大丈夫。

したがって、前回のレッスンでは、各通貨の値を持つテキストウィジェットを囲むドロップダウンメニュー項目ウィジェットを作成する方法を説明しました。

そして、私たちはそれらの多くを持っていることを知っています。

したがって、これを手作業で行うことは、怠け者のプログラマーにとっておそらく選択肢ではありません。しかし、幸いなことに、ほとんどすべてのプログラミング言語にループと呼ばれるこの構造があります。

Dartの機能について説明したときのことを思い出してください。buyMilkと呼ばれるこの機能は、小さな家事用ロボットによって実行され、毎日外出して牛乳を買ってから戻ってきます。

週に7日それをしなければならないことを想像してみてください。そして、毎日牛乳を買うようにロボットに伝えることを忘れないでください。

したがって、牛乳を購入してから、もう一度牛乳を購入し、週に7回、または何回も実行する必要があることがわかっていても、この関数を実行する必要があるたびに何度も何度も呼び出します。

これを行うより良い方法があります。いくつかの機能またはコードの一部を複数回実行する必要があるときはいつでも、ループを使用して実行できます。

そして、最も便利なタイプのループの1つはFORループです。現在、FORループにはさまざまな種類があります。

そして、C言語から来る最も一般的な型はこのようなものです。

その後に来るのはFORループであると言うキーワードforがあり、その後、通常iと呼ばれる慣習により新しい整数を作成します。 0の開始値に等しくなるように設定し、セミコロンを設定して、ループの実行時間を定義します。

したがって、私は7歳未満ですが、ループを実行し続けます。そして、2番目のセミコロンの後、各ループの後にiを増やす方法を指定します。

最後に、中括弧の内側で、7〜0回の処理を行います。したがって、これが開始値であり、これが終了値であるかのように見ることができます。これが、ゼロから7に進む方法です。

したがって、1ずつ増やしていきます。

ゼロから始めて、コードを1ずつ増やしてからコードを実行し、コードを1ずつ増やしてから、iが6になるまでコードを実行します。しかし、コードを何度も実行するだけでなく、実際に行うことができます。私との何か。

だから、これを使用することができます。このiは、ゼロに等しくなり、次に1および2などになります。

FORループ内で使用して、コードに影響を与えることができます。

さて、これはどういう意味ですか？

具体的に見てみましょう。

メイン関数内のDartPad内で、今紹介したFORループの1つを作成します。

そして、新しい整数を作成することから始めます。そのため、iという名前の変数

ゼロに等しくなることから始めます。そして、セミコロンを追加し、10歳未満の間はコードを実行し続け、コードを実行するたびにiに1を追加すると言います。

したがって、このFORループの中括弧内で、ここにあるコードすべてが10-0回実行されます。何かを印刷しましょう、たぶん印刷しましょう

こんにちは。実行をクリックすると、コンソールに10回印刷されます。そして、これを実行する回数は、これらの括弧内にあるすべてによって決まります。

したがって、helloを出力する代わりに、iの実際の値を出力すると、何が起こっているのかを少し裏で見ることができます。

もう一度runを押してみましょう。FORループが初めて実行されたとき、それがゼロに等しいことがわかりました。

だから、私がiを印刷したとき、それはコンソールにゼロを印刷しました。

しかし、それが実行された後、再びチェックするために戻ってきます。

さて、私はまだ10歳未満ですか？

0は10より小さいので、1を追加して、もう一度実行して、この出力を実行します。そして、私が9に等しくなるまでこれを続けます。

したがって、今回はiが9に等しくなり、そのチェックを再度行います。

さて、私は10歳未満ですか？

ウェル9は10未満なので、もう一度実行してみましょう。

そして、このコードが終了するたびに1を追加するため、10に等しくなります。

だから今私は10に等しいし、最終的にそのチェックを行います。 10は10未満ですか？いいえ、10は10です。

したがって、これはもはや真実ではなく、FORループは終了します。したがって、このコードはもう実行しません。

そのため、ループの実行方法を変更する方法はいくつかあります。

そして、最も簡単な方法の1つは、実際に増分方法を変更することです。

ループが実行されるたびにiに1つずつ追加する代わりに、2を追加するとどうなりますか？

プラス2を使用しましょう。

だから今、私は毎回2ずつ増加するつもりです。

[実行]をクリックすると、ゼロから開始します。これは、開始するべきだと言ったからです。

しかし今では、ループが実行されるたびに、ループに追加します。

したがって、10に到達するまで0から2から4から6から8になり、10未満になり、ループが停止します。ここで、FORループについて変更できる他のことは、開始値です。

