In the Classroom!

by
Brendan Smith, Camden Education Trust, Ireland
# Table of Contents

Project 1 – Add, Subtract, Multiply & Divide ....................................................... 3
Project 2 - Building a Calculator ........................................................................ 12
Project 3 - Drawing Shapes.................................................................................. 27
  Exercise ............................................................................................................. 31
  Other Shapes ..................................................................................................... 31
Project 4 - Pick and Choose you own Polygon Shape ...................... 35
Project 5 – Why Trees are important ................................................................. 39
  Scene 1: The Introduction ................................................................................ 42
  Scene 2: Trees & Oxygen ................................................................................ 51
  Scene 3: Trees & Food ..................................................................................... 54
  Scene 4: Trees & Wildlife ............................................................................... 55
  Scene 4a: Trees & Human Health ................................................................. 60
  Scene 5: Trees, Air Pollution & Urban Heat .............................................. 65
  Scene 5a: Trees, Flooding & Soil Retention .............................................. 69
  Scene 6: A World without Trees ................................................................. 71
Project 6 – A Wildlife Quiz ............................................................................... 80
  Exercise .......................................................................................................... 99
Project 7 – A Geography Quiz .................................................................... 102
  Requirements ................................................................................................ 103
  Exercise .......................................................................................................... 108
Project 1 – Add, Subtract, Multiply & Divide

Mathematics is the science of the study of numbers, shapes and patterns, dealing with the logic of quantity, shape and arrangement. It is the building block for everything in our daily lives including mobile devices, architecture, art, finance, engineering and sports.

Mathematics is divided into three main branches:
• Arithmetic
• Algebra
• Geometry

This project will be based on:

Arithmetic
which is about numbers and their basic operations of addition, subtraction, multiplication and division

Algebra
the part of mathematics in which letters and other symbols are used to represent numbers and quantities in formulae and equations.

We will show the children how to build an interactive programme that will add and subtract numbers that are inputted. Then we will allow them the opportunity to enhance the programme by providing both a multiplication and a division element.

So let’s start!

First upload Scratch

Delete the cat sprite.

Go to the Sprites Library

Select an appropriate sprite from the People folder such as
Position the sprite two thirds up from the bottom of the stage on the left-hand side.

If the sprite is facing the wrong direction (away from the rest of the stage), then go to Costumes and select Flip Horizontal.

Now we start to build the script.

Go to Events category and select the block.

Drag and drop into the Scripts area (also known as Workspace).

Go to the Variables folder.

Ensure that the children understand the definition of a variable in mathematics, namely a generalised representation of a value (number) that can change within the context of a mathematical formula. Typically, we use a single letter or symbol to represent a variable number. So we can use X and Y to represent the numbers that we are going to create.

In algebra, the use of letters such as X, Y, A, B helps us to generalise the formulas and rules that we write.

Select Make a Variable option.

Type in X under variable name and choose For all sprites.

The X box will appear in the workspace at the top left-hand corner under the Green Flag icon. Move it to the top right-hand corner of the screen.

Drag and drop into the Workspace under and connecting to the previous block.
Make another variable

Follow the same procedure.

Once again select the *Make a Variable* option.

Type in Y under variable name and choose *For all sprites*.

The Y box will appear in the workspace at the top left hand corner under the Green Flag icon. If you have already moved the X box to the top right hand corner of the screen, repeat the process with the Y box and position it under the former.

Drag and drop the `set Y ▼ to` block into the Workspace under the

The sprite needs to say a few words to briefly explain what the project is all about.

Go to the *Looks* category, select the *Say* block twice, drop both blocks into the script and input a few words of introduction.
Go to the **Sensing** category. This folder contains blocks of code that will allow the user to interact with the programme.

![Ask block](image)

Select **ask** and drag and drop it to the script.

Type in the text *Input a number in the X box* into this block.

Go to the **Variable** category

![Set block](image)

Select **set** and choose X from the drop-down menu and place this block at the bottom of the script.

Go to **Sensing**, choose **answer** and place it into the white box of the aforementioned block.

Repeat the process once again of **Ask** (Sensing) **Set** (Variable) and **Answer** (Sensing) blocks with the text this time stating *Input a number in the Y box*.

When this is completed, go to the **Looks** category.

![Say block](image)

Select and drop the **say Hello! for 2 seconds** block into the script.

Replace the word **Hello!** with *What is the addition of these two numbers?*

Place another **say Hello! for 2 seconds** from the **Looks** category at the bottom of the script. Increase the seconds from 2 to 3.

If you expect the children to answer verbally before it appears onscreen, then input a higher number.

We need now to locate a block that will allow the operation of addition between the two separate variables to take place.

Thus, we go to the **Operators** category.
Select and drop it into the first white box of the last Say block

Type into the first white box *The answer is*

Then go return to Operators, select the block and drop it into the second white box

Go to Variables, select and and place them appropriately into

Increase the seconds from 2 to 4.

Go to full screen and press on the Green Flag icon

The first of four arithmetical operations has been included. So save the programme by going to File in the top bar and choose the option Save to your computer in the drop down menu.

Give the file an appropriate name before saving

Now we move onto subtraction.

To do so, first we place the cursor onto the block. Select the top left of the mouse or the relevant part of the keypad and drag slightly downwards. This separates the Green Flag from the rest of the script.
Hover over the disconnected large set of blocks, top right click on the mouse/keypad and select *duplicate*.

Re-connect the first sets of blocks to the Green Flag block and connect the new set of blocks to the end of the script.

**Split the script at**

```
say "In this project, we going to add, subtract, multiply and divide numbers" for 6 seconds
```

and remove this specific block of code by dropping it anywhere in the **Blocks’ Palette** area.

Replace it with

```
wait 1 seconds
```

and reconnect the blocks into the script.

**Change 1 to 2 in the block**

Change the wording of the text *First we will add two numbers to Let’s subtract!* (or similar text)

**After the amendments have been completed, run the programme to see if it works.**
However, as the exercise ends after the single subtraction is done, we can input a *Forever* loop so that the programme keeps on running.

Before doing so change the text

![Code block](image)

So that the narrative reads better in the script as below

![Code block](image)
If the script works fine, now is the time for the children to experiment.

If they are of sufficient age and level of knowledge, get them to undertake the operations of multiplication and division as part of the programme being enhanced. Tell them to revisit what they and you have done together before starting the process. Give them a period of time to complete the exercise.

The additional blocks of code should be placed in the Forever loop and look similar to

Once the time period is up, get one or more of the students, who have successfully completed the exercise, to come to the top of the class and explain step by step what they have done. If the classroom has an interactive whiteboard or a projector, get the volunteer(s) to guide the class through the process on the big screen.

After these tasks are completed, as a novelty piece you can get the project to *talk*.
Click to the Addition icon located at the bottom left hand corner of the Scratch Interface.

In the Add Extension that now appears, select Text to Speech and return to the Scratch Interface.

In Text to Speech, select and replace Hello with What is the addition of these two numbers?

Change the set voice to and set language to that required.

Substitute with

Run the script.

This will obviously excite the children. However, it is best to leave the introduction of speech until the end of the class as it can become overused and whilst it will be great fun for the students it can be a very noisy distraction!

By the Way...
The completed project is located on the ACW Mentors Scratch account at https://scratch.mit.edu/projects/411572136/
Project 2 - Building a Calculator

As a follow up from the last project which was based on coding a series of simple but powerful interactive arithmetic functionalities, this lesson will use Scratch to build a calculator to undertake addition, subtraction, multiplication and division. It is most suitable for older pre-teens and teenagers.

If you wish, the structure of this project will allow students the opportunity to learn from the tuition provided in the first part of this project so that they can, as an exercise, deduce what code would be required to make elements of the calculator operational that may not be activated by and with you as the instructor. In other words, you can guide them through coding the addition, subtraction and possibly multiplication and allow them then to independently undertake the division process.

To start, first select New under File in the Menu bar.

In the Scratch User Interface, go to Backdrops in the Stage section.

Then click on Backdrops in the menu to the left of the screen

Then choose the rectangle icon in the Tool Box

and pick a suitable light colour
Starting on the top left-hand corner, drag the cursor out to approximately two thirds of the width of the stage and then down to the bottom. The rectangle will represent the outline of the calculator.

