大丈夫。先ほど言ったように、最初にやるべきことは、スコアキーパーを構築して、ユーザーが正解した質問の数と、間違った質問の数を追跡することです。

これを行うために、列の最後にコードを配置します。そのため、このコラムには多数の子があります。

いくつかの拡張ウィジェットがあり、1つは質問用で、2つは回答ボタン用です。

ここの一番下に、To Do が表示されます。これが、スコアキーパーのコードがどこに行くべきかを示すために、これらの開始ファイルに入れたものです。あなたが大規模なプロジェクトで作業しているとき、あなたがおそらくやりたい最初のことはそれをより扱いやすい一口サイズのチャンクに分解することなので、dosは本当に役立ちます。そして、最終的なプロジェクトが構築されるまで、それらのそれぞれに1つずつ取り組みます。

したがって、大規模なプロジェクトを一連のタスクに分割するのは非常に簡単です。Android Studioでは、タスクを作成するために、コメントを書くのと同じくらい簡単です。

したがって、2つのスラッシュを入力し、次に実行する単語を作成し、コロンを追加して、その後、自分自身または他の同僚に、協力していること、何をすべきかを伝えます。

そのため、To Doを作成したら、To Doタブ内でそれらを見つけることができます。

あなたはそれをクリックするのであれば、それはあなたのすべてが表示されますに、プロジェクトのドス。

現時点では、Quizzler プロジェクト内に1つしかありません。それを見つける方法を紹介します。

libフォルダー内、main.dart ファイル内にあります。しかし、はるかに簡単な方法は、単にやるべきことをダブルクリックすることです。そうすれば、カーソルが必要な場所までずっと移動します。

まず最初に、これを削除してみましょう。すぐにやるからです。将来の課題では、to doの各ステップであなたに課題を設定します。

これで仕組みが理解できたので、これを使用して、完了に必要なすべての手順を把握することができます。

大丈夫。

そのため、やるべきことを削除したので、スコアキーパーを作成するのに完璧な位置にいます。そして、私たちのスコアキーパーは単に、列に入る行ウィジェットになります。そして、この行には、アイコンウィジェットである子がたくさんあります。

私たちは、アイコンウィジェットを作成しようとしているように、我々はのアイコンと言うつもりだつもりに表示さは材料のアイコンから来ています。そして、私たちがしていることを特定の1 つもりショーはチェックと呼ばれるものです。

そして、あなたはここに小さなプレビューがあるのを見ることができます。

ENTERキーを押すと、ガターにも表示されます。

そして、これは単なる目盛りです。

アイコンの色を緑色に変更して、チェックマークが表示されたときに最後の質問が正しいことをユーザーがさらに確認できるようにします。

ただし、何か問題が発生した場合にアイコンを追加することもできます。

そのため、close というアイコンを追加します。

そして、これはユーザーに最後の質問が間違っていることを伝えるための良い小さなクロスです。

そして、これはもちろん赤くなります。

だから今我々が保存ヒットした場合、緑色のチェックマークや赤い十字を持って、で、私たちのプロジェクトをチェックアウトしたので一瞬、その後、私たちは私たちの小さな行が下部に表示さご覧ください。

これがスコアキーパーになります。

そして、ユーザーが物事を正しくするか、物事を間違えると、ここにアイコンを追加してスコアを追跡します。

この行は、ユーザーがすべての質問を完了するまでこのような新しいアイコンを追加することで構築されます。その後、この行をクリアして、すべてのアイコンウィジェットを削除します。

行と列に複数のウィジェットを含めることができることがわかりました。

たとえば、このコラムでは、現在、3つの拡張ウィジェットと1つの行ウィジェットがあります。この行には、5つのアイコンウィジェットがあります。ただし、セーフエリアやパディングウィジェット、センターウィジェットなど、子を1つだけ持つことができるものもあります。

それらの違いは、列ウィジェットがリストの形式で子を受け入れることができることです。

現在、リストは、データをコレクションにグループ化する方法にすぎません。

これは、さまざまなプログラミング言語で実行できる一般的なことです。

したがって、異なるプログラミング言語を使用している場合は、リストを配列として知っているかもしれません。しかし、両方で例、それはだ、私たちは1つのリストにデータのグループに関連する作品のことができるようにするための方法。

その後、リストを操作したり、そのコンテンツをタップしたりできます。

次に、空のリストを作成して、ここでそれを実行します。Dartでリストを作成するために、キーワードリストを作成します。

そして、これは私たちが後で作成しているタイプのリストであることを行っていること、およびそのどんな変数と言い、その後、我々はこの呼びますので、それに名前を付けるつもりだ記録係は、それはだつもり等しくなるように設定され、空のリストを。

