All right.

So in the last lesson, we saw how we have to create drop down menu item widgets that encloses a text widget that has a value for each of our currencies.

And we know we have a lot of those.

So manually doing this is probably not an option for us lazy programmers. But luckily we have this construct called loops in pretty much all programming languages.

Now remember back when we covered functions in Dart and we had this function called buyMilk which was to be carried out by a little housekeeping robot where it would go out every day buy us a bottle of milk and then come back.

Well imagine having to do that seven days a week then you would have to remember to tell the robot to buy milk every single day.

So buy milk, then buy milk again then calling this function over and over again every single time we need it to happen even though we know that we need it to be carried out seven times or however many times per week.

There's a better way of doing this. Whenever we need some functionality or some piece of code to be executed multiple times,we can use a loop to do that.

And one of the most useful types of loops are FOR loops. Now FOR loops comes in many different flavors.

And the most common type which comes from the C language looks something like this.

We have the for keyword to say that what comes afterwards is a FOR loop, and then we create a new integer which is usually by convention called i. We set it to equal a starting value of 0 and then we have a semicolon and we define how long we want our loop to run for.

So while i is less than seven, we're going to keep running our loop. And then after a second semicolon we specify how we want i to increase after each loop.

Now finally inside some curly braces, we're going to do something: 7 - 0 times. So you can look at it as if this is the starting value and this is the ending value and this is how we're going to go from zero to seven.

So we're going to go in increments of 1.

We're going to start out at zero, run our code increase by 1 then run our code then increase by 1 again then run on code until i is equal to 6. But more than just running code many many times, we can actually do something with i.

So we could use this i which we know is going to start out being equal to zero then to 1 and 2 etc.

We can use it inside our FOR loop to affect our code.

Now what is all of this mean?

Let's look at it in a concrete way.

Inside DartPad inside my main function, I'm going to create one of those FOR loops that I showed you just now.

And we start out by creating a new integer, so a variable that's going to be called i

wnd it's going to start out being equal to zero. And then we add a semicolon and we say that while I is less than 10, we're going to keep running our code and after every single time we run our code, we add one to i.

So now inside the curly braces of this FOR loop, our code everything, that's inside here, is going to be run 10 - 0 times. So let's print something, let's just print maybe

Hello. Now when I hit run, it's going to print that 10 times into the console. And the number of times that we run this is determined by everything that's inside these parentheses.

So if instead of printing hello, we printed the actual value of i, then we can peek a little bit behind the scenes as to what's going on.

So let me hit run again and you can see that for the first time that my FOR loop ran, I was equal to zero because that's the starting value we gave it.

So when I printed i, it printed zero into the console.

But after that's been executed, it comes back to check again.

Well now is i still less than 10?

Well 0 is less than 10 so let's add one to it and let's run it again so it runs this printing out one. And it keeps on doing this until i is equal to now nine.

So this time i is equal to nine and it makes that check again.

Well, is i less than 10?

Well 9 is less than 10 so let's run it one more time.

And now i is going to be equal to 10 because we add one to it every single time this code is finished.

So now i's equal to 10 and we finally make that check. Is 10 less than 10? Well no, 10 is equal to 10.

So this is no longer true and our FOR loop exits. So we don't run this code anymore.

So there's a number of ways that we can change how our loop will be executed.

And one of the easiest ways is actually to change how I increment.

So we could say instead of adding one to i every single time after every single time the loop runs, what if we added 2?

So let's use plus equals 2.

So now i is going to increase by 2 every time.

And if I click run, i still starts with zero because that's what we said it should start as.

But now after every time the loop is run, i get to added to it.

So it goes from 0 to 2 to 4 to 6 to 8 until it goes to 10 which is now no longer less than 10 and so on loop stops. Now other things that you can change about our FOR loop is what the starting value is.

So let's say that it started with a value of two and we increased by one each time.

Well now when we print it, it's going to go from two to nine instead of from zero to nine.

Now we can also change when our code ends.

So if instead of i being less than 10, we put an i is less than 15 maybe. Then our code is going to run from 2 until it reaches 14, which is the last value before i is equal to 15.

Now let's see this in a real example.

So let's say that I had this function called buyMilk and I was able to specify how many days I want to buy milk for.

So maybe buy milk for seven days right?

So now let's write this function down here.

So it's going to be called buyMilk and we can specify an integer which is going to account for the number of days that we're going to buy milk for. Now inside this function what I want to happen is something really simple.

My robot is simply going to say buying milk on day 1, then day 2, then day 3 then day 4 to show which day it's actually buying milk for.

And I want this to run for as many times as we specified in this input.

So in order to do that, we can use a FOR loop. So we can say for int i is equal to and we're going to start it out with day one right?

So i is equal to one. And then while i is less than or equal to the number of days that we've specified in our input here, we want our code to keep running. And we're going to increase

i by one each time.

So now, we're going to enclose our print statement inside this FOR loop and we're going to use the value of i to be printed in here.

