



KS2 maths quiz - adding a score activity

Using variables in programs

Recommended Year Group: Any key stage 2

Activity Duration: 1 hour

Concepts and approaches



Variables



Selection



Programming

Curriculum links

Computing

- work with variables
- write and debug programs that accomplish specific goals
- use logical reasoning to detect and correct errors in programs

Maths

Lower Key Stage 2

- The principal focus of mathematics teaching in lower key stage 2 is to ensure that pupils become increasingly fluent with whole numbers and the four operations, including **number facts** and the concept of place value.

Upper Key Stage 2

- At this stage, pupils should develop their ability to solve a wider range of problems, including increasingly complex properties of numbers and arithmetic, and problems demanding efficient written and mental **methods of calculation**.

Introduction

In this activity pupils learn about variables and how they can be used in programs. Pupils learn to use variables in Scratch to make a scoring system for the maths quiz they made in the introduction to selection activity ([KS2 Use selection in programs – maths quiz](#)).

Prior knowledge

Pupils should already have an understanding of algorithms and be familiar with Scratch. This could be through completing the following activities:

[KS1 understand what algorithms are](#)

[KS2 Scratch tinkering](#)

[KS2 Viking raid animation for upper key stage 2](#)

[KS2 Dinosaur fossil animation for lower key stage 2](#)

Note – pupils should have also completed:

[KS2 Use selection in programs – maths quiz](#) – as this activity builds upon the maths quizzes that pupils made. *However, an example maths quiz Scratch file has been provided for any pupils who don't have access to a quiz created in that activity.*



Pupil objectives

- I can explain what a variable is
- I can use variables in a program

Resources

- A projector to display Scratch and the presentation as you work through this lesson.
- MIT's [Scratch 2.0](#) or [Scratch 1.4](#). (Please refer to [this guide](#) on the ways to download and use our Scratch resources in your school).
- Pupil access to the maths quiz programs pupils created in [KS2 Use selection in programs – maths quiz](#). For pupils without access to their maths quiz, use the example Scratch file ([see download link at bottom of webpage](#)) or Scratch 2 version online at [example maths quiz](#).
- Maths quiz score presentation (see download link at bottom of webpage).
- Scratch commands help sheet – Up to one per pair if required (see download link at bottom of webpage).
- Timer extension sheet – up to one per pair if required (see download link at bottom of webpage).
- Maths quiz extension Scratch file (see download link at bottom of webpage) or Scratch 2 version online at [maths quiz extension](#).

Before you start

- Read the 'Teaching Notes' for this activity below.
- Download the resources required for this activity from the list above.
- Load the presentation ready for the lesson.
- Print the paper based resources required, as explained above.
- Ensure that Scratch is working on your school's computers and the files for this activity are downloaded or accessible.
- Ensure pupils have access to their maths quiz they created within the [KS2: use selection in programs activity](#), or the example maths quiz Scratch program provided with this activity.

Organisation

- This activity is organised with pupils working in pairs. These should be the same pairs that pupils worked together in for [KS2 Use selection in programs – maths quiz](#). Each pair will need access to a computer running Scratch and access to their maths quiz (or the example file provide for pupils who did not complete the selection activity). If you do not have a sufficient number of computers to make this possible, you may wish to group pupils in 3s or deliver this lesson with half of your class at a time.

Introduction (5 mins)

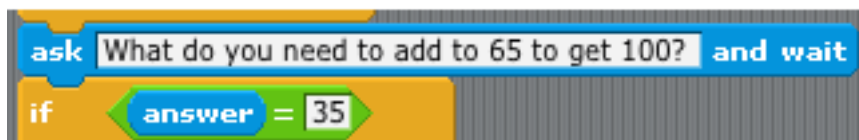
- Remind pupils that previously they created a maths quiz in Scratch which asked a series of questions and responded to the player's answers. Ask pupils if they can recall what type of command they used to get the program to respond to the player and how this worked? [Selection](#) commands. Pupils used the If...then... and the If... then...else... commands. These use a condition being met or not met to trigger different commands.
- Load one of the pupils' maths quiz files or use the example maths quiz on the interactive whiteboard.