したがって、2の値から開始し、毎回1ずつ増加したとしましょう。

さて、これを印刷すると、0から9ではなく、2から9になります。

コードの終了時に変更することもできます。

したがって、iが10未満になる代わりに、iが15未満になる可能性があります。次に、コードは2から14に達するまで実行されます。14に達すると、iが15に等しくなる前の最後の値になります。

これを実際の例で見てみましょう。

たとえば、buyMilkと呼ばれるこの関数があり、牛乳を購入する日数を指定できたとします。

牛乳を7日間買うのではないでしょうか？

それでは、この関数をここに書きましょう。

したがって、buyMilkと呼ばれ、牛乳を購入する日数を考慮した整数を指定できます。さて、この関数内で私がしたいことは本当に簡単なものです。

私のロボットは、1日目、2日目、3日目、4日目に牛乳を買うと言って、実際に牛乳を買っている日を表示します。

そして、この入力で指定した回数だけこれを実行したいです。

そのためには、FORループを使用できます。 int iが等しいと言えば、1日目から始めましょう。

だから私は1に等しいです。そして、iがここでの入力で指定した日数以下である間、コードを実行し続けたいと思います。そして、我々は増加するつもりです

私は毎回一つずつ。

そこで、このFORループ内でprintステートメントを囲み、ここに出力されるiの値を使用します。

コードを実行すると、関数buyMilkへの入力として7を入力していることがわかります。

7がここに入り、ここで指定した回数だけFORループを実行します。

だから私は、1の値から始めて、iが7以下になるまでここでコードを繰り返して、1が7になるまで続けるように言った。そして、このコードが繰り返されるたびに、iの値を1。

そして、私のコードが実行されるたびに、i日目に牛乳の購入を印刷するので、1日目、2日目、3日目まで7になります。

これで、基本的なCスタイルのFORループがどのように機能するかを確認したので、次は挑戦の時です。

99-bottlesof-beer.netと呼ばれる非常に古い学校のようなWebサイトがあり、ここに99本のビールの歌の歌詞があります。

そのため、曲の歌詞をクリックすると、この曲がどのように進行するかを確認できます。

それは壁に99本のビール、99本のビールから始まります。

1つを下ろして渡します。

壁に98本のビール。

そして、歌詞が繰り返されるたびに、その数字は下がります。

したがって、壁にビールのボトルがなく、ビールのボトルがなくなるまで、99から98 t 9に移動します。

店に行って、壁にさらに99本のビールを買います。

このウェブサイトは、この曲の歌詞を印刷できるようにプログラムを記述したさまざまな言語を集めており、言語別にソートされているため、ユニークです。

つまり、aからzです。

たとえば、Javaのコードを見ると、ここで投票した人に基づいてスコアが評価されていることがわかります。そして、あなたは彼らのプログラムがここで書かれているのを見ることができ、人々が提出した様々な異なる代替バージョンがあります。

挑戦として、この歌の簡単なバージョンを印刷してほしい。

したがって、これが開始ベースラインになり、この数が毎回1ずつ減少する必要があります。

この数は1減少し、この数は1減少します。

ですから、ビールのボトルはもうないという最後の行を持つ必要はありません。

コードはこの行で停止する必要があります。この行では、壁にビール1本、ビール1本しか入っていません。

したがって、FORループに基づいて学習したことを試して、これらの歌詞をコンソールに出力してみてください。ビデオを一時停止して、このチャレンジを完了してください。

大丈夫。

そのため、このためのプログラムを可能な限り簡単な方法で作成してみます。

それで、私が知っているコードの最初の行をほぼ100回繰り返すことをコピーします。そして、それをDartPadに入れます。

これが私の歌詞の最初のベースラインになります。この曲で変わるものは、この99、99、98であることがわかります。

したがって、FORループを作成して、ループを数回ループする必要があります。この場合、99に等しい整数iから再び始めます。そして、ゼロよりも大きい限り、ループを実行します。