Now if I run my code, you can see that I'm putting 7 in as the input to my function buyMilk.

So 7 goes inside here and then it's going to run my FOR loop as many times as I specified in here.

So I said to start out with a value of 1 and to keep repeating the code inside here until i is less than or equal to 7, so until 1 becomes 7. And after every time this code is repeated, increase the value of i by one.

And every time my code is run, it's going to print buying milk on day i, so day 1 then day 2 then day 3 until 7.

So now that we've seen how a basic C style FOR loop works, it's time for a challenge.

There's a very old school looking website called 99-bottlesof-beer.net and here they've got the lyrics for the song for 99 bottles of beer.

So if you click on song lyrics you can see how this song goes.

It starts out with 99 bottles of beer on the wall, 99 bottles of beer.

Take one down and pass it around,

98 bottles of beer on the wall.

And then each time the lyric repeats that number goes down.

So we go from 99 to 98 t 9 until we get to no more bottles of beer on the wall, no more bottles of beer.

Go to the store and buy some more 99 Bottles of Beer on the wall.

So this website is unique because it collects together various languages that have written programs to be able to print out the lyrics for this song, and it's sorted by language.

So it's from a to z.

So for example if we take a look at the code for Java, you can see that it has a score rating based on people who have voted through here. And you can see their program being written in here and there's various different alternate versions that people have submitted.

Now as a challenge, I want you to print out a simple version of this song.

So this is going to be the starting baseline and then we need this number to decrease by 1 each time.

This number to decrease by 1 and this number to decrease by 1.

So you don't have to have the very last line saying No more bottles of beer.

Your code should stop at this line where we have only one bottle of beer on the wall, one bottle of beer.

So try and use what you've learned based on the FOR loop to try and print out these lyrics into the console. Pause video and try to complete this challenge.

All right.

So we're going to try and write a program for this in the simplest way possible.

So I'm going to copy the first line of code which I know to repeat pretty much 100 times. And I'm going to put it into my DartPad.

So this is going to be the starting baseline of my lyrics and you can see I've got this 99, 99, 98 which are the things that change in this song.

So let's create a FOR loop where we have to loop through it a couple of times and I'm again going to start with my integer i equal to 99 in this case. And I'm going to run my loop as long as I is greater than zero.

And finally I'm going to change i not add, but I'm actually going to reduce it by 1 each time,so i--. So now inside this FOR loop, I now have access to this thing called i.

And if we come in our lyrics for now, let's try and print I each time.

So let's hit run.

And you can see that inside my console,

I'm printing i from 99 all the way down to 1, which is what I need. Now instead of printing i,

I'm going to print my lyrics.

So I'm going to uncomment my lyrics, cut it and put it in here as a string, so inside some single quotes. And note that when we copied and pasted, it's actually on two lines.

So we need this to be considered as one line of code.

And now I can substitute 99 for my variable i, so that's gonna be i, that's gonna be i, this is going to be i - 1.

So we have to use a dollar sign and then we have our curly braces and inside our curly braces we can say i - 1.

So now if I run my code, you can see that it starts out being 99 bottles of beer on the wall, 99 bottles of beer.

Take one down pass around, 98 bottles of beer on the wall. And then it repeats until we get to 1 bottle of beer on the wall. 1 bottle of beer.

Take 1 down pass it around, 0 bottles of beer on the wall.

Now this is not perfect because we don't have the full lyrics because we haven't considered the singular version of bottles is bottle.

So once it reaches one, it should probably change to bottle. And once it reaches 0, it should probably say no more bottles of beer.

And there's also that final line that needs to be printed.

So if you're up for a challenge then feel free to try and write some code that will actually print out exactly the same lyrics as you see here.

But the fundamental thing I want you to learn here is how you can use for loops really flexibly to determine how many times code should be repeated and to use this value i inside that FOR loop for various things such as using inside a print statement or using it as an index for something like a list for example. Now while we're talking about FOR loops, I want to show you another type of FOR loop and this is called the for in loop and this is really helpful when we want to loop through all of the items in an iterable,so something like a list.

In this case let's say I have a list of fruits and I want to loop through all of the values inside that list and then do something with each of the fruits inside that list.

So if I use a for in loop, it looks something like this.

So instead of using that i to iterate through and to determine when our loop starts and when it ends and how it iterates, we instead create a new variable that is of the type that's contained in a list.

For example in this case we have a list of strings called fruits.

And for each fruit in fruits we can do something with that variable fruit.

And it's going to run this loop for every single item inside that list.

So if there's five items then it's going to run it five times. So heading again into DArtPad, here I've created a list of strings which is called fruits and it's set to equal a new list that contains five fruits:apple, pear, orange, grape and banana.

Now let's say that I want to make a function called maybe pieMaker and it's going to take each of those fruits and it's going to add the word pie to it.

So I want to be able to make apple pie, pear pie, orange pie which I'm pretty sure nobody makes, grape pie and banana pie. And I want to print each of those into the console.

Well what I could do is I could simply use a for in loop so I could say the keyword for, inside the parentheses,

I'm going to create a new string that is going to represent each item inside that list.

So because this is a list of strings then this is why I'm creating a new string that represents a single item in that list, so a string I'm going to call it fruit,

The singular form of fruits. And I'm going to say for every fruit inside the fruits list, I'm going to do something with that list of fruits. I'm going to print my fruit which is the single item and then

I'm going to add the word pie to it.

So now if I call my method pie maker inside my main function and I hit run, then you can see it goes through my list of fruits and for every fruit inside that list of fruits, it adds the word pie to the end and it prints it into the console.

So I now have apple pie, pear pie, orange pie, grape pie and banana pie.

Now it's again time for a challenge to see if you've understood how the for in loop works.

And I've got a starting DartPad for you which you'll find in the course resources.

Once you head over to that URL you should see a list of numbers this time list of integers called winningNumbers and it contains six winning lottery numbers.

Now inside the main function, there's two tickets that you've bought ticket 1 and ticket 2.

And these are your numbers that you've chosen for the lottery.

Now we want to create a function called checkNumbers which is going to take each of these tickets that you pass in here and check against the winning numbers to see how many matching numbers you have. So you're going to write your code right here inside this check numbers method and we're going to call it inside the main function. In order to solve this challenge,once you run the code, it should be able to check your ticket which currently is ticket 1 against the winning numbers and tell you how many matching numbers you have.

So we can see quite easily that this ticket has both number 9 and also number 12 which match with the winning numbers in the lottery ticket.

Now if we change to ticket 2 as the input to check against, then now we only have one matching number which is the number 41.

So you're going to need to use what you've learned about for in loops and it's going to take a little bit of thinking to figure out how you can create the code that performs this task.

So pause the video and try to complete this challenge.

All right.

So we're going to create our code inside this method right here, so let's delete that comment. And we know that we have access to this thing called myNumbers which is going to pass in ticket 1 to begin with.

So we now have access to this list of integers inside our check numbers method.

Now we also have access to the winning numbers.

So let's first iterate or loop through my numbers.

So to do that we have to create a for loop and the easiest way is to use our for in loop.

Let's create a new integer variable that is going to represent each number inside this list of numbers.

So this is going to be called, let's say myNum right?

To represent each of the numbers.

So it'll start out being equal to 45 then it'll be 2 then it'll be 9.

And for each number in mynumbers,so this is are for in loop syntax,we can now print out each of the my num numbers.

So we already know that we can do this.

So ticket 1 gets passed in and we create a new variable that represents each of the numbers in my numbers and then we print them out.

So it's now printing them out one by one from our ticket 1. The next step is if we have access to each of these numbers well why don't we check each of these numbers against our winning numbers? In order to do that, we have to again separate the winning numbers.

So what if we could loop through each of the winning numbers wo start out with twelve and see is 12 equal to myNumber?

No.

Is 6 equal to myNumber? Is 34 equal to myNumber?

You get the point. And then once we've done that for the first number in our list, well we move on to the next one. We check to see if 2 matches 12 matches 6 matches 34.

To do that we have to create another for in loop inside this for in loop, so a nested one.

So here if we create another for in loop that's going to loop through each of the winning numbers.

So call it winningNum in the list of winning numbers which comes from here. And inside here we can print both my number equals, it'll will pass in my num. And we can also print the winning num, so winning number equals winning num. So let's go ahead and delete this print statement and run our code and you can see we live through firstly, my numbers.

So we're starting out with 45 which is the first number and then inside here we looped through each of the winning numbers. So we can now start checking to see is 45 equal to 12? is 45 equals to 6? Is 45 equals 34? etc..

So instead of printing it out we can say well if the winning number is equal to my number,well in that case it means I have a match right?

One of my numbers matches with the winning numbers.

So let's print got a match.

All right.

So now let's run our code and we can see that we get this called only twice.

That means we have two matches.

So how can we create a value that stores the number of matches we get?

Well we could create a new variable outside the for loop. So let's create a new integer called matches.

And it's going to start out being 0 because we'll have no matches to begin with.

And every time the winning number equals my number, I can increase matches by 1.

And finally once the for loop has completed and I've checked all of the numbers against my numbers, I can now print how many matches I have.

So you have and then we'll put on matches variable in here, you have say two matches. And now if we run our code as it is, you can see that we have, 'You have two matches' and it's going through both of these for loops to be able to tell us that.

And if we change this to ticket 2, then it will look through all of the numbers in ticket 2 and it will again run those for loops and tell us that you have one match. This challenge to help you fully understand how the for in loop works and if you haven't been able to figure out the solution, then write out the code yourself and use and some print statements to see what's going on at each step of the for loop being carried out.

This is the only way to really get to grips with what the code is doing behind the scenes.

Once you're done we're going to head back into our Flutter project and we're going to use what we learn about for loops to complete the challenge there.