

Pīkahu Name: What is Programming

Video Name: Programming in Another Language (EMP07-7)

Presenter: Professor Tim Bell and Joanne Roberts

Tim: Now that we've looked at programming in Scratch, I thought it would be good just to look at it in a different language so that we can see what's common between different programming languages, which part of it's programming, which part is the language. This language we are going to use is called Python.

Joanne: I know Scratch quite well, but I don't know Python at all, so this will be interesting for me.

Tim: Okay, cool. Python is quite popular for education. In the last couple of years in universities it actually became the most popular introductory language. People will learn multiple languages, but it's quite a good one for getting people started. Before Python was around there was another language called Java, that was very popular but it would put off some students because you have to jump through a lot of hoops to use it, so Python is a slightly more gentle introduction.

Joanne: Why's it called Python?

Tim: Python, interestingly, it's named after Monty Python's Flying Circus.

Joanne: So nothing to do with snakes?

Tim: No snakes.

Joanne: That makes me feel a little more comfortable.

Tim: The guy that invented it was a fan of Monty Python. So what I've got here is (refer to computer screen), this is being accessed through the web, and the reason I'm using this system is it means that after people have watched this they can actually just go straight to the web and use the same system. A lot of people when they program will be using something that has been downloaded to the computer to program on, but this works fine, particularly for getting started. With Python we've got this area over on the left which is where I'm going to type my program, and the program's going to run on the right-hand side.

Joanne: Okay. In Scratch we've got where we type the program on the right, and where inputs and outputs end up happening is on the left.

Tim: Yeah, sort of the other way round, but, yeah. Basically there's an area for the programmer and there's an area for the person using the program. I'm actually going to start the demonstration with a very simple program, which is the classic first program, which is to display the words 'Hello world'. In Python the command for displaying something on the screen is 'print', which is kind of weird, it just harks back to the days when computers mainly had printers on them for input and output rather than screens, but we still use 'print'. One thing I just want to emphasise is we are not going to teach how to program in Python, so most of this you just have to take for granted. You can do that [learn Python] more carefully later on, separately from this.

So here's my program; to run it I'll press the green button up here and then over on the right-hand side, that's the output of my program, which is not a big deal, 'Hello world', but it is a big deal in some ways because I've written a program, I've run it, and I've seen how it works. But what I want to do is get back and actually implement the program that we were looking at before, which is displaying numbers in te reo. So we'll implement that in Python. The first thing that happens there ...

Joanne: So, it's asking 'What number shall I display?'

Tim: Right, which is, in Python, *input* is the command for getting an input. Some things make sense! And it says 'What number...'

Joanne: 'Shall I display?'

Tim: 'Shall I display?' That basically says: get an input from the user by asking them that question. So it's completely equivalent to the 'ask' command, except that we need to store [the answer] somewhere. In Scratch the "ask" command stores it in a variable called 'answer'. We are going to store it straight into the variable which is ...

Joanne: 'Number to convert'

Tim: 'Number to convert'. Now, again, I don't want to over explain it, but the right-hand side says 'get a value from the user'. The equals sign means put it into the variable (or a box) with this name on the left-hand side.

Joanne: So that's like the 'set' thing in Scratch. In Scratch it will set the 'number to convert' to the answer.

Tim: It's exactly equivalent. In Python it's an equals sign, in Scratch it's the word 'set'. I'm just going to throw in another thing here, which we don't really need to worry about, but it just says I'm expecting an integer from the person - I want them to type in a number. Scratch kind of hides that from you - you don't have to worry about that so much. Now, just to check that it is working I'm just going to display that 'number to convert'. It's not really what the

program is going to do, but it is a little bit of what we call *scaffolding* to see if it's actually working. Let's run it: it said 'what number shall I display?' I can type in a number, type in '23' and it's printed the '23' on the screen for me. So you can see the input and output are both over here. Now it's not what I actually want it to do, but like most scaffolding, it was useful to have it there but we will get rid of it. It saves time in the long run to be checking things as you go along. Instead, what's the next thing we did once we got a number?

Joanne: We've got the 'if' box, or block.

Tim: Okay, so in Python 'if' is actually 'if'!

Joanne: Yay!

Tim: Nice when it's the same isn't it?

Joanne: So, it's the 'if number to convert is equal to 1'

Tim: Right, so we are saying 'if the number to convert is equal to 1', and in Python, saying "Do two things have the same value?", that kind of equals, we do a double equals sign (==). If I'd done a single equals sign it would say 'work out the right-hand side and put it in the left-hand side', which is not what we want. The colon means 'I'm going to show you after this line what I want you to do.' The next thing it does, if it did match...

Joanne: 'Say tahi'

Tim: And of course, in Python it's 'print' to get it on the screen. We're pretty much there. You'll notice it's indented a bit, it's just the same in the Scratch blocks, it's actually indented, or it's inside it, it only belongs to the case where it is equal to 1. Let's run it and see what happens. 'What number shall I display?' Type in '1'. Tahi.

Joanne: Yay!

Tim: It worked. Now, just to really be sure that it's working as it should, I'll type in number '2' and hopefully, it doesn't print anything. If it had displayed something then my logic wouldn't have been quite right. So we are getting there now. I just need to say 'if the number to convert is equal to 2', and actually I can copy most of that from there, and in that case we are going to print...

Joanne: Rua.

Tim: Rua. And we'll run that. Test that it works, so I put display '2', it displays 'rua'. It looks like all the structure is right. You can see a pretty close correspondence between the structure here, and it will be easy to add the next couple, but we will leave that as an exercise for the reader I think. The only other thing that's in here is there is this 'forever' loop around it because at the moment I have to run my program every time I want to do a conversion. But I can put a loop right around this. 'Forever' loops, Python, it's not that natural

to make stuff happen 'forever', normally it's going to finish at some point, at the end of the day, or when the last customer's typed something in or you get to the last thing, but one of the main looping commands in Python is called 'while'. It's 'while some condition is true' it will keep on doing it. We actually want it to always do it, so we actually just put in the word 'true' which just means 'always keep on looping'. I just need to indent these things down here because this is what is going to be inside my loop. Whoops, that one shouldn't be indented, but that one should. You can see that the indentation here does correspond to [Scratch] because the 'forever' thing, it's got all that stuff inside it. When students are learning these structures [in Scratch], it's kind of been done for them, but they are used to that kind of layout. Python uses a similar kind of idea. So now we can test our program. I'll run it, and it asks for a number. I'll try '1', 'tahi' but now it's looped and it keeps on asking 'What number shall I display?', type in '2'. If I type in '3' nothing happens because neither of those 'if' statements are true and it just gets to the end and just loops around again. But now we've essentially got that program in a different language. Some things are common, some things are different.

Joanne: I can see how my students learning Scratch it really does scaffold them up to a more complex language, so yes, learning Scratch really is worthwhile.

Tim: Yes.