PĪkau Name: First steps in programming: CT PO1 (EMP05)

Video Name: Why sequences are important in Computational Thinking

Presenter: Tim Bell

Commentary: Shortly we'll look at an activity where students are giving instructions to each other. Now, it seem as a like bit of a simple activity, so I wanted to give you a bit of a spoiler, and show you how it applies to programming first, and then we will go back and look at how we can work with students at a fairly introductory level.

The concept we are looking at is called a "sequence". So let's have a look at sequence in programs.

So this programming language is called "Scratch", and I'm going to use it just for a very simple purpose of getting the cat to display three phrases, and I'm going to put them in the right order. So when the thing starts, when we click the green flag; it's going to display the word 'kiwi', and I'm going to display 'like to eat', and then I'm going to get it to display the word 'worms'. So I will click on the green flag, and there it is: 'kiwi', 'like to eat', 'worms'. Fantastic. Now, the concept of sequence is simply that the first instruction is the one at the top and then the next one and so on, it goes from top to bottom. Not really a foreign concept, but it's a really important one in computer programs. If I move these around and I say something like "Worms", "like to eat", "kiwi", I've changed the sequence and if we run it now indeed the cat says "Worms like to eat kiwi", which I don't think is quite the case. Probably doesn't pay to overthink that one.

Now, we want to skip to some other programming languages, and at this point we are getting way beyond what junior primary school students would be doing, but just to show you how it applies. Here's a programming language called Python. It's quite popular in computing education, and this is what the equivalent looks like. Print the word 'kiwi', print 'like to eat', and print 'worms'. And if I run it then over here the three phrases come up. Now, the reason it uses the word 'print' is because very early computers, there used to be a teletypewriter so to display something they would print it on a typewriter. Different terminology, but you can see it's doing the same thing, and if I change the sequence, if I put the word 'like to eat' at the end then when I run it over here it says 'kiwi' 'worms' 'like to eat'. Not terribly gramatical!

Here's another language, this one is called JavaScript. It's used quite a bit in web pages and on online systems, and the process for getting a message on the screen (or one of them) is the 'alert' function. And that pops up a box. So I've got a program that does an alert called 'kiwi', and then an alert that says ‘like to eat’, and so on. So if I run this here’s the first alert, the word ‘kiwi’, and then the next alert: ‘like to eat’, and then the word ‘worms’. So you can see the sequence is just top to bottom, one after the other.

And one more example. This language is called Java and it’s a little bit different to JavaScript; they are not totally related, but it’s a popular language in industry for a lot of purposes. You can see that the way that the wording is used is a little more complicated but the sequence still appears here. It says ‘do an output of “kiwi”, do an output of “like to eat”, do an output of “worms”. And if we run that then over here, hopefully, there it is: ‘kiwi’, ‘like to eat’, ‘worms’.

So that’s what sequence looks like in computer programs. Now, you probably won’t be using those programs with young students, but I just wanted to show you that this principle applies across all kinds of programming languages: educational languages, ones that are used across the web, ones that are used for other purposes. So your students are going to encounter this in the future, and now you’ve got a bit of an understanding of why this exercise we are about to do is actually quite important.