大丈夫。

それではまず、Android Studioを開いて新しいFlutterプロジェクトを開始しましょう。

これを完全にゼロから構築します。

Flutterアプリケーションを選択したら、[次へ]をクリックして、プロジェクトにtodoey\_flutterという名前を付けます。

ここで完了したら、[次へ]をクリックして終了します。

大丈夫。

これが、通常の機能を備えた新しいアプリケーションです。

そのため、基本的に削除するサンプルカウンターアプリを含むmain.dartを取得しました。

だから私は私のホームページを取り除くつもりです。

また、myAppを削除し、代わりにほとんどゼロから作成します。

そこで、ショートカットを使用して新しいステートレスウィジェットを作成し、これをMy Appと呼び、これが実行するアプリに一致するようにします。

ここに赤いボックスが表示されると、Android Studioによって作成され、タブで移動したり、特定のものを簡単に変更したりできます。

しかし、それを消したい場合は、タブを押すだけで消えます。

コンテナを作成する代わりに、アプリ内で実際にマテリアルアプリが必要になりました。

そこで、ホームルートを持つマテリアルアプリを追加します。最初に表示する画面です。そして、これをTaskScreenと呼び、ここで作成します。

明らかに、まだタスク画面がないため、これには赤い波線があります。先に進んで作成してみましょう。

そのため、libフォルダー内に新しいフォルダーを作成し、screensと呼びます。

そこで、ここでスクリーンファイルを整理します。次に、新しいDartファイルを作成します。このファイルをタスクと呼び、アプリの最初の画面に表示するものをTasksScreenと呼びます。

そして、丸括弧で作成します。

TasksScreenがまだ存在しないため、赤い波線ができました。

それでは、すべてのスクリーンファイルを作成する新しいフォルダを作成して、作成してみましょう。

ここですべてのスクリーンファイルを整理し、最初のファイルを作成します。これは、「tasks\_screen」と呼ばれるDartファイルで、Enterキーを押します。

これで、tasks\_screen.dartができました。

ここで、UIを作成するときに常に必要になるマテリアルをインポートします。

それから、TasksScreenと呼ばれる新しいステートレスウィジェットを作成します。

この場合、ここにSが表示されていることに注意してください。複数表示されているため、画面には多数のタスクが表示されますが、正確に一致することを確認してください。

必要に応じて、コピーと貼り付けも常に簡単に行えます。

したがって、screens / tasks\_screen.dartをインポートするだけで赤い波線が表示されます。これですべての矢印がなくなり、Android Studioで問題がなくなりました。

最後に、トップ画面に戻って、この画面の構築を始めましょう。

コンテナだけを作成したくありません。

それで、アプリをそのまま実行して、ビルドしているものを見ながらビルドできるようにします。そうすることで、画面にピクセルをペイントできます。

ビルドが完了したので、この黒い画面は非常にエキサイティングではありませんが、まもなくジャズアップする予定です。

それでは、アプリはどのようになるのでしょうか？

ダウンロードセクションまたはこのレッスンのリソースには、モックアップを行うためのダウンロードがあります。

そこで、これらをSketchで作成しましたが、残念ながらMacでしか利用できないアプリケーションです。

モックアップを実際にPDFにエクスポートしたので、Android Studioでそれをダウンロードし、ズームインして見回したり、色を選んだり、デザインに必要なものを見たりすることができます。

しかし、私もあなたと一緒に一歩ずつやっていきます。

それでは、このデザインを使用して、ウィジェットに転送してみましょう。

そのため、まず、明るい青のアクセント色である背景色が必要です。

画面にコンテナを置きたくありませんが、代わりにマテリアルアプリが一番上のウィジェットであることに注意してください。

それからまもなくタスク画面が表示され、ここで私たちを引き継いで、いつものように足場を構築します。

したがって、足場の背景色は明るい青のアクセントになり、保存ボタンを押すとすぐにシミュレータでレンダリングされた色が表示されます。

この場合、scaffoldにはアプリバーはありませんが、もちろん本体があります。

モックアップにアイコンがあり、テキストがあり、さらにテキストがあり、この他のコンポーネントがここにあることに注目してください。そのため、足場の本体は列ウィジェットになります。だから、すべてが多かれ少なかれ垂直に積み上げられているので、このシナリオではコラムが非常にうまく機能すると思います。

そして、私たちのコラムの中で、私たちはたくさんの子供を持つことになります。最初の子供はこのアイコンこちらです。

次に、アイコンを取得するために、アイコンウィジェットをタップします。アイコンデータは、ここにあるicons.listになります。これは、やることリストのように見えます。