- Run and play the quiz (select to display in full screen to hide the code with the correct answers) to remind pupils of how it operates.
- Ask pupils for suggestions for ways in which they could improve this quiz. If your school subscribes to any game-based learning software prompt pupils to consider how their quiz and these might differ. Guide a discussion to determine one improvement would be by having a scoring system of some kind.
- Explain that in this activity pupils will be adding a scoring system to their maths quiz and to do this they will use **variables**.
- Introduce the learning objective on slide 2 of the presentation if this is your normal practice.

Introducing variables (5 mins)

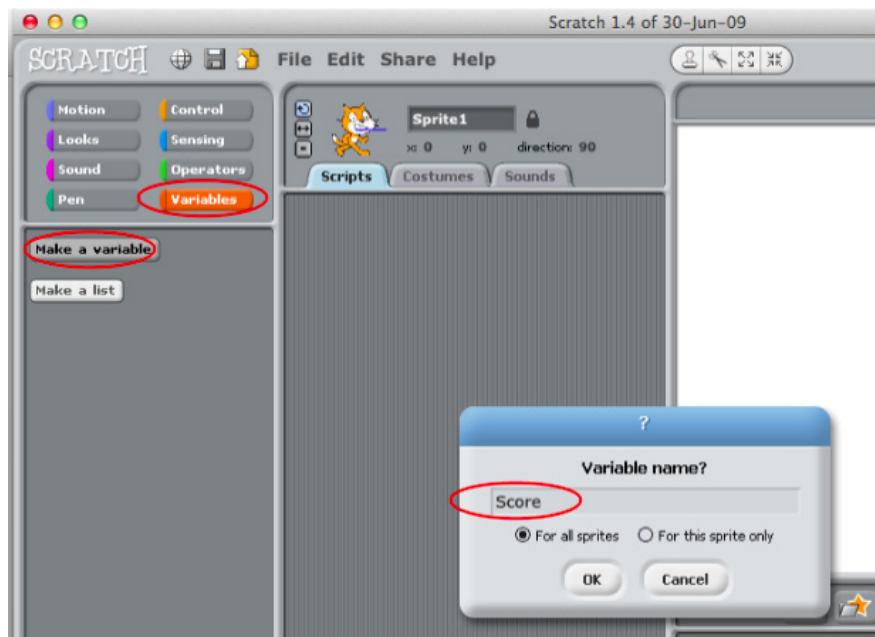
- Guide a class discussion to elicit if pupils have any existing understanding about variables: Have pupils heard of the term variable before? Perhaps they heard the term used in other subjects, such as science? What can pupils recall about a variable in science? *It is something we change.*
- Depending on pupils' existing understanding, use slide 3 to briefly explain the concept of variables, covering the following points:
 - Variables are one way in which computer programs can use part of a computer's memory to store, retrieve or change simple data, such as numbers.
 - We can think of a variable as a box within our program that we can place data in. We can then use the data when we need it within our program. The data in the variable might change as our program runs (similarly variables in science are things we change).
 - We can look to see what the data in our variable is at any time. Mention to pupils that when we look at the data in our variable we do not take it out of the box, it is still there as a value (see Teaching Notes – common misconceptions about variables below).
 - Ask pupils to think about what information is entered into the maths quiz program by the player when it is run? The player's answers. Explain that this information is held in a temporary variable called 'answer' – so pupils have used a variable in their quiz already



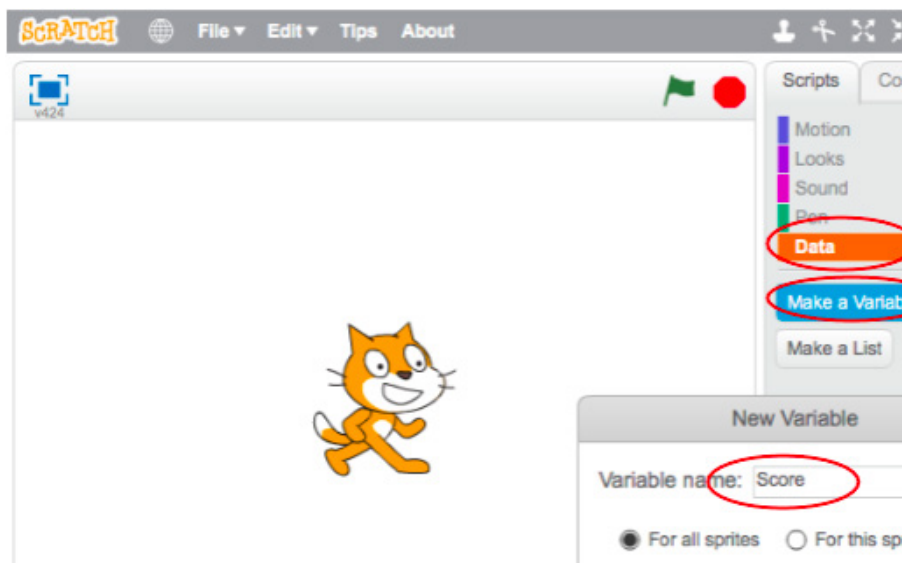
The 'Ask and wait' command stores the answer to the question in a variable called 'answer'.

