<upbeat music>

[Lia] Hey! I'm Lia and welcome to coding@home TV.

Over the last couple of episodes we've been teaching you coding skills and tips to work towards your very own coding project.

The aim is to create a digital solution to help Queenslanders learn more about cybersecurity.

These skills will also help with any coding project you'd like to develop.

Here's what we covered last episode.

We learnt about algorithms, and how to represent them with a flow diagram.

Plus we caught up with drone experts and discovered how coded algorithms assist drones to be smarter and more useful in everything they do.

Today, we are ready to start coding.

But there is a big question we need to answer first.

If this isn't a question you need answered, it's probably a question you've definitely heard some adults in your life ask.

What is code?

Code is a way of conveying a set of instructions.

It might be a secret code, so nobody knows what your next mission is.

Or it might be a series of flags used to send messages across very long distances.

Code has been used in different ways for hundreds of years.

When we talk about coding nowadays, we most often mean coding using a computer.

The most common form of code for a computer is text-based coding.

This is when the programmer uses lines of numbers, text and symbols to

represent the instructions that they want the computer to perform.

The lines of code have to be very precise.

One keystroke in the wrong place could mean that 1000s of lines of code just won't work at all.

Fortunately there's a way to learn how to code that isn't so strict.

It's called block coding.

With block coding, the instructions that we want to give to the computer are represented by pictures of different blocks.

Just like we might use lots of different types of toy blocks to build up a tower, different coding blocks can be used to assemble a computer program.

Now that we know how we're going to be programming, our next step is to figure out what we're going to program.

There are two main ideas that we're going to be looking at today.

User input and Variables.

If you are using the program, then you are called the User, and programs are designed for Users.

User input is when our program asks the user to do something.

That something might be to click on a picture, to control a character by moving around the screen, or even to enter some text.

There are many other forms of User input, but these are just some of the main ones.

Sometimes when we ask for User input, the computer does something straight away.

For example, as soon as I press the left arrow key, my character moves to the left.

Other times when we ask for User input, we want the program to remember that information, so that it can be used in another part of the program.

The way that computers keep track of this information is through using a Variable.

Think of a Variable like a jar.

We can put things inside the jar, but we can also label the outside of the jar, so we know what's inside.

Let's look an example.

I want to keep track of how many red cars are driving past my window.

I take an empty jar and write on the outside, 'red cars'.

That label helps so I don't get confused with other information I'm trying to store - like blue cars and green cars.

Then I watch outside my window for cars.

There's one, two, three red cars going passed.

So now I can take a piece of paper and write '3' on it, and then I can pop it in the jar to use later.

I have stored the number '3' as my data in my Variable.

Later that day my good friend Rose calls me and asks,

"How many red cars came passed your window today?"

Because I recorded my data in the morning, I've managed to forget what the answer was.

But because I saved the data, I can go back to the jar, open it up and take out the paper.

I can read the data and let Rose know exactly how many cars came passed.

Now that we know all about Variables, let's get block coding!

Today we're going to be coding using Scratch.

Scratch is a programming language that was created in 2007, by Professor Mitch Resnick from the media lab at the Massachusetts Institute of Technology.

It's free for everyone and you can download your own version at the address on your screen.

Remember to ask permission from a parent or guardian before you do.

It also works offline, so you don't have to stay connected to the internet.

A key focus of Scratch is the use of sprites.

They're the characters you will see on-screen, like these ones.

We can use our block coding to give each one of these sprites a list of instructions to perform.

This is a sprite. This is a sprite. Even this is a sprite.

It's a bit like a movie set, where each one of the sprites is an actor, and they have a list of instructions for what to do in each scene.

Here's a quick example. Let's choose two blocks.

I'm going to choose one block from Events, 'when the space key pressed'.

I'm gonna choose the next one from Looks, 'say "Hello!" for 2 seconds'.

My blocks now read, 'when space key pressed, say "Hello!" for 2 seconds'.

So, I'm going to press my space key, and my cat says 'Hello!' for 2 seconds.

Now, that was a pretty simple example, but Scratch can actually be used to create a whole different range of programs.

I've seen Scratch used to program quizzes, games, musical instruments, stories, and even artworks.

Let’s recap. Today we looked at:

Block coding is a form of programming that uses colourful blocks to represent the instructions we want our program to run.

Programs are designed for users.

