



Guess the flag

Introduction

In this resource, you will create a flag quiz to test yourself and your friends. Six flags and the name of a country are displayed, and you have to click on the correct flag to match the country.

What you will make

Click on the flag belonging to the country for which you are being asked.

What you will learn

This project covers elements from the following strands of the [Raspberry Pi Digital Making Curriculum](http://rpf.io/curriculum) (<http://rpf.io/curriculum>):

- [Apply basic programming constructs to solve a problem](https://curriculum.raspberrypi.org/programming/builder/) (<https://curriculum.raspberrypi.org/programming/builder/>)
-

What you will need

Hardware

- A computer capable of running Scratch

Software

- Scratch 2.0 (either [online \(https://scratch.mit.edu/projects/editor/\)](https://scratch.mit.edu/projects/editor/) or [offline \(https://scratch.mit.edu/scratch2download/\)](https://scratch.mit.edu/scratch2download/))
-

Draw the flags

- Open the starter project.

I'm using Scratch online

- Open the 'Guess the flag' Scratch starter project at jumpto.cc/flags-go (<http://jumpto.cc/flags-go>).
- If you have a Scratch account, you can click on **Remix** in the top right-hand corner to save a copy of the project to your account

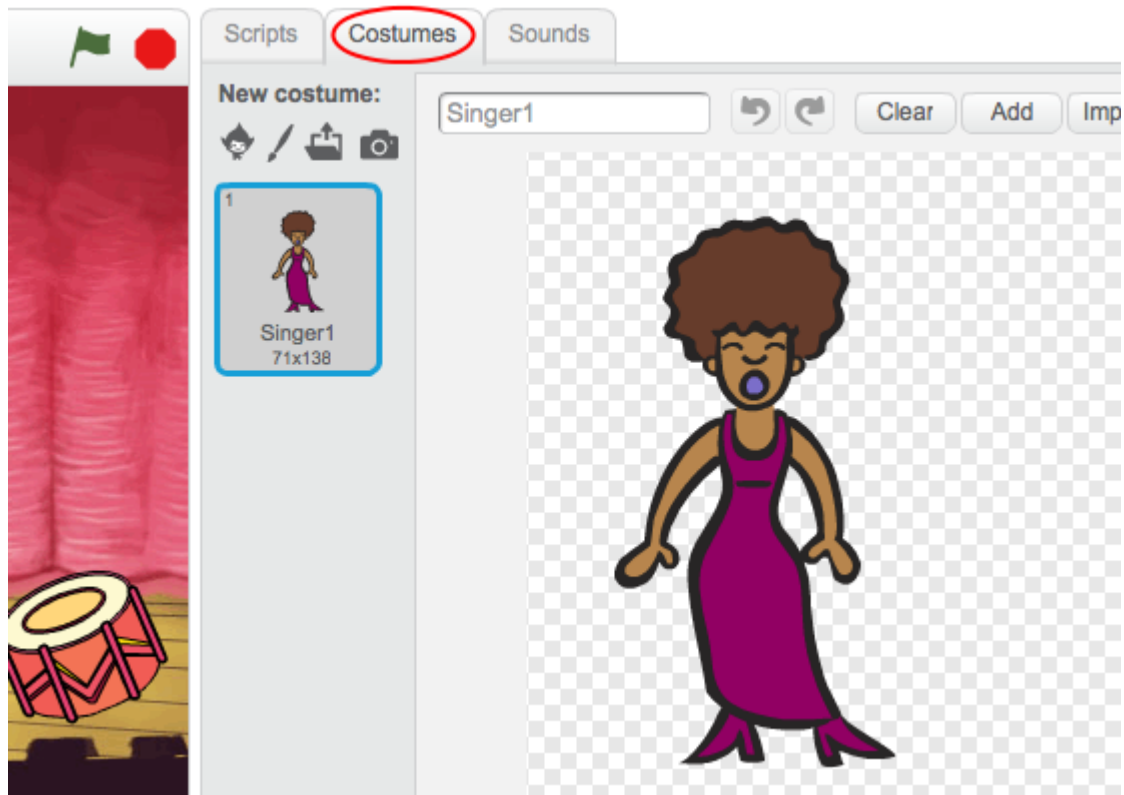
I'm using Scratch offline

Download the [flags.sb2 \(https://s3.eu-west-2.amazonaws.com/learning-resources-production/projects/guess-the-flag/d4b42ded3627f3f31b550692c71358e9ff1f058d/en/resources/flags.sb2\)](https://s3.eu-west-2.amazonaws.com/learning-resources-production/projects/guess-the-flag/d4b42ded3627f3f31b550692c71358e9ff1f058d/en/resources/flags.sb2) Scratch starter project and open it using the offline editor.

