最後のレッスンでは、Dart列挙型についてすべて学びます。そして、私たちの列挙型はコードをずっと読みやすくしており、それがずっと意味をなし始めています。

しかし、いくつかの機能を実現するための非常に多くのコードがまだありますが、実際には非常にシンプルで、カードの色を切り替えます。

なぜそんなに多くのコードが必要なのですか？コードの量を減らしながら同じロジックと動作を維持するには、Dartが提供する他の何か（三項演算子）について学ぶ必要があります。

これまで、IF、ELSE IF、ELSEステートメントを使用して、条件が何であるかに応じて異なる動作を実行できることがわかりました。

そして、私たちはそれを鉄道線路の観点から話しましたか？現在プレイ中の条件に応じて、どの方向に進むかを決定します。

現在、Dartの三項演算子は、実際には非常に長くて非常に冗長なものを作成するための構文糖と呼ばれるもののほんの一部です。

そして、多くの括弧、多くのキーワード、そしてこれを達成するための多くのコード行があります。しかし、代わりにDart三項演算子を使用した場合、条件をチェックする1行に単純化し、その条件が真であれば、これを実行します。

そして、条件が偽であれば、これを実行します。ここで必要なのは、疑問符とコロンだけです。それが少し魔法のように思えても心配しないでください。信頼できるDartPadに行き、肉付けしてどのように機能するかを確認します。

したがって、メイン関数に新しい変数があり、ブール型であるため、trueまたはfalseにできるとしましょう。

そして、名前を付けましょう。

jackBauerIsAwesomeを確認するために追跡しようとしているとしましょう。これは明らかに真実です。

だから誰かがショー24を見ていないなら、私を無視してください。

これが通常のブール変数であるふりをしてください。したがって、必要に応じて、これを別の変数に変更することができます（durianIsAwesomeなど）。

本当に問題ではありません。

これは、単にtrueまたはfalseにできる変数です。

プログラムのある時点で、これがfalseに変更されるか、trueに変更される可能性があるため、IFステートメントを使用して確認する必要があります。

jackBauerIsAwesomeの場合、trueに等しい場合、印刷したい場合があります。

'それは素晴らしいことです。'

または、そうでない場合は、印刷したいかもしれません

'大野。'

そして、必要な場所にすべての引用符があることを確認し、そこで引用符をエスケープします。

そのため、今すぐコードを実行すると、jackBauerIsAwesomeがtrueに等しいため、素晴らしい結果が出力されます。

そして、実際にそうであるかどうかを確認します。

そして、これらの括弧に分類され、「それは素晴らしい」と表示されます。

したがって、これは古典的なIF ELSEステートメントの動作です。これで、三項演算子を使用するだけで、これらのコード行をすべて1行に削減できます。そしてそれを行うために、最初に来るのはチェックする条件です。

だから、jackBauerIsAwesomeは等しいと言っているので、double equalsは、左側が右側と等しいかどうかを確認するためのチェックを記憶しています。

そして、これが本当なら、私たちがやることは素晴らしい印刷です。

そして、疑問符を追加して、これが確認対象であることを示します。

それが本当なら、それを印刷します。

しかし、それが偽の場合は、コロンを追加して、これが偽だった場合に行うべきことを示します。

それで、印刷します。このコード行は、これらすべてのコード行とまったく同じことを行います。

そして、私はそれをコメントアウトし、実行をクリックして、まさにそれが何が起こるかを確認することができます。

これをfalseに変更すると、この部分がトリガーされ、「Oh no」と表示されます。かなりクールだよね？

多くのコードを削減します。

これを初めて目にするのは初めてですが、これはすべてのレイアウトほどレイアウトされていないため、少し混乱する可能性があります。また、コンピューターは疑問符の前のすべての部分を検索して、それがtrueに等しいかどうかを確認するため、この行をさらに削減することもできます。

したがって、実際にtrueに等しいかどうかを確認する必要はありません。

IFステートメントのここと同じです。

実際には、その等価性チェックは必要ありません。これがtrueの値を持っている限り、この部分をトリガーしますが、falseの値を持っている場合、この部分をトリガーします。今では、コードの行数を減らし、物理的に可能な限り文字数を短くすることが唯一の目標であるコードゴルフをすることを常に避けたいと思っています。

これは実際のゲームであり、コードゴルフですが、それは実際の目標ではありません。

ほとんどの場合、コードを簡潔にしたいが、それでも表現力があります。

それは実際に何が起こっているかをまだ示しています。

ですから、コードの行を切り詰めるのか、それともそれを切り詰めて読みやすく、読みやすく、混乱を少なくできるのかを選択するときはいつでも、それが常に選択したいものです。

