Hi, I'm Lia and you're watching coding@home TV, a regular educational program that will take you through the steps of creating your very own coding project.

The aim of this project is to create a digital solution to help Queenslanders know more about cybersecurity.

In our last episode, we researched the topic of cybersecurity.

Learnt about how chat bots might function as a digital solution and discussed more about identifying our audience and designing a product specifically for them.

Everything you do each day follows a series of steps.

Not just the kind we take when walking around the house, either.

I'm talking about each separate action that you do.

When we brush our teeth, we are following a series of steps.

When we put our shoes and socks on, we are following a series of steps.

We might make some slight changes to those steps, but mostly they stay the same steps each and every time.

Now when we do this, we are following an algorithm.

We can also make some decisions and repeat some of those steps as part of our algorithm.

An algorithm is a precise ordered series of steps that can be represented in words, or as a diagram.

And in this episode you'll be learning how writing an algorithm is the next step for your project.

Once completed, you can actually use this as a draft for your entry in the Premier's Coding Challenge.

How many algorithms have you followed today?

Probably a lot more than you realise.

An algorithm is any ordered series of steps that you follow to do something.

If you made your breakfast this morning, you followed an algorithm.

If you got dressed, you followed an algorithm.

If you brushed your teeth, then,

Yep, you guessed it.

You followed an algorithm.

Let's take a closer look at getting ready in the morning to leave the house.

What kind of steps do we need to do?

While the order that you do, these steps might be slightly different.

They probably go a little something like this - brush your teeth, grab your clothes, get dressed.

We can make more steps by breaking down some of the things we're doing into smaller steps.

For example, we could make more steps by thinking about all of the steps we need to brush our teeth.

Get toothbrush.

Get toothpaste.

Put toothpaste on toothbrush.

Wet toothbrush.

Brush teeth.

Get clothes.

Get dressed.

Or, we could look more closely at some of the steps and break down what is happening to provide some more detail.

For example, we could try and explain how to put toothpaste onto a toothbrush.

Here, we need to think about all of the steps involved in getting the toothpaste onto the toothbrush.

Here is what our algorithm would look like now.

Get toothbrush. Get toothpaste. Take off toothpaste cap.

Squeeze toothpaste onto toothbrush. Replace toothpaste lid.

Wet toothbrush. Brush teeth. Grab clothes. Get dressed.

There are many ways to write an algorithm.

Even if they all have the same result.

Do these algorithms solve a problem?

Well, if the problem is how to get ready in the morning, then I think they're a great solution.

Thinking about our Getting Ready algorithm, we can put the steps into a diagram and connect the shapes with arrows.

This is called a flow diagram.

Our first step is to start with an oval, we've written the problem that needs solving inside the oval.

We've written the next step on a rounded rectangle, and now we need to connect the oval and the rectangle with an arrow, showing the way the steps go.

The arrow will show the direction of flow.

All of the steps of our algorithm can be added to a flow diagram.

Let's see what our whole Getting Ready

algorithm looks like as a flow diagram.

Notice the different colour and shape of the final step.

This is to help show that we have reached the end.

The arrows flow from the first oval to the next step and so on until we've reached the final step.

This is to keep all of the steps in order.

Now, this algorithm is quite simple.

It's really easy for us to follow, but did you know that any algorithm can be made into a flow diagram?

Let's make another flow diagram for getting ready for school,

this time we need to make a checklist of all of the things we need to do before we leave the house in the morning.

Have you brushed your teeth?

Have you grabbed your clothes?

Are you dressed?

Have you noticed that each step is now a question?

But we can't just ask any question in an algorithm.

We can only ask Yes or No questions.

When we make our flow diagram

this will make a lot more sense.

But for now, let's just note that each step now requires us to make a decision.

We represent this decision-making part of the process with a diamond box.

This is what the first decision will look like in our flow diagram.

We can see the arrow coming into the box

and two arrows coming out of the box.

Yes and No.

Remember that a decision block should only ever have a Yes or No outcome.

Meaning the question that we asked inside the box should only have two possible answers.

Yes or No.

Let's add this next step and see what our flow diagram looks like now.

We have just added a decision block to our flow diagram.

Adding a decision is also called branching.

This is because it makes our flow diagram look a little bit like the branches of a tree.

If we trace the diagram with our finger, we can see that when we get to the first decision block, there are two ways to go depending on the answer to the question.

If we answer Yes, we end up at the next decision block.

If we answer No,

we follow the line to the block that tells us to, Brush our teeth.

Then if we keep following the arrows,

we will end up at the second decision block.

It's important to always follow the arrows in a flow diagram.

So make sure you've got them pointing the right way when you make it.

Let's complete our flow diagram with all of the blocks we need.

Now we can see that there are many ways to get from our green Start block to our End red block.

The branches allow us to make decisions and our path along our algorithm is dependent on those decisions.

We can always add in extra steps later if we feel like it and we can add in extra decision blocks to help make our algorithm more clear.

Can you think of any other steps that we could add in?

Maybe we could add in a step to check that we've got our shoes on after we get dressed, or a step to check the weather before we leave the house.