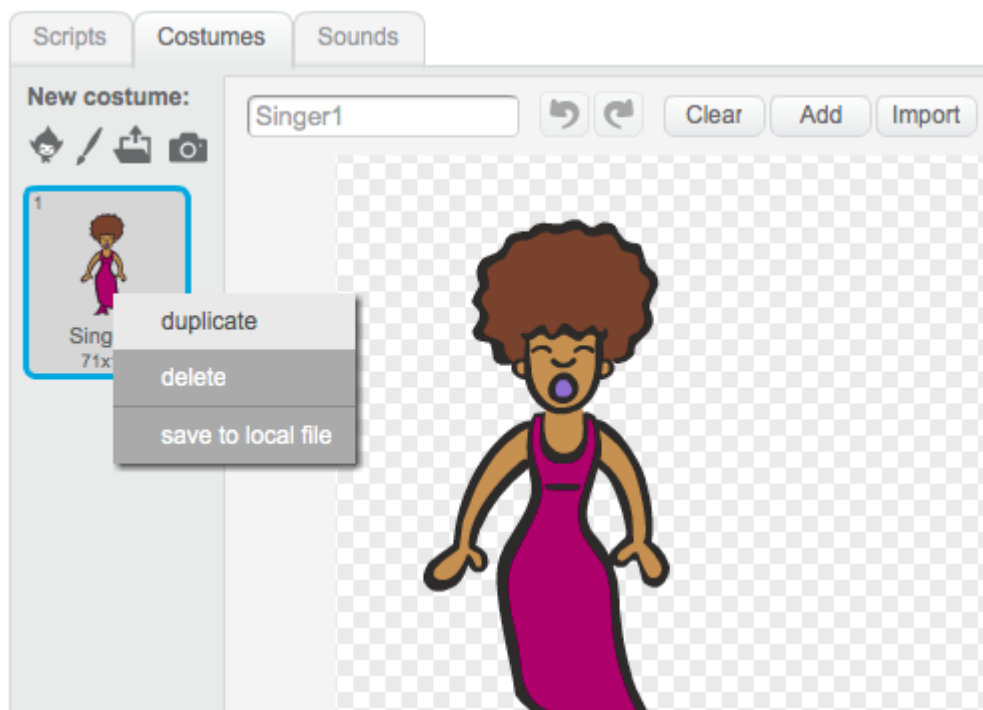
- Click on the flag sprite and then click on the Costumes tab. You will see a blank flag template. Create nine duplicate costumes of the blank flag template, so you have ten flags in total.

Duplicate costumes in Scratch

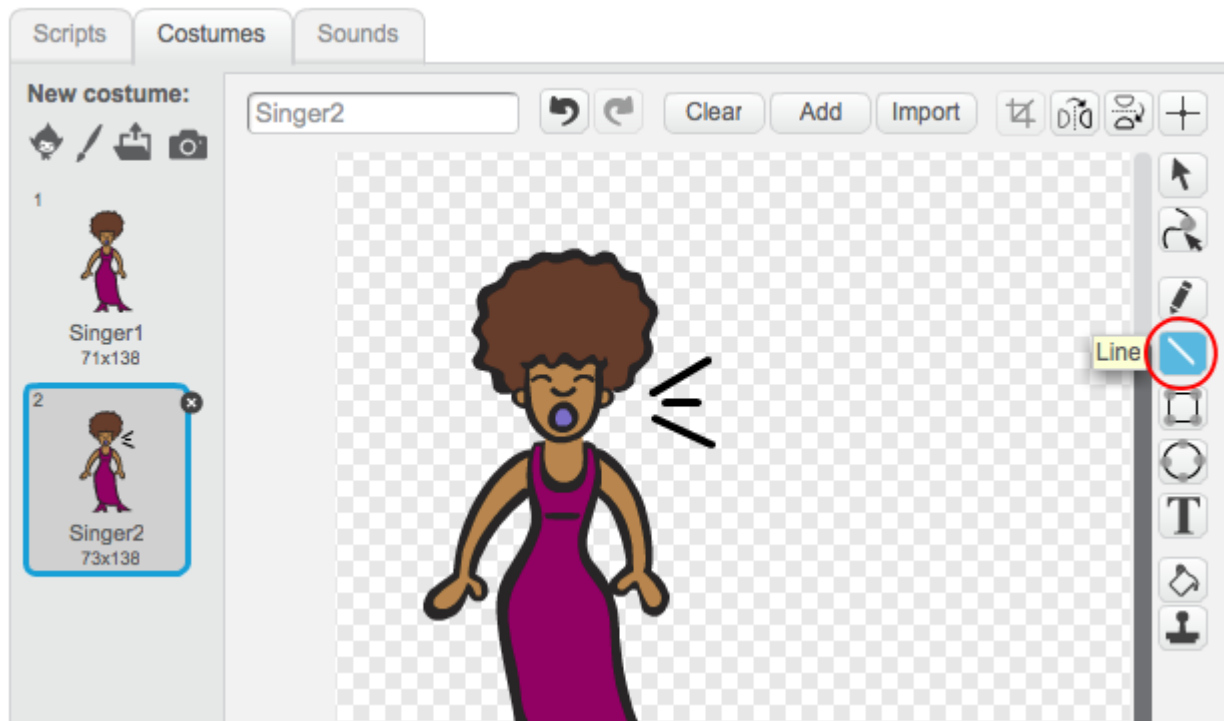
- Click the **Costumes** tab, and you'll see your sprite's costume.



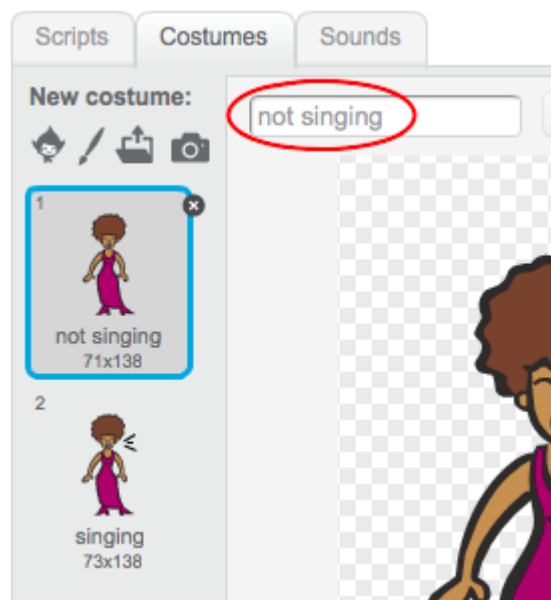
- Right-click on the costume and click **duplicate** to create a copy of the costume.