As the cat sprite is not required in this programme, remove it by hovering over its icon, right click on the mouse and select Delete.

We now go to the Variable category to create four different variables using the Make a Variable option.

The first will be labelled Number 1 to represent the first number to be inputted on the calculator.

The second variable will be labelled Number 2 to represent the second number to be inputted on the calculator.

The third variable will be labelled Operation to represent the actual action of addition, subtraction, multiplication or division that will take place between the two numbers.
The fourth variable will be labelled *Answer* to represent the outcome of the mathematical action between the two numbers.

There is no need to have Operation and Answer on view to the user of the Calculator. So hide both by ticking on the appropriate blue boxes in the Variable category.

Furthermore the wording “Number 1” and “Number 2” are not required in their appearance on the Stage as we are only interested in the actual mathematical numbers that appear in the course of operations.

So separately right click over Number 1 and Number 2 sprites and select the large readouts option in both cases.

Then drag both towards the top left-hand side of the calculator design, dropping one directly on top of the other.

We now need to populate the calculator with the standard function keys including digits and mathematical operations.

To start the process, select the paint brush icon in the *Choose a Sprite* section.

Select the circle icon in the paint *Tool Box*. 
Choose a colour, and should you wish, a border outline (second box) with its width controlled by adjusting the number in the third box below

Its size can be adjusted moving the blue lines

Position the sprite in the calculator background.

Duplicate the circle or key sprite twelve more times. Position the duplicates so that three columns of four and one column of five sprites are created.

Return to the first sprite and go to Costumes.

Select (T)ext in the Tool Box.

Choose a different colour

Then bring the cursor to the centre of the circle and input a number between 0 and 9.

Use the Reshape key to move the number to within the circle that you are happy with.

Repeat this process in each sprite until all numbers between 0 and 9 as well as a decimal point (full stop symbol) are inputted.

Copy the layout of a simple calculator when positioning each button icon.
For four of the sprites, select a new colour and input the symbols for addition, subtraction, multiplication and division.

If you wish, choose a different colour for each of the last two keys (sprites), namely \(= \) and \(C\) (clear). Position these key sprites as per your agreed layout.

Two small rectangle sprites can be positioned towards the top of the rectangle to give it an authentic calculator look.

The first step in building the script is to input the settings for the four variables, namely number 1, number 2, operation and answer, which will be placed in the \(= \) (equals) key sprite.

Go to **Events**

Drag and drop into the workspace

Go to **Variables** and drag and drop set operation, set answer, set number 1 and set number 2 blocks into the workspace connecting to the \(= \) block.

By default the value in these blocks will be 0 (zero). But delete the zero in both Number 1 and Number 2 block as we want the number slot to show on the calculator as blank.
As we want the Number 1 input value always to appear first onscreen during the start of an operation, we have to initially hide the Number 2 variable.

This we achieve by using the *show variable* and *hide variable* blocks, placing them at the end of the current script and using Number 1 and Number 2 options from the drop-down menu.

Now click on the sprite representing the digit 1 key

Go to the Control category, select and then
Drag both into the workspace and connect

The purpose of these two linked blocks of code is that when key 1 is pressed it sends a broadcast or message to another part of the programme to implement a change.

Repeat this process for all other nine digits (0 to 9) changing the label message1 to message2 to message3 and so forth.

To do so just click on the message1 dropdown menu, select New message and type in the new name.

For a broadcast to work it has to have a corresponding I receive command somewhere else in the programme. In this case we will position these blocks in the workspace of the Backdrop.

Click on the Backdrops icon in the Stage section

We will start the process with message1

From Events, drag and drop

What we want to happen in the programme is that when we click on Key 1, the number 1 will appear onscreen on the two occasions (first input and second input) involved in each operation of adding, subtracting, multiplying and dividing.

As each operation will involve two separate inputs of numbers, once in the Number 1 variable and once in the Number 2 variable, we have to ensure that the script can differentiate between both.

The script already inputted in the Backdrops Scripts area (Workspace) ensures that the digit 1 will always appear if it is the first (or part of) number clicked. The second number selected should only appear on the calculator after an operation (add, subtract, multiply and divide) is started e.g. 3 (first number 1) + 4 (second number)
So to ensure that it appears before and after each operation when it is selected, go to Control and choose

As we have the operation variable already in the programme and set to a value of zero, we get the code to take cognisant of this.

Go to Operators category.

Select \( \text{Operation} \) and place it between \( \text{if } _____ \text{ then} \).

Place \( \text{Operation} \) from the Variable category into the first white spacing and input the value 0 in the second white spacing.

Place \text{set number 1 to} \ and \text{set number 2 into} \ the script as follows:

If we change the value in both \( \text{Operation} \) and \( \text{Number 1} \) and \( \text{Number 2} \) from 0 to 1, it means that when we click on the key on the calculator the digit 1 appears. But the script doesn’t allow the multiple entry of 1 as in 11, 111, 1111 etc.
To achieve this, we have to have a script that will allow the joining of multiple entries of 1, as in for example 1 followed by 1 followed by 1 to give 111.

This we achieve by selecting \textbf{join} \textbf{apple} \textbf{banana} from the Operations category and placing it in the script as follows:

\begin{itemize}
  \item \textbf{set Number 1} to \textbf{join apple banana}
  \item \textbf{set Number 2} to \textbf{join apple banana}
\end{itemize}

Place \textbf{Number 1} and \textbf{Number 2} in the first white spacing and the digit 1 in the second white spacing of both blocks to give:

\begin{itemize}
  \item \textbf{set Number 1} to \textbf{join Number 1 1}
  \item \textbf{set Number 2} to \textbf{join Number 2 1}
\end{itemize}

Duplicate this script nine times replacing the digit 1 with 0, 2, 3, 4 5, 6, 7, 8 and 9.

To help in simplifying and understanding the role of each digit sprite, it is best to rename them. This can be done by going to the label Sprite directly under the Stage on the left side and change the name appropriately.
This renaming process applies not only to the digit sprites but to all sprites (+, -, = ...).

To activate the four operations of addition, subtraction, multiplication and division, we first need to enter a script in each of the appropriate four sprites.

To activate the four operations of addition, subtraction, multiplication and division, we first need to enter a script in each of the appropriate four sprites.

Click on the \textbf{add} sprite

From Events, drag and drop

Then go to Variables.

Select the block set \textit{Operation to ______} and input the text \textit{add} and then connecting it to the previous block.

As the use of any of the four Operation keys takes place only after the clicking of Number 1, then we have to input code that will hide the Number 1 sprite whilst showing the Number 2 entry.

This is achieved by the addition of two further blocks from Variables to give the following script

Repeat this process for the three other Operation sprites representing subtraction, multiplication and division. The easiest way to do this is to copy this script into the other four sprites and replace \textit{add} with the appropriate word \textit{subtract}, \textit{multiply} and \textit{divide}.

For the final part in activating the calculation process, we return to the \textit{= (equals)} sprite.
Drag and drop from the Events into the workspace.

To show the mathematical result of the +, -, X or / action between the two numbers, we have to first hide the Number 2 entry which would be what is showing before the user clicks on the equals key.

So place from Variable under the

We commence the input of the four separate operations by starting with addition.