Introducing variables in Scratch (10 mins)

- Ask pupils if they know how to create a variable in Scratch. If any pupils do, let them demonstrate to the class the steps below or the equivalent.
- Demonstrate that to create a variable in Scratch you select the variable tab (Scratch 1.4) or data tab (Scratch 2) from the command block palette and select 'Make a variable'.

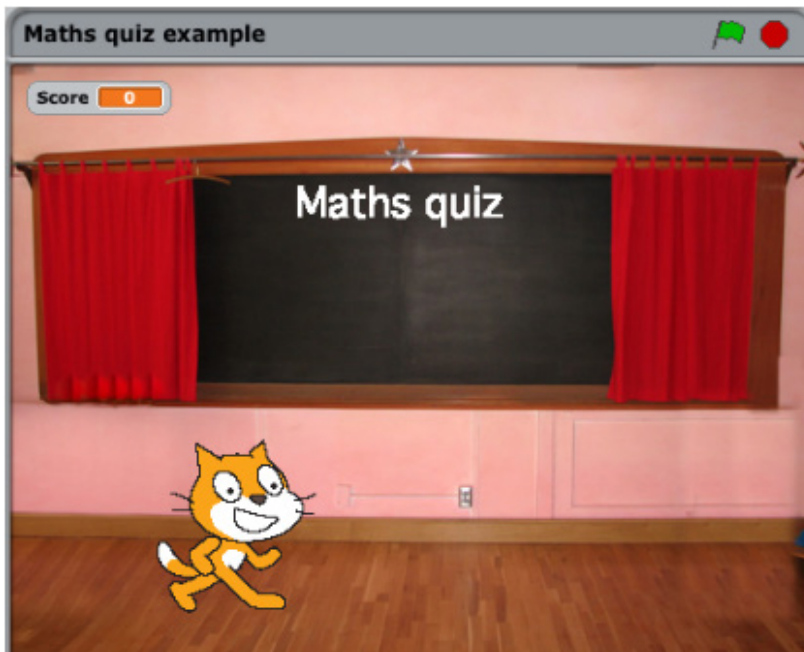


Creating a variable in Scratch 1.4



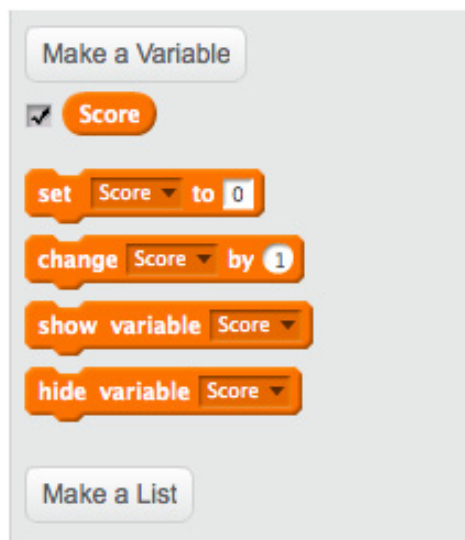
Creating a variable in Scratch 2

- Show pupils that they then need to name the variable. Ask pupils why they think it is important we give the variable a sensible name, such as 'score' as you have just modelled as opposed to just 's'? *To help us to remember what the variable is for within our program.*
- Click OK and highlight that once we have created the variable 'score' this now appears within the stage area, as shown below. Remind pupils that this variable allows us to store data, such as a number, in the computer's memory, and use this within our program. Can pupils see what number currently is in the variable score? 0



Once you have created a variable it is displayed in the stage area with its value.