- Click on your new costume, and then use the drawing tools to edit the costume.



- Give your costumes more helpful names.

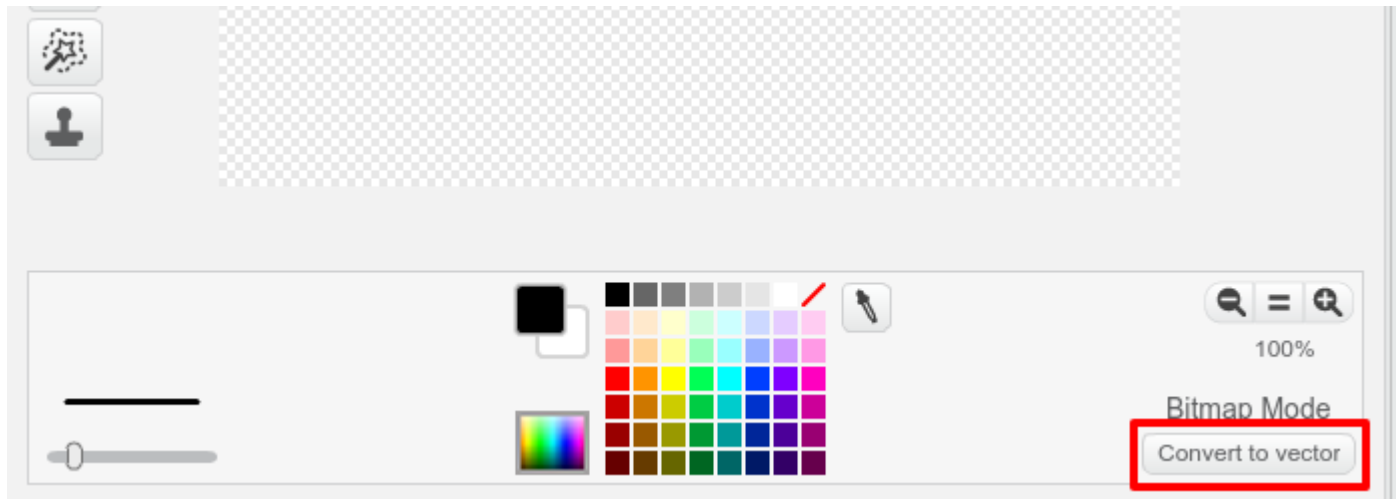


- Now we will draw the flags. Click on the first costume and add the name of a country as the costume's name.
- Use the **vector drawing tools** to draw that country's flag. Make sure your drawing is exactly the same size as the flag template.

