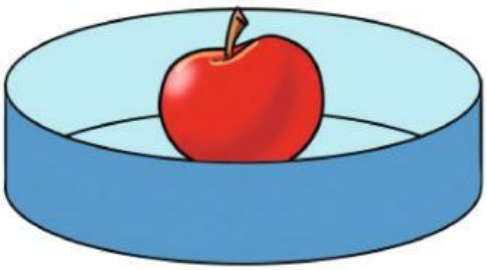
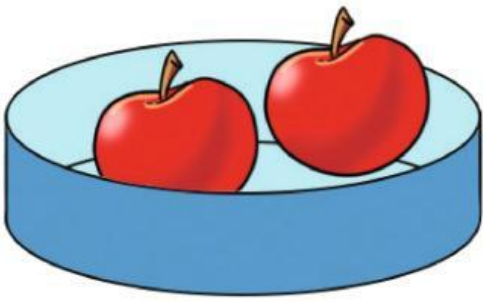
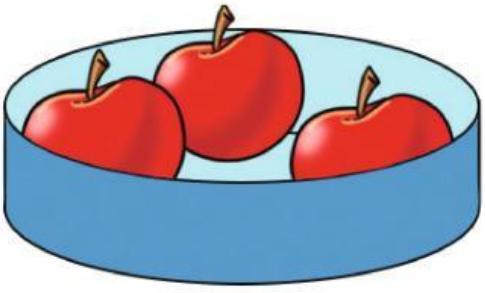
Algorithms to code: Understanding variables

Put the correct number of containers and apples in the table.

	Containers	Apples
a 		
b 		
c 		

1 What changed?

2 What remained the same?

Cambridge Primary Computing Stage 5 © Hodder & Stoughton Limited 2023

LIVEWORKSHEETS

Algorithm to code

Understanding variables

Keywords

constant: a value that does not change

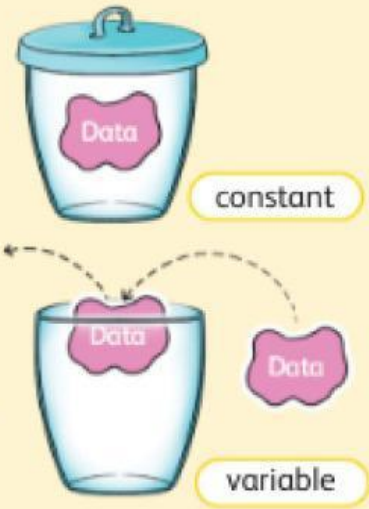
variable: a value that can change depending on certain conditions

Learn

There can be different types of data in algorithms and programs. For example, data can be numbers, text, dates and so on.

Data that does not change while a program or algorithm runs is called a constant. A constant may be typed into a program or assigned a name to identify it.

A variable is a value that can change when the program runs. Think of a variable as a container that holds different data at different times. For example, a variable may hold the value 7 at one point but later in the program, it holds the value 3.



Variables can be used to store numbers such as the score in a game, a count of something or the value of calculations. Variables can also store text such as people's names and addresses.

Look at the example below where one variable, called Name, is being used in an algorithm that displays a Hello message to the user.

Example of an algorithm with a variable

Step	Instruction
1	Start
2	Create a variable called Name
3	Ask user to enter their name
4	Set variable, Name, to user's input in Step 3
5	Join 'Hello' + space + the variable name together
6	Output the result
7	End

In this algorithm, the user can enter any name. The variable stores whatever the user enters. The message that is displayed combines the word "Hello" with the data that the user enters.

For example, if the user enters "Jack", then the display message would be "Hello Jack". If the user enters "Jill", then the display message would be "Hello Jill". Each time the program is run, the value of the variable can change.

	User's input	Program's output
1st time program runs	Jack	Hello Jack
2nd time program runs	Jill	Hello Jill



Variables allow us to store and change data as an algorithm or program runs.

LIVEWORKSHEETS

We must ensure that we create variables with clear names. It is important to use a clear and meaningful name for a variable so that:

- it correctly describes the variable's purpose
- it is easy for anyone else to understand what your program or algorithm is doing
- we can easily tell the difference between multiple variables.

Variable names should be unique and not too long. A variable's name should be exactly the same every time it is used in an algorithm or program. You need to take note of which letters in the name are uppercase, which are lowercase and if there are spaces.

Practise

1 Write an algorithm that displays a message to the user. The program should ask the user to enter their eye colour and then display the eye colour that was entered.

Step	Instruction
1	Start
2	Create a variable called _____
3	Ask user to enter _____
4	Set variable, _____, to user's input in Step _____
5	Join "You entered" and _____
6	_____
7	End



2 Create a variable in Scratch by following the steps below.

a Choose a clear name for a variable to be assigned to text that a user enters as their name.

b Open a new project and choose any sprite.

c Click on the **Variables** group.

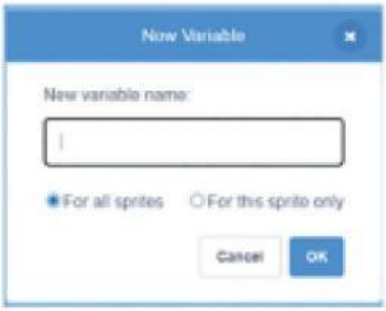
d Click **Make a Variable** and type the name of your variable.

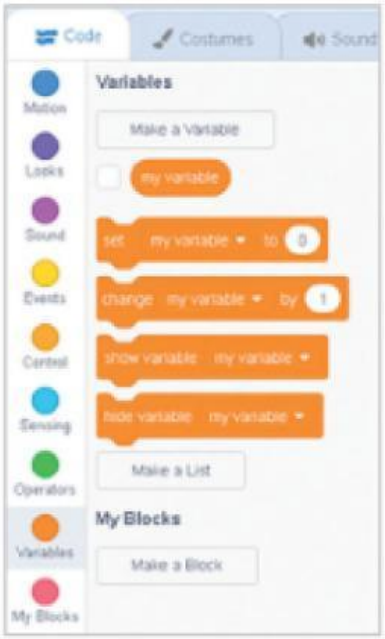
e Click **OK** and check that the variable you created is listed.

The image shows a variable created called **Name**.









LIVEWORKSHEETS

Algorithms to code: Understanding variables

Drag to the red or blue box the missing word.

A variable is a value that can change when the program runs

Data that does not change while a program or algorithm runs

Constant **Variable**

Cambridge Primary Computing Stage 5 © Hodder & Stoughton Limited 2023

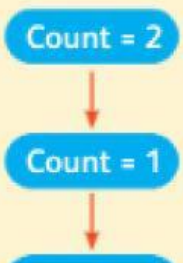
LIVEWORKSHEETS

Variables in programs

Learn

We can create a variable called **Score** to store a player's points during a game. When the player scores a point, we can tell the program to increase the variable by one. As you collect more points, the value of this variable keeps changing.



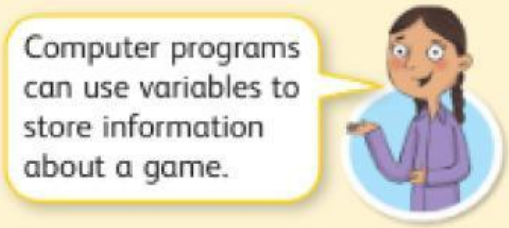


We can also create a variable called **Count** to store the number of lives a player has. Every time a player dies, we can decrease the variable by one.

If the total number of lives is 2, then the variable can change from **2 to 1 to 0** and then the game ends.

Example of a variable in a program

Here is an algorithm for a **Duck** Sprite bouncing on a trampoline in a computer game. The algorithm includes a variable called **Score**, which stores the player's points. The first part of the algorithm sets the variable **Score** to 0. The second part of the algorithm increases the score by 1 every time the space key is pressed and the **Duck** jumps.



Duck Sprite (Part 1)	
1	Start program when the Green Flag is clicked
2	Create a variable called Score
3	Set variable, Score, to 0



Duck Sprite (Part 2)	
1	Start program when the space key is pressed
2	Change y position by 40
3	Wait 0.2 seconds
4	Change y position by –40
5	Change variable, Score, by 1



We can create this algorithm in Scratch. We can create a variable called **Score** in Scratch. The code for the **Duck** Sprite is shown above.

The two sprites in this program are **Duck** and **Trampoline.** The **Trampoline** Sprite does not do anything. It is a static object and so it has no code.

We can see that if a player keeps pressing the space key, the score keeps increasing – from **0 to 1,** from **1 to 2,** from **2 to 3,** and so on.



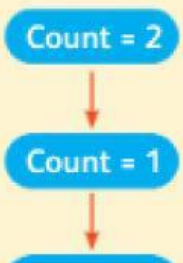
LIVEWORKSHEETS

Variables in programs

Learn

We can create a variable called **Score** to store a player's points during a game. When the player scores a point, we can tell the program to increase the variable by one. As you collect more points, the value of this variable keeps changing.



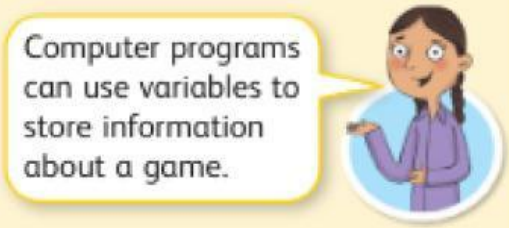


We can also create a variable called **Count** to store the number of lives a player has. Every time a player dies, we can decrease the variable by one.

If the total number of lives is 2, then the variable can change from **2 to 1 to 0** and then the game ends.

Example of a variable in a program

Here is an algorithm for a **Duck** Sprite bouncing on a trampoline in a computer game. The algorithm includes a variable called **Score**, which stores the player's points. The first part of the algorithm sets the variable **Score** to 0. The second part of the algorithm increases the score by 1 every time the space key is pressed and the **Duck** jumps.



Duck Sprite (Part 1)	
1	Start program when the Green Flag is clicked
2	Create a variable called Score
3	Set variable, Score, to 0



Duck Sprite (Part 2)	
1	Start program when the space key is pressed
2	Change y position by 40
3	Wait 0.2 seconds
4	Change y position by –40
5	Change variable, Score, by 1



We can create this algorithm in Scratch. We can create a variable called **Score** in Scratch. The code for the **Duck** Sprite is shown above.

The two sprites in this program are **Duck** and **Trampoline.** The **Trampoline** Sprite does not do anything. It is a static object and so it has no code.

We can see that if a player keeps pressing the space key, the score keeps increasing – from **0 to 1,** from **1 to 2,** from **2 to 3,** and so on.



LIVEWORKSHEETS

Practise

1 Create the program on the previous page as follows.

a Open a new project in Scratch.
b Add any backdrop.
c Search and select the **Trampoline** Sprite and position it in the middle of the stage.
d Select the **Duck** Sprite.
e Create a variable following the steps on page 48.
f Add the code on the previous page to the **Duck** Sprite.
g Test your program and check that you get the required results.





When you run your program, you should notice that the Score value on the screen increases every time you press the space key.

2 Create a program in Scratch with a variable assigned. It should do the following:

- The **Jordyn** Sprite should move one (1) step when the right arrow key is pressed and it should add one (1) to the count every time this happens.
- The program should start counting from 0.
- The **Basketball 2** Backdrop should be used.

a Choose a clear name for your variable and state what value is assigned to it.
b Write an algorithm for the sprite for:
  - when the Green Flag is clicked
  - when the right arrow key is pressed.
c Open a new project in Scratch.
d Add the correct backdrop.
e Add the correct sprite.
f Create the variable you chose in **a.**
g Create the code for your algorithm.
h Run the program and check the results.



LIVEWORKSHEETS