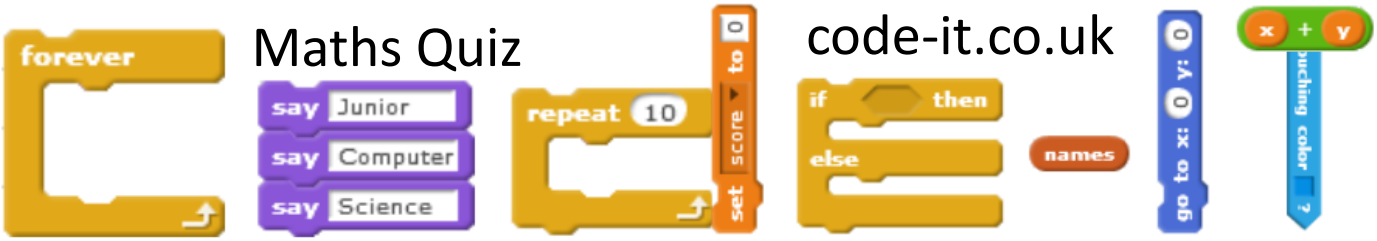


Train your computer to solve real maths challenges

Maths Concepts
-add
-subtract
-Divide
-Multiply
-Multi stage problems

Program Aim Write algorithms and convert them into programs to solve real maths challenges

Learning Path

1, Share challenge

2, Write Addition Algorithm as class

3, Convert Addition Algorithm to programming using key

4, Pupils write more algorithms and convert to code

5, Pupils express the answers in different ways

5, Pupils use their new understanding to solve interesting maths based challenges. They are using principle of abstraction to remove the calculation layer allowing them to solve harder more interesting problems.

Resources
-Class Instructions Slides http://code-it.co.uk/wp-content/uploads/2015/08/maths.pdf
-3 Plastic beakers to demonstrate with
-Whiteboard or paper recording sheet

Top Tips
Stick to low amount numbers, as it is easier to use with pencils.
Don't remove the pencils from the N1 & N2 pots to put in the T1 pot as these numbers remain in the variables.
Only input numbers when using one of the four operations. If using measures, leave the units of measure out.

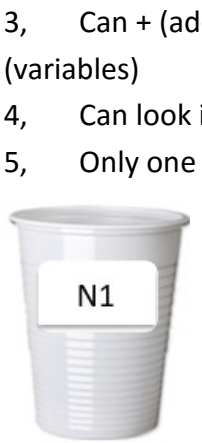
1, Share Challenge

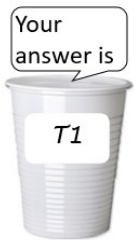
Use the slides to explain how we can train our computer to do maths, explaining that computers know nothing unless humans tell them what to do and that all programming is an algorithm turned into code. Stop at slide 10.

2, Write Addition Algorithm as a class

Use slide 10 (class instruction slides) to go over the rules

1, That all numbers must be in pots (variables)
2, All pots (variables) must have different names
3, Can + (add) -(subtract) / (divide) & * (multiply) pots (variables)
4, Can look into a pot (variable) to see what is inside
5, Only one operation per line (abstract BODMAS issues away)

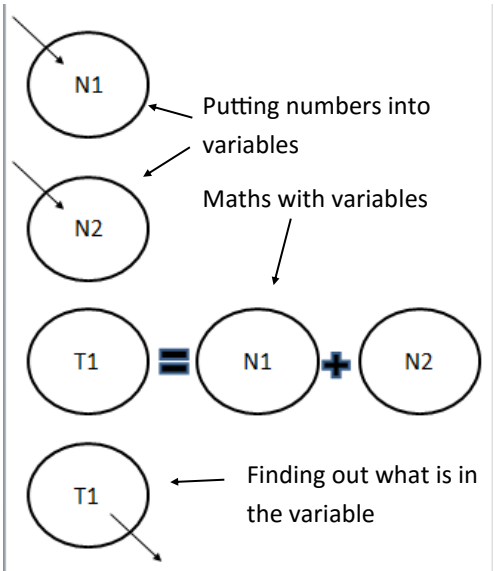




Computer Science Concepts
-Converting user input into a variable
-Writing Algorithm
-Converting Algorithm into program
-Working with multiple variables

2, Write Addition Algorithm as a class continued

Notation to create on a class board alongside the pots example

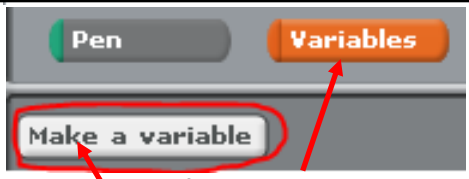


Number two pots (variables) as N1 and N2 explain that these can hold our numbers to be added. Get two pupils to hold these. On a class board add two circles with an arrow notation to put numbers into variables.