- Show pupils that once we have created a variable, a list of commands now appear within the variables tab, as shown below. Ask a pupil to read out the labels on the commands and lead a short discussion with pupils on what they think the commands might do?



The commands available within Scratch which can be used to manipulate variables

- Ask pupils how they think the commands could be used in the maths quiz, covering ideas such as how they could be used to change the number of points you score for answering a question correctly or the points you lose for answering incorrectly, for example. Add that you aren't going to show exactly how to use these commands as this will be pupils' challenge shortly.



- Explain to pupils that the process above can be repeated to create another variable. For example, we might want to have a 'Lives' variable that changes during the quiz also.

Main task: pupils create their scoring system (30 mins)

- Ask pairs of pupils to open their maths quizzes from the location they were saved. For any pupils who do not have a maths quiz demonstrate how to locate and open the example maths quiz from the your school network or by accessing it online at this link: [maths quiz example](#). (Pupils using the example maths quiz should save their own copy of this).
- **Pupil task (5 mins):** Give pairs 5 mins to run their maths quiz programs and explain to each other how their program works to recap their learning from the previous activity. Write the following questions up to guide pupils' discussion during this time:
 - Can you explain to your partner:
 - The overall algorithm on which your program is based?
 - What each command in your program does?
 - Why you have used each command?
 - Which is a selection command?
 - How the selection command works?
 - Why the selection command requires a condition? What condition have you used?
 - What happens when the condition is/is not met?
- **Pupil task (5 mins):** Explain to pupils that before adding the scoring system they need to design it, and create the rules to say how it will work. Ask pupils if they can remember what we call a rule that makes something happen? (algorithm). Pupils should have 5 minutes to make notes describing how they want their scoring system for their quiz to work, including writing the rules. Slide 4 provides prompts for pupils to consider during this activity.
- **Note:** pupils scoring systems might be as simple as just having one variable 'Score' with one algorithm: If correct add 1 point and if incorrect subtract 1 point. However, pupils can also make a more complex scoring system with a number of variables, 'Score' and 'Lives' for example, with different algorithms for different questions – the player loses more lives failing to answer a simpler question correctly for example.
- **Note on initialisation:** a prompt for pupils to consider that their scoring system must be reset each time their program is run has been left off slide 4 as a miniplenary discusses this during the main task, however see if any pupils mention this in their scoring system design (See Teaching Notes for more information on initialisation).
- Ask a selection of pupils to feedback to the class their scoring system design and guide a short discussion to identify how groups' ideas are similar/different.
- **Pupil task (15 mins):** Pupils should now be given time in pairs to add their scoring system to their maths quiz. Slide 5 & 6 (depending on the version of Scratch you are using) of the presentation can be displayed to remind pupils of how to create a variable during this task.
- Provide support to pupils who require it. A Scratch commands help sheet has been included with this activity which provides a recap on how to create variables and hints on using the variable commands. This also includes a short explanation of how to make variable values reduce using a negative value – lower key



stage 2 pupils may require some additional explanation.

- Remind pupils that it can be useful to run their program as they are writing it to check it does as they expect and matches their algorithm design. This might be after adding each new command, or after a group of commands. If it doesn't do as they wished they should debug their program. Write the four points below up as a guide to help with debugging, these steps are explained in greater details [here](#).
 - Identify what the program should do?
 - Observe what the program does do?
 - Use your knowledge of the Scratch commands to work out where it went wrong.
 - Fix it.
- **Mini-plenary on initialisation (5 mins):** Once pupils have had a good opportunity to start the main task, stop pupils and ask the following questions in order to guide a brief discussion around initialisation:
 - Who's managed to get their variables changing in value in their quiz? Does anything strange happen when you play the quiz again and again? *Some pupils variables will continue with the values they ended with when the program was last run.*
 - Can anyone think what we might need to do at the beginning of our program if this is happening? *Set the initial values for our variables.* What commands could we use for this? *'Set variable to'.*
 - Can anyone remember from previous activities (Viking raid animation for example) what we call the process of setting things at the beginning of a program? Can anyone recall other examples? *Initialisation – other examples include setting the starting position of sprites in animations.*
- Pupils should now have time to finish programming their scoring system for their maths quiz.
- If pupils complete the main task an extension has been provided, see below. However before attempting this, pupils can also be encouraged to develop their scoring system further if they haven't already done so, by adding 'Lives' that reduce when players answer incorrectly, or by having additional 'Bonus' points that can be scored for example. Alternatively, pupils could also try adding a simple timer to their quiz – a sheet to support pupils in tackling this challenge has been included.

Extension

- A Scratch maths quiz extension file has been provided which encourages pupils to tinker with a more complex use of variables. Pupils explore a quiz that ends if the player runs out of lives. Pupils could be asked to add comments to this file, to explain how the code works and how variables are used. They could tinker with the code which changes the variables and see how this changes the game when it is played. Pupils can then share what they found out either with another pair or with the whole class.

Plenary

- Explain to pupils they will now get the opportunity to play each others' maths quizzes and as they do so they will use logical reasoning to work out how the game has been programmed.
- Pupils should have time to play several other groups' quizzes. Each time they do,



they should:

1. Play the quiz in full screen mode, as shown below.
 2. Make a prediction about the variables and code used to alter the variable values based on their experience in playing the quiz, e.g. *"I think you've added a variable called 'playerscore' and when I get the answer right the score goes up by 2. You've done this using a 'Change score by 2 in the if ... then block..."*
 3. Check their predictions with the program's code to see if they are correct.
- Finally, use some of the following questions (slide 7) to guide a class discussion:
 - How would you describe a variable in a program?
 - What variables have you created for your program?
 - How have you used variables in your program?
 - When and how have your variables changed their values in your program?

Differentiation

Support:

- An additional adult, if available, may work with a small group of pupils to provide additional support when required. The Scratch commands help sheet can also be provided to those pupils that require it during the main tasks.

Stretch & Challenge:

- An extension activity has been provided above for pupils completing the main task.

Assessment opportunities

- Informal, teacher assessment of progress during main task, class discussions and plenary.
- Formal, summative assessment of Scratch projects if required (note however these are completed in pairs).

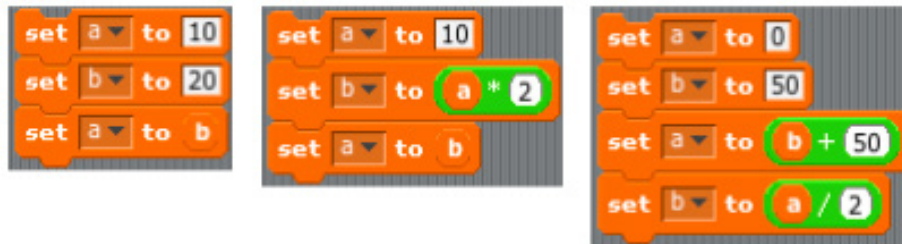
Area	Questions	Ideas for evidence
Variables	What is a variable? Why would you use variables? How have you used variables? How do your variables change as your program is run? Why is it important to set the variable values at the start of your program?	Pupils can give a concise explanation of what variables are. Pupils have used variables correctly in their programs. Pupils can explain how and why they have used variables within their program when questioned or during class discussion. Pupils included commands in their programs to set initial values for variables. Pupils are able to make accurate predictions about the code used in other pupils' programs during the plenary.
Selection	What is selection? Which commands within your program are selection commands? How is selection used in your maths quiz?	Pupils can explain how their maths quiz uses selection and highlight the selection commands. Pupils will add commands to alter the values of variables within the correct position within the selection commands within their program i.e. Increasing score when the answer is correct.



- Guidance on assessment areas appears in the table below:

Reading code

- Another way to explore pupils' understanding of variables, along with their ability to use logical reasoning to determine the outcome of a program, is to provide short sections of code and ask pupils to determine what the final values for the variables will be, as shown below.



Can pupils determine what the final values will be in the variables 'a' and 'b'?

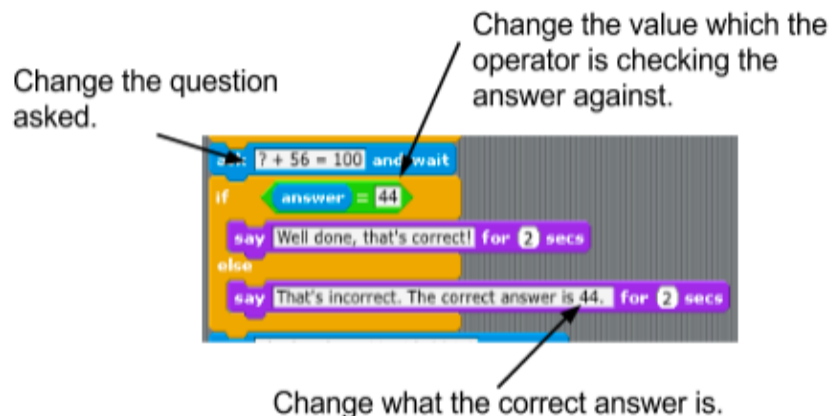
Teaching Notes

What is initialisation?

If the starting conditions of a program are different each time it is run, so too will the output differ. It is therefore important to include commands to set starting conditions. This might, for example, be setting the starting position and direction of a sprite in an animation or game. The process of ensuring a program starts with the same initial conditions is called initialisation. Initialisation is covered in this activity as we need to specify the initial starting values for our scoring system, e.g. setting 'score = 0' and 'lives = 3'. As if we neglect to include this code, a player might find themselves starting the game with no lives and the previous player's high score.

Adjusting the sample maths quiz

For pupils who didn't create a maths quiz within the KS 2: Use selection in programs activity, an example maths quiz Scratch file has been provided. The level of maths questions within this quiz targets number bonds to 100. As such, depending on the mathematical ability of the pupils using this program, you may wish to change the questions. Even better would be to get pupils to change the questions themselves before completing this activity. The image below briefly recaps how such changes can be made. Only provide this to pupils if they need support, as they should be encouraged to use logical reasoning to determine the changes to be made.



Altering the example maths quiz if required.

Concepts and approaches

Variables

In computing, variables are one way in which computer programs can access the computer's memory to store, retrieve or change data. They may be thought of as a box into which we can place data and then use, change or view this data as we need to as our program runs. In this activity pupils create variables to store data as part of a scoring system for their maths quiz.

Common misconceptions about variables

Pupils sometimes think that by reading a value from a variable that the value is removed from the variable. This is not true. When we read a value from a variable the value remains. It is as though we just looked to see the value. Pupils sometimes think that by changing a value in a variable that they are adding a second value to the variable. This is not true. When we change or update a value in a variable only one value remains in the variable, the new value. The old value no longer exists. This is like 'rubbing out' the old value and replacing it with the new value.

Selection

Selection is a concept used in algorithms and programs to choose which commands are executed (meaning they are run). Whether commands run depends on whether certain conditions are met. In this activity pupils review how they used selection in their quiz to execute different commands depending on whether the player's answer is correct or not (the condition).

Programming

In this activity pupils create a scoring system for a maths quiz program in Scratch. In doing so they use their knowledge of the Scratch programming language to implement the algorithm they have written describing how their scoring system will work. Programming is the process of implementing an algorithm as code.



Taking this further

Pupils could use their experience from this activity to make quizzes for a range of subjects from across the curriculum. The variable 'answer', which stores the user's response to the 'Ask..... and wait' command, can store both text as well as number data, meaning pupils' questions could ask for text responses, such as the names of countries in geography for example. Pupils in key stage 2 could create quizzes to support the learning of key stage 1 pupils.

Variables could also be used to alter the questions asked. For example a user might be asked if they want an easy or hard quiz. A variable easy_or_hard_indicator could be set according the user's response. Then within the program an IF... THEN...selection uses the variable to control the complexity of the quiz.

Pupils in key stage 2 could create quizzes to help younger pupils revise their learning. By hosting the quizzes pupils create on the school network they can be used across the school. Alternatively, by uploading pupils' quizzes to the Scratch 2 on-line community (<http://scratch.mit.edu>) they can be posted on or embedded in your school's website or blog for others to play. In order to do this you or your pupils (with parental permission) would need to sign up for a Scratch community account, which is free.

Related activities

[KS2 Scratch tinkering](#)

[KS2 Use sequence in programs Viking raid animation for upper key stage 2](#)

[KS2 Use sequence in programs dinosaur fossil animation for lower key stage 2](#)

[KS2 Use repetition in programs shapes and crystal flowers](#)

[KS2 Use selection in programs maths quiz](#)