アイコンが追加されたので、ここの左上隅に並べて表示できます。そして、もう少し簡単にするために、ここにもテキストを追加しましょう。

Todoeyのタイトルを付けませんか？

テキストウィジェットを追加しましょう。テキストウィジェット内にある文字列は、もちろんTodoeyです。

そして、保存ボタンを押すと、左上隅にすべてが壊れているのがわかります。

1つの選択肢は、もちろん、通常使用するセーフエリアウィジェット内に列をラップすることです。これは、ユーザーが実際に操作できない電話の部分から安全を保つために使用します。

そのため、iPhoneでは、ノッチの周りにあります。 Androidスマートフォンでは、ステータスバーの周りにあるため、バッテリーと時刻などが表示されます。

また、iPhoneのこの部分は一種の新しいホームボタンです。したがって、これも安全な領域の外になります。

しかし、私にとってこの種のマージンは十分ではありません。

私たちのコンテンツはすべて、すべてのエッジからかなりパディングされているので、もう少し押し下げたいと思います。

したがって、安全な領域を使用する代わりに、実際にはコンテナを使用するだけです。コンテナには常にpaddingプロパティがあることを思い出してください。

それでは、列を埋めて、エッジの挿入を行いましょう。

今、私が通常使用する「すべて」であるものを使用する代わりに、4辺すべてに同じ量のパディングを与えることで、代わりにこの場合は「のみ」を選択します。

これは、上部と残りの3辺に異なる量のパディングが必要だからです。

だから最初に、私は実際になんと6になる最高値から始めます。私はそれをたくさん押し下げたいからです。そして、30になる予定の左と、30になる予定の右と、30になる予定の下部があります。

それでは、saveを押してみましょう。このような忙しさのすべてをはるかに超えて押し下げられており、実際のアプリのbodyの内側にあります。

ここで必要なのは、これをもっとこのようにすることです。

あなたはそれは本当に遠く離れていると思いますが、実際には、すべてのウィジェットはウィジェット内にいくつかのものをラップするだけなので、それほど多くのコードを必要としません。

最初のことは、この小さな丸い円の中にアイコンがあることです。

そして、これらのプロファイルアイコンまたはアバターアイコンを生成する最も簡単な方法、またはこの場合は単なる円であることがわかっている場合は、円のアバター内にラップすることです。

これは以前に見たことがありますので、あまり詳しく説明しません。しかし、サークルアバターの機能や、実際に私が話しているウィジェットを簡単に改訂したい場合は、常に更新されているFlutterのドキュメントをご覧ください。

そして、これらのウィジェットのどれよりも多くの情報を持っています。

サークルアバターを作成したので、その中にアイコンがあります。

しかし、サークルアバターに関する他のプロパティ、つまり多分background colorも変更したいと思います。

それをcolors.whiteに変更しましょう。

そして今、私たちは素敵な白いアイコンを持っているので、かなり似ていますよね？あとは、少し大きくするだけです。

円の半径を変更して、30ピクセルとしましょう。

そのため、おそらく直径は約60になります。

だから、ずっと大きく見えます。

そして今、対照的に、アイコンが少し小さすぎるように見えるので、アイコンのサイズも変更しましょう。

ただし、アイコンにはsizeというプロパティがあり、親circle thatavatarと一致するように30に変更しましょう。

スタックしているときやウィジェットのプロパティを思い出せないときは、MacでCOMMANDを、WindowsでCONTROLを押したままウィジェットにカーソルを合わせると、すべてのプロパティが表示されるので、を変えることができます。

これらのプロパティの詳細を知りたい場合は、ウィジェットにカーソルを置き、クイックドキュメントに移動します。これはMacではF1ですが、システムでは異なる場合があります。

そのため、[表示]クイックドキュメントを参照して、ショートカットを確認してください。そしてもちろん、ドキュメントだけでなくすべてのプロパティが表示され、何ができるかを見るだけでなく、ウィジェットを少しよく理解できるようになります。

最後の最後のステップはもちろん、MacまたはCONTROL WindowsでCOMMANDを押しながらクリックするだけで、ソースを表示できます。Flutterのオープンソースですよね？

何かがどのように機能するか、またはそれをどのように修正するかを理解したい場合、それが最終的な方法です。

大丈夫。

アイコンはかなり見栄えが良いですが、デザインに鋭い目を持っている場合、この青がこの青とは少し異なることに気づいたかもしれません。これは、デフォルトですべてのマテリアルアプリが青である色であるためです。メインテーマの色として青。しかし、私たちの場合、それを切り替えたいと思います。