Go to Controls and select

In Operators

Take and place in the spacing between If and then

In Variables select
and then

```
set Answer ▾ to 0
```

placing both separately in the script as follows

```
if Operation = 50 then
  set Answer ▾ to 0
else

As the operation in this case will be addition, input the word *add* to where the number 50 is in the script.

As the answer will be the addition of :Number 1: and :Number 2:, place from *Operators*, the addition (+) block on top of the digit 0 and

we now implement a similar coding for Subtraction

```
when this sprite clicked

hide variable Number 2 ▾

if Operation = add then
  set Answer ▾ to Number 1 + Number 2
else
```
In the spacing below else in the script, place in the block from Controls.
Repeat the process for subtraction as was done with addition by using the word 

*subtract* instead of *add* and replacing with

Once this is completed,

place

under the to give:

Replicate the addition and subtraction operations for multiply and divide in the *if then*_________ *else* block within the script.
Click on the Green Flag to test the script.

The programme does not yet work as it is unfinished.

For the answer to appear onscreen, we have to set the last number used (Number 2) to blank status (no numbers) and be hidden from view (hide variable) whilst having the other number (Number 1) become the answer and being seen on screen (show variable).

So attach the following blocks to the end of the other blocks
to give the following

By the Way...
The completed project is located on the ACW Mentors Scratch account at
https://scratch.mit.edu/projects/411571246/
Project 3 - Drawing Shapes

*In this project students learn to understand how to draw shapes using a series of scripts.*

The project will be based on: **Geometry**

which is the branch of mathematics that deals with shapes and sizes of figures and their properties.

The basic elements of geometry are points, lines, angles, surfaces, and solids

What is a Polygon?

In geometry, a polygon is any two-dimensional shape formed with straight lines. The word polygon derives from the ancient Greek words for many (poly) and angle/corner (gon). Triangles, hexagons, pentagons, squares and rectangles are examples of polygons. Their names can often tell you how many sides the shape has, for example a tri(three)angle and an octa(eight)gon.

However a circle is not a polygon as it does not have straight lines

In order to draw in Scratch, go to the **Pen** category.

However, as Pen is not one of the standard onscreen categories, we have to click on the **Addition** icon located at the bottom left-hand corner of the Scratch Interface.

Under *Choose an Extension* click on **Pen**.
Then choose the Pen element which will automatically add it to the thematic block categories on the (Home) Interface.

**Creating a Script to draw a Square**

This time we do not need to use a sprite. But as all of the Scratch commands cannot function without a sprite, we need to hide it (rather than delete it).

Place the following commands in the Workspace to make the sprite disappear:

![Robot block](image)

From Pen, select a pen size and colour.

We also have to give instructions for the pen to a) start and b) finish drawing.

![Robot block](image)

It is also important to clear previous drawings from the stage area and to recommence the drawing process on a blank canvas once the Green Flag is selected to restart the project.

So, use the command

![Robot block](image)

Adding on *wait 1 secs* block from Control will allow the viewer to better appreciate visually both the commencement of the drawing and the actual process of the formation of a new object.

To ensure that the geometric object that we are about to draw, namely a square, is of sufficient dimensions for easy viewing, use a sizeable number of steps from the Motion folder e.g. 100 steps.

To code in the command that will make the four lines form a box, we first use the Repeat command. In the construction of a square it is **Repeat 4** times.

| Please note also that for each geometric shape, the angle is proportionate to the number of sides i.e. 360 degrees divided by the number of sides. |
| --- | --- | --- |
| For instance, a square is 360 divided by four = 90; |
| A triangle is 360 divided by three = 120 and |
| A circle is 360 degrees divided by 360 = 1 |
We turn the direction of the lines using the *turn _____ degree* block from the Motion category.

Using the [image of block diagram]

will stop the square being lopsided.

We can also change the colours for each drawing by combining [image of block diagram]

with the random option in the Operators category to give [image of block diagram] which can replace [image of block diagram] in the script.

The spectrum of colours goes from 1 to 200. Hence choosing a high random range e.g. 1-200 (see script below) for colour variation will allow the programme to randomly select from the full range of colours each time that it is run.

Use [image of block diagram] to set a Pen size of your choice.
Let's draw a triangle

How many sides in a triangle?

What would the angle (degree) settings be?
Let's draw a circle.

How many turns (degrees) in a circle?

So, turn one degree at a time. Do not use the block in the script.

This is because using it will mean that it will take 360 seconds for the circle to complete.

Add inside the loop:

![code_snippet]

to give

![code_diagram]

and enjoy the colourful effect.

**Exercise**

1. Draw a Pentagon
2. Write a programme that draws three different shapes that appear at different times at different locations on the Stage.

**Other Shapes**

Experiment with different angles and 'repeats' in the programme. For instance, select the code below and admire the results.
Input the following code:
to give
Create a four-petal flower

The effect is achieved by making four half circles (180 degrees) and turning right (90 degrees) at the completion of each one.

By the Way...

The completed project is located on the ACW Mentors Scratch account at https://scratch.mit.edu/projects/411572612/
Project 4 - Pick and Choose you own Polygon Shape

In this project, participants will be able to create an interactive programme that will allow users to ask the computer to draw different types of polygons by telling it how many sides it has.

In the previous project we learnt how one can write scripts to draw different shapes.

For instance, the key blocks to create a square are:

![Diagram of square creation blocks]

Based on

that for each geometric shape, the angle is proportionate to the number of sides namely 360 degrees divided by the number of sides.

For instance, a square is 360 divided by four = 90
A triangle is 360 divided by three = 120 and
A circle is 360 degrees divided by 360 = 1

So let’s write an interactive script that will allow the user to answer a question asked by the computer on how many sides he/she would like in their polygon.

For the same reasons as heretofore, we use the same script as we used in the previous lesson to draw a square.

![Diagram of square drawing script]
However we will replace the block with an arithmetic action of 360 (degrees) divided by 4, which represents the number of sides of a square.

So go to Operations.

Select

And insert it into the script as follows:

Place the number 360 into the first oval shaped white space and the number 4 into the second white spacing. These numbers are chosen because 360 divided by 4 gives an answer of 90 (see grey block above).

Click on the Green Flag. The result should be a square appearing on the stage.

Now we are going to transform the project into one of human interaction by coding in the capability for the script to ask the user to choose the amount of sides that the polygon should have.

First change the Y coordinate in the Motion block from 0 to 100 (to provide better positioning for the range of different shapes that will be created).

Go to Sensing

Select

ask What's your name? and wait
Change the text of *What’s your name?* to *How many sides do you want in your polygon?* Select a number between two and eight.

Return to Sensing

Select and place this block into the two oval shaped white spacings occupied by the number 4.

Press the Green Flag.

The addition of a Repeat block would improve the performance of the programme as it would increase the level of engagement between the user and the script.

Input as follows:

![Diagram of block code]

However a problem now arises with the new polygons appearing on top of the older selections.

So input two extra pieces of code:

![Diagram of block code]

and

![Diagram of block code]

to allow the polygon to remain on screen a few seconds before it is replaced.
The final script should be