Explain that we need a pot (variable) to hold our total. Label a third beaker T1 short for total1. Create two more notes to say + and = and position them between the pots. Explain that we can now get the computer to add what is in N1 to N2 and put the result in T1. Model this with small amounts of pencils in the pots. Don't remove the pencils from N1 and N2 to make the T1 total as these stay in the variables. Draw out that this algorithm can add any two numbers by demonstrating this with different amounts. On the board add the sum as notation T1 = N1 + N2.

Now explain that we need to look into the T1 pot variable to see what is inside and add the final circle with the arrow pointing out as shown.

Give pupils time to copy this into their pupil recording sheet or whiteboard.

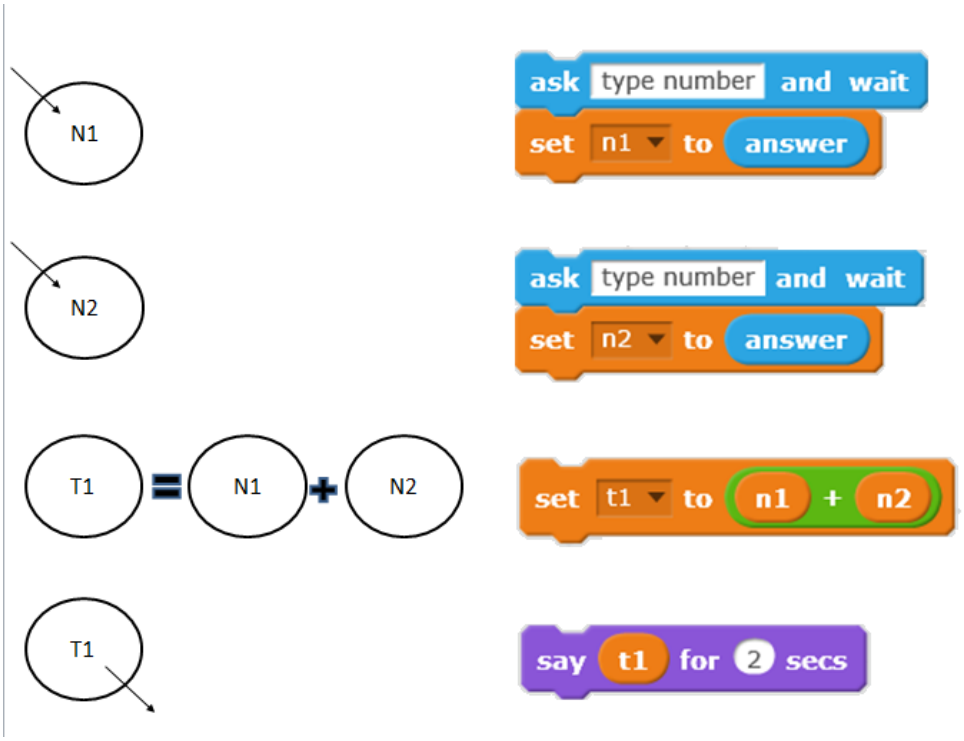


1, Left click on variables
2, Left click on make a variable
3, Name variable n1 etc

Remind pupils how to create variables

3, Convert Addition Algorithm using Key

Load Slide 11 which shows the key to converting the algorithm into Scratch programming code.
As pupils are logging on and loading a blank copy of Scratch explain that they now need to convert their algorithms into computer programs. Warn pupils that the computer language will be different from our written algorithm.



Give pupils time to convert their algorithm into Scratch code and test it.

Remind them that the order of their algorithm should stay the same in their Scratch code

Pupils need to make sure code joins up not open like the example.