それで、アイコンの色をcolors.lightBlueAccentに変更して、一致させましょう。見て、ずっと良く見えます。

次に、次に行うことは、アイコンとTodoeyタイトルの間に少しスペースを追加することです。

そのため、ここには10ピクセルのギャップがあります。

したがって、列内で作業しているので、それをsizedBoxとして列に追加するのは非常に簡単です。

パディングウィジェットの内部を深くラップすることで混乱する必要はありません。

それで、高さのあるサイズのボックスを追加しましょう。それは、それが私たちが心配している唯一の次元だからです。それは垂直な列の中です

そして今、私たちはユーザーインターフェースアイコンの間に少しのスペース、少し息をする部屋があります。

それでは、テキストウィジェットを見てみましょう。

そのため、Todoeyのことを書いた文字列があります。しかし、テキストのスタイルを変更しましょう。

それでは、colors.whiteの色を持つようにテキストスタイルウィジェットを変更し、他に何があるのでしょうか？

フォントサイズはかなり大きく、フォントの太さも少し大きくなりますよね？

フォントサイズを50に変更します。フォントを50に変更します。フォントの太さも変更します。これを.w700に変更します。太くて太字よりも少し重いです。

だから、大胆で、普通の、軽い等を使用できることを知っています。

しかし、100、200を900まで使用することで、よりきめ細かな制御を行うこともできます。したがって、タイトルに少し太さを加え、非常に重要に見える700を選択しましょう。

だから今、あなたはすべてが正しいけれども私たちのコラムでシフトされていることがわかりますか？

どうしたの？

それでは、Flutterインスペクターの内部を見て、デバッグペイントを表示して、すべてがどこにあるかを見てみましょう。

もちろん、列の背景色を変更したり、コンテナにラップしたり、ファンキーなことをしたりすることもできます。

しかし、これは私たちのレイアウトで何が起こっているのか、そしてWeb Devから来ている人たちを見る最も簡単な方法です。

これは、Chrome開発者ツールまたはFirefox開発者ツールを使用する場合の2番目の性質です。

これは私たちが期待しているものです。

ここに列があり、その中に2つの項目があることがわかります。

1つはサークルアバターで、もう1つはTodoeyのタイトルです。

したがって、列内では、これらの両方がデフォルトで中央に配置されます。

しかし、列は、子供が必要とするだけの幅を占有します。

これがタイトルをするために巨大なものだったとしたら、200に変更すると言うと、足場内にある親の範囲内でできる限り、列が強制的に拡大するのがわかります。

しかし、私たちの場合、実際に子供を変えたくありません。

私たちは、このタイトルの寸法をそれに満足して変更したくありません。

しかし、代わりに、列内のすべてを列の左端に揃えるようにすることです。これは列の開始とも呼ばれます。

それをオフにして、列のプロパティを変更してみましょう。もちろん、これはcrossAxisAlignmentであり、開始を選択します。

それでは、保存をクリックして、列のサイズが変更されていなくても、すべてが列の左側に移動することがわかります。

よし、かっこいい。

それで今比較すると、かなり近づいていますよね？そして、それはかなり似ています。

では、次のテキストを追加しましょう。このテキストウィジェットの下に、別のテキストウィジェットを追加しましょう。これにより、残りのタスクの数が表示されます。

そこで、12タスクのハードコードされた文字列の既定の並べ替えを追加します。次に、最初に色を変更して、このテキストウィジェットのスタイルの一部を変更します。

そのため、これも白にしたいので、フォントサイズを18に変更します。

だからほんの少しだけ大きい。

そして、それは今のところ完璧に動作するはずです。

これで、アプリの上部がレイアウトされました。

次に見るべきことは、この下半分で、角が丸くなったこの白い領域があり、上に座っているちょっとしたリストカードを持っているように見えるので、本当にきれいに見えますリスト名を実行するか、このタイトルを実行します。

では、これをどのように作成できますか？

コンテナを作成できます。

したがって、私たちのコラムでは、現在4つのことがあります。

そして、そのテキストウィジェットの最後に、コンテナを追加してみましょう。そして、コンテナは色を取得します。

そして、それは白い色になります、そう、colors.white。

ただし、ホットリロードすると実際には何も表示されないことに注意してください。これは、コンテナにディメンションを持つ子がある場合にのみ表示されるためです。

この場合、このコンテナにはまったく子がありません。

ただし、いくつかのディメンションを指定することで、強制的に表示させることができます。

300の高さにしましょう。

そして、ここでコンテナを高さで見ることができます。また、カラムの幅全体を占めています。

それでは、この白い正方形をこの素敵な小さな丸いコーナーカードにするにはどうしたらいいでしょうか。そのため、コンテナが持つプロパティの1つは、装飾と呼ばれるものです。

