Unit 4: An Introduction to Programming for Teachers
Technoteach

This Task Book contains tasks that are designed to be completed during the course. You should aim to get all tasks completed. These tasks should develop your understanding of core programming concepts, which will help you not only with Scratch, but with programming in general.

You already have some experience in using Scratch from the previous sessions when creating our Maze games. If not, these can be found in the Resources Folder.

Tasks 1 –

These tasks should get you used to the Scratch user interface and more familiar with the various blocks available and putting them together.

- **Task 1.1 –**
  Open Scratch and get a sprite to say "Hello World!" when the green flag is clicked

- **Task 1.2 –**
  Adjust your code so the character says "Hello World!" when the space key is pressed

- **Task 1.3 –**
  Add a second character to your game and use a "broadcast" block to get this character to automatically say something in response to the first character. Hint: Remember when using broadcasts, you need to use a block for receiving the message.

- **Challenge Task 1.4 –**
  Get a character to calculate and say the result of the expression:

$$\frac{42 \times 67}{89 - 4}$$

  Hint: Remembering the order of operations is key here i.e. BODMAS/BIDMAS/PEMDAS (whichever one is familiar)

- **Challenge Task 1.5 -**
  Get a character to calculate and say the result of the expression:

$$\sqrt{5} \times \sqrt{6}$$

  Hint: Can you find a square root function?

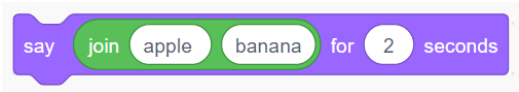
Tasks 2 – Variables

Variables are used to store data in programs. You can then refer to the data by the variable name such as “age”, “name” or “colour” etc. rather than the actual values of each.

- **Task 2.1 –**
  Store your name in a variable called name and store your age in a variable named age.
  Get a character to say the following:

  My name is <name> and my age is <age>. In 2 years, I will be <age + 2>.

  Where the angular brackets represent the variables.
  Hint: You will need to use a combination of “say” and “join” blocks to combine a message with variables. You can use a join block to stick two things together like this:



- **Task 2.2 –**
  Imagine you are running a tuck shop, each sweet you sell costs £2. Write a program that asks the person using your program to enter a number of sweets and then shows them the total cost of the sweets.
  You may want your program to do something like this:

- Task 2.3 –

A shop is finding it tough to count how many apples they have left. Write a program to help them work out how many they have. Your program should:

  - Ask how many packs they have
  - Ask how many apples are in each pack
  - Say how many apples they have in total

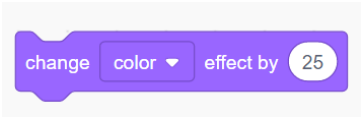
- **Challenge Task 2.4 –**

You go out for a walk with 4 of your friends and play a game to see who can collect the most pinecones on your walk. You collect 8 pinecones, all of your friends collect the same amount as each other but you’ll have to ask them how many it was. Write a program that calculates the total amount of pinecones you’ve collected.

Tasks 3 – If Statements:

If statements are a way to make decisions in programs.

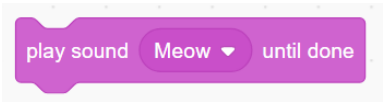
- **Task 3.1 –**
  Ask the user to enter a word, if they enter the word "magic" then change the character's colour. Hint: You can find the change color effect block in looks.

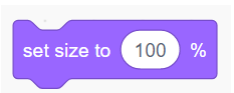


- **Task 3.2 –**

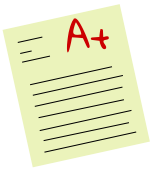
  Ask the user of your program to write their school year. If their school year is less than 7, then say they are in Primary School, otherwise say they are in Secondary school.

- **Task 3.3 –**
  Ask the user of your program to write their favourite colour. If they enter "blue" get the cat to make a meow sound. If they enter any other colour, change the size of the cat.



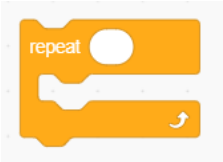


- **Challenge Task 3.4 –**
  Ask the user to enter their score from their test, if their score was more than 17 say "Amazing, well done!". If their score was 17 or less then check if their score was more than 10, if so say "Not bad! Try again!", otherwise say "You need to study more!"

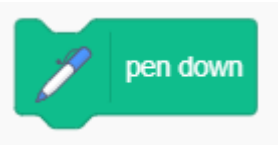


Tasks 4 – Iteration (Repeats and Loops)

You can use loops to repeat any instructions that you might want to happen more than once. This is known as iteration.



- **Task 4.1 –**
  Get the character to say "Hello" for 2 seconds, 5 times.



- **Task 4.2 –**
  Using the Pen extension, make the character draw a square, using a repeat block. Think about which instructions you would need to draw a square and how many times you need to repeat instructions to make it.

- **Task 4.3 –**
  Make a timer that counts down from 30 seconds until the time reaches 0. It should only count down once per second!

- **Task 4.4 –**
  Let the user enter words and get the character to say them, if they typed "stop" then stop the program, but if they say anything else then get the character to say what they typed.

- **Challenge Task 4.5 –**
  Get the character to say all the even numbers between 0 and 10 (including 0 and 10).

- **Challenge Task 4.6 –**
  Get the character to sing the 99 green bottles song. Think about how you would change just the number each time so every time they sing it reduces by 1.

- **Challenge Task 4.7 –**
  Using loops, create a disco in Scratch with one or more characters all dancing to music and possibly get different colour lights involved too!

Tasks 5 – Lists and Loops

Lists can be very useful ways of storing multiple bits of data together which we can then refer to through the list itself, instead of having to use a lot of variable names to store each bit of data.

- **Task 5.1 –**
  Create a list called names in scratch and add 5 names to the list.

- **Task 5.2 –**
  Get a character to say each of the names in the list, one after the other.

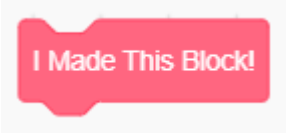
- **Task 5.3 –**
  Write a program that inserts all the numbers from 0 to 10 into a list called numbers.

- **Challenge Task 5.4 –**
  Write a program that makes the character sing (just through text) Old McDonald had a farm, using a list to store all the animals and another list to store all their sounds.

- **Challenge Task 5.5 –**
  Write a program that has a list of Questions and a list of answers and asks the user these randomly. If the user is correct the character should say correct, if not it should say they were wrong. (Extension: Try adding a score variable which increases every time they are right.)

Tasks 6 – Functions

A function is a small section of code which is used to perform a specific task. Functions are very useful whenever your program performs a similar task in different places as you can define that function once and reuse it as many times as you'd like, without needing to re-write each step of the function again.



- **Task 6.1 –**
  Make a move block which allows you to move the character in any of the 4 directions.

- **Task 6.2 –**
  Make a Question block which allows you to input the question and correct answer and gets the character to reply if you are correct or not. Use your block to get your character to ask 3 different questions.

- **Task 6.3 –**
  Use the pen extension to create a block which draws a square. The block should let you choose how big the square is.

- **Challenge Task 6.4 –**
  Edit your block so that it can make any regular shape, not just a square.