Physical model with pots

4, Pupils write more algorithms and convert to code

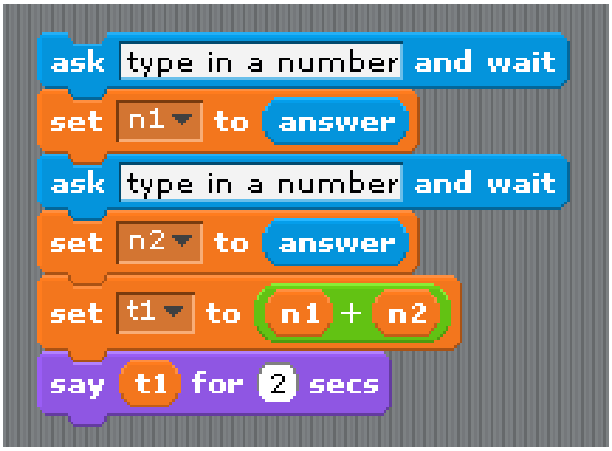
Once they have written and tested their code to add two numbers challenge them to write an algorithm to subtract, multiply or divide one number from another. They must write their algorithm and show it to you before converting it into code. It can be a good idea to force a move away from the computers to do this.

When they have finished this, move on to adding three numbers and then adding two number before multiplying by a third. With multi stage number challenges insist that they only do **one** calculation on each line as Scratch doesn't have brackets* and the order can change the outcome. Save each sub total in a new variable (t1, t2 etc). You may wish to find simple two stage problems in their maths text books and convert these to work without numbers.

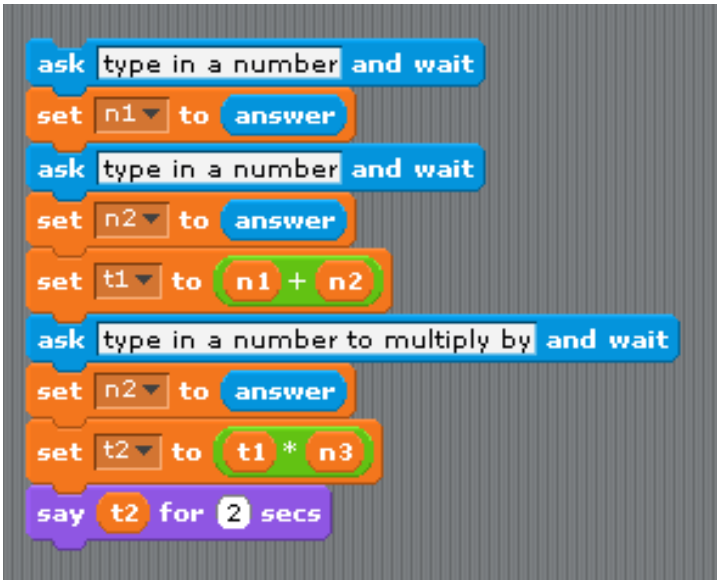
There are a few examples in the maths problems slides which you can find at http://code-it.co.uk/wp-content/uploads/2016/06/samplemathsprobs.pdf

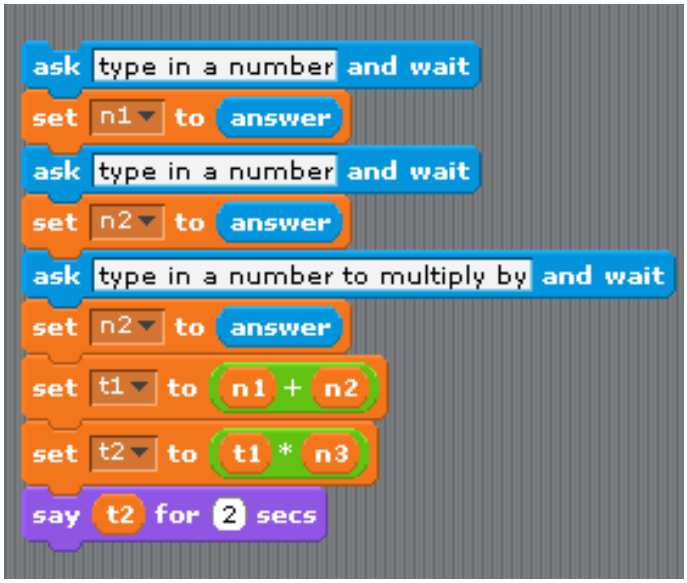
Common Bugs

Pupils only create one number into variable input block. Refer them back to their algorithm in the first place and ask them to point out what code goes with which part of the algorithm, it can help to hold their algorithm up alongside the code. Pupils use **change** the variable instead of **set** the variable. Get them to compare their code with example on the board.



Correct add two numbers code





These solutions both add two numbers and then multiply a third from the total of the first two numbers. Both solutions are correct and equally efficient as they use the same amount of code.

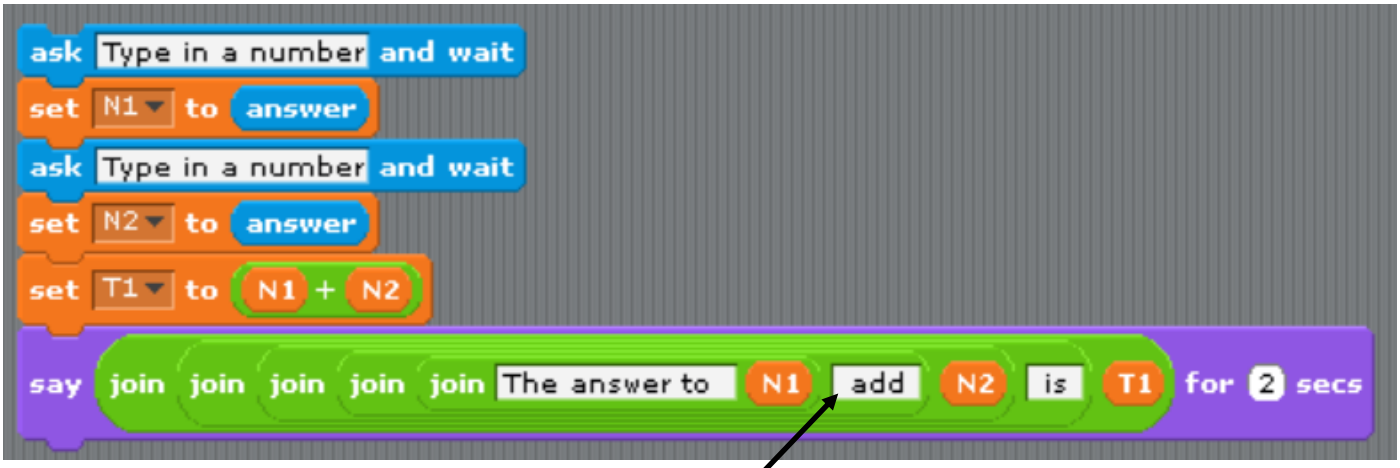
Because Scratch can cope with other mathematical operations such as greater than or less than this can be used as a form of digital literacy where pupils prove that they understand the stages of a problem by writing a program to solve them. For some children this might be the practical hook that they need to see that Maths is useful outside of their text books. **See Section 6**

5, Pupils express the answers in different ways

Go back to your pot model. Get the pupils out to be the variables and symbols. Prepare three slips of paper with **add, is** and **the answer to** on them. Get pupils to move themselves around so that it reads.

The answer to N1 add N2 is T1

Now show them how to make this in Scratch



Notice how you snap lots of joins into each other.

Can pupils express their answers is different way

NOTE how there is a space between N1 and add. This gives a space between the words in the sentence.

6, Pupils use these ideas to solve bigger real world maths problems

Why not start of with something like

How much water would it take to turn our classroom into a swimming pool or aquarium?

You can investigate volume length x breadth x height You can calculate it in Metres, CM or mm

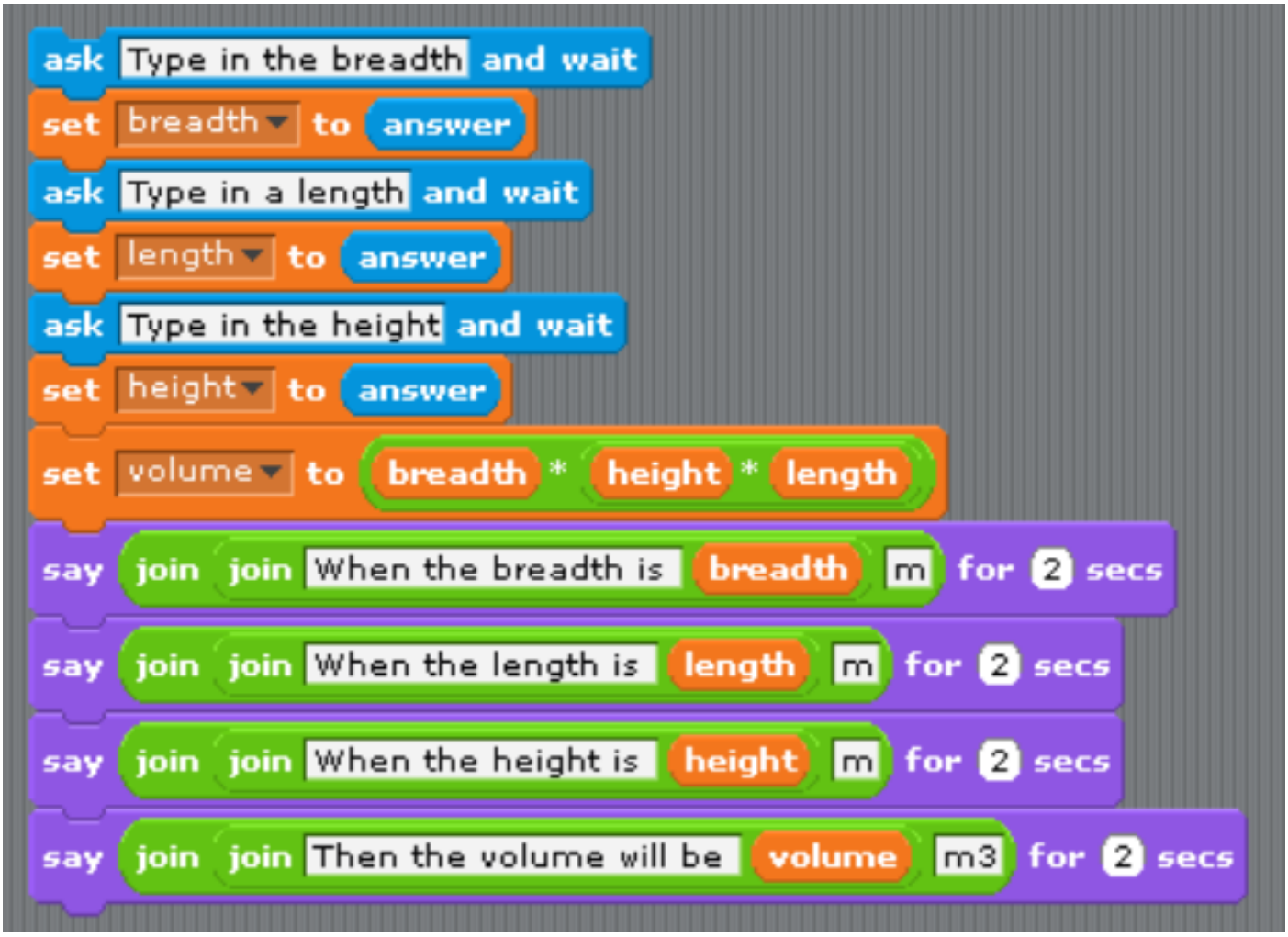
You can investigate how long it would take to fill up the classroom using a 1 litre jug.

How much it would cost. 1 cubic meter costs are online.

Your more able pupils can research what formulas to use if the room is cylindrical.

When your pupils do this they are abstracting out the manual calculations and concentrating on the higher thinking skills around solving a problem. It also helps them to see that maths has real purpose.

When pupils write programs to solve maths challenges the solutions remain with them. I have had pupils come straight out of a Maths SATS test and say they remembered the maths because they wrote a program.



Possible class swimming pool volume solution.

Once your class find out how many litres there are in a cubic metre and how long it takes them to fill one litre up and return to class they can work out how long it will take to fill the pool.

Of course you will be able to think of much better real world maths problems that match your pupils maths knowledge. If you think of any please let me know and I will add these as examples for others.

My thanks to Conrad Wolfram whose excellent keynote at the CAS conference 2016 encouraged me to adapt my work to encourage pupils to solve more open problems.