログテーマがあることがわかります。または、アプリバーを使用している場合は、アプリバーのテーマ、テキストテーマ、アイコンテーマ、またはタブバーテーマを変更できます。

基本的に、これらの材料コンポーネントはすべて、テーマデータを使用してさらに微調整できます。

それでは十分ではない場合はどうなりますか？現時点では、私たちが達成しようとしているデザインと非常によく似ているため、スライダーを使用しているためです。

では、なぜ車輪を再発明するのでしょうか？

なぜカスタマイズしないのですか？

しかし、それで十分でない場合はどうでしょうか？

本当に違うものを作りたいとしたらどうでしょうか？ まあ、円や正方形などの基本的な形状など、非常に基本的なFlutterコンポーネントから独自のウィジェットを構築することもできます。 次に、標高を変更するか、背景色を変更して動作を変更し、独自のウィジェットとその動作方法を完全に定義します。

そして、私たちは次のレッスンでまさにそれを探求するつもりです。

それで、それ以上のこと、あなたに会いましょう。