```
when green flag clicked
point in direction 90
hide
erase all
pen up
set pen color to purple
go to x: 0 y: 100
wait 1 seconds
pen down
repeat 10
    ask "How many sides do you want in your polygon? Select a number between 2 & 8" and wait
    wait 1 seconds
    repeat answer
        move 100 steps
        turn 360 / answer degrees
    wait 2 seconds
    erase all
```

By the Way...

The completed project is located on the ACW Mentors Scratch account at [https://scratch.mit.edu/projects/411572712/](https://scratch.mit.edu/projects/411572712/)
Project 5 – Why Trees are important

In this project, participants will be introduced to the science of trees, the importance of this type of plant in regulating the atmosphere, in supporting biodiversity, in managing water and in sustaining life on planet Earth.

The project will help children come to a basic understanding of the interconnections between some of the great problems impacting on the natural world, namely Climate Change, deforestation, biodiversity loss and soil degradation.

The medium used is a story with a narrator. But it is important to have elements of interaction included in the Scratch project in order to have some level of hands-on engagement with the young students. Furthermore, consider including a creative arts dimension by getting the participants to draw a tree and/or an animal sprite.

As a follow up to this project, the teacher can use a similar template (story) and theme to undertake a Scratch project on for instance Why Water is important, or Why Rivers are important, or Why Oceans are important, or Why Soils are important.

It could also form part of a wider school, local community or state environmental programme such as a tree planting initiative.

Before the project commences, get the children, under your guidance, to carry out background research and collect information related to the importance of trees in people’s lives and that of the planet. This can be preceded by a ‘question and answer’ session on the topic of the ‘Why Trees are important’ to gauge or prompt their level of understanding of this key topic. Include if you wish some aspects of local, national, or regional mythological, religious, cultural stories and beliefs associated with trees such as the Baobab.
The material collected can form the basis for planning out the project’s storyline.

This project will be based on good scientific knowledge of the role of trees in human society and its wider role as a key part of the planetary biosphere. It is a guideline only. So please use as much or as less of its content as you feel is appropriate to your classroom.

However whatever approach that you take, there is no doubt that this project will last for two, three or more sessions.

To start

First upload Scratch

Delete the cat sprite.

Go to the Sprites Library

Select an appropriate sprite from the People folder who will function as the narrator such as

Position the Sprite towards the bottom left of the stage.

If the sprite is facing the wrong direction (away from the rest of the screen), then go to Costumes and select Flip Horizontal.

The next step involves developing a story line of the subject matter, based on short simple-to-understand scientific facts on trees. This can be done in association with the children or prepared in advance by you to be delivered as part of the science curriculum.
For this project we have looked on the role of trees in producing oxygen, food, biodiversity, medicines, filtering out toxic gases from the atmosphere, Climate Change, flooding and soil protection.

Time should be given to choose suitable backgrounds and the sprites for the project.

**The Backgrounds (backdrops)**

Select from the library a background that is a natural area of landscape devoid or trees or river.
Should a suitable type not exist in your estimation in the library, draw a background or alternatively download an appropriate copyright-free online image (example below).

Give it a suitable name e.g. Grassland

Duplicate it.

Give the new background a name (e.g. Forest).
Use the appropriate tools to draw and colour in a simple blue river in the foreground.
Type in an appropriate name such as Forest in the section

Copy the second backdrop (Forest) four more times.

In one of these new backdrops (Forest4) draw in a light sprinkling of dark clouds. In another backdrop (Forest5) draw in some additional thicker clouds. These two costumes will represent different levels of atmospheric pollution.

Forest4

Forest5

Scene 1: The Introduction

The Opening Background
Go to Backdrop1 (white) which will be the first background that will be used in the project.
Select **T** for text in the Tool menu.

Select the colour Black in the palette by bringing the three sliders to the far left.

Use the Text tool to type in *Why are Trees important?* and drag the text to the top left side of the screen in the Costumes work area.

**Why are Trees important?**

**The Sprites**

Draw a series of sprites and/or use sprites from the Scratch library.

In this Scratch demonstration, we will use sprites representing:
  * Interactive Box Label
  * Trees
  * Fruits (3)
  * Insects (2)
  * Monkeys (2)
  * Fish
  * Bird
  * Frog
  * Woman
  * Boy
  * Buildings (2)
  * Vehicle
  * Sun

Try to provide enough time in the first session to allow the children to draw at least one sprite (e.g. tree) so that they can feel personally connected artistically to the project.
The ‘Tree of Introduction’

Position the tree sprite towards the centre of the stage.

Go to the code icon.

Go to Events category and select the block Drag and drop into the Scripts area (also known as Workspace).

Go to Looks and connect show to the previous block.

Then place an X and Y block from Motion onto the script.

The Interactive Box Sprite
To bring in an interactive element to the project, use the paint brush icon in

to draw a new sprite

Select a Red shading in the palette

Africa Code Week is an initiative by SAP and UNESCO. All content here is the intellectual property of SAP Africa Code Week and was developed by Brendan Smith for the Camden Education Trust.
Then click on the rectangle icon in the tool kit to draw a red box.

Select the colour white from the palette, click on the Text Tool and type in the text 'Click here to find out.'

Position the sprite on the bottom right side of the stage.

Now we start to build its script.

Go to Events category and select the block.

Drag and drop into the Scripts area (also known as Workspace).

Go to Motion category.

Drag and drop the X and Y block into the workspace and connect to the proceeding block.
The X and Y coordinates of the sprite should automatically be picked up by Scratch and appear in the two white boxes of the Sprite Editor.

Go to **Looks**, chose **show** and place it between the two existing blocks in the script.

**The Tree of Introduction Sprite**

We want the tree sprite to disappear when the programme moves onto the next background in response to the user clicking on the `·` sprite.

This is achieved by the getting it to hide once it responds to a broadcast message telling it to do so.

---

### Broadcasting

Broadcast is used in Scratch coding to send a message (a communication or broadcast) from one element such as a sprite to some other part(s) of the programme, which could be another individual sprite, multiple sprites, backdrops or even another script in its own workspace, instructing it or them to implement an action.

The response to the message from the recipient sprite is initiated by placing in a block of code in the sender’s script.

The broadcast blocks, located in Events, will be used extensively in this project.

To create a new broadcast message, first go to the broadcast block and click on message1 to select **New Message**.
Type in forest

Place the block in the workspace and connect it to from Looks

The Interactive Box Sprite
To activate the change in the backgrounds, that is from backdrop1 to forest, go to the section of the .

To have it appear a few seconds after the Green Flag is clicked by the user, input the following script

The opening screen should look as follows
Why are Trees important?

To start the process of having this opening screen disappear and the next screen (forest) appear when the sprite is clicked, we use from **Events** the block and the broadcast command in combination with **hide**

This script will send out (broadcast) messages to a number of appropriate sprites to either hide or to show when they have embedded the block into their code.

**The Backdrops**

Go to the code section of Backdrops.

To ensure **backdrop1** background always appears when the Green Flag is clicked,

To ensure the second (forest) background always appears when **is clicked**, input

Africa Code Week is an initiative by SAP and UNESCO. All content here is the intellectual property of SAP Africa Code Week and was developed by Brendan Smith for the Camden Education Trust.
The second background has to be populated with trees.

The Tree Sprites
Copy the Tree sprite by right clicking on its small icon and selecting duplicate.

In the second tree sprite, delete the motion block and reverse the *Show and Hide* blocks in the code to give

![Tree sprite code]

Reduce it in size to ensure that it does not take up too much of the stage area.

Duplicate this tree sprite multiple times and reposition them across the stage.

![Duplicate tree sprite]

In order to give the impression of depth (layers) we need to have some trees positioned towards the front, others in the middle and still others towards the back of the forested area that we are creating.

So go to Looks and select
Drop this block into the script of the tree sprites using a mix of settings from their drop-down menus as the following examples show in the script from two different tree sprites.

Tree A

Tree B

The Sun Sprite
Select a Sun sprite from the Scratch library

Make a second costume.
In this version, use the drawing tool and colouring option to increase the proportions of the sun’s rays.

Sun2 costume will be used later in the programme.

Position the sprite in the sky of the forest backdrop using the following script.
Please note that your X and Y coordinates will no doubt be different from those shown above.

The **sun** is used as we want the sun to always appear in the background when bird sprites are flying across the sky.

The second forest backdrop in this demonstration project now appears as follows:

![Scene 2: Trees & Oxygen](image)

**Scene 2: Trees & Oxygen**

Stagger the appearances of the sprites. In the first forest backdrop introduce the narrator, static sprites (trees) and one moving talking sprite. The latter is sufficient as we want to focus of the viewer's attention onto the storyteller.

**The Narrator**  
Input the following script
as the narrator will be hidden in the opening screen and should always be positioned in the foreground (front) with other sprites such as trees, animals and fish appearing behind her.

In the example used, the costume (avery b) of the narrator will show her hands in a relaxed non-gesturing pose.

We will now build up a narrative.

The narrator will be a constant throughout each element of the story providing different pieces of scientific information as the project progress, often accompanied by a changing background and/or other sprite(s).

What is a Tree?
A plant that is generally characterised by having a hard woody stem

In Scene 2 (forest), unhide (show) the narrator and have her briefly refer to the life-giving characteristic of plants.

Once her scientific fact is spoken, the script will change the backdrop to go to scene 3 (forest2).

In this case there is no actual visual distinction between the Forest and Forest2 backdrops; they are used only to facilitate a new thematic comment by the narrator and the introduction of sprites by way of the broadcast blocks.
The Bird Sprite
The bird sprite is hidden in Scene 1, is positioned in the sky by using an X and Y motion block, moves only upright by way of a 90 degree angle (moves upright) block and is always at the front of other sprites when it is flying across the screen thanks to the *go to front layer* code.

The appearance of the bird in Scene 2 is delayed by a few seconds to allow time for the narrator to finish her speech on how trees provide oxygen to fauna. The bird should allude to that in its script.

To allow the bird to constantly (*forever*) move back and forth across the screen (*if on edge bounce*) and to animate the movement of its wings (an ongoing change between costumes with different wing settings), attach the blocks of code below to the above script.
Scene 3: Trees & Food

In this scene (Forest2) the narrator will talk of the food aspects of trees.

Introduce the fruits that provide food to humans and wildlife.

From the Scratch library, select an orange, a banana, and an apple.

Attach each fruit to a different tree.

As all three fruits will first be seen in the Forest2 backdrop and will appear in front of the tree, input the following script in all of these sprites:

Duplicate each of the three sprites multiple times and populate their individual tree with additional fruits.
Before duplicating you can, if you wish, use a X and Y motion block for each fruit whose coordinates are correct once the sprites are placed in their final position.

The scene will now show as

### Scene 4: Trees & Wildlife

In this scene (Forest3) the narrator will start by briefly mentioning the importance of trees to wildlife.

![Scratch code block]

Input a variety of wild fauna sprites from the Scratch library as well as from, if you wish, a free online image gallery.

In this demonstration, two insects, two monkeys, one fish and one frog join the bird and trees that are already in the project.

**The Monkey Sprites**

Choose

![Monkey]

Duplicate this sprite
Go to Costumes and use the paint tool and colour options to paint the second monkey in another colour (just to give some variety!).

To have these new fauna sprites only appear onscreen when this third (Forest3) comes onscreen, input

```
when backdrop switches to forest3
    show
```

However to ensure that they are hidden when the project starts, use the following script with a different X and Y coordinates for each

```
when clicked
    go to x: 132 y: 80
    hide
```

Depending on the specific sprite, a layer positioning block can be used. For instance, in the case of the monkey sprites, they need to appear in front of the tree rather than behind it. So the front option in the `go to _____ layer` block should be introduced.
To have the monkeys move (jumping/gliding) from branch to branch, use a combination of the switch costume to (Looks category), glide (Motion) and forever and wait ___ seconds (Control) blocks.

In the other fauna scripts, move ___ steps will be used in some of these sprites instead of the glide ___ secs to x: y: block (as with the monkeys) along with if on edge bounce and next costume.
To ensure that the elephant does not walk in front or behind Avery the narrator (positioned on the left of the stage), input a script that will have it change direction by way of a 90 degree right turn.

For variety, special effects and reflecting the interrelationships between different species within an ecosystem (see below), code in the two insects to disappear (hide) when they come close to the frog and fish. This represents them being eaten by their natural predators. Have them reappear (show) a short while later.

Ecosystem
A large community of living organisms (plants, animals and microbes) in a particular area. The living and physical components are linked together through nutrient cycles and energy flows. Ecosystems are of any size, but usually they are in particular places.
To give the impression of the stationary frog catching the insect, code in both of its costumes:

---

Let some of the animals give some short informative talks to commence after the introductory piece by the narrator. This can be done by a broadcast command or alternatively using a `wait ___ seconds` block. End the script with a broadcast block to activate another animal to talk.
Using the wait _____ seconds, the flight of an insect can be synchronised with the frog’s talking so that it disappears from the screen immediately after the frog says (see script above) “I love eating insects”

Thanks to the broadcast command in the script above and below, the monkey is programmed to talk also.

Scene 4a: Trees & Human Health

The block in the script of the monkey above is picked up by the complimentary block in the narrator’s script.
The Sick Boy Sprite
Introduce a sprite, either drawn, downloaded from the copyright-free online gallery or from the Scratch library (e.g. Ten80’s 8th costume) which can be represent a young person feeling sick.

For example

The new sprite will be hidden by its opening script of

until it is activated by the block in the previous sprite’s (narrator) script in association with its own script of

The Mother Sprite
Introduce a woman (singer) sprite from the Library to represent the boy’s mother and position her in the front not far from the boy sprite
As with so many other sprites, the new sprite will be hidden by its opening script

Introduce another sprite that will represent a bowl containing a herbal medicine mix that should be positioned with a X and Y motion block in front of the mother’s sprite.

The mother sprite will show onscreen when it is activated by

in the boy sprite’s script in association with its own new script commencing with
However it needs the addition of a block so that it appears in front of the flora and fauna sprites that are already on the stage (screen).

Input a **Say** for **seconds** block with a relevant comment preceded by a short **wait** **seconds** block and followed by a new broadcast (mom2) that will help activate a script in another sprite.

---

**The Medicine Bowl**

The medicine bowl sprite will show onscreen when it also is activated by

```
when I receive mom
show
```

in the boy sprite’s script in association with its own new script commencing with

```
when I receive mom
```

However it needs the inclusion of a **wait** **seconds** block before

```
go to front
layer
```

is added to the script. Only then will the show block be used. This is because this medicine bowl sprite should appear in front of the mother sprite which also has to appear in front of the flora and fauna spites already onscreen. The last sprite to use the block will be the one that appears in front of all other sprites.

Hence the code for the medicine bowl should be as follows
In response to the mother sprite’s block the bowl will use the glide block from the Motion category to move towards the boy sprite. After a short delay (wait___ seconds) the bowl’s script will activate the boy sprite to respond (new broadcast block) to the medicine he takes and then instruct him to disappear (hide) from the screen.

The will be responded to by that will commence a new script in the boy sprite.

Select a costume for the boy sprite that will personify happiness or contentment.

Complete a script that will change costume and say words of joy and thanks (Looks category), as well as provide a broadcast (Control category) to be sent to the mother sprite to hide.
The mother sprite will need one more short script to complete this present part of the storyline.

Scene 5: Trees, Air Pollution & Urban Heat

The Backdrops
Within one backdrop (forest4) there is the presence of a few dark clouds in the sky. In forest5, there are additional thicker clouds.

These two Backdrops represent different levels of atmospheric pollution and are activated in response to the block in the boy sprite’s script.

Go to the Code element and input the following
The Narrator

In response to the block in the boy sprite’s last script, input the following script:

```blocks
when I receive mom3
wait 2 seconds
say Trees keep our air clean by taking from it many poisonous gases such as sulphur dioxide for 6 seconds
```

In response to the block in the Backdrop’s last script, input the following script:

```blocks
when I receive co2
say Trees absorb the CO2 gas that causes our planet to get very hot for 5 seconds
wait 1 seconds
```

The also activates the Sun sprite to enlarge in order to give the impression of increased global warming.
With the positive message of

a broadcast block now is included in the script to get the Sun sprite to return to its normal smaller size.

This is done by the addition of a new message which is received by the Sun sprite (see next section).

Go to Controls and make a new broadcast message entitled which should then be attached to the previous script.

Go to Looks and select two of the Say ______ for ___ seconds blocks. Increase the duration (amount of seconds) of both.

Connect these blocks to the current script with the two Say _____ blocks interspaced by wait ___ seconds blocks as below

The project now needs the addition of a number sprites to represent life in a city. These can be found in the Sprite library.

In this project, two building sprites and one truck sprite are used
All these sprites will have the following three scripts (with differences of course for their X and Y coordinates).

**The Sun**
As mentioned in the Narrator’s section above, the block activates the Sun sprite to enlarge in order to give the impression of increased global warming.

This is done by making a script comprising *When I receive* __________* broadcast (Events), a `switch costume to Sun2` (Looks) and a further block to increase the size of the sprite (Looks)

In order to return to normal size as a response to the block in the Narrator’s broadcast, the following script will be inputted
Scene 5a: Trees, Flooding & Soil Retention

Duplicate one of the tree sprites

Increase its smaller than normal size from what is probably 20 (this reduction was undertaken earlier on in the project) to 70 which can be undertaken in the sprite’s Properties section directly underneath the stage.

Go to the Costumes section of this sprite.

Using the Fill option (colour) and the paint brush, draw in extended roots

Position the tree in an appropriate position on the landscape amongst the other smaller trees.

Go to Events category.
Create two new messages in the broadcast block, firstly *tree roots* (or similar name)

![Broadcast block for tree roots](image1)

and secondly *hideroottree* (or similar name)

![Broadcast block for hideroottree](image2)

In the Script area, input the following three scripts

![Script block for tree roots](image3)

The two *when I receive* blocks are in response to two corresponding broadcast commands in a new script to be inserted in the Narrator’s code, namely
Scene 6: A World without Trees

At this stage of the project the participants have built up a great story on the positive contributions of trees to the planet.

It is time now to complete the narration by looking at what would happen to the climate, biodiversity and humans if all the trees were cut down.

As well as the human narrator, different fauna sprites will tell us one after the other of how the disappearance of trees will impact on their lives and that of the planet.

The connections between the different narrators and the consequences of what they say will be activated in the programme by the use of the broadcast blocks from the Events folder.

The viewer will also be involved by having to interact with the programme in order for the project to continue.

The Narrator
Return to the Narrator’s Workspace.

Change the costume of the sprite to give the impression of the narrator engaging with the viewers.
Attach onto the last script which ended with the block, the following

**The Black Monkey**
As the above narrator’s script makes clear the user must click on the black monkey sprite for the programme to continue

Click on the black monkey icon

From Events, drag and drop into the workspace

Go to Control

Select the *Stop* ________ block, connect it to the previous block and choose the *other scripts in sprite* option in its drop-down menu

This will freeze all the other scripts of this sprite.

We will now build a script that will have the monkey come down from a tree, start the story of what happens when trees are cut down, and to have the fruit sprites, commencing with the apples and bananas, begin to disappear from the screen.
The Apples
The broadcast message *applesgo* in the script above needs a complimentary script in each of the apple sprites which will make them disappear.

It also activates the script in some of the tree sprites starting the process of them disappearing also

The Bananas
The broadcast message *bananagone* needs a complimentary script in each of the banana sprites which will make them disappear.

As with the previous broadcast message *applesgo*, *bananagone* also activates the script in some of the tree sprites.
The Trees
The trees in the forest will start to disappear on a phased basis as each of the flora and fauna scripts disappear.

To achieve this, place different options in each of the blocks placed in the individual tree scripts until all the trees are gone.
The Brown Monkey
The scripts are now following a pattern begun by the Black Monkey which is based on the send and receive structure of the Broadcast commands (blocks).
The Elephant

The Butterfly (1)
The Butterfly (2)

when I receive insectsgone
stop other scripts in sprite
say When the jungle disappears, insects will have no leaves, fruits or flowers to eat for 3 seconds
say They too will disappear for 2 seconds
say Without insects, most plants will not be able to pollinate and reproduce for 3 seconds
say They too will become extinct for 3 seconds
hide
broadcast fish gone

The Fish

when I receive fish gone
stop other scripts in sprite
say The loss of trees that leads to the disappearance of many insects will have a bad effect on creatures that live in rivers and lakes for
say For insects are an important part of the diet of many species of fish and other water life for 5 seconds
say So we too will die for 3 seconds
wait 1 seconds
hide
broadcast watergoes

The Backdrops
The watergoes broadcast message is also received by a Backdrops script in order to revert to the original grasslands backdrop before the water feature (river) was drawn in for the benefit of the forest backdrops.

when I receive watergoes
wait 1 seconds
switch backdrop to grasslands2
The Frog

- when I receive watergoes
- wait 1 seconds
- say The cutting down and burning of trees leads to more CO2 in the air. for 4 seconds
- say CO2 traps the Sun’s heat in the atmosphere and the world becomes a lot hotter. for 4 seconds
- say Then the rivers and lakes get smaller and smaller as they dry up. for 3 seconds
- say It is not only fish that call water their home. We frogs also live in rivers, lakes and ponds. for 3 seconds
- say So once they dry up, it means that we too also die. for 3 seconds
- wait 1 seconds
- hide
- broadcast birdsgone

The Bird

- when I receive birdsgone
- stop other scripts in sprite
- wait 1 seconds
- say Birds need trees to build their nests in. for 3 seconds
- say It is where we give birth, nurture our young and it is where we live. for 5 seconds
- say Take away trees and we will no longer exist. for 3 seconds
- wait 1 seconds
- hide
- broadcast humans gone
The Human (Narrator)

```
when I receive humans gone
wait 1 seconds
say With the chopping down of the jungles and the drying up of the waterways, fruit trees so beloved by humans will also be gone for 4 seconds
broadcast orangegone
say Without tree roots to hold the soil together, the winds will blow it away for 4 seconds
say Without soil and in a very warmer world, we will not be able to grow many vegetables and herbs for 5 seconds
say So all of us, young and old, male and female must help protect the jungles, and plant lots of trees for 4 seconds
say Otherwise humanity will also disappear from the planet for 4 seconds
wait 1 seconds
hide
```

By the Way...

Please note once again that all the sprites, backdrops and scripts do not have to be used for this project. If you wish, pick what you feel are best suitable for a shortened version of this powerful environmental message on the importance of trees to life on planet Earth.

The completed project is located on the ACW Mentors Scratch account at [https://scratch.mit.edu/projects/411747582/](https://scratch.mit.edu/projects/411747582/)
Project 6 – A Wildlife Quiz

‘Question and Answers’ are a key part of everyday life in a school classroom. The quiz game format is a simple but highly effective and indeed enjoyable way to test knowledge on a subject and to assess students’ progress.

Getting the youngsters to co-create a quiz in Scratch will not only improve their coding capabilities but also allow them to develop a template that can be used across the educational curriculum. As well as enhancing their learning capabilities and interest in the specific theme being used in this project, the process involved should encourage teachers and students to use it in other subjects ranging from languages, geography, history to science.

It is recommended that this exercise be followed up by getting the students individually or in small groups to develop their own quiz based on another topic or topics that they are currently studying. Researching the content matter and coming up with suitable questions should be a challenging but nevertheless rewarding and fun undertaking for all involved. The completed quizzes can then be tested out amongst all the students of the classroom.

Project Research & Design Summary

The Wildlife Quiz to be developed in this project will be based on questions related to the behaviour and habitats of four animals. By clicking on any of the fauna sprites, the user will be shown a number of interesting facts on the specific species. Then the user will go to a new section where he/she will be asked a question with the answer to be chosen from multiple choice options.

In preparation for the coding of this demonstration project, the necessary research into the lifestyles of four animals and selection of appropriate quiz questions was carried out.
Upload Scratch from your computer by clicking on

![Scratch Desktop](image)

Delete the cat sprite.

**The Backdrops**
Go to **Backdrops** in the **Stage** section.

Then click on **Backdrops** in the menu (below) to the top left of the screen

Duplicate **backdrop1** by right clicking on the mouse/keypad

Stay on **backdrop2**

In the **Costume** name label, change the name from **backdrop2** to **homepage**

In the **Canvas** area, type out the wording **Wildlife Quiz** by first clicking on **T** in the **Tools Menu** of the **Vector** option

Select a suitable colour in Fill
and a suitable font in

![Font Selection](image)

The text can be decreased or increased in size using the tool at the top left hand first column of the Tools Menu.

![Font Size Tool](image)

and then, using the same tool, it can be moved around the stage to secure a suitable location.

Whilst the text should indeed be large, nevertheless ensure that enough space on the stage is allocated for the placement of animal sprites.

As aforementioned, four wild animals (monkey, elephant, lion and octopus) will be chosen for this Wildlife Quiz.

Select four suitable sprites and manually position them on the stage leaving a large gap towards the bottom third of the stage. This space will be filled by an instructional box sprite.

![Animal Sprites](image)

The ‘Instruction Box’ sprite

Go to the Make New Sprite option of the drop down Sprite Menu located towards the bottom right side of the Scratch Interface.
Click on the **Bitmap** option below the Canvas

Select **Square** in the Tools Menu

Then choose colour *(Fill)*, **Outlined** structure and line **Thickness** in

The depth (thickness) of the boundary line can be adjusted by changing the number in the latter box (it is 20 in the sample shown above)

**Draw box outline**

Move the rectangle box whilst still in edit (box) mode

Go to the **Vector** option

Click on **T** in the Tools Menu

Select a suitable colour in Fill
and a suitable font in

Type the following (or similar text): *Click on an animal to find out some interesting facts about it and to take part in our wildlife quiz.*

Move the sentence into the rectangle box and adjust the text to fit using the blue pointers.

The text can be decreased or increased in size using the tool at the top left (first column) of the Tools Menu

At present the text and box move independently.
To unite the two elements together, select

Drag this tool from top left hand corner to the bottom right hand corner and move the object. The text and box are now operational as one entity.

The Monkey's 'Facts Box' backdrop

Click on ☑ (Make New Backdrop option) in the drop-down Backdrops Menu located towards the bottom right side of the Scratch Interface
In the Costume name label, type in *MonkeyFacts*

Click on **T** in the **Tools Menu** of the **Vector** option

Select a suitable colour in **Fill**

and a suitable font in

The text can be decreased or increased in size using the tool at the top left of the **Tools Menu**. But endeavour to have it fill up the stage screen.

Creating a boundary line surrounding is an optional feature with the procedure the same as in the instructions of the previous sprite.

*Interesting Facts on Apes & Monkeys*

1. Are our closest living relative
2. Eat mainly fruits, leaves, flowers and sometimes insects
3. Some use stones to crack open nuts
4. They help forests grow. The seeds from the fruits that they eat are dropped onto the forest floor to become new trees over time
5. Apes sleep in nests made from branches or foliage on the ground or in trees...
The Monkey’s Question & Multiple-Choice Answers backdrop
As with the previous backdrop, click on (Make New Backdrop option) in the drop-down Backdrops Menu located towards the bottom right side of the Scratch Interface.

Follow the same procedure to compose a question with multiple choices (three) from one of the five facts on monkeys and apes that comprised the last backdrop.

Name this backdrop MonkeyQuiz

However leave appropriate spacing to the left of the three options (stones, wood & hammers) as shown below.

To crack open nuts, some monkeys use

stones

wood

hammers

This is because three coloured circle sprites representing answer icons will be placed in these spaces.

The Monkey’s Coloured Circles of Answer sprites
Create a different coloured circle sprite beside each of the three possible answers.

To do so, go to the Make New Sprite option of the drop down Sprite Menu of the Scratch Interface.
Click on the \[\text{square}\] in the \textbf{Tools Menu} of the Vector option.

Select a suitable colour in Fill.

To get rid of any outline boundary of a different colour, bring the number to zero (0) in box beside Outline and Fill.

Adjust the circle to a suitable size.

Position the coloured circle to the left of the first text answer of the \textbf{Multiple Choices Backdrop}.

\textbf{To crack open nuts, some monkeys use} 

\begin{itemize}
  \item \textcolor{purple}{\textbf{stones}}
  \item \textcolor{green}{\textbf{wood}}
  \item \textcolor{green}{\textbf{hammers}}
\end{itemize}

Follow the same procedure to create two more circle sprites, each one positioned against separate Answer options.

If one wishes to do so, each circle could have a different colour.

\textbf{To crack open nuts, some monkeys use} 

\begin{itemize}
  \item \textcolor{purple}{\textbf{stones}}
  \item \textcolor{green}{\textbf{wood}}
  \item \textcolor{green}{\textbf{hammers}}
\end{itemize}
The ‘Arrow Instruction Text Box’ sprite
To have the user move from the page displaying the facts on monkeys and apes to that containing the Question with multiple choices, we need to create two new sprites. The first will contain brief instructions on how to go to the latter page; the second will be the Arrow icon that the user touches to undertake the move.

To build an arrow instruction box, we will once again follow the same procedure as was done with the ‘Instruction Box’ sprite.

Go to the Make New Sprite option of the drop down Sprite Menu located towards the bottom right side of the Scratch Interface.

Click on the Bitmap option below the Canvas

Select in the Tools Menu

Then choose colour (Fill), Outlined structure and line Thickness in

Draw box outline

The depth of the boundary line can be adjusted by changing the number in the box labelled Thickness in the Line bar.

Move the rectangle box whilst still in edit (box) mode
Go to the Vector option
Click on T in the Tools Menu

Select a suitable colour in Fill

and a suitable font in

Type the following (or similar text): Click on Arrow to go to Quiz

Move the sentence into the rectangle box and adjust the text to fit using the blue pointers.

The text can be decreased or increased in size using the tool at the top left of the Tools Menu

At present the text and box move independently.

To unite the two elements together, select
Drag this tool from top left hand corner to the bottom right hand corner and move the object. The text and box are now operational as one entity.

The Arrow Sprite

Go to the Make New Sprite option of the drop-down Sprite Menu and select the library (magnifying glass) icon

Then click on the Arrow sprite

Select the costume with the direction pointing right

Go to Fill, and the paint bucket icon in the Tools Menu should you wish to change the colour of the arrow.

At this stage it is time to commence inputting the code (instructions)

The ‘Homepage’ Backdrop (costume)

Drag and drop from the Control category into the Workspace

connecting to it the following block from the Looks category
The ‘Instruction Box’ sprite

Ensure that the Instruction Box sprite is in the correct location on the Stage.

To have it appear when the quiz starts, go to Control category and drag and drop the Green Flag block into the Workspace.

From the Motion category select the go to X____ and Y___ block. Scratch automatically picks up the present coordinates of the sprite.

Go to Looks category and drag and drop the show block into the Workspace. This is because this sprite (as with the four animal sprites) should be on view when the quiz starts.

The script should appear as follows

The Monkey sprite

Follow the same procedure using a similar script as was undertaken with the Instruction Box sprite.

Once again ensure that the sprite is in the correct location on the Stage.

As per the onstage instructions, clicking on the monkey needs to bring up a new screen displaying interesting facts about this type of animal

To allow this to happen, first go to Events and select
This command needs to be followed by an instruction that will lead to the Interesting Facts on Apes and Monkeys backdrop appearing.

As heretofore, we use the broadcast command blocks for this task.

Go to broadcast block and under **new message**, type in *monkeyfacts*

![Broadcast monkeyfacts](image)

Drop it into the Workspace placing it underneath the previous block.

Then take *hide* from the Looks category and connect as follows:

![Hide](image)

The Monkey should also disappear when any of the other three animal sprites are clicked.

So return to Events and select the when I receive __________ block

Under **new message**

![When I receive monkeyfacts](image)

Type in *elephantfacts* and place this block in the Workspace.

Go to Looks category, select *hide* and connect it to the previous block.

![Hide](image)

Repeat this process to create broadcast command blocks for *octopusfacts* and *lionfacts*.
The ‘Instruction Box’ sprite
The ‘Instruction Box’ sprite should also disappear when any of the four animal sprites are clicked.

So likewise input the following short scripts

![Script for 'Instruction Box' sprite](image)

The Monkey’s ‘Interesting Facts’ Backdrop (costume)
As a result of the script

![Script for Monkey's backdrop](image)

the stage should now switch to
Interesting Facts on Apes & Monkeys

1. Are our closest living relative
2. Eat mainly fruits, leaves, flowers and sometimes insects
3. Some use stones to crack open nuts
4. They help forests grow. The seeds from the fruits that they eat are dropped onto the forest floor to become new trees over time
5. Apes sleep in nests made from branches or foliage on the ground or in trees.

To ensure that this screen appears in response to the broadcast *monkeyfacts* block in the Monkey script, we need to input a corresponding broadcast *when I receive monkeyfacts* block in a script within its Workspace

The ‘Arrow Instruction Text Box’ sprite
The Arrow Instruction Text Box sprite should be hidden when the Home Page is shown; appear only with the Monkey’s ‘Facts Box’ backdrop as a result of the script in the Monkey sprite; and should likewise disappear when the Arrow sprite is clicked by the user to move forward to the Question and Answer Quiz section. So use the following three scripts to achieve the desired results
However, as it is a sprite (rather than a backdrop), it needs the addition of an X and Y coordinate block to ensure that it appears in the same location every time. (Just in case that it was accidently moved manually).

The Arrow sprite
The Arrow sprite should function very similarly to the Arrow Instruction Text Box sprite, though having different X and Y coordinates.

However the touching of the → sprite is the mechanism that brings the user to the Question and Multiple Choice Answers section.

Hence will form part of the following script
The Monkey Quiz Backdrop (costume)

In response to the

in the Arrow script (above)

the following script should be inserted in the Backdrop code

which will lead to the appearance of MonkeyQuiz backdrop costume:

To crack open nuts, some monkeys use

- stones
- wood
- hammers

The Coloured Circles of Answer Sprites
There are three choices offered as the answer with only one being correct.

Hence we have to delineate in the code the two wrong answers from the correct one. Furthermore, the students need to be encouraged to continue until they select the right option in order to improve their knowledge of the subject matter.

The ‘Correct Answer’ Sprite
Let’s start with the correct option, namely the sprite beside the stones text.

Rename this sprite MonkeyYes in the section directly under the stage in the Scratch Interface
As this sprite should only appear onscreen in response to the

in the Arrow script, then the following needs to inputted into

its Workspace

The inclusion of the X and Y coordinates block above is critical as the sprite has to be
positioned beside the word *stones*.

Once the user selects this option the project needs to return to the Home Page so that
he/she can choose another animal to continue with the quiz.
Hence the following will form the third and final script for this sprite

The block above confirms for the user that he/she has
chosen the correct answer.

The ‘Wrong Answers’ Sprites
Rename these two sprites *MonkeyNo1* and *MonkeyNo2*

Both sprites will have two scripts similar to that in *MonkeyYes*
However, as clicking on either of these two sprites represents the wrong answer, the following needs to be inserted in both

```
when this sprite clicked
say (Sorry. Wrong answer) for 2 seconds
say Try again for 2 seconds
```

When the user gets the correct answer by clicking on the *MonkeyYes* sprite a broadcast needs to go to *MonkeyNo1* and *MonkeyNo2* in order to have them disappear from the stage.
This is achieved by the inclusion in both of the following script

```
when I receive gotohomepage
hide
gotohomepage
```

**The MonkeyQuiz Backdrop (Costume)**
The final script for the Backdrop in this series linked to the Monkey sprite is to have the MonkeyQuiz screen (costume) replaced by the HomePage (with Wildlife Quiz title)

```
when I receive gotohomepage
switch backdrop to homepage from Looks
```

**The Monkey Sprite**
The final script for the Monkey is a code that will have this sprite reappear after the multiple-choice quiz question is answered correctly.
To have it show onscreen along with the Home Page backdrop costume as well as with the other three animals and the Instruction Box sprite, combine the *when I receive gotohomepage* command with the *show* block from Looks
The ‘Instruction Box’ Sprite
In this sprite, the same *when I receive gotohomepage* and *show* blocks are required to have the Instruction Box show onscreen when the Home Page appears.

Coding the Elephant, the Lion & the Octopus – Follow the Monkey!

A similar coding process to the monkey, from clicking on the animal sprite to moving to the ‘Interesting Facts on Monkeys’ screen to taking part in the subsequent ‘Question & Multiple Choice Answer Quiz’ and finally returning to the Home Page (backdrop), has to be undertaken for the elephant, the lion and the octopus.

Each animal sprite has to have new associated (connected by broadcast commands) elements:
- ‘Interesting Facts’ backdrop
- Arrow sprite
- ‘Question & Multiple-Choice Answers’ backdrop
- Three circle answer (one correct and two incorrect) sprites.

The ‘Arrow Instruction Text Box’ sprite and the Home Page backdrop also require additional coding blocks to be integrated into the process for each animal.

So the message is quite clear – Learn from the Monkey and follow its example!

Exercise
The students can now independently or under the teacher’s guidance undertake the coding of the three other animal sprites and associated elements.
To help in the process, here are the *Interesting Facts and Question & Multiple-Choice Answers* screens for the elephant, the lion and the octopus.

**Elephant**

**Interesting Facts on Elephants**

- 1. Are the world's largest land animals
- 2. Have the best sense of smell of any mammal
- 3. Have long memories
- 4. During the dry season, they use their tusks to dig for water which allows other animals also to survive in harsh dry climates
- 5. Create gaps in forest vegetation that allows new plants to grow and provides pathways through the trees for small animals to use.

Elephants use their tusks to dig for:

- ants
- vegetables
- water

**Lion**

**Interesting Facts about Lions**

- 1. Are symbols of strength and courage
- 2. Have terrific night vision. Are six times more sensitive to light than humans
- 3. Lionesses are caring mothers who will even take care of a neglected cub
- 4. 50 years ago, there were 100,000 lions in the African wild. Now there are less than 20,000 due to poaching, poisoning and habitat loss
- 5. Are the only big cats to live in family groups, called *prides*
A family group of lions is called a:

- herd
- pride
- swarm

Octopus

<table>
<thead>
<tr>
<th>Interesting Facts about Octopuses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Have eight limbs</td>
</tr>
<tr>
<td>2. Have no bones</td>
</tr>
<tr>
<td>3. Have three hearts</td>
</tr>
<tr>
<td>4. Can change colour and skin texture to blend in with their surroundings</td>
</tr>
<tr>
<td>5. When attacked, can release a cloud of black ink to hide their escape</td>
</tr>
</tbody>
</table>

To protect itself from an attacker, an octopus can:

- fire a spray of water
- release a cloud of black ink
- use its limbs to throw stones from the ocean floor

By the Way...

The completed project is located on the ACW Mentors Scratch account at https://scratch.mit.edu/projects/414137108/
Project 7 – A Geography Quiz

As with the example shown in project six, the quiz game format can be an enjoyable way for young people to test their knowledge on a subject and for teachers to assess their progress.

The sample project below shows one way of how it can be used in geography. It represents a tour of Africa with the questions based on the names of the countries. Of course, it could instead involve naming the main languages, capitals, mountains, rivers, lakes, etc.

Project Play - Coding Plan Summary

The tour guide is Gobo who travels across Africa stopping off at different countries. At each stop he asks the user to name the country.

The respondent cannot move from the country until he/she types in the answer correctly. Once the correct answer is inputted, Gobo travels to the next country to ask the same question until all highlighted countries are visited.
Requirements

One sprite (Gobo) and one backdrop (Map of Africa) with only scripts being used in the former.

Map of Africa Backdrop
The map should consist of the continent of Africa with only the national boundaries of each country outlined.

Go to the Costumes option of the Backdrop and use the paint brush or the fill options in the Tool Box to colour in a number of countries, giving each one a different colour.

It is the actual colour that will define the country in the coding process.

This is achieved by first selecting the touching color block in the Sensing category and dropping it into the relevant script of the sprite. (all code is placed in the Workspace of the Gobo sprite).

The specific colour that the country is shaded in is chosen by first right clicking on the colour shading in the block.
Select the eye drop icon at the bottom of the drop-down box that now appears.

Then position it over the coloured country that you have selected.

Click once and the result is that the colour will now appear in the box within the touching colour block.

Repeat this process for each country.

**Warning!**

Ensure that as the sprite moves from country to country, it does not touch another coloured country as the colour will trigger the code.

Blocks used in this project will be located in the *Control, Looks, Motions, Sensing* and *Operations* categories

Unlike the detailed text instructions of the previous five projects, we will use image representations of the scripts.

Below are the stages of building the scripts
Script 1:

Part 1

when flag clicked
go to x: 104 y: 102
say Hello! for 2 seconds
say My name is Gobo for 2 seconds
say Please join me on a tour of Africa for 2 seconds
glide 2 secs to x: -127 y: 49

Part 2 (connected to the Part 1 blocks of code above)

forever
if touching color ? then
ask What is the name of this country? and wait

Part 3 (connected to Part 1 & Part 2 above)

forever
if touching color ? then
ask What is the name of this country? and wait
if answer = Nigeria then
think Hmm... for 2 seconds
wait 1 seconds
say Correct answer. Well done! for 2 seconds
broadcast goto angola
else
say Please try again for 2 seconds
Part 4

which is combined within Part 3 above to give
The complete script (Part 1 to 4) will then be
Script for the country **Angola**

Follow the same procedure for all the other colour highlighted countries.

**Exercise**

Get the students, as individuals or in groups, to make a similar project based on the countries of another continent. This may involve them independently or together with you undertaking advanced research and planning in order to obtain the necessary relevant information (e.g. capitals of countries, or languages spoken, or geographical features such as main rivers, mountains, etc).

**By the Way...**

The completed project is located on the ACW Mentors Scratch account at [https://scratch.mit.edu/projects/414173543/](https://scratch.mit.edu/projects/414173543/)