



Maths Quiz Part 1

1. Learn about, write everyday examples of & role play conditional selection
2. Learn about the ask and answer input to a variable block
3. Explore three quiz programming examples
4. Pupils plan their own quiz in Word, Publisher or on paper
5. Optional teacher demo
6. Program Quiz
7. Consider Maths quiz part 2 & 3

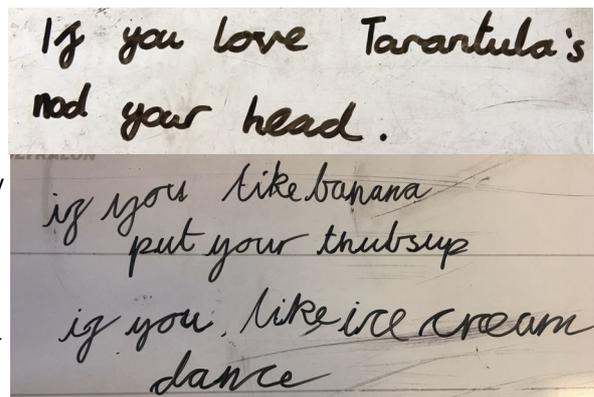
Prior to the module

- Print out a copy of conditional selection fact sheet and Programming input for each child or upload to their digital storage.
- Print out two copies of Role play conditional selection, cut up and laminate.
- Choose your classes method of planning and download the relevant templates see section 4

1. Learn about, write everyday examples of & role play conditional selection

Start with the slides. As you go through slides 2, 3 & 4 which outline each different type of conditional selection ask pupils to act out the simple everyday conditions. Ask them if they can identify which part is the condition and which are actions and share this with their partner. Once the slide shows what the everyday condition will look like in Scratch code form get pupils to write their own condition on a whiteboard or paper if you wish to keep these.

Offer pupils one point for a statement that makes sense and uses the condition that you are learning about, two points if the condition, and actions are on separate lines as shown in the slide examples and three points if pupils have indented. As pupils are writing these go round and tell pupils how many points they would get and read some of their examples to the class. The most important aspect is the meaning but the indentation used follows the indentations that many programming languages as well as some pseudocode use so gentle encouragement to work towards good forms may help



Program Aim

Design and Create a Maths quiz
Understand how to use all three types of conditional selection

Computer Science Concepts

- condition starts action
- condition stops repeated action
- Condition switches between actions
- Algorithm to code
- User typed input into program

National Curriculum Programs of Study

Pupils should be taught to:

design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts

use sequence, selection, and repetition in programs; work with variables and various forms of input and output

use logical reasoning to explain how some simple algorithms work and to **detect and correct errors in algorithms and programs**

Conditional Selection

Programming that starts or is affected by a condition

Condition starts action
If it starts to rain put your coat on.
Programming condition starts action

Condition switches between actions
If it is cold outside put a jumper on otherwise wear a t-shirt

Condition switches between actions
forever loop means the condition is checked over and over again.

Condition stops repeated action
Keep jumping until I say stop.

Condition stops repeated action
Turn right 15 degrees until X button is pressed
Repeat until is a loop

Only one action can be run (if or else)

The forever loop means the condition is checked over and over again.

Slide 5

Drag out an ask and answer block. Left click on the ask block and a long text input bar will appear on the screen. Type into the text box and when the tick is pressed or return is pressed on the keyboard the text will be input into the answer variable.

Left click on the answer block and it will show what information it is holding.

Information will also be shown on screen if the answer tick box has been selected.

The answer block is really a special type of one use variable. Explain that you are going to use this to collect the users answers and check to see if they are correct later. There is a support sheet you can give to pupils to build up their programming knowledge about inputs.

4. Explore three quiz programming examples

Before they can write their own maths questions it helps to see how someone else has used all three conditional selection blocks with the text input to write quiz questions.

Explain to the class that they are going to use the same

them in KS3/4.

Slide 5 includes some simple Scratch examples of the three types of conditional selection outlined earlier. Getting pupils to go through and answer some simple questions about these is a next step towards moving from the everyday to the programming specific. E.g. What would happen if the key right arrow is not pressed in the condition switches between actions example (It would turn left continually 15 degrees at a time).

Ask pupils to get into pairs and give out a the role play cards one for each pair. Can they take it in turns, one person acting as the condition and the other person acting out the programming. Can they act out the programming without the forever loop and with the forever loop?

Swap cards regularly so everyone gets a chance to reason how these work.

2. Learn about the ask and answer input to a variable block

The other complexity that we are introducing is the idea of inputting information that a user has typed and storing it in the special answer variable.

Ask and answer input blocks

conditional selection to design and program their own maths quizzes.