Where would you add another decision block?

Maybe you could add something that you like to do in the morning to make this algorithm your own.

Let's recap. An algorithm is an ordered series of steps that can be used to help us solve problems.

Algorithms can be represented as a flow diagram.

Branching is when we make a decision in an algorithm.

The path we follow through the algorithm is dependent on each of the decisions we make.

Later in this episode you'll get to meet some drone experts who are using algorithms every single day, and you'll even get some practice at writing your own algorithms with educational drone specialists.

Before you know it, you'll be an expert at writing algorithms for very own coding project!

You're watching coding@home TV! Before the break,

We learnt what an algorithm is, and how to create a flow diagram to represent an algorithm.

If I was looking at a flow diagram for presenting this episode, then the next step would be... Introducing Drones!

An aircraft that doesn't carry a pilot on board is known as a drone.

They are either controlled by a computer which can be on board, or by a remote control from a ground-based operator.

Drones can be large or small and have many uses, including photography, racing and surveillance.

Let's catch up with Stacey who is taking us to meet some industry professionals, who solve problems with drones by using algorithms.

[Stacey] Hey, everyone, it's Stacey here, and today we're taking coding@home to new heights.

That's right, we're flying high to get a new perspective on algorithms.

[Stacey] Dr. Catherine Ball - Lovely to meet you

Now, you're obviously very passionate about this technology.

How did you decide that a career in drones was for you?

[Catherine] So being an environmental scientist using drones to capture imagery of turtles on offshore islands, gave us real insight into how those animals were behaving without us being there, standing amongst them and disturbing how they might naturally behave.

So we were able to fly long-range Australian drones for hundreds of kilometres and get real imagery about whether those turtles were nesting or not nesting.

And we could also tell from their footprints -their tracks- what species that had actually crawled on those beaches.

So we were able to do full turtle habitat surveys and that was when I realized that drones were something quite special when it came to monitoring and looking after Planet Earth.

[Stacey] So how are drones helping people, especially here in Queensland?

[Catherine] So, Google chose to use Google Wing's first commercial services at

Logan, so you can get bread and coffee and medicines delivered.

Then if you think about the Great Barrier Reef, which is very important to Queensland, you can actually have drones that are swimming under the water that can take imagery of the corals.

And especially, you know, people that get lost in our National Parks.

Around the world we've had people be rescued using drones in National Parks when they've gone missing.

So drones really can be used for pretty much anything you put your mind to.

[Stacey] Today we're exploring how algorithms are used to solve problems.

Why is it important to understand this before you start to write code?

[Catherine] Okay.

So before you ever do anything with drone technology,

you need to understand three fundamental things.

Why? What? And Which?

So - Why are you flying the drone?

Because you need to plan your mission.

You need to plan, how the drone's going to fly.

And then What. So what are you actually capturing?

Do you want to take really beautiful high-resolution photography?

Are you flying because you're wanting to?

And so you actually just want a very quick and dirty video that shows you that's what the road looks like.

And then the final thing is the Which.

So which drone are you actually going to be flying?

Are you flying one with battery power?

Petrol power?

There's even solar-powered drones now.

So, what kind of equipment are you putting your sensors on?

So Why are you there?

What kind of payload are you flying?

And which drone are you using?

[These] are the three fundamental things you need to think about.

[Stacey] Catherine Ball began her career in drones through a passion for environmental science, but for James Miller, it was a different story.

Let's meet a software engineer from Insitu Pacific, an aviation company that operates drones to collect all sorts of information.

[Stacey] So, you're using algorithms to code drones and we've been exploring branching and repetition.

How do you use those steps in the drones that you use?

[James] An example that comes to mind, is we have a drone similar to this one in the background that searches for things, buried underground.

So the control software would program the drain to actually fly back and forth in a pattern to cover the entire area.

So we use repetition in the code to do that same pattern over and over again.

Another example -say for branching- would be those drones have also got sensors on them that detect obstacles and stops the drone from flying into them.

So, in programming languages we call that like an IF statement or a branch.

So, if an obstacle is detected, then the drive will stop.

[Stacey] Like Catherine mentioned, safety always comes first.

I wonder what safety features James codes into his drones?

[James] Drones have a lot of systems in them to make sure that they're safe.

We call these backup systems or redundancy.

They could be something as complicated as an obstacle avoidance algorithm that uses computer vision or LIDAR scanning to stop the drone from running into things.

Or it could just be a simpler part of the code that just monitors the fuel to make sure the drone has enough to get back home safely.

This is an algorithm that basically takes the current position of the drone and also remembers how far and how the drone got to where it is.

You can then work out how much fuel it needs to get home.

And when it reaches that limit, the code will branch into a new section of code, that will take the aircraft back home safely.

[Stacey] How can students learn and practice coding while they're still at school?

[James] I like to think of coding as just problem solving, but doing things in steps, but instead of writing those steps down in English,

So if you want to start practicing coding, you can start to think like a coder.

You can start to think in steps and use branching and repetition, and there's other techniques to come up with an algorithm.

So, if you have an idea to solve a problem or write a cool new game, you can start by taking those ideas and breaking them down into steps.

