Hi, I'm Isla and you're watching coding@home, a regular educational program that you can check out right here on 10 Peach during Term 2 and online.

Our mission is to show you just how fun and awesome coding can be.

Last week we learnt all about algorithms: what they are, why they're useful, and how we can use them in our coding projects.

Plus, we spoke to some inspiring drone specialists who are using coded algorithms to make their drones smarter and safer, to help Queenslanders every day.

Coming up today: we're getting our first look at writing some actual code and finding out firsthand what the journey to a career in coding looks like.

Today we're going to start learning about using coding elements to write our solution. We'll learn what user inputs and variables are and we'll revisit branching.

We'll also be catching up with some amazing Queenslanders who are coding experts and they'll tell us about their career pathways.

And, don't forget about entering your project in the Premier's Coding Challenge and you could share in a $10,000 prize pool of some amazing prizes like robotics and electronics kits.

Coding can mean many things and at its heart is about representing information in a way that we can analyse or obscure data.

In other words, we can use code to hide or encrypt information to keep it safe and secure, and we can use code to classify and identify patterns and information, so we can learn more about the world around us.

Computer coding is about using a programming language to get a computer to do something.

Computers do lots of things for us, from entertaining us with games, and playing music, to educating us and helping us find information.

Computers are everywhere and can be in fridges, cars, watches, phones, even vacuum cleaners.

Did you know that there are over 700 different programming languages?

Each language has the same purpose of programming a computer for it for it to do almost anything you can imagine.

Each line of code is read by the computer as a command to make our program do things like print something on the screen, to increase your high score, to move a character in a game, or to ask the user for information.

Each programming language may have different commands and different syntax.

Syntax is a bit like grammar.

In English, we capitalise the start of sentences, use quotation marks to show when someone is speaking and show that a sentence is complete by using a full stop, exclamation mark or a question mark.

We also use words in a certain order.

Imagine if those grammar rules didn't exist.

"Hi name Isla my is."

That would be very confusing. When we use a programming language, we need to use the syntax of that language so that our computer can understand our instructions. We'll learn a bit more about that later.

If we want the program to be interactive, it should ask the user to make a decision, or to supply some information and in turn the program should provide feedback to the user.

Whenever you interact with an app you are making decisions and supplying information to the app. In turn, the app gives you feedback in the form of new information, sound or vibrations.

The data that the program reads from the user is called the user input and this needs to be stored in the computer's memory so that it can be used in the program. To keep track of all this data we use variables.

Think of a variable as a container.

We can store whatever we want in our container, but we should label it so that we remember what's inside. A good label helps you remember what the variable is supposed to be when you have lots of lines of code and also helps if someone else is going to read your code later.

Now that we've covered some key concepts, let's go write some code.

For our project we'll be using a language called Python.

Did you know that Python is not named after a snake?

Its creator Guido van Rossum named it after the British sketch comedy troupe, Monty Python.

Python is used by Google, YouTube, Spotify, Instagram... the list just goes on and on.

You can download your own copy of Python that'll work offline.

So, once you've installed Python onto your device, it'll work without being connected to the internet.

To start with, let's look at the interface. It's called IDLE.

Short for Integrated Development and Learning Environment and it's what will be using to write and save our Python code.

It opens in the Shell interface, which is good for writing simple lines of code or copying and pasting chunks of code to see how they run, but it doesn't save our work.

To write some code that we can save and come back to later,

We need to create a new file. To do this you navigate to File and then click on New File.

Did you know there's a tradition that dates back to the 1970s for when coders write their first program?

The first program is called the Hello World program and it's used to demonstrate the syntax of that language.

We're going to follow the same tradition and print some text on to our screen that says, "Hello world".

To print to the screen we need to know the command or statement to print.

Statements are predefined instructions that Python uses to perform tasks.

The print statement is print followed by some brackets.

Python will then print whatever is placed inside the brackets.

The part that's inside the brackets is called an argument.

Not the kind you have with someone you disagree with, but that's just the name for that part of the code.

Let's see it in action so it makes more sense.

You can watch as I type the code, and see what happens when I run it.