したがって、コードの行数を削減するためだけに行っているわけではありません。

ここでできることは、三項演算子を使用してIFステートメントで使用できるものと同じ比較を使用することです。

したがって、myAgeという整数を作成し、12であるとしましょう。

私は12歳ではないので心配しないでください。

大丈夫だよ。ただし、canBuyAlcoholというブール値を作成し、年齢の値に応じてtrueまたはfalseの値を割り当てたい場合は、右側の三項演算子も使用できます。

したがって、myAgeが21より大きい場合、次のように言うことができますか？

したがって、この条件が真であるかどうかを確認します。

その場合は、canBuyAcoholをtrueに設定します。

ただし、そうでない場合は、falseに設定します。これは、IFおよびELSEステートメントではできないことです。等号の右側に複数行のコードを置くことはできません。

しかし、三項演算子は単なる1行の式であるため、この場合は可能です。

したがって、canBuyAlcoholを印刷すると、年齢が12歳のときに偽になることがわかります。これを少し大きくすると、真実になることがわかります。

したがって、この条件に応じて、基本的にこの値をこの変数に割り当てます。

そして、三項演算子を使用してこれをすべて行っています。

そのため、ほとんどの場合、これが発生します。

したがって、これは何かをチェックする必要があるときに非常に便利であり、そのチェックに応じて、変数に異なる値を与えて作業します。コードに戻りましょうそして、この挑戦​​は少し難しいですが、基本的に、三項演算子について学んだことを適用してほしいです。そして、この関数を取り除いて欲しいです。代わりに、三項演算子を使用して、ここで色がある場所を確認し、男性カードが選択されているかどうかを確認し、アクティブな色を与える必要がありますが、女性のカードが選択されている場合は、色が非アクティブな色で、ここでは反対になります。

そのため、現在の機能とまったく同じ機能を備えている必要がありますが、この機能は不要です。そして、設定状態を少し変更する必要があるかもしれないカスタム変数を追加する必要があるかもしれません。また、ここで三値演算子を使用して、値の色を確認する必要があります。

ですから、少し難しい挑戦ですが、あなたならできると思います。ビデオを一時停止して、試してみてください。

大丈夫。

まず最初に。

この関数を削除すると言いました。

とりあえずコメントアウトします。これは実際には必要ありませんが、代わりに性別を保持する新しい変数を作成し、それをselectedGenderと呼びます。

そもそも性別が選択されていないため、最初はnullになります。

しかし、男性カードが押されたとき、または女性カードが押されたとき、選択した性別の値を変更します。このカードが選択されると、選択されたアジェンダはgender.maleと等しくなります。

このカードが選択されると、選択された性別はgender.femaleになります。そして、私たちが持っている列挙型のおかげで、この表現力豊かで本当に明確なコードを書くことができます。

そして、これが、整数や任意の何かではなく、性別の値を保持するこのプロパティを持つことができる方法です。

選択した性別を設定し、selectGenderという変数を使用してその値を追跡できるようになったので、カードの色を変更する方法に対処できます。

したがって、条件を評価して値を割り当てる方法として三項演算子を使用できることを知っています。

この場合、男性カードの場合、selectedGenderがgender.maleと等しいかどうかを確認できます。その場合、reusableCardの色はactiveCardColorになります。

ただし、これが当てはまらない場合、つまりselectedGenderとmaleがfalseの場合は、inactiveCardColorに変更する必要があります。そして、私たちの女性カードの再利用可能なカードでは、逆のことが当てはまります。

したがって、selectedGenderがfemaleに等しいかどうかを確認します。次に、この場合、この再利用可能なカードをactiveCardColorの色に変更しますが、男性の性別が選択されていてこれがfalseの場合は、inactiveCardColorに変更します。

これは、アプリが最初に起動したときにもメリットがあり、性別には価値がありません。

したがって、これらのステートメントは両方とも偽と評価されます。 selectedGenderは男性と等しくなりません。また、女性と等しくなることもありません。

したがって、両方のカードはinactiveCardColorで始まります。これはまさに今私たちが望むものです。

先に進み、このコードをすべて削除しますが、ここで、これらの数行のコードに置き換えられます。

超簡単、超簡単。

そして、式を評価し、その条件に応じて異なる色の値を同じ行にすべて割り当てることができるので、これをすべて行うことができますIFとELSEに依存します。先に進んでテストしてみましょう。

ホットリスタートをクリックして、アプリの状態を最初の状態にリセットします。

つまり、この性別であるselectedGenderはnullから始まり、両方のカードが非アクティブになることを意味します。しかし、カードの1つを選択するとすぐに、アクティブな色になります。

そして、反対のカードを選択すると、そのカードは無効になります。

したがって、まったく同じ機能があり、わずか5行のコードを使用して作成されています。