Once you have that coder brain working, you can then take those ideas and use some of the online tools to make it a reality.

[Stacey] Private Mackenzie Togo and Gunner Lewis Day are drone pilots in the Army Drone Racing Team.

And if you think that's impressive, they build and code these drones too.

[Mackenzie] The best part about being on this team is building your own drone.

So we start from all the basic components, solder them together and eventually code it to be able to fly the way that we want it to.

Before we even start building, we've got to think about how our drone's going to fly and we start by planning how we're going to code it and changing the rates for the roll, the pitch and the yaw to make sure it flies how we would like it to.

It's very important to be able to change the code

between the freestyle drones and the racing drones.

We do this using some coding software on a computer and fine-tuning those algorithms to improve the flight characteristics of our drones.

[Stacey] They're not just fun to fly,

[Lewis] So, drones are used for a lot of different types of things in the Army.

[They're] used for filming ourselves in training out in training areas like this.

Where they get mainly used [are] for looking at areas that are forward of soldiers.

So that before we send our soldiers to those areas, we know that they're safe.

So the Black Hornet is one of the most recent drones the Australian Army has acquired.

It has an approximate flight time of 25 minutes and about a range of up, to two [kilometres], weather dependent.

Its noise signature is extremely low.

It can sit about 10 metres above your head and you would not know it was there.

[Stacey] Wow, talk about, inspiring!

Our industry professionals are proof that when it comes to coding and drones, the sky is the limit.

[Lia] We now know what an algorithm is, and how to create a flow diagram which will help us with writing one.

We've also caught up with some industry professionals who are using drones to solve problems. to help our own drone solve problems.

Someone who knows all about talking to tech is Damien from STEM Punks.

[Damien] At STEM Punks, our programs are all based around the mindset of having innovation, creativity and entrepreneurship.

If we have a problem, it can be really tempting to try and solve that problem by jumping in straight away, and trying to write some code or for later on because we consider more of the options.

We can make a step by step process that can give us a better idea of what we're looking at.

And also, some of the things that we may not have seen in first place.

Today we're going to learn how to solve a drone problem using an algorithm and represent it using branching and repetition.

Today we're going to be talking about air-based drones and how they're coded.

Let's think of a scenario where drones would be coded with different types of algorithms.

An interesting situation where a drone could be used, would be helping keep students safe at a cross-country carnival.

Maybe if a student falls over, the drone could be used to detect the incident when monitoring the students from the air.

Let's plan how this might work.

We can start with the monitoring algorithm.

So the simple steps might be, the drone takes off, monitors the students over the carnival and then lands.

The algorithm might be - Take off.

Follow a pre-planned route. Land.

Now that might work for one lap around the circuit, but what if we had to send it time and time again, we don't have to keep pushing it on and pushing it off.

That's why we need to learn about repetition. we can actually incorporate a Repeat. By adding this Repeat in the code, we don't have to keep writing the same thing over and over again.

The code on the drone will not change in the Repeat.

If it's asked to do 100 times of the circuit, it's going to do 100 times of the circuit.

The only way you can change it -

Stop it manually or it runs out of battery.

Let's add repetition into our algorithm.

Repetition is great if you want the drone to do the same thing over and over again.

Now, this is where things can get pretty cool.

Let's think of a scenario.

Our drone is cruising over the cross-country carnival.

It sees a student injured.

Let's see how branching would be involved.

A drone has senses that receive input.

It has a camera that can take photos or record video.

There's audio sensors so it can hear things.

It also can have a small servo arm that can drop a first aid kit that includes an ice pack.

"Thank you! Ahhh..."

We can include some steps that allow the drone to make some decisions based on the input that it receives.

IF the drone hears a cry for help, it can move towards the sound.

IF the drone finds a student, crying out for help it can send footage to a first aid officer.

IF the first aid officer can see that the student has a minor injury, they can activate the drone servo arm to drop the first aid kit and send the location of the student for someone to provide further assistance.

Each time there was a step that started with IF, a decision needed to be made.

This is what we call branching.

Maybe you could think of another way that this drone could have helped other students stay safe in our scenario.

You can plan the branching steps to make your idea work.

Show your idea to a friend, or a family member and ask them to give you feedback on your algorithm.

Make sure you listen to that feedback because you might be able to improve your design.

Planning to code is an awesome way to start problem-solving with drones.

And once you get the basics, right...

The sky's the limit!

Today you've seen how algorithms are everywhere, as well as just how easy they are to write out using words or using shapes and arrows in a flow diagram.

This week -

You can start thinking about how your digital solution will work and the kind of steps needed to teach a user about cybersecurity.

Start with the problem you want to solve and then go from there.

Once you've added a few steps, give it to someone else, to have a go of following your algorithm.

This way, we'll know if it makes sense to them.

They might have some suggestions on how to make your algorithm better or some steps that you could add, remember feedback is important.

So consider their ideas.

Well, that's it for our show today and you are one step closer to making your big coding ideas...a reality.

Catch you guys next time!

Authorised by the Queensland Government, Brisbane.