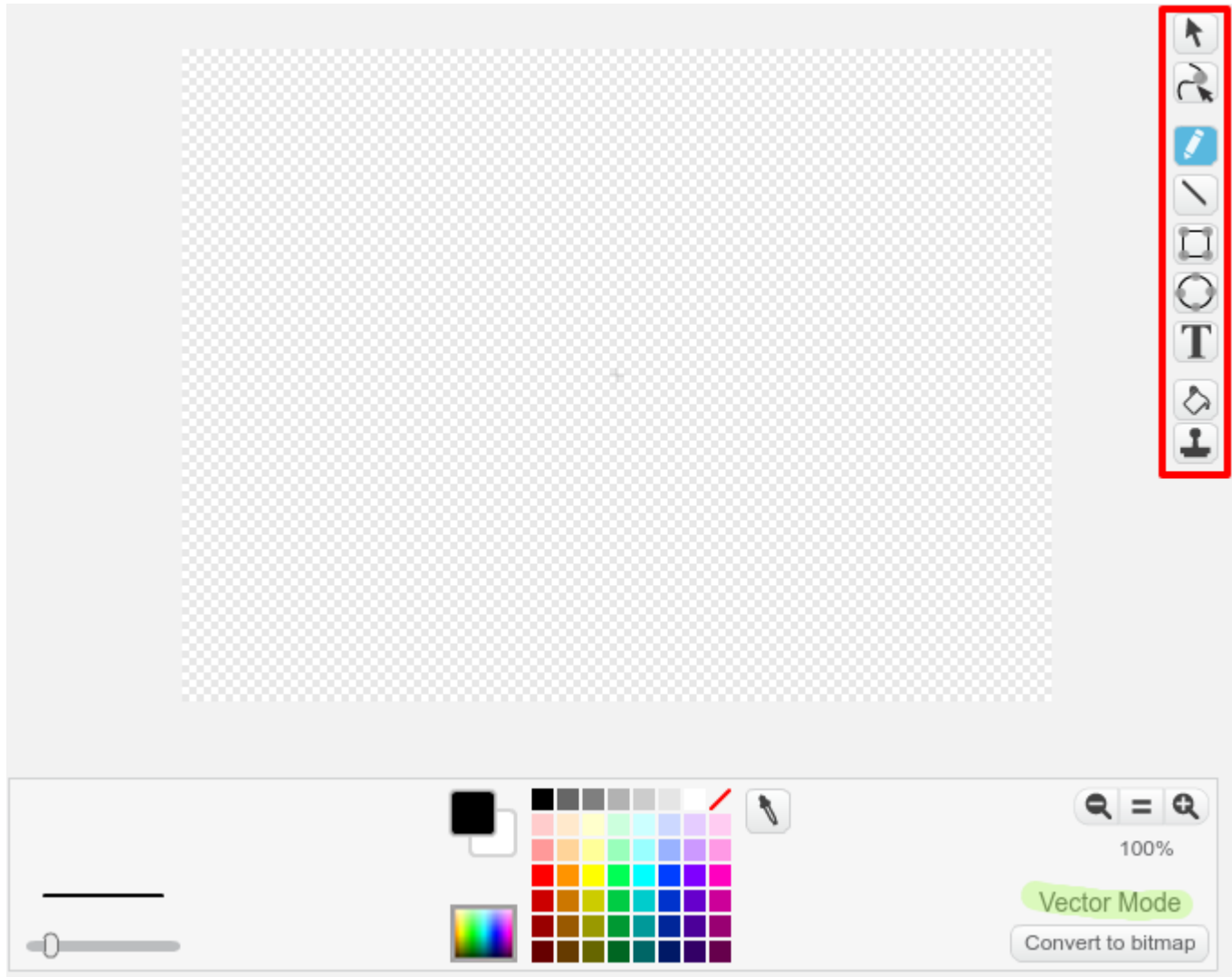
Scratch paint tool vector mode

The Scratch paint tool has two modes: bitmap and vector. Vector mode stores your picture as shapes and lines that you can move around individually. This is really useful!

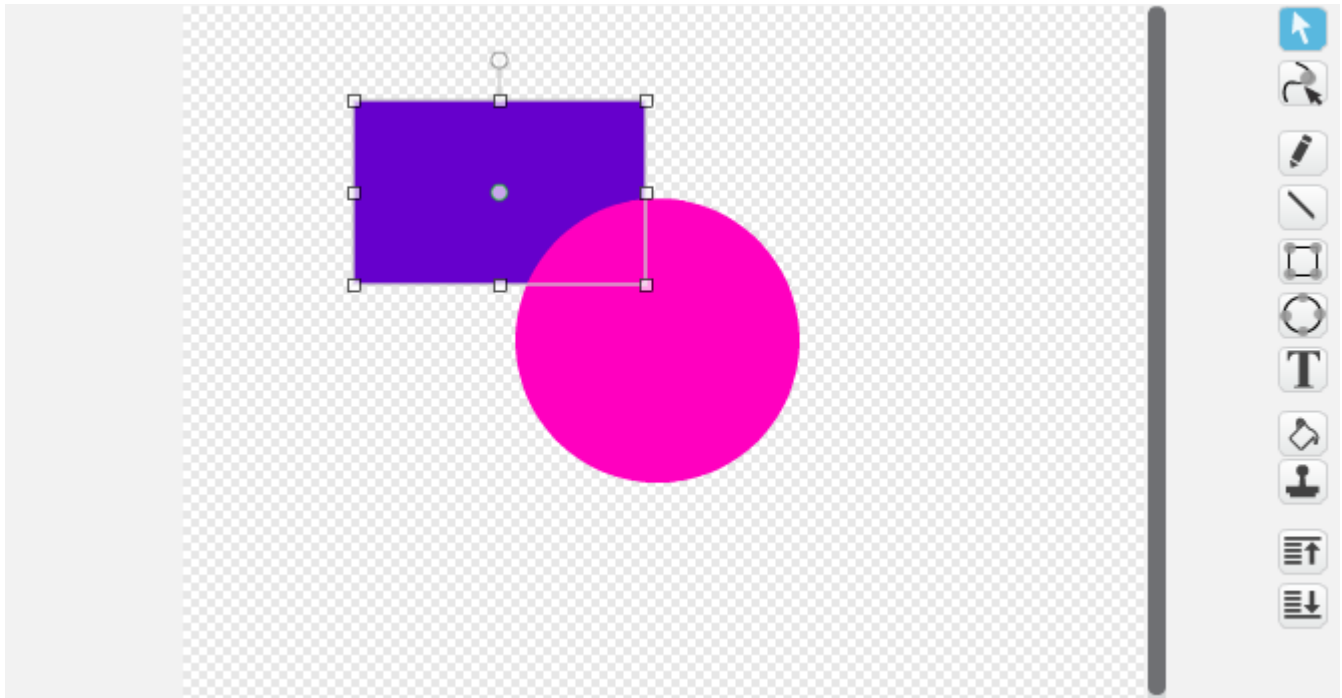
When you are in bitmap mode you can click **Convert to vector** to switch to vector mode.



In vector mode, the tools appear on the right side of the drawing area.



Choose the select tool (the arrow) to move, resize, and rotate shapes and lines.