最後に、追加せずに変更しますが、実際には毎回1ずつ減らします。このFORループ内で、iという名前のものにアクセスできるようになりました。

そして今のところ歌詞に来たら、毎回Iを印刷してみよう。

それでは、走りましょう。

そして、あなたは私のコンソールの中でそれを見ることができます、

私は99から1まで印刷していますが、これが必要です。今、私を印刷する代わりに、

歌詞を印刷します。

そこで、歌詞のコメントを外し、カットして、ここに文字列として挿入します。単一引用符で囲みます。また、コピーして貼り付けると、実際には2行になります。

したがって、これを1行のコードと見なす必要があります。

そして今、変数iを99に置き換えることができるので、それはiになります、それはiになります、これはi-1になります。

したがって、ドル記号を使用する必要があります。次に、中かっこを使用し、中かっこ内にi-1と言うことができます。

したがって、コードを実行すると、壁に99本のビール、99本のビールが入っていることがわかります。

ダウンパスを1周し、98本のビールを壁に置きます。そして、壁にビール1本が届くまで繰り返します。ビール1本。

1杯下ろし、壁に0本のビールを渡します。

ボトルの単一バージョンがボトルであるとは考えていないため、完全な歌詞がないため、これは完全ではありません。

そのため、1つに達すると、おそらくボトルに変わるはずです。そして、それが0に達すると、おそらくビールのボトルはもうないはずです。

また、印刷する必要のある最終行もあります。

そのため、挑戦が必要な場合は、ここで見たのとまったく同じ歌詞を実際に印刷するコードを試してみてください。

しかし、ここで学びたい基本的なことは、コードを繰り返す回数を決定するために本当に柔軟にforループを使用し、printステートメント内での使用やたとえば、リストのようなもののインデックスとして。 FORループについて話している間、別のタイプのFORループを示したいと思います。これはfor inループと呼ばれます。これは、反復可能なすべてのアイテムをループする場合に非常に役立ちます。リスト。

この場合、果物のリストがあり、そのリスト内のすべての値をループしてから、そのリスト内の各果物で何かをしたいとしましょう。

したがって、for inループを使用すると、次のようになります。

そのため、そのiを使用してループを繰り返し、ループの開始時、終了時、および反復方法を決定する代わりに、リストに含まれる型の新しい変数を作成します。

たとえば、この場合、fruitsという文字列のリストがあります。

そして、果物の各果物について、その可変の果物で何かをすることができます。

そして、リスト内のすべてのアイテムに対してこのループを実行します。

したがって、アイテムが5つある場合、5回実行されます。そこで再びDArtPadに進み、ここでは果物と呼ばれる文字列のリストを作成しました。これは、リンゴ、ナシ、オレンジ、ブドウ、バナナの5つの果物を含む新しいリストに等しくなるように設定されています。

多分pieMakerと呼ばれる関数を作成したいとしましょう。それはそれらの果物のそれぞれを取得し、それにpieという単語を追加します。

だから、アップルパイ、洋ナシパイ、誰も作らないと確信しているオレンジパイ、グレープパイ、バナナパイを作りたいと思っています。そして、それらのそれぞれをコンソールに印刷したいと思います。

私ができることは、単にforインループを使用して、かっこ内でキーワードforを言うことができることです。

リスト内の各アイテムを表す新しい文字列を作成します。

これは文字列のリストであるため、リスト内の単一の項目を表す新しい文字列を作成するのはこのためです。そのため、文字列を「フルーツ」と呼びます。

単数形のフルーツ。そして、果物リスト内のすべての果物について、その果物のリストで何かをするつもりです。単一のアイテムであるフルーツを印刷し、

パイという言葉を追加します。

したがって、メイン関数内でメソッドパイメーカーを呼び出してrunを押すと、果物のリストを調べ、その果物のリスト内のすべての果物について、最後にpieという単語を追加して印刷しますコンソールに入れます。

アップルパイ、梨パイ、オレンジパイ、グレープパイ、バナナパイができました。

ここで、for inループがどのように機能するかを理解したかどうかを確認するための挑戦の時間です。

そして、コースリソースにあるスタート用のDartPadを用意しています。

このURLに移動すると、winningNumbersと呼ばれる整数のリストが表示され、6つの当選番号が含まれます。

メイン関数の中に、チケット1とチケット2を購入した2つのチケットがあります。

そして、これらはあなたが宝くじに選んだ数字です。

ここで、ここに渡される各チケットを取得し、当選番号と照合して一致する番号の数を確認するcheckNumbersという関数を作成します。したがって、このチェック番号メソッド内にコードを記述し、メイン関数内でコードを呼び出します。この課題を解決するには、コードを実行すると、現在チケット1であるチケットを当選番号と照合し、一致する番号の数を確認できる必要があります。

