BY NC ND

code-it.co.uk

Maths Quiz Part 2 Personalising the Quiz using a variable

1. Learn about how variables store and retrieve information
2. Plan and create a simple personalising example
3. Plan how to add personalisation into the quiz
4. Adapt the quiz to include personalisation including testing and debugging new features
5. Learning summary and skills developed

Prior to the module

- Print out a copy of the variable fact sheet and for each child or upload to their digital storage.
- Make sure pupils have access to their maths quiz planning from part 1. If you haven't done part one do it first.
- Download the slides to help you with section 1 or access the videos part 1 and 2.
- Print out the variables role play cut up and laminate if possible

Program Aim

Collect the name of the user and assign it to a variable before using the variable to personalise the quiz

Computer Science Concepts

-Variables are used to hold information

-The computer can display the variables value by reading the label

-A variable can be used many times

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

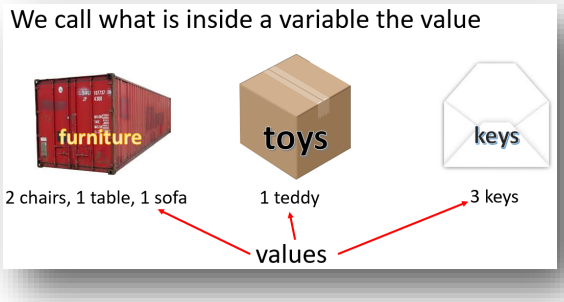
use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

Rationale for this planning

We know that variables are a difficult concept for pupils to understand. This planning only attempts to explain a small part of the concept before pupils get to plan and use it within their quizzes. Part three will deal with collecting and presenting a score variable which will extend the complexity by introducing the idea that a variable can be changed after it has had a value assigned.

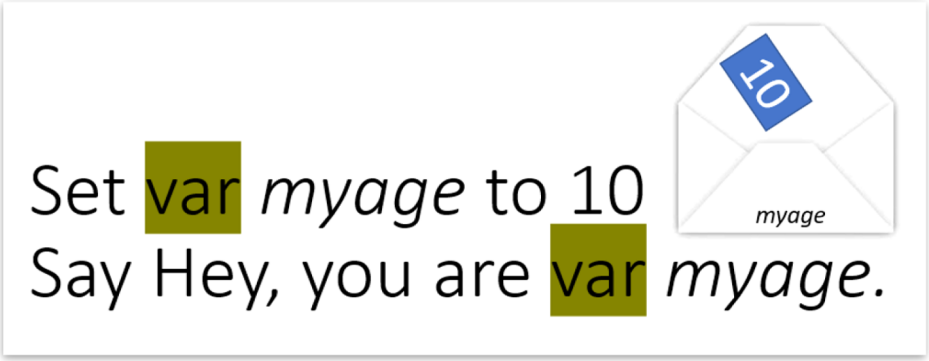
1. Learn about how variables store and retrieve information

Explain to pupils that they are going to adapt their quizzes so that every user has the question and feedback addressed to them individually. So instead of 5x5=? We might have what is 5x5= Dave? Ask them what the problem with that might be? Exactly, no one knows the names of every user that might use the quiz. Today they are going to learn about variables so that we can achieve this individual personalisation without typing every name in for every quiz.



You can use the slides as described here or use the new videos

parts 1 and 2. Open the slides. Slides 2, 3 and 4 introduces what variables are and establishes that variables have two parts. The name or label which the program uses to identify it and the value that is attached to the label.



Slide 5 introduces the idea that when a programming language reads a label it displays the value attached to it. We use a written algorithm to do this and introduces the idea of writing VAR before each variable so they can be identified as variables in a text algorithm. We recommend that teachers explain that we name variables after the information that is assigned to them and if we use two words we combine them.

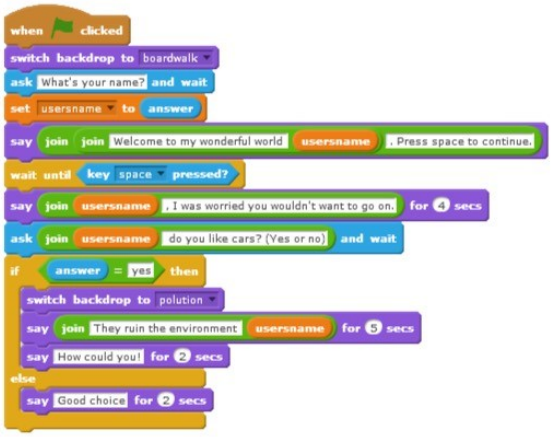
Create three variables with the following labels and your chosen values

Label	var bestfriend	var favnumber	var footballteam
Value	(you choose)	(you choose)	(you choose)

Last night I ate var *favnumber* pies.
I hate watching var *footballteam* on TV as they always lose by var *favnumber* goals.
I am thinking of changing my name to var *bestfriend* because it sounds much better than my name.

Slides 6-13 introduces the idea of setting a variable to assign it a value before using the variables in simple sentences. Pupils are given an example to learn from and then are asked what the sentence would read for each example. The answers are on the next slide.