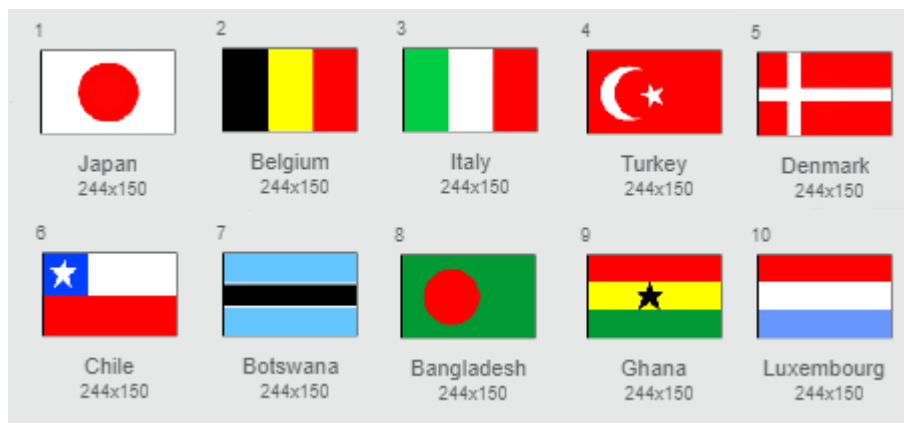
For example, you can draw the flag for Italy by drawing three equally sized rectangles using the rectangle tool. Position them using the selection tool.



You could easily create a similar flag by duplicating this costume and then changing the colours. For example, to draw the flag for Belgium, duplicate the costume designed to look like the the Italian flag, and use the fill bucket tool to fill each rectangle with a different colour.

- Draw another nine flags so that you have exactly ten different flag costumes in total. Make sure that the name of each costume is the name of the flag's country.

Here are the costumes we chose, but you can choose any flags you like for your game.



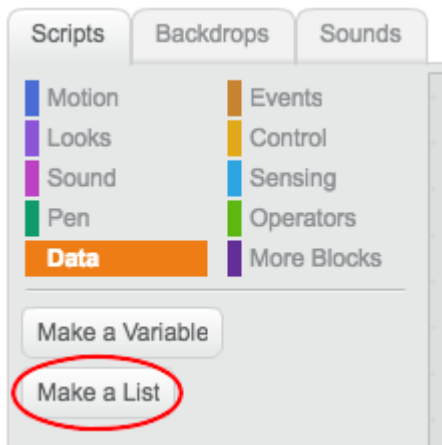
Create a list of flags

- Click on the Scripts tab, and then on the flag sprite.
- Create a list called `Flags`. This will be where we store the names of all of the countries whose flags we have included in the game.

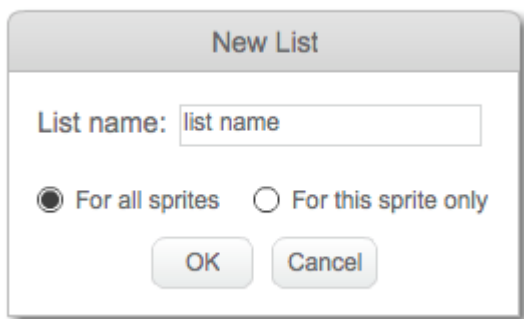
Make a list

Make a list

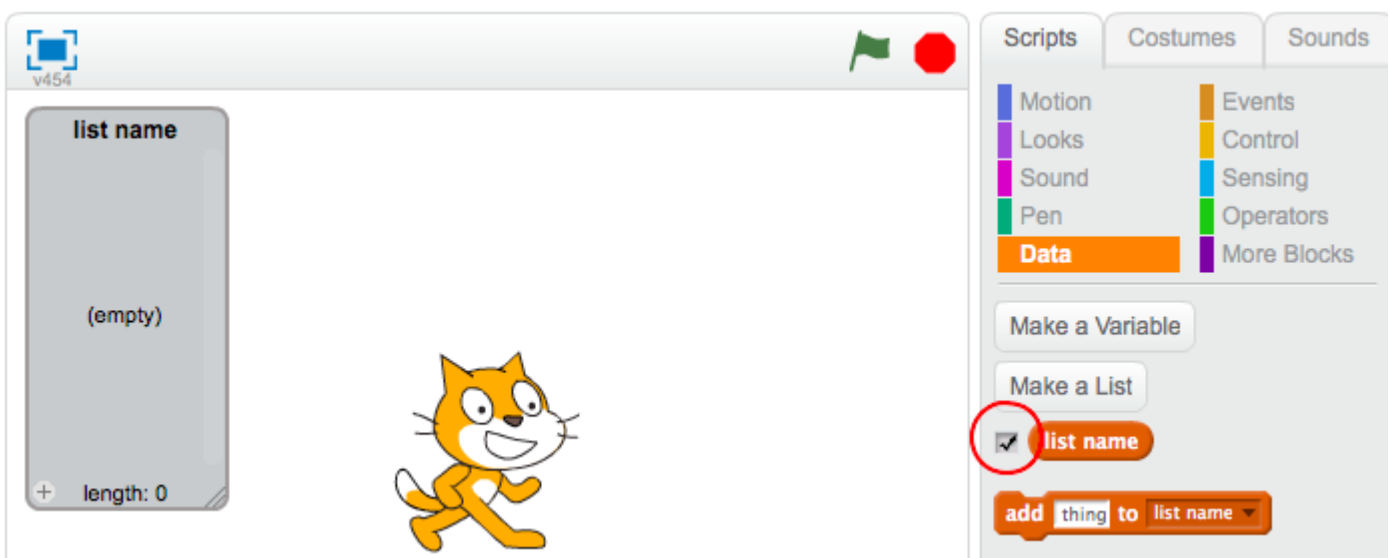
- Click on **Data** in the **Scripts** tab, then click on **Make a List**.



- Type in the name of your list. You can choose whether you would like your list to be available to all sprites, or to only a specific sprite. Press **OK**.