これで、コレクションを操作しているときに、これらの角かっこがよく表示されます。

そしてこの場合、これらの空の角括弧が表すものは空のリストです。

現在、リスト内にアイテムはありません。ただし、すべての行と列に対して行ったのと同じ方法でリストを作成するときに、リストにアイテムを追加することもできます。

したがって、この行が多数の子ウィジェットを受け入れると言ったとき、開き角括弧と閉じ角括弧があり、その間にはこの特定のリストのすべての項目があります。

これで、scoreKeeperでも同じことができます。

これらのアイコンのいくつかを移植します。ここに単純に切り取って貼り付けます。

これで、scoreKeeper というリストがあり、5つのアイコンが含まれています。

後でリストを使用することに決めた場合は、名前でリストを参照するだけです。

したがって、このchildrenプロパティの右側のすべてを削除します。

代わりに、ここで作成したスコアキーパーと呼ばれるリストを参照します。ので、今、この場合には、記録係は、リストのデータ型を持っているとの子プロパティ行に必要なリストを、その後、これらの二つの側面が左側と右側を左に、なぜこの作品である互換性があります。

この場合、実際にはウィジェットのリストが必要ですが、ここでは、ストアキーパーは動的データ型のデータのリストです。

さっきも言ったように、動的なデータ型を使用していて、多くのことが起こっていると、非常に簡単に毛深い状況に陥ることがあります。

したがって、最良の方法は、作成するリストのタイプを実際に明示することです。

そしてそれによって、どの種類のデータが実際にリストに入ることができるかを意味します。

たとえば、文字列のリストを作成する場合、それを文字列と呼ぶことができます。

そして、それを作成すると、a、b、c などの文字列が含まれます。

さて、この場合、私はこの特定のリストが呼び出されるように指定することができること方法myStrings のみ受け付けることができ内部の文字列のリストは、私は単に単語リストの後にアングルブラケットを追加することによってそれを行うことができ、およびデータ型指定リストに入ると。

この場合、このようになります。

これで、たとえば、リストを作成して、ここに数字を入力できるようになりました。というか、int 型をString型のリストに割り当てることができないと言っているからです。

したがって、これによりリストがより安全になり、間違ったデータ型を誤って入力することがなくなります。

ここまでで、同じことです。

ウィジェットを含むリストのみが必要であることを指定したい場合は、リストの直後にある山括弧内に「widget」という単語を追加するだけです。

この場合、このリスト内にあるものはすべてアイコンであることがわかっているため、これはアイコンのリストであるとも言えます。アイコンは実際にはウィジェットであるため、これはここでも機能します。

そのため、この特定の行がウィジェットを持つリストしか取得できない場合でも、互換性があります。

代わりに、ウィジェットではない文字列のリストがある場合、ここで非互換性エラーが発生します。文字列のリストをウィジェットのリストに割り当てることはできませんが、アイコンのリストは割り当てることができるためです。

それを元に戻しましょう。そのため、保存をクリックしてアプリをチェックアウトすると、ほとんど何も変わっていないことがわかります。ここにある私の行は、私の行内に直接作成するのではなく、scoreKeeper というリストを使用して作成されます。アイコンを取り出して別のリストに入れることの利点は、アイコンをその場でアイコンのリストに追加できることを意味します。

したがって、たとえば、私は真のユーザーがボタンを押すが、その後、内部のときと言うことができonPressed 、私はそれがアイコンの私のリストにアイコンを追加するために取得するつもり。

まず、scoreKeeper リストをタップする必要があります。次に、.add と呼ばれるものを使用します。これは、括弧内の値をリストの最後に追加します。そして、リストの長さは1ずつ増えます。

そのため、scorekeeper.add と言いますが、追加したいのは新しいアイコンです。

スコアキーパーにはアイコンのみを含めることができます。

そこで、新しいアイコンを作成します。このアイコンはチェック記号になります。そして、緑色になります。

アイコンを作成し、scoreKeeperに追加したので、ウィジェットツリーの再構築をトリガーするように設定状態に入れるだけです。

そして、その新しいアイコンがスコアキーパーの行内に表示されます。

それでは、saveをクリックしてコードを更新しましょう。そして、真のボタンをクリックすると、小さな緑色のチェックマークが私の行に追加されるのを見ることができます。Dartリストの力により、これを行うことができました。

このリストの概念が初めての場合は、次のレッスンに進むことをお勧めします。このレッスンでは、Dartリストについて詳しく説明し、その仕組みについてさらに学習します。

リストまたは配列がどのように機能するかを既によく知っていて、この概念が新しいものではない場合は、次のレッスンをスキップして、その後レッスンに進んでアプリを作成し続けてください。