User Input is when the user interacts with the program

Variables are used to store User Input so that the program can use it whenever it needs to.

And we also had a quick look at Scratch.

This is the program we're going to use to start coding our very own chat bot.

Coming up we'll meet some experts who will share the many different way you can turn a love of coding into a career.

And we'll further explore the basics of visual coding.

Coding is awesome, and I'm not just saying that because I'm the host of an awesome coding show.

There is nothing like the feeling you get when you spend time refining a coded algorithm, and at the very end you've created your very own chat bot that can respond to users' questions...

...told a drone exactly how you want it to fly...

...or maybe instructed a robot to follow a sequence.

Once you've completed something like this, there is no doubt that you are going to love to code. You might love it so much that one day, you end up getting paid to code.

To find out exactly what that's like,

Stacey is going to catch up with some experts who have transformed their love for coding into a career.

[Stacey] Hi everyone it's Stacey here, and I've just been brushing up on my gaming skills not only are these games a lot of fun to play, but it turns out that it's also a great place to start if you're looking for a career in coding.

Looks like I might need a bit more practice.

Today we're going to meet some professionals who have taken their love of gaming to the next level and have turned these skills into a career.

Meet Sam Sinnamon, he's a senior developer for a company called Technology One, but we'll get to that later.

Let's go back to the beginning to understand where his love for technology all started.

[Sam] From when I was a kid...as long as I can remember,

I've loved computers, I've [been] really keen on them and trying to learn all about them in primary school [and] high school.

So joining computer clubs, stuff like that and making little video games even when I was in primary school.

[Stacey] Sam's not the only one whose gaming skills led them to a career in technology.

Natasha Moore's love for programming began with a game about pirates, when she was in Year 6.

[Natasha] Well I just loved playing games, I still do Stacey, in Year 6 I went to a small school a small town school, and we had a shared computer for Years 5 and 6, and on that computer we played a game called, Pieces of Eight, and it was a survival pirate treasure hunting game, and I was just so thrilled with it, because I ended up finishing the game.

I ran home to my mum, I was so excited, and I told her, "Oh Mum, I would like to study computers at college."

[Stacey] So, gaming is where their stories started, but what happened next?

[Sam] So, I went did a degree of Information Technology at University of Queensland and

I knew that I wanted to do technology stuff, but I didn't really know what industry I'd end up in.

In my last year, I applied for internships at big companies, and landed an internship at Technology One, which was awesome.

And the internship was basically a really good way for me to get a taste of what it was like to be real software developer and see real software developers working day-to-day.

I can still remember walking into the office here and seeing this space and I was like...

I was absolutely gob-smacked at the space that we had here.

I thought that you could only get offices like this for Google and Facebook overseas.

And so I was so surprised to see a company like Technology One who actually had their headquarters in Queensland and it looked like this.

[Stacey] It does look like a really cool space to work in and it must be so much fun, but tell me more about your internship role.

[Sam] So, as an intern at Technology One, you basically work in a small team with other university students, and you try and mimic a software development process as closely as possible and as an intern you only work on really, really exciting, bleeding-edge technology.

So, as an intern I looked at how we can make our code secure before it even became software, and it was actually a really good way to get a job here.

[Natasha] I went on to study programming at Central Queensland University in a Bachelor of Information Technology. There we learnt to code in multiple languages.

Namely Pascal, Borland Pascal, C, C++.

I bought my first PC when I was at university in my second year and I was so excited.

I was so excited that I opened the box to have a look inside and checked out the CPU and memory and hard discs.

It really developed, or assisted me in developing further skills in PC support at a later stage.

Following that, I applied for a traineeship with New South Wales Health, and that involved some coding, but it involved technology exposure to different areas.

While I was at New South Wales Health in the cadetship, I was assigned to the programming team and I was actually responsible for the bed management systems.

[Stacey] Looks like gaming, isn't the only thing Natasha and Sam have in common.

Their interest in helping people also led them to a career in coding and technology.

[Sam] At Technology One, I am a senior developer in the research and development team.

I work on budgeting software mostly, and with that we are looking at big businesses.

So a lot of our customers are councils, big organisations and helping them to make decisions with their money, so we're looking at potentially millions and billions of dollars worth of money and trying to make sure those numbers come out correctly, and that people can actually make decisions based off of those numbers.

[Stacey] What else do you love about your job?