そのため、このチケットには宝くじの当選番号と一致する9番と12番の両方があることが非常に簡単にわかります。

照合する入力としてチケット2に変更すると、一致する番号は41だけになります。

したがって、ループで学んだことを使用する必要があり、このタスクを実行するコードを作成する方法を理解するために少し考える必要があります。

そのため、ビデオを一時停止して、このチャレンジを完了してみてください。

大丈夫。

このメソッド内にコードを作成しますので、コメントを削除しましょう。そして、最初にチケット1を渡すmyNumbersと呼ばれるものにアクセスできることを知っています。

これで、チェック番号メソッド内でこの整数のリストにアクセスできます。

これで、当選番号にもアクセスできます。

それでは、まず、数字を反復処理またはループ処理してみましょう。

そのためには、forループを作成する必要があり、最も簡単な方法はforループを使用することです。

この数値リスト内の各数値を表す新しい整数変数を作成しましょう。

これが呼び出されます。myNumが正しいとしましょう。

各番号を表すため。

したがって、45に等しくなり始め、2になり、9になります。

そして、mynumbersの各番号に対して、これはfor in loop構文であるため、my num番号のそれぞれを出力できます。

したがって、これができることはすでにわかっています。

そのため、チケット1が渡され、自分の番号の各番号を表す新しい変数を作成してから印刷します。

チケット1から1つずつ印刷します。次のステップは、これらの各番号に十分にアクセスできる場合は、これらの各番号を当選番号と照合してみませんか？そのためには、当選番号を再度分離する必要があります。

では、勝利番号のそれぞれをループして、12から始めて、12がmyNumberに等しいことがわかるとしたらどうでしょうか。

番号。

6はmyNumberと等しいですか？ 34はmyNumberと等しいですか？

ポイントを取得します。そして、リストの最初の番号についてこれを行ったら、次の番号に進みます。 2マッチ12マッチ6マッチ34を確認します。

そのためには、このfor inループ内に別のfor inループを作成する必要があります。

したがって、ここで別のforインループを作成すると、勝者番号ごとにループします。

したがって、ここから来る当選番号のリストで、winningNumと呼んでください。そして、ここでは、両方の数字が等しいものを印刷できます。それは、numに渡されます。また、当選番号を印刷することもできるため、当選番号は当選番号と等しくなります。それでは、このprintステートメントを削除してコードを実行してみましょう。まず、私の数字を確認してください。

したがって、最初の番号である45から始めて、ここで各勝利番号をループしました。では、45が12に等しいことを確認するためにチェックを開始できますか？ 45は6に等しいですか？ 45は34ですか？等..

それで、それを印刷する代わりに、勝った番号が私の番号と等しいかどうかをよく言うことができます。

私の番号の1つが当選番号と一致します。

一致するものを印刷してみましょう。

大丈夫。

それでは、コードを実行してみましょう。これが呼び出されるのは2回だけです。

つまり、2つの一致があります。

取得した一致の数を格納する値を作成するにはどうすればよいですか？

さて、forループの外側に新しい変数を作成できます。それでは、matchesと呼ばれる新しい整数を作成しましょう。

そもそも一致するものがないので、0から始めます。

そして、当選番号が私の番号と等しくなるたびに、マッチを1増やすことができます。

そして、最後にforループが完了し、すべての数値を自分の数値と照合したら、一致した数を出力できるようになりました。

したがって、ここに一致変数を設定します。2つの一致と言います。そして、コードをそのまま実行すると、「2つの一致があります」ということがわかり、これらのforループの両方を調べてそれがわかるようになります。

これをチケット2に変更すると、チケット2のすべての番号が調べられ、それらのforループが再び実行されて、一致が1つあることがわかります。この課題は、for inループがどのように機能するかを完全に理解するのに役立ちます。ソリューションを理解できなかった場合は、コードを自分で記述し、forの各ステップで何が起こっているかを確認するためにいくつかのprintステートメントを使用しますループが実行されています。

これは、コードがバックグラウンドで実行していることを実際に把握するための唯一の方法です。

完了したら、Flutterプロジェクトに戻り、forループについて学習した内容を使用して、そこで課題を完了します。