Variables

Introduction

You will aim to learn the following objectives and keywords during this lesson.

Learning Objective	❥ To describe what a variable is ❥ To demonstrate the use of variables
Keywords	❥ Variable ❥ Store

Setting the Scene

Concept	How it is used
A variable is a storage location within the code, where the value can be changed and stored to be called upon.	When you write code, it is not always straightforward as aspects of the code may change. A variable is like a box, you put a value inside and when it is needed the box is opened and the value used.

Can you think of when you have remembered something that you have used in other situations?

Activity 1

Scenario 1	Scenario 2
You have received some money for your birthday, and you are going shopping. ❥ How do you know how much you have? ❥ How do you know how much you have left after the first purchase? ❥ How to you know when you have spent it all?	You have decided to buy the latest game console but need to do some price searching? ❥ Where can you buy it from? ❥ When you look at one shop how much is it? ❥ How do you know which the cheapest option is?

Where else do you remember a value to use again?

Small Group Activity

In small groups, play a quick couple of games!

What did you learn after playing?

You use **_variables_** without thinking on a daily basis. You may find a time where you must remember (**_store_**) a value and recall it when asked – how about maths equations or science experiments?

Activity 2

The blocks available for **_variables_** are located within the 'variables' icon.

The default for the name of the variable is 'item' and this can be changed by clicking the dropdown list and selecting 'rename variable'.

It is good practice when naming a variable, to use suitable names to ensure you know which variable to use within the code; calling them variable1, variable2, variable3 etc will confuse you as the programmer.

Why should you use a suitable name for a variable?

How can we get MiRo to use a variable to help him move around the arena?
We are going to create and use a variable to store what MiRo senses.

What is a sensor and how can we use it to stop MiRo falling off the table edge on the simulator or bumping into objects?

The sensor on MiRo is constantly reading what is around it and storing this information. We need to set a place to store this information so we can use it; we are going to create a variable.

Can you write some instructions out for the algorithm to do this below?

1 ....

2 ....

3 ....

     a. ....

     b. ....

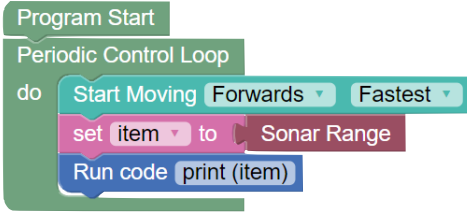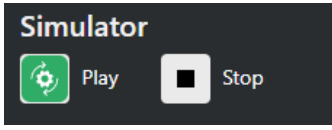*What would the name of your variable be?*
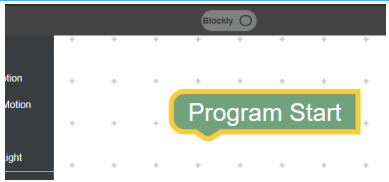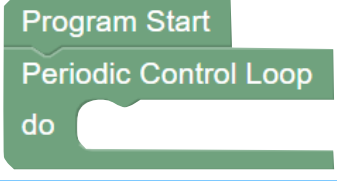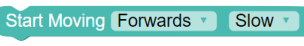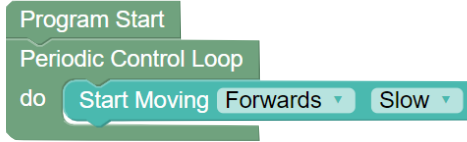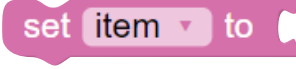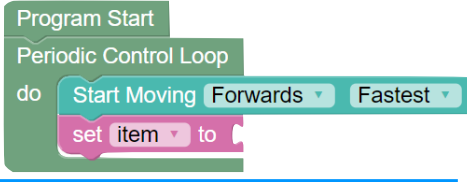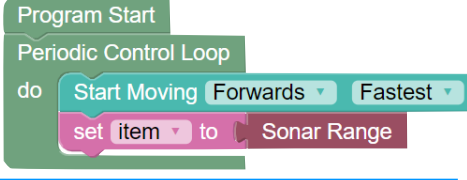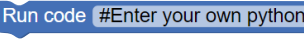
Why is the name of the variable just as important as the content?