[Sam] So, at Tech One we have a lot of opportunities to try and give back through our foundation and one of the things that we do is 'SteamLabs'.

So, we go to state schools, we've been to Morayfield State High School for example, and we actually run a little program at the school showing them what it's like to be a software developer and work in a software company.

So, as part of that, I run the coding portion, where we actually teach kids how to program in an hour and I find that really rewarding.

[Stacey] Natasha you've had some really cool experiences in the industry, what does your job look like now?

[Natasha] Well Stacey, I work at the Aboriginal and Torres Strait Islander Legal Service in Queensland and it's my job to ensure that our staff are cyber safe, but I also manage our IT systems across the state.

[Stacey] Even though you're not coding on a daily basis in this job, your skills in coding have helped in your career progression?

[Natasha] Coding skills are extremely valuable skills to have, underlying for any IT professional, so coding has provided me really valuable troubleshooting and root cause discovery skills.

[Stacey] Wow, there are so many different career paths you can go down with coding.

So what do our professionals have to say to aspiring coders?

[Sam] My biggest piece of advice would be, "Just jump in and give it a go."

Programming is all about solving problems, so look around your life, your family, friends, see if they have any problems that you can solve with programming.

You might not get all the way there, but the things you learn will be really useful to go back next time, and the time after that.

It's really, really important in tech industry and in software to be able to communicate your ideas effectively, as well otherwise -you know- it's not very useful to be able to write good software, if people can't understand what you're writing or why it's so great.

[Natasha] Well my biggest piece of advice would be to keep documentation, so I'm a really, really big believer in documentation and keeping notes on your work. That's a really big point.

Secondly, I think don't be scared to give it a go.

You know, even if you find that the programs failing, or it's not compiling, keep having a go and remember you always have your fellow coders to help you.

[Stacey] Sam and Natasha have inspired me to keep on practicing my gaming skills.

Who knows? Maybe one day,

I'll code something absolutely amazing!

[Lia] Thanks, Stace! If a job like one of those has you excited, then you don't want to go anywhere!

[Lia] Up next, a special guest shares some skills to get started in the visual coding world of Scratch.

[Lia] Welcome back to coding@home TV!

Today is all about getting started in the basics of coding.

So, you have a great idea for a coding project?

Maybe it's an entry for the Premier's Coding Challenge where you will code an app, a game or a chat bot to help Queenslanders know more about cybersecurity.

So, so far you've thought of who your audience for this project is going to be and what you want your design to look like, you've also planned out how it's going to work using a flowchart to represent your algorithm.

Now for the actual coding.

How to get started? Well allow me to point you in the right direction.

That would be towards Dr Damien Kee, who is going to share with us some of the basics of visual coding,

[Damien] Hi - I'm Dr Damien Kee, and for the last twenty years, I've been working with students and teachers all around the world, introducing them to robotics and coding.

Today, I'm going to show you my favourite way to get into programming - Scratch.

As Lia showed you, we can create programs by assembling Scratch programming blocks in very specific ways.

These blocks represent the instructions that we want our sprites -or our characters- to perform.

Let's start with our first program.

I'm going to grab a, 'say "Hello!" for 2 second' block and drag it out.

And when I click on this block, I can see that my cat does indeed, say, "Hello!" for 2 seconds.

Now let's grab a second block.

I'm going to use the same type of block, a 'say "Hello!" for 2 seconds', but this time I'm going to change the words inside and make it say, "What is your name?"

Now when I click on this stack of blocks, my cat does indeed say,

"Hello!", and then, "What is your name?"

Great that's looking fantastic so far!

Now let's add in a third block.

I'm going to put in another, "say 'Hello!'",

but because my name is Damien, I'm going to have my cat say, "Hi Damien!"

I'll connect this block underneath the first two blocks and let's give it a go.

Cat says, "Hello!"

"What is your name?"

"Hi Damien!"

Great that's looking fantastic now!

You'll notice that every time I add in a new block, I check it by running my code, just to make sure everything is working as I expect.

I've been programming Scratch for a very long time, and

I still make lots and lots of mistakes.

So it's a really good habit to get into - to check your code every time you make changes. So that's looking really good.

But if I decide to send this to my friend, Kim, and she ran it - it's still going to say, "Hi Damien!" and she's not going to be very impressed by that all.

We need to make it a little bit smarter, and to do that we're going to add in some User Input.

At the moment, when the cat asks,

"What is your name?"