- Once you have created the list, it will be displayed on the stage, or you can untick the list in the **Scripts** tab to hide it.



- Click the + at the bottom of the list to add items, and click the cross next to an item to delete it.



- New blocks will appear and allow you to use your new list in your project.



- Drag some blocks into the scripts area to add all of the countries to the **Flags** list. You can start off like this:



- Check that, when you press the green flag, the countries appear in the list.

If you press the green flag more than once, you will see that the countries are added to the list again, so you end up with a list of 20 countries and not 10.

- Add a block to, before adding the countries, **delete all** of the countries in the list. This will stop the countries from being added to the list more than once.

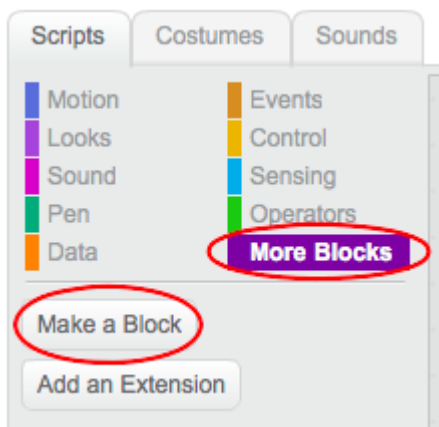
Let's make a custom block. This is a special block with a name, and we will be able to create a list of flags using only this one block, instead of having to use lots of blocks.

- Call your custom block **Create flags list**, and drag all of the code (except the **when flag clicked** block) to become part of this new block.

Making a block

Making a block

- Click the **Scripts** tab, then on **More Blocks**, and then click **Make a Block**.



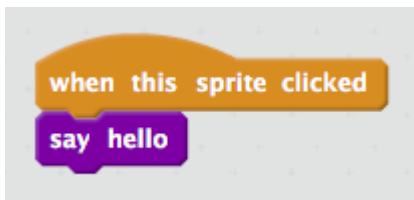
- Give your new block a name and then click **OK**.



- You will see a new **define** block. Attach code to this block.



- You can then use your new block just like any normal block.

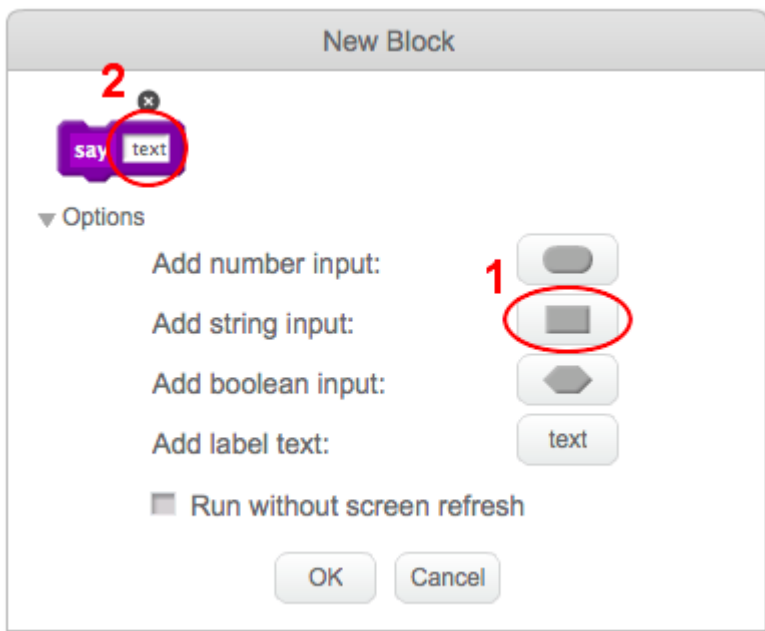


- The code attached to your new **define** block is run whenever the block is used.

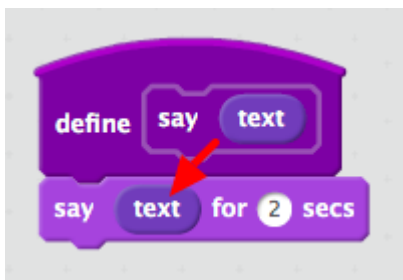


Making a block with parameters

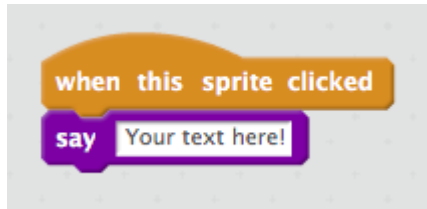
- You can also create blocks that have 'gaps' for adding data. These 'gaps' are called 'parameters'. To add parameters, first make a new block, and then click on **Options** to choose the type of data you want to add. Then give your data a name, and click **OK**.



- You will see a new `define` block as usual, except that this one contains the data gap you added and which you gave a name.



