大丈夫。

したがって、このレッスンでは、Dartに不慣れな皆さんのために、Dartの条件をすぐに取り上げたいと思います。

より具体的には、IFおよびELSE。現在、IFステートメントは基本的に、条件が真であるかどうかを確認するだけです。そして、条件が真の場合、中括弧のセット内で命令を実行します。

したがって、この場合、IFステートメントは、トラックが明確な場合、二重等しいとは、左側が右側と等しいかどうかを評価することを意味しますが、単一の等しいは割り当てです。

つまり、左側と右側を等しくすることを意味します。これは、コーディングの際に注意すべき微妙な違いです。

この場合、英語で言っているのは、トラックがクリアである場合は行ってください。

ステートメントを調べてみると、これが条件の場所であり、これが命令です。

そのため、コンピューターが声明を評価し、この場合、トラックが実際に明確になったとき、それはただ行きます。

しかし、これを評価したときに実際には正しくない場合、トラックは明確ではありません。この場合、巨大な岩があり、中括弧内の指示は実行されません。

代わりに、このすべてをスキップして、実行する必要があるコードの次の行に進みます。

したがって、この場合、列車はどこにも行きません。前回のレッスンでは、単純なIFを使用して条件が満たされたときにステートメントをトリガーするだけでなく、IFおよびELSEも使用できることを確認しました。

線路が空いている場合、列車は直進する必要があると言っているとしましょう。

この場合、コードは次のようになります。テストする条件は、トラックが明確であるかどうかです。

そしてその場合、この中括弧のセットの間にあるすべての命令を実行する必要があります。まっすぐ行きます。

しかし、この条件が当てはまらない場合は、elseステートメントに進み、代わりにこの中括弧のセットの内容を実行します。より一般的には、IFステートメントとELSEステートメントが次のように構造化されており、条件があり、次に中かっこのセットがあり、その条件がfalseの場合は他のすべての処理が行われます。これが前回のレッスンでも行ったことです。

プログラマーと一緒に住んでいる場合、それは本当に危険であり、物事をどう表現するかに非常に注意する必要があります。

たとえば、プログラマーである夫または妻に「卵を6個買ったらスーパーに行って牛乳1本を買ってください」と言った場合。

あなたが言っていることは、牛乳1本を購入することであり、卵が6個ある場合です。

しかし、彼らが聞くかもしれないのは、haveEggs = trueの場合、6本の牛乳を購入し、そうでなければ1本の牛乳を購入することです。

プログラマと一緒に暮らすすべての人に大声で叫ぶ。

このコードが実際にどのように機能するかを示すために、愛の計算機を作成します。

だから、80年代からの先祖返りの1つで、あなたが好きな人との互換性をテストします。

ここで、愛の計算機を本当にシンプルに保ちます。

loveCalculatorと呼ばれる関数を作成します。

そして、単に1から100の間の乱数を生成するだけです。

そのため、これまでのレッスンで既にその方法を知っています。最初にDart数学ライブラリをインポートする必要があります。

数学ではなく数学であることを忘れないでください。

そして、インポートしたら、乱数ジェネレーターを使用できます。したがって、loveScoreという新しい変数を作成できます。これは、整数を生成しているため、乱数ジェネレーターの.nextIntと等しくなります。

そして、100にします。

したがって、0から99までのすべての数値が生成されます。その後、1を追加して範囲を1から100に変更します。

乱数loveScoreを生成できるようになったので、愛の計算機を本当に信じているすべての人に謝罪します。

ただし、この場合は少しランダムになります。

loveScoreが生成されたので、それを出力して、得られるものを見てみましょう。

したがって、ここでメイン関数で関数を呼び出し、愛の計算機を呼び出し、コードを実行すると、コンソールに結果が表示されます。

したがって、29％を初めて取得したときは18％でしたが、引き続き22％でした。

今日は愛の日ではないいくよ

96％を得ました。ついに高得点を得ました。そのため、1〜100の乱数がたくさんあることがわかります。

この数値に基づいて、ユーザーに愛のスコアの解釈を提供したい場合は、IFステートメントを使用できます。だから、愛のスコアが大きい場合は70としましょう。そうすれば、彼らはお互いが本当に好きなのかもしれません。たぶん彼らは...あなたはカニエがカニエを愛しているようにお互いを愛している。

そして、彼らが70より低いスコアを得た場合、それはあなたがお互いが欲しいと言うだけです。

大丈夫。

だから、今、私たちの愛のスコアに応じて、コンソールに別の何かを印刷する必要があります。

そして、実行すると、最初に30を取得したことがわかります。したがって、それぞれが好きなテキストを取得できます...あなたはお互いが好きです。

どうしてそんなにたくさんの手紙を逃したのか分かりません。

そして、もう一度実行すると、9が得られます。そして、最終的にはうまくいけば...ああすごい！

私たちは実際に100を得ました。

ジャックポット！

カニエがカニエを愛するように、あなたはお互いを愛しています。したがって、ランダムジェネレーターから取得したラブスコアに応じて、コンソールに異なるテキストを印刷できることがわかります。

そして、これはすべてIFステートメントによるものです。

現時点では、70を超えるまたは70未満の異なる解釈しかできないため、複数のブラケットが必要な場合はどうでしょうか。

もっとニュアンスにしたい場合はどうしますか？

70〜100と50〜70と0〜50が必要な場合はどうでしょうか。

どうすればいいですか？

IFステートメントを使用しているときは、使用できる演算子もたくさんあります。

たとえば、左辺が右辺と等しいかどうかを確認するためにdouble equalsを使用できることを既に見ました。

しかし、感嘆符、等号を使用して、等しくないかどうかを確認することもできます。または、左側が右側よりも大きい場合、より小さい、より大きい、または等しい、またはより小さいまたは等しい。

そして、これらの比較を使用して、IFステートメントでこれらの条件を組み合わせることができます。

したがって、たとえばANDには2つの&&を、2つの||を使用できます。 ORの場合、およびNOTの場合は単一の感嘆符。それで、これがどのように機能するかをお見せしましょう。 loveScoreが70を超える場合のIFステートメントがある場合、これを実行します。ここに別のIFステートメントを置くこともできます。これで、愛のスコアが50を超えており、愛のスコアが70未満であるかどうかを確認したい場合、「You like each other」と印刷します。しかし、愛のスコアが50未満の場合、私たちはあなたがお互いを好きではないことを印刷します。大丈夫。

そのため、ここで、位置引数が多すぎるというエラーが表示されます。

1つは予想されていましたが、5つ見つかりました。

これは何ですか？

単一引用符について言ったことをよく覚えておいてください。

これは私たちが印刷したいものであると考えており、最後にこのコードがすべてある理由はわかりません。

そのため、文字列内で一重引用符を使用する場合は常に、バックスラッシュを使用してエスケープする必要があります。

「これをコードとして見ないでください」と言ってください。

これは英語の文法で使用するものであり、コードの一部ではありません。」

ここで、コードを実行してみましょう。71を取得すると、70を超えると、KanyeがKanyeを愛しているように、お互いを愛していることがわかります。

しかし、異なる数、たとえば50未満の8を取得した場合は、お互いが好きではないことがわかります。

そのため、これらの演算子を使用し、さまざまな条件を組み合わせて複数のことを確認できます。

したがって、ラブスコアが50を超えているかどうか、およびラブスコアが70未満かどうかを確認できます。または、ラブスコアが50を超えているか、ラブスコアが70未満かどうかを確認できます。私たちの場合、これは、これも真になる場合に、これが真になることを意味するためです。

したがって、22の場合、70未満であり、これはORであるため、これがトリガーされ、22も50未満になります。

したがって、これもトリガーされます。

これは機能しますが、さまざまなものを組み合わせてこれらの方法を使用してステートメントをより具体的にすることができますが、これを行うはるかに良い方法があります。

IFおよびELSEステートメントのラダーを構築し始めることができます。

これがどのように機能するかです。

トラック1（この1つ）が明確かどうかを確認できます。

それから私たちは列車を追跡するために行くつもりです。

これが最初の選択です。しかし、それが間違っていた場合、そのためそこに岩があった場合、次のステートメントに進むことができます。そして、ELSEIFトラック2がクリアであると言っているので、トラック2に進む必要があります。しかし、track2にもロックが付いている場合はどうでしょうか？

どこに岩があるのか​​、誰がこれらの鉄道を維持しているのかわかりません。

track2もブロックされている場合、track3がクリアされているかどうかを確認するために別のElse Ifステートメントを使用できます。

これで、コードは次のようになります。

したがって、愛のスコアが70より大きい場合、Aを実行できます。

しかし、そうでない場合は、2番目の選択肢に進みます。

そうでなければ、愛のスコアが30を超える場合、Bを行います。

最後に、それも偽の場合は、Cを実行します。

したがって、50のLoveスコアを取得した場合、それは明らかに70以下であるため、Aをスキップします。その後、次のステートメントをチェックします。

愛のスコアが30より大きいかどうかを確認します。

だから、それは先に進み、Bを行います

しかし、愛のスコアが75だった場合、最初の声明はすでに真実です。

そのため、この場合は単純にAを実行します。

これがELSE IFではなく、単なるifである場合、これも真実であることに注意してください。

75も30より大きいです。

しかし、これはELSE IFであるため発生しません。したがって、IFステートメントの後にELSE IFがあり、その後に他のELSE IFまたはELSEが続く場合、条件の1つがtrueであることが検出されると、他のすべての条件がスキップされます。

ただし、これらのすべてがIFステートメントである場合は、それらを1つずつ確認します。

したがって、愛のスコアが70を超えていて、愛のスコアが30を超えていた場合、最初の条件が評価されます。

だから、それは70以上です。

Aにします。そして、これがIFステートメントの終わりであることがわかり、別のIFステートメントがあります。

まあ、これも本当です。

75も30より大きいです。

それで、代わりにBに行きます。

コードをこのように動作させたい場合もありますが、代わりにELSE IFを使用したい場合もあります。

したがって、コードに戻って、実際に変更することができるので、比較する必要がなく、複数の条件を持つ必要さえありません。愛のスコアが50より大きく70より小さいかどうかを確認する代わりに、これをELSE IFに変更できます。そして、愛のスコアが50を超える場合は、単にelseを書くことができます。最後のスコアはifを必要とせず、単にelseになります。

実行をクリックして、別の新しい乱数を生成し、35のスコアを取得すると、最初に起こるのは、愛のスコアが70より大きいかどうかを確認することです。35は70以下です、したがって、これはスキップされ、次のものに進みます。

35が50より大きいかどうかを確認するようになりました。これも事実ではありません。

したがって、このブロックはスキップされます。

最後に、これはelseブロックであるため、上記のすべてが当てはまらない場合は、ここで指示を実行します。これは、「お互いが好きではない」と印刷することです。

もう一度試してみると、今回は94になり、最初のIFステートメントをチェックすると、それはすでに真であり、この関数を実行し、他のすべてをスキップします。

また、50より大きいかどうかを確認するために行きません。

しかし、もしこれを別のものに変更した場合、クリックして実行し、75のような多数を取得したら、これは事実であるため、これを実行します。しかし、これも真実です。

これが、IFとELSE IFの違いです。そして、実際にELSE IFステートメントを作成して、それらをたくさん持つことができます。

たとえば、愛のスコアが30より大きい場合、別のメッセージを使用できます。この場合、30〜50の別のメッセージが必要な場合があります。そして、50から70については別のメッセージがあり、ここでは70から100については別のメッセージがあります。

したがって、このコードをいじって、IFステートメントの束全体とELSE IFステートメントのラダーの違いと、これらのタイプの条件ステートメントがDartでどのように機能するかを十分に理解してください。完了したら、Quizzlerアプリの作成に戻ります。

それでは、次のレッスンでお会いしましょう。