Get them to either open file <https://scratch.mit.edu/projects/230350130/> from Scratch online or open the template file you have saved on the network or use the example you have on slide 7.

Can they match the types of condition selection they have learnt earlier with the ones they have seen?

Go through in detail what happens if they get the question right or wrong with every example?

What part of each script is the condition, text input and selection? What does the = do? (checks to see that the text input from the user in answer is the same as the right answer) What part are outputs (say commands) Could any other outputs be added? (Sounds are the easy ones)

Are there any blocks that they have not seen before?

Green = operator block

Formative Assessment

At any point if you suspect that some pupils are not sure how this works use the conditional selection question cards starting with a general example before moving onto more specific examples.

If there are pupils who are still struggling they might be candidates for the supported planning worksheet in the next phase of the lesson.

4. Pupils plan their own quiz in Word, Publisher or on paper

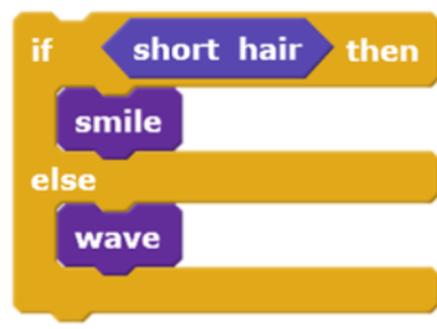
There are four different options for your class to use.

Plan on the computer using the Publisher Template called *Publisher Pupil Planner as a ZIP* This document allows pupils to drag their chosen algorithm template and edit it to create their chosen type of questions.

Plan on the computer using the Word Template called *Word Pupil Planner.docx*. Pupils will have to type out the whole algorithm.

Plan on paper using the PDF document called *PDFPupilPlanner.PDF*

There are also **scaffolded options** but save these for pupils who really struggled with the formative assessment cards.



- 1, If you have short hair what will you do?
- 2, If you don't have short hair what will you do?
- 3, Point to the condition.
- 4, What does this conditional selection block do?
- 5, When this code is run, how many times will it check the condition?
- 6, What could you do to check the condition over and over again?
- 7, Which blocks are outputs?



Formative Assessment Questions

Ask $6 \times 6 = ?$

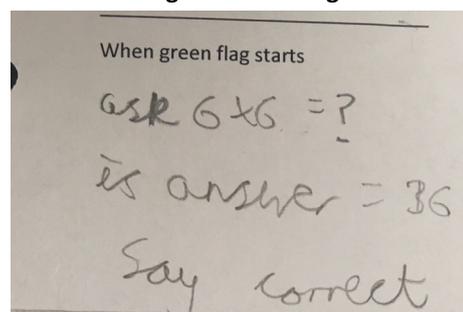
If answer = 36

Say Well done
Play sound Cheers

Else

Say wrong this time
Play sound rain drops

Algorithmic Design



Hand written example

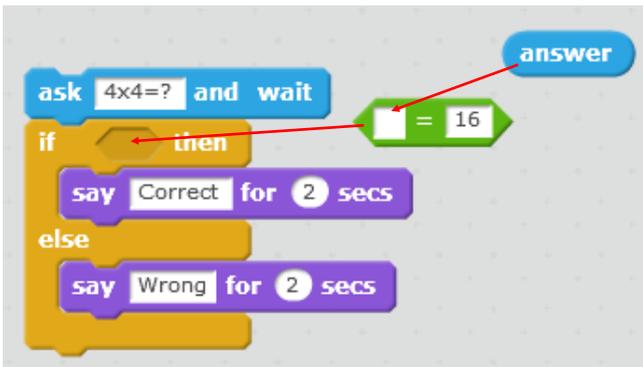
Word Pupil Planner Scaffold

WordPupilPlannerPrint.pdf

Appearance Design

There is also space to plan what the stage and sprites will look like. We call this appearance design and it is important that pupils get to think about this aspect as well as algorithmic design.

While they are planning this is a great time to assess more pupils understanding using the conditional selection question cards mentioned above.



5. Optional teacher demo

Some pupils understand the concept but struggle with the order blocks snap together inside each other. Demonstrating some of these can be helpful.

6. Program Quiz

Give pupils plenty of time to program their quiz and turn their algorithmic and appearance designs into reality in their programs.

Ask questions about their code. Does it match their planning algorithm? It is ok if they adapt their planning and do it differently from their algorithmic planner? This is fine as long as they know why they made changes.

7. Consider Maths quiz part 2 and 3 which introduce variables

Part 2 introduced a name variable which can be used to add personalisation and part three adds a score.

Please see over the page for problem solving attitudes used.

Draft 2.0 17th September 2018

Phil Bagge



What computational attitudes have they used during this module?



Inspired by Behaviour Rubric created with [@MarkDorling](#) and linked at <http://code-it.co.uk/attitudes/>