そして、ボックスの外観を操作するためのさまざまな方法を提供するボックス装飾ウィジェットに設定できます。

最初に、色のプロパティをここに移動できます。実際には、ボックスの装飾プロパティとコンテナの色のプロパティの両方に色を設定しようとした場合、「保存」をクリックすると、実際にクラッシュすることに注意してください。

そして、色のプロパティをnullにするか、装飾をnullにする必要があるため、問題があることを教えてくれます。

色の引数（この1つ）は単なる装飾の省略形であるBoxDecoration色であるため、色と装飾の両方を使用することはできません。

したがって、これは基本的に、色付きの新しいボックスデコレーションを作成するための省略形です。

だからそれは排他的です。したがって、どちらか一方しか持てません。

それでは、保存をクリックして、赤いボックスに変わるのを見てみましょう。しかし、白いボックスが必要です。

ボックスの装飾ができたので、次に境界線の半径を指定します。

したがって、境界線の半径により、ボックスは正方形のボックスから角の丸いボックスに変わります。

しかし、四角すべてが丸くなっているのは嫌です。そうすれば、画面上に何らかの大きなアイコンが住んでいるように見えるからです。

代わりに、上部のエッジのみを湾曲させます。

そのため、新しい境界半径ウィジェットを作成し、「only」メソッドを使用して、左上と右上のみを設定します。

したがって、これには半径が必要になるため、radius.circularを使用します。

そして、左上と右上に20ピクセルの境界半径を指定します。これにより、20の別の円形半径になります。

それでは、保存してみましょう。

そして、上部に小さな湾曲したエッジがあり、小さなコンテナに対して完全に正方形または90度の底部があることがわかります。

かっこいいですね。

しかし、現時点では少し切り取られているように見えるので、どうやってそれを一番下まで延ばすことができますか？

1つの方法は、高さを増やすことです。

しかし、高さを大きくしすぎると、画面から消えてしまい、オーバーフローの問題が発生したとしましょう。

そのため、代わりに、ハードコーディングされた高さを削除し、代わりに、その子であるコンテナを展開する拡張ウィジェット内にコンテナをラップして、存在する限り多くのスペースを占有できるようにします。

しかし、私たちのカードには現在、これらのすべての面にパディングがありますので、まだ問題が1つあります。しかし、私たちのモックアップでは、すべてのエッジに至ることがわかります。

どうしてですか？

なぜパディングがあるのですか？

それは、そのコンテナーが列内にあり、列がいくつかのパディング、つまり左に30ピクセル、右に30ピクセル、そして下部に30のコンテナーがあるためです。

そのため、コンテナをこのパディングから解放するには、実際にこの列からコンテナを取り出す必要があります。代わりに、別の列を作成します。この列には、Todoey、タスク、アイコンがあり、内部にネストされます別の列。

そのため、コンテナを別の新しいコラムにラップします。

そのため、この列の内部には、これらすべてを含むコンテナがあります。しかし、白い湾曲したカードを持つ拡張ウィジェットも持っています。

そして今、デバッグペイントを外すと、ここで得たような湾曲したエッジカードである、私が探していたことを達成したことがわかります。

ただし、この列にはデフォルトですべてが中央に配置されているように見えるため、1つの問題があります。これは、この新しい列のクロス原点が中心に設定されているためです。

したがって、この新しい列ですべてを再び左に戻すには、開始するようにクロス軸の配置を設定する必要があります。鮮やかさ。

最終的に望ましいモックアップにかなり近づいています。

あとは、フローティングアクションボタンを追加するだけです。これは非常に簡単です。足場には、背景色と体があり、FABまたはフローティングアクションボタンを追加します。

また、Android StudioでFABと入力するだけで簡単に選択できます。これは、フローティングアクションボタンを書くのが面倒だからです。

多分、彼らはそれをフローティングボタンまたはフローティングサークルと呼ぶべきだと思います。

しかし、とにかくこのフローティングアクションボタンの内側に、色、背景色のプロパティを指定します。これは、淡いブルーのアクセントであるお気に入りの色になります。また、子を持つことになります。記号が追加されるので、別のアイコンになります。

icons.addと呼ばれます。

シンプル。

これで、フローティングアクションボタンとほとんどのユーザーインターフェイスがレイアウトされました。

ここでやらなければならないことは、このコンテナ内に埋め込むことができるスクロールリストを作成する方法を理解することです。そして、そのリストには、タスクの名前とチェックオフできる小さなチェックボックスが付いたリスト項目全体が含まれます。そのすべてについて、次のレッスンで説明します。