Pīkau Name: Communicating well when programming (CT P0 5)

Video Name: Variables of Different Data Types (EMP10-2)

Presenters: Tim Bell and Joanne Roberts

Tim: For Progress Outcome 5 we are starting to get into some of the details of programming that are really important for making sure the programs are working correctly. One of the ideas in there is called the 'type' of a variable. To illustrate that we will use a couple of programs. This one here (I'll put this together), I'm going to imagine that we are working out how much change we are going to give someone. We are going to ask them what the item is that is being purchased (I'll store that in a variable), how much it cost, and how much money they tendered so we can work out what the change should be. If I were to run this program then we can load up these variable values. So the cat says 'What's the item being purchased?'
What are we buying today?

Joanne: Grapes.

Tim: Okay. Why not. What's the purchase cost?

Joanne: $9.99/kg.

Tim: Okay, so you are buying 1kg?

Joanne: Yip.

Tim: Okay, $9.99. How much money are you going to offer?

Joanne: I've got a $20 note.

Tim: A $20 note. So I'll type in '20'. Now it's asked all the information and it's stored all the variables. We can actually have a look at those variables. One way of doing things in Scratch is you can double click on things. If I double click on the 'item purchased' it's grapes, 'cost' was 9.99, and the 'amount tendered was '20'. Each of those is a different type of data. What makes it a type is what we can do with it. In particular the two numbers, the 20 and the 9.99 (being a number is a type of data), we could subtract them. We have a subtraction operator here. We want (actually how do we work this out), the 'amount tendered' ' - ' the 'purchase cost' is how much I owe you. I double click that: $10.01. With 'number' types, if the type is a number, you can subtract them, you can add them, you can

multiply them. On the other hand with ‘grapes’, I could try and add them but you can see it doesn’t really make any sense to say ‘grapes + 3’. In Scratch if I double click that it actually says ‘3’. It treats text as the number zero.

Joanne: Okay.

Tim: Scratch has got this thing where it doesn’t like to say that you’ve made an error which I think a lot of people would find that just … It really probably should say ‘hang on…’. Most programming languages have a sense of types where they say ‘I don’t think you should do this.’

Here I’ve got the equivalent program that’s written in Python. It’s got three inputs that are going to go into our three variables.

Joanne: I can see them on the three separate lines of input, input, input. What’s that word ‘float in front of it?’

Tim: That’s the type of the variable. Python wants to know what kind of value to expect.

Joanne: So numbers or words?

Tim: Yes. Without anything in front it assumes it’s a word - that it’s text.

Joanne: Okay.

Tim: But the word ‘float’ actually means that it’s going to be a number.

Joanne: Okay.

Tim: ‘Float’, ‘number’ don’t seem that related but…

Joanne: No!

Tim: It’s actually an abbreviation for ‘floating point number’.

Joanne: Okay. Still doesn’t make sense.

Tim: Fair enough. So the point is when you have the $9.99 it was 9.99.

Joanne: Okay.

Tim: It’s called ‘floating point’ because that can go anywhere. You could have 1,000,000.99 or 0.00009 and anything, all sorts of numbers.

Joanne: So if it didn't have the decimal point, can I just write just '17' there or do I have to write '17.0'?

Tim: If we don't have decimal numbers then there's another type of number called an 'integer'.

Joanne: Okay.

Tim: Integers are whole numbers. You can do slightly different things with integers than you can with floating numbers. Mainly you can do the same things but they behave slightly differently. Let's run our program. What item's been purchased?

Joanne: Grapes.

Tim: Grapes. Okay. Purchase cost was?

Joanne: 9.99.

Tim: Okay. And the amount tendered was?

Joanne: $20.

Tim: $20. Excellent. I can ask it to show what the value of that variable is. So 'item_purchased' is 'grapes'. But notice it's got inverted commas around it. That's not actually stored that way. It's just saying 'what I'm actually storing is the text'. Inverted commas are the signal in Python that we've got text. The technical name is a 'string', sometimes abbreviated 'str'. It's a string of characters, that's why it's called a 'string'. Strange names but simple concepts. Spreadsheets do the same sort of thing. You can type a word into a cell in a spreadsheet. You can type in a number. Certain things you can do with words, other things you can do with numbers. That's what the types are. I'll extend my program a wee bit. We'll work out what the change should be. That's going to be the 'amount_tendered' less the 'purchas_cost'. This is where we'll have the output. I'll get it to 'print' the value of the change. We'll run this program. Once again our item: grapes. Still $9.99?

Joanne: Yep.

Tim: Okay. Still got your $20 note?

Joanne: Yes.

Tim: And, it's just printed out the 'change'. From a user interface point of view not so good. I should have said 'please give $10.01 change' or something like that. The output of that is a 'floating point' number. We can do things to numbers such as add them (2 + 3 will give you 5), but with text we can't. We can actually add text, it's really interesting. There's a piece of text because I put the inverted commas around. If I say 'add that to this piece of text

here' ['Tim' + 'Bell'] then mathematically that makes no sense but it says that 'since it's text I'm going to do a different kind of thing.' [TimBell].

We can do that in Scratch. It's just that the command we use in Scratch is called 'join'. You can say 'join' and the two words that you want, double click it and it joins them together. If I were to 'join' two numbers like 12 and 13…

Joanne: Will it add them?

Tim: Will it add them? Well actually it will treat them as text because 'join' always works on text. So the result of joining 12 and 13 is 1213. So you can see that the type of a variable is very important for what the programming language can do with it. The types we have looked at they are the main ones you are likely to encounter at this level. 'Strings' (text) that's one type, 'integers' (whole numbers), and then 'floating point' numbers, or 'float'. One other type that is likely to come up, and that's the result of these calculations here [<]. If I say is '2 < 3' then that gives us a result but the result in this case is 'true'. The only output I can get from this is either 'true' or 'false'. If I say '5 < 3': false. That is another type, and that type can only have 'true' or 'false' as it's value. That's called a Boolean type because Boole is the guy who invented true/false logic.

Those are the main basic types that are really common. On top of that and in more advanced programming we start building our own types of data. For example you might have a type that is a person. So if I was storing information about you I would have your first name and your surname. That's two strings. That's text. Then I might have your bank account balance if it's for my banking system. That would be a number. But I have to think carefully what kind of number it is.

Joanne: A huge one, it's my bank account balance.

Tim: It's your bank account balance. The way that different types behave, even between 'floating point' and 'integers' can be complicated. That's for another day. But you can see that the type is a fundamental idea and we can build up quite complex things. A date could be made out of three numbers: the day of the month, the month of the year and the year. Each of those is an integer but not a floating point number. You can't have month number 12.1. Or at least you might decide that you are not going to allow that. So that's the role of types.

Joanne: Thank you.