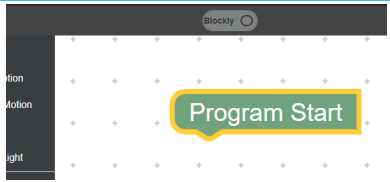**Using this algorithm as your plan, create and run the program in the MiRoSIM.**
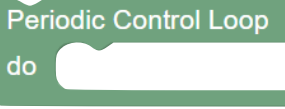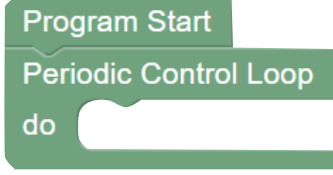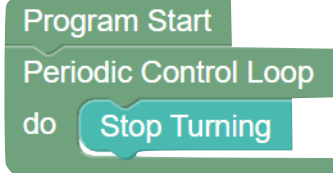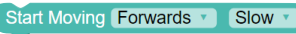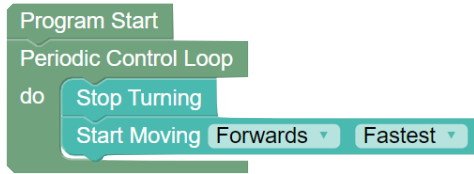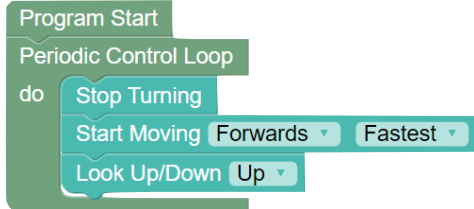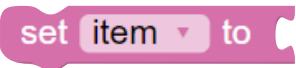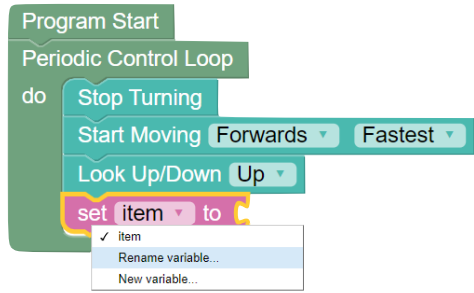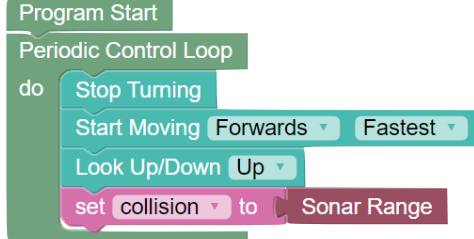
Summary Self-Assessment

PART 1 Step-by-Step - Use MiRo simulator to output the range detected by the Sonar Range Sensor.

Step	Block	Category	Image
1 Add 1 x **Program Start** block	Program Start	🔑 Setup	Program Start
2 Add 1 x **Periodic Control Loop** block	Periodic Control Loop do	🔑 Setup	Program Start / Periodic Control Loop do
3 Add 1 x **Move Forwards** block	Start Moving Forwards Slow	Simple Motion	Program Start / Periodic Control Loop do Start Moving Forwards Slow
	Change speed from slow to fastest.		
4 Add 1 x **Set Item** block	set item to	Variables	Program Start / Periodic Control Loop do Start Moving Forwards Fastest set item to
5 Add 1 x **Sonar Range** block	Sonar Range	Sensors	Program Start / Periodic Control Loop do Start Moving Forwards Fastest set item to Sonar Range
6 Add 1 x **Run Code** block	Run code #Enter your own python	Code	Program Start / Periodic Control Loop do Start Moving Forwards Fastest set item to Sonar Range Run code print (item)
	Change text to '*print (item)*'		
7	Click **Simulator Play**		Simulator ▶ Play ■ Stop
	Follow a Program Script at bottom of screen shows code running and the initial values of the sonar range is output each time MiRo moves.		User script log 0.336808651686 1.0 1.0 1.0 1.0 When the Sonar Range detects an object, observe how the sensor is now outputting a '1.0'

PART 2 Step-by-Step - Simulate MiRo to move around and stay within the area given.

Step		Block	Category		Image
1	Add 1 x **Program Start** block	Program Start	🔑	Setup	
2	Add 1 x **Periodic Control Loop** block	Periodic Control Loop do	🔑	Setup	
3	Add 1 x **Stop Turning** block	Stop Turning	⬌	Simple Motion	
4	Add 1 x **Move Forwards** block	Start Moving Forwards Slow	⬌	Simple Motion	
	Change speed from slow to fastest.				
5	Add 1 x **Look Up/ Down** block	Look Up/Down Up	⬌	Simple Motion	
6	Add 1 x **Set Item** block	set item to	⏱️	Variables	
	Rename the variable to 'collision' by: 🐰 Click the arrow next to the word 'item' 🐰 Select 'rename variable' 🐰 Rename variable to: 'collision'				
7	Add 1 x **Sonar Range** block	Sonar Range	◉)))	Sensors	

Step		Block	Category	Image
8	Add 1 x **If Do** block		🧠 **Logic**	
2	Add 1 x **Equals** block		🧠 **Logic**	
3	Add 1 x **Collision** block		⏱ **Variables**	
4	Add 1 x **Maths '0'** block		➗ **Math**	
	Change from '0' to '1'			
5	Add 1 x **Stop Moving** block		✥ **Simple Motion**	
6	Add 1 x **Set Item** block		✥ **Simple Motion**	
	Change speed from slow to fastest			

Step	Block	Category	Image
7	Click **Robot** OR **Simulator Play**		
8	**Follow a Program** Script at bottom of screen shows code is ready and waiting for the 'clap'	The clap button or a physical clap will need to be heard by MiRo at the appropriate time for the program to proceed. The script at the bottom will show the program has ended.	

	Extension Click the button above the block code with the word 'Blockly' on it and switch the screen to 'python'. Can you narrate anything that happens in the code?		