I could have quite easily said, "Bananas", and it will still say, "Hi Damien!"

What we really need to do is to gather in that information, and use that further in our response.

To do this, we're going to swap the, 'say, "What is your name?" block', for an 'ask block', you can find the 'ask block' in the light blue Sensing section.

When we now run this stack, the program pauses at the 'ask block' and brings up what we call a 'text input field'.

The cat says, "Hello!" says, "What is your name?"

And you'll see that box down the bottom there.

The text input field allows the user to enter the information.

I type in, "Damien" and press Enter.

And I get my, "Hi Damien!" result.

However, let's try this with a different name.

If I run the same stack of blocks it says, "Hello! What is your name?"

and if I type in "Kim" it still says, "Hi Damien!"

Hmm... that's not really what we're after.

What we need to do is get that User Input and use that information in the last code of our block.

When the user types in their name, Scratch remembers this User Input inside the 'answer block', and you'll find the 'answer block' in the light blue Sensing section as well.

To store that User Input in the program to be used later, we're going to use what's called a Variable.

To use a Variable in Scratch, we firstly need to create one.

We'll go to the dark orange section named 'Variables'.

Click on, 'make a variable', and let's give it a name.

Just like Lia's jar example,

We need to get the variable a label, so we can

remember what type of information is being stored inside.

I'll keep it nice and simple to start with, and call this one, 'Name'.

Now that I have my variable, I can put some information in it, in our case, I'm going to take the answer to the question that was asked, and use that and store it inside my variable.

This is done using the 'set variable block'.

In our case. I'm not setting my variable, I'm setting the 'Name' variable, and we are going to set the 'Name' variable to be the answer that was typed in.

Let's review how these blocks will work.

First, our program will say, "Hello!" for 2 seconds, then we'll ask the user for their name and wait for a response.

Once a response is received, the program will store the User Input in the variable, 'Name'.

Finally, the program will say, "Hi Damien!" for two seconds.

Now that that User Input is stored as a Variable, we can continue with our program, and when we need that information, we can dive into the variable to retrieve it.

So now that 'Name' is stored in our User Input, we can use that with our 'say block'.

Let's see what happens when I use the Variable by itself.

I'm going to grab the 'Name' variable from my Variable section and place it inside the 'say block'.

When I run this stack of blocks now, the cat says, "Hello!"

It asks, "What is your name?"

I'm going to type in, "Damien".

Hit the Enter, and it does indeed say, "Damien".

When I use a variable by itself, it just says, "Damien"

I'd much prefer it to say, "Hi Damien!"

To do that, we're going to use a new block.

It's called a 'join block' and you'll find it in the green 'Operators' section the 'join block' takes two separate words and combines them together.

Our first word we're going to put in here, "Hi" - and the second word, instead of writing in, "Damien" - I'm going to use my Variable.

Let's run our program now and see how it goes.

I click on my stack of blocks. Cat says, "Hello!"

Asks, "What is your name?"

I type in, "Damien". Hit Enter.

It does indeed say, "Hi Damien", and just to make sure it's working correctly, let's try it again with a different name.

I run the stack of blocks again.

The cat says, "Hello!". It asks, "What is your name?".

This time, I'm going to type in, "Kim". And when I hit Enter, it does in fact say, "Hi Kim". So just to recap:

We used the, 'say block' to have our character say a few lines of dialog.

We used the 'ask block' to capture some user data.

We then stored that user data in a Variable.

And finally, used that Variable in the final part of our program.

Why don't you try to add some more code to ask some more questions of your user?

Don't forget to save your answers as Variables and give those Variables a name, so you remember how to use them later in your program, and most importantly have fun!

[Lia] Fantastic advice! Now you're ready to start experimenting with coding.

Start simple, and do not give up if you don't get your code working the first time.

Figuring out how to get your code to work is part of the fun and makes it even more rewarding when you do succeed.

If you do end up with a great coding project that helps Queenslanders learn more about cybersecurity, then you have an entry for the Premier's Coding Challenge.

You can find more info at the address on the screen.

It's open for Year 3 to Year 10 students, and there are some amazing prizes up for grabs, like robotics and electronics kits.

Well, thanks for watching.

We'll catch you guys next time for more tips and ideas to help your coding projects.

Until then, keep coding, have fun, and we'll catch you next time.

Authorised by the Queensland Government, Brisbane.