Slides 14 & 15 are a fun exercise where pupils create three values before reading them in a sentence with a funny outcome. They are then invited to write their own simple sentence which uses a variable.



2. Plan and create a simple personalising example

Slides 16 & 17 outline how pupils can create a simple paragraph of text using one name and then adapt this to accept a variable. You can then create one using Scratch as an example and a reminder of how to create variables. Data tab in Scratch 2.0 and variables tab on Scratch 1.4. Or you can use slides 18 and 19 to go through the process. They can then turn their paragraph into code similar to slide 19. It can be wise to re-

vise the text input and output block as well.

3. Plan how to add personalisation into the quiz

Slide 21 outlines this approach. Pupils examine their original algorithm planning that they made in part 1. They look for words or phrases that could be adapted to use a persons name. They could underline these words or highlight them and add notes if there is room.

4. Adapt the quiz to include personalisation including testing and debugging new features

Pupils need access to their adapted, highlighted algorithmic planner and the example program they examined earlier. How much personalisation can they achieve? Which part of the algorithm should not be personalised? (The condition as personalising it would stop it from working)

5. Learning summary and skills developed

Pupils can choose which learning attitudes they have developed from the list on the last page and the last slide on the PowerPoint.

Optional Extra Activity Slide 20

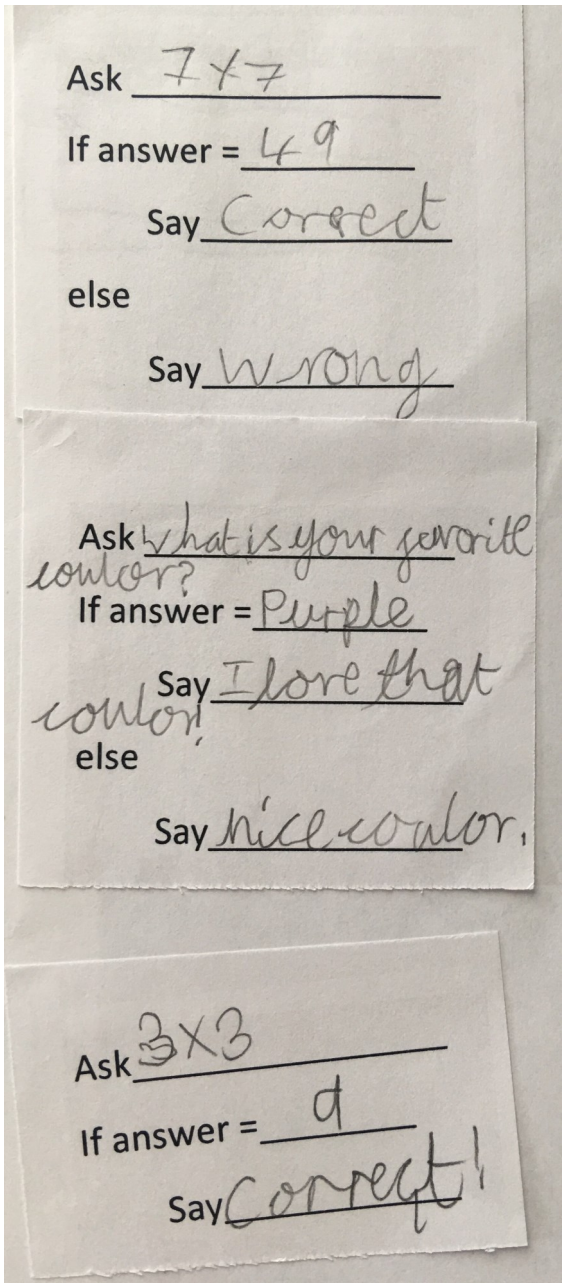
I have included the extra example on slide 20 so that you can work with any pupils or small group of pupils who are still struggling with the basic concepts. This could also be used as a way of waking up pupils brains, as an algorithmic thinking starter.

Waking pupils algorithmic brains up from week to week

I have included a simple role play activity for pupils to use as a lesson starter for week two or three. If you like it better than some of the starter activities you could swap them round. This is on slide 22 and is available as a download.

Why not try part three where we introduce a score into our quiz.

Phil Bagge Draft 1.0 17th September 2018



I recognise there is more than one way to solve/describe a problem
I don't just accept the first solution
I look for a range of solution to the same problem
I look for how a project can be extended
Handles Ambiguity
Open Ended Problem Solver
I can break complex problems into parts
I can evaluate my solutions against a set criteria
I can design criteria to evaluate my creations
Evaluates
Insert picture of your students here
Copes with Complexity
I can discover / concentrate on the most important part of a problem
I can identify patterns in problems & solutions
I can contribute useful ideas to a partner or group
I can encourage others to share their ideas
Communicates
Adapts
I can adapt existing ideas to solve new problems
I can develop, test and debug until a product is refined
I lead using all the people talent in my group
Investigates
I make predictions about what will happen
I learn from setbacks and don't let them put me off
Perseveres
I can persevere even if the solution is not obvious
I repeatedly experiment through predicting, making, testing & debugging
@baggiepr
Code-IT
Inspired by Behaviour Rubric created with @MarkDorling and linked at http://code-it.co.uk/attitudes/