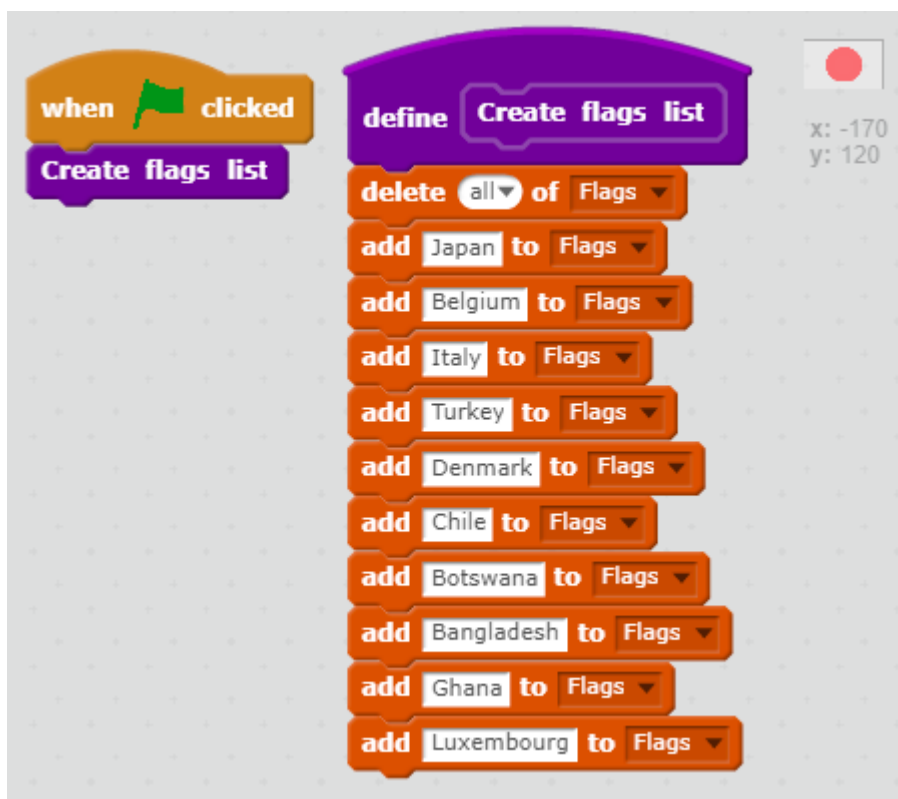
- You can then use your new block, filling in data in the gap.



- As usual, the code attached to your new `define` block is run whenever the block is used.



Your code for when the green flag is clicked should now look similar to this:



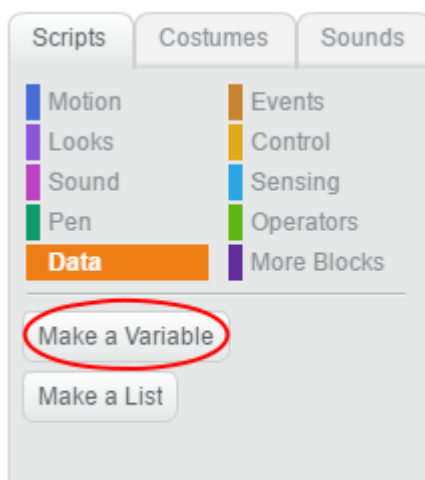
Choose random flags

We need to select six random flags from the flags list to be the possible choices in the game.

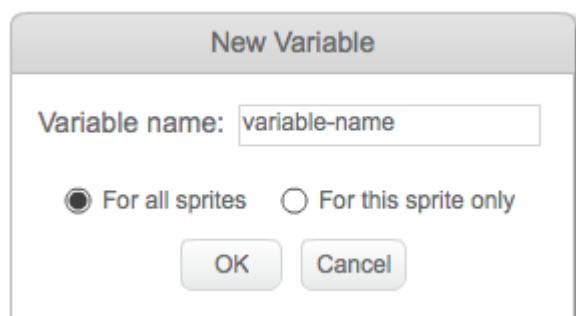
- Create another list called **Chosen flags**. This list will store the six chosen flags.
- Now create a variable called **Flag number**.

Add a variable in Scratch

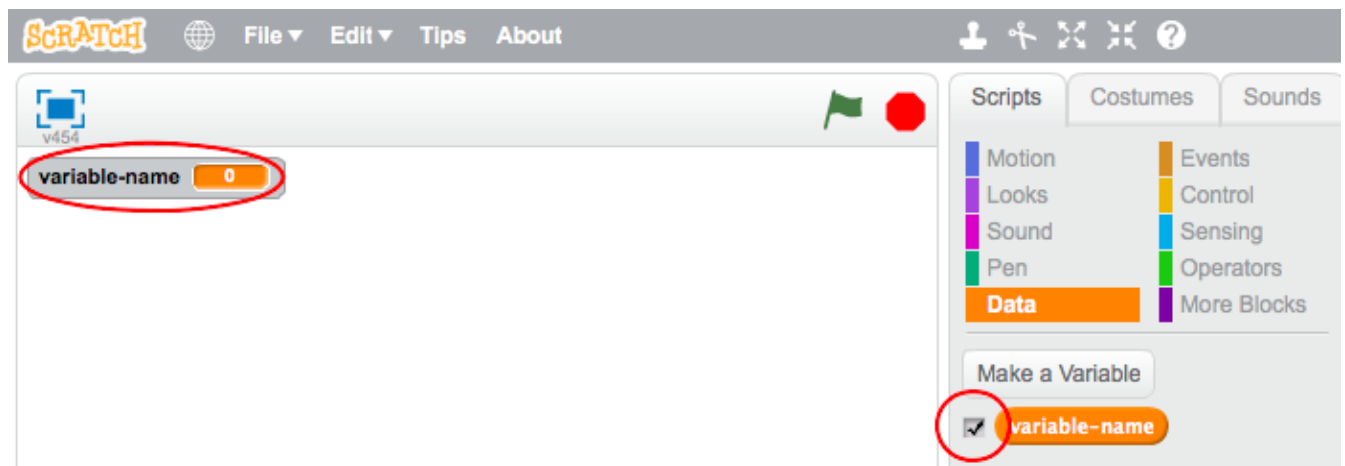
- Click on “Data” in the Scripts tab, then click on “Make a Variable”



- Type in the name of your variable. You can choose whether you would like your variable to be available to all sprites, or to only this sprite. Press OK.