You can see the print statement and in the brackets,

"Hello, World" is the argument.

Now, I want to run my code.

So I go to Run, then Run Module.

I'm then prompted to save my file. I need to save my code with a file name that helps me to remember what it does.

I'm going to call it helloworld.py. IDLE will add the extension at the end, so I don't need to put the .py there every time.

The shell opens up and: hello world.

We're now Python coders.

This code isn't interactive though. Let's give our users something to do when the program runs.

It's time to learn a new statement.

I'm going to write it into our file and we can break it down.

Let's start with the argument and work back.

Our code is asking, "What's your name?"

We used an input statement because we want the program to wait for the user to input some data.

This data will be stored in a variable called name.

We can label our variables whatever we want, as long as they start with a letter.

Now, let's do something with our name variable.

What do you think this code will do when I run it?

What will we see on the screen?

The great thing about coding is that we can evaluate it and test it at any time to find out if it's doing what we intended.

It's actually a great way to learn.

Our code printed "Hello, World" and then "What's your name?"

Then when we entered some data, it printed the name variable.

It's pretty good for our first go at text coding.

But, we can make our code a little bit smarter.

What do you think will happen this time when we run the code?

Wow, that's so cool! The code can say hello, remembers my name, and can print it all in one line.

I think this is going to be the start of the beautiful friendship.

When we join arguments together, it's called concatenation.

Here, we use the plus sign to join our arguments.

Note the extra white space after hi so that our final output also has a space.

Let's recap the key concepts and highlight some of the important syntax.

Python is a text-based coding language used all over the world.

Code syntax is all about using the exact spelling of statements and following the rules of a language.

The print statement is used to write an argument on the screen.

The argument goes inside the brackets. The input statement is like print, but asks the user a question. It waits for user input.

Variables are used to store data.

In Python, variables must start with a letter.

Concatenation is how we join arguments together.

Today, we joined our data with the plus symbol.

Now that you've met Python you can ask a parent or guardian for permission and download it onto your own device.

Then you can have a go at writing the start of your chat bot.

You can re-watch this episode on TENplay, or you can record this episode and watch it again for some extra help.

Plus there are loads of Python tutorials available online and in the Python help files.

Again, make sure you ask for permission before going to the help files.

After the break we'll be speaking to some special guests who are using coding to solve problems every day, and we'll find out what it's like to be a professional coder.

[Isla] Welcome back to coding@home.

Before the break we had our first go at writing some code using Python.

Later on we're going to delve a little deeper into that but first let's chat to some tech industry professionals who code every day.

[Stacey] Let's meet Amber Coffey.

She's a Cloud Support Manager at one of the world's largest technology companies.

[Amber] When I was in high school I really didn't know what angle I wanted to take later on in life.

So I really tried to keep my options open and I followed the things that I really enjoyed which were maths, science and the arts.

[Stacey] Amber's not the only one whose interests in high school led her down a tech and coding pathway.

Peter Roberts isn't just playing games.

He's helping students to design them.

[Peter] I work for Autism Queensland where I'm a mentor at Studio G, which is one of our programs where I help mentor other people on the spectrum to learn about how to make games so that they can transition into either higher education or employment.

[Stacey] Did you learn about this sort of thing while still in high school?

[Peter] Not really. I didn't do any computer subjects in high school.

I wasn't... I liked games, but I hadn't really discovered that I love them.

What I wanted to do was creative writing.

So when I finished school I had applied for a creative writing course, and then I ended up deferring that and doing a Diploma of Multimedia for a year as it's a fun sort of side thing to do for a bit. And during that I started reading about games, talking about games with other people and just learning about game development a lot more and found that it was really really interesting to me.

[Stacey] Now Amber, you had a broad range of interests growing up.

What did you decide to study after high school?

[Amber] After high school I wanted to find an opportunity where I could practice my creative thinking along with my maths and science skills.

And technology was just starting to emerge, the internet was starting to become more prevalent.

And I found an opportunity to move to the Gold Coast and study IT there.

[Stacey] Now you mentioned being creative. Do you think being creative

is important for someone like you who wants to have a coding technology career?

[Amber] Being creative is so important in technology.

There's always more than one answer and more than one way to solve a problem.

And when I was first starting in coding I found my ability to think creatively really allowed me to come up with some really unique solutions.

[Stacey] So, both Amber and Peter started off a little differently in their coding journeys, but both ended up in some interesting places.

[Peter] I've got a few different students that I work with and they've all got their different projects.

I've got... one of them's making a platforming game where the main character is a mouse... the character has to go through different levels, fight enemies, beat a boss at the end.

And I got one guy, he's making a mod for a game.

So he's not actually making his own game.

He's got a game that he really enjoys, and he's learning about how to modify it and make his own version of it using it's already existing software.

[Stacey] Amber, what exactly is a Cloud Support Manager?

[Amber] As a Cloud Support Manager I have a team of industry professionals where we work with businesses to provide support services for the technologies that they use to build applications on.

[Stacey] Well it certainly sounds like you have a very important job and it's a real team effort.

And a lot of people rely on you for their organisations and individuals as well.

[Amber] Lots of organisations do rely on us to make sure that we can provide them with the infrastructure that they need to provide a secure reliable scalable solutions.

To put that in context, if you're shopping on a website, and that website is going to have a major sale.

It's going to have lots and lots of people hitting that website and that's called high traffic.

And what my team does is work with those organisations to make sure that we can provide a seamless experience for the customer.

[Stacey] What do you love about your job and all of the amazing experience you've had along the way?

[Peter] Well, one of the things I really love about it is all the problem-solving skills involved.

I had a few different projects that I worked on that have been really interesting.

One of them... the one I'm most proud of was -- also education related -- was helping some people develop a game that interestingly had a live orchestra that would play music in time with the events of the game.

As well involved a lot of problem solving and helping people.

So my job at the moment as a mentor, I think that's one of the things that I really get out of it is working with a variety of people to help solve problems.

It's one of the things that actually has in common with programming and game development is the solving these problems for people, so that's been really good.

[Amber] What I love most about my job is the ability to connect with so many people and make such a big difference. So even though I'm based in Brisbane I have a world of opportunity available to me.

So I connect with people right across the world to solve really complex situations for customers based both here in Queensland, and around Australia and around the world.

[Stacey] Now Amber, as a previous judge on the Premier's Coding Challenge, what are you looking for and what advice can you give to students?

[Amber] The advice that I would give to the entrants is to really have a story about, that they want to take their end user on.

So to be really clear and to map out step-by-step what they want the journey

to look like for the person that's using their application.

[Stacey] There are so many different roads you can go down for coding.

Do you have some top advice for aspiring coders or game developers?

[Peter] To learn and get better at coding I think the thing that has helped me the most is definitely patience. Anyone who has tried to learn how to code

I think you'll find quite quickly that patience is the skill you need because there's a lot of problems that will pop up.

Patience, persistence, and just trying different things.

You know, I didn't start with an interest in game development.

I sort of fell into that, and I've tried a variety of different things and I've sort of gotten better at it over time until it's finally become the thing that I do.

[Amber] My advice for students looking for a career in coding is to really keep their options open, and to think about their skill sets from the more holistic point of view.

Coding is certainly a really important fundamental of the industry, but so is having great communication skills and being a really creative thinker, so it's really important to have a very diverse range of skills and all of those skills are very much in need in the industry.

[Stacey] Back when I was at school I actually wanted to be a vet when I grew up and look where I am today. It just goes to show you should follow what you're interested in and who knows what interesting twists and turns will happen along the way.

[Isla] Don't go anywhere. After the break we'll be looking at the next steps for your coding project.

Welcome back to coding@home. We've had an introduction into writing some code, and we've gained an insight into the career paths of some coding professionals.

If you remember last week, we talked about algorithms.

[Robot] I remember. Branching was really important in allowing our code to make decisions.

[Isla]. Yeah, that's right. So, let's talk to another coding professional about branching in Python.

[Robot] That sounds really fun.

[Isla] You're probably familiar with movie and TV versions of coders and hackers madly typing away on a keyboard, but in reality it's not quite like that.

Sasha Morrissey is a real-life professional coder and can show us exactly what writing code actually looks like.

[Sasha] Hi, I'm Sasha. When I'm at work, I'm the head of engineering at a medium-sized software company.

My other passion is being one of the lead coordinators of The Girls Programming Network.

Today I'll be showing you how to develop your chat bot code to include branching.

Until now our programs can only do one thing.

They execute the same way every time we run them.

What if we want the program to run differently each time?

We can do that by using if statements.

An if statement tells the computer to only run a portion of code if a certain condition is true.

This part of the code is always run no matter what.

This is an if statement. It is also known as a conditional because it tells the computer to do something depending on a condition.

This is the condition.

The condition is what the computer checks before running the next piece of code.

In this case it's checking if the name of the user is equal to Alice.

Notice that we're using two equals signs here.

When we use one equals sign, it's called assignment.

We did this earlier when we assigned data to a variable.

When we use two equals signs, we're checking whether two pieces of data are equal.

This is the branch.

It runs only when the condition above the branch is true.

You can tell which part of the code is branched by the indents.

If the code is indented look up to the first line that isn't indented to see what the condition is.

This tells you what condition must be true in order to run this code.

Indentation might not look like it does anything, but it has special meaning in Python.

If you get errors when running your code, check that your indentation is correct.

When we run our code, our program prints this message when Alice enters her name.

What if we want to do something different if somebody else enters their name?

For this we can use an else-if statement.

This is an else-if statement, or elif for short.

Elif is very similar to if, but it's only run if the previous if statement is not true.

So the if statement will check these conditions in this order.

First it will check if the entered name is equal to Alice.

If that's not true it will then check if the entered name is equal to Bob.

If it finds one of them is true, it enters the branch for that conditional and the code is run.

In this case because we enter the name Bob,

this condition was true and it enters this branch, and then prints, "Hi Bob".

You can expand this to include the names of as many people as you like.

Try adding the names of some people that you know.

Wait, why didn't Kass print a greeting? What happened here?

We entered a name that we hadn't thought of yet.

So there was no response programmed for it. In order to display a message we'll have to add another branch for Kass.

But if we keep doing this, we're going to need to know everybody's name before they can get a greeting using our program.

If only there was a way to display a generic message, even when we don't know the person's name yet.

For that, we can use an else statement.

Now our program can display a message, no matter which name we enter.

This is great. Now we don't have to think of all the names in advance.

And for the ones we do know in advance we can program a specific message.

Now, there's a response for everybody.

The else statement doesn't have a condition.

It only runs if none of the previous conditions are true.

This means that if you have an else statement,

exactly one of the indented blocks will run in your program.

We can use this to tell at a glance what our program is going to do.

This is very similar to how a chat bot works, except instead of asking for your name it asks you which command you would like to run.

And instead of checking which person entered their name, the chat bot checks which command the person asked them to do.

Instead of printing a customised greeting the program performs the command the user requested.

They are the same program.

Remember that the indented code is only run if the conditional for that branch is true.

To find the conditional for a branch, look up from where the indented code is.

Only one branch will be run for your set of if-elif-else statements.

If comes first, then elif, then else, in that order.

You can use if statements to create powerful branching code that runs differently depending on the user's input.

Now, you can write your own chat bot using this structure and add any commands that you like.

[Isla] Well, "if" I wasn't already excited about coding, "then" I definitely am now.

But seriously, that was some great advice, and I can't wait to learn even more.

While you're at home, why not try having a go at starting your first coding project.

Try to write your own "hello world" program and include branching so that your code recognises you when you enter your own name multiple times.

You can even have a go at coding it so it will respond to any name that is entered.

If you don't get it right the first time, remember part of the fun is to look back and evaluate your code and trying to figure out a solution to your problem.

And don't forget to think about taking part in the 2020 Premier's Coding Challenge.

It's fun and we'll continue to provide you with tips and inspiration along the way.

So until then stay positive, look out for each other, and we'll catch you next week.

Authorised by the Queensland Government, Brisbane.