- Once you have created the variable, it will be displayed on the stage, or you can untick the variable in the Scripts tab to hide it.



- New blocks will appear and allow you to change the value of the variable.

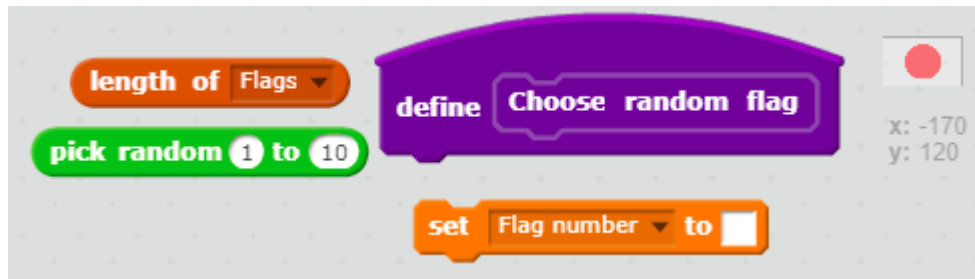


- Add a custom block, just like you did for the `Create flags list` custom block. Call this new custom block `Choose random flag`.
- Add code to this block to set the `Flag number` variable to a random number between 1 and the number of items in the `Flags` list. You will find that there is a special block in the Data tab for finding the number of items in a list.

I need a hint

Set the `Flag number` variable to a random number between 1 and the `length` of the `'Flags'` list.

Here are the code blocks you'll need:



This is what your code should look like:



This block selects an item from a list, by number:

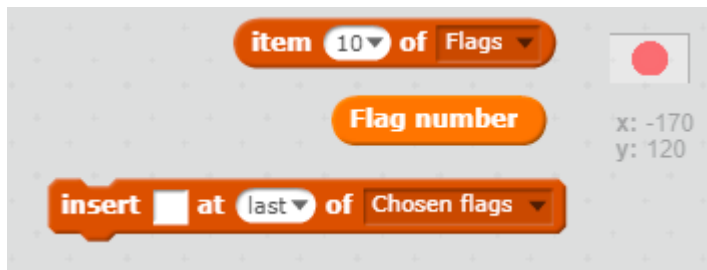


- Combine this block with the `Flag number` variable to get the text of the randomly chosen item in the `Flags` list. Insert the item text into the `Chosen flags` list. Add this code to your custom block.

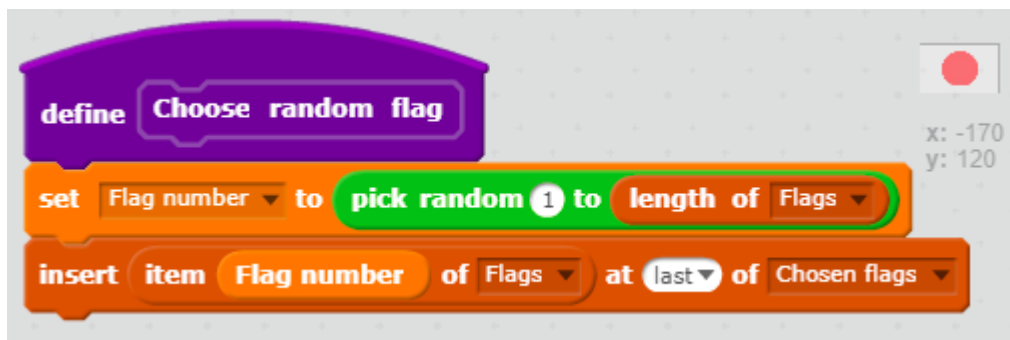
I need a hint

Insert the text from the item at the `Flag number` position on the `Flags` list into the `Chosen flags` list.

Here are the code blocks you'll need:



This is what your code should look like:



- Add your custom block to the code that will be run when the green flag is clicked.
- Test that your code works by clicking the green flag several times and checking that different countries are added to the **Chosen flags** list. (If you have hidden the list, tick the box next to it to make it visible.)

You will notice that, if you press the green flag lots of times, your **Chosen flags** list quickly fills up with more than six choices.

- Add blocks to delete all of the countries from the **Chosen flags** list before choosing six flags for the game.

I need a hint

Delete all of the chosen flags. Repeat choosing a random flag six times.

Here are the code blocks you'll need:



This is what your code should look like:



- Test your code by clicking the green flag and checking that the **Chosen flags** list is filled with six countries each time.

You might notice that sometimes the same country gets chosen more than once in the list.



- Add a block to the end of your custom block code to delete the randomly chosen `Flag number` from the `Flags` list after it has been added to the `Chosen flags` list. This will stop it from being chosen more than once.



Choose a correct answer

Now that we have a list containing six chosen flags, let's choose which of them will be the correct answer this time.

- Create a new variable called `Correct answer`.
- After you have chosen the six flags, set the `Correct answer` variable to be a random item from the `Chosen flags` list.

Add your code like this:



You can use the same block we used before:



However, you will need to change the drop-down menu to choose a random item instead of a specific item number.

Show the flags

The player needs to be able to see the six possible flags to make their choice in the game, so we need to display the pictures of the flags that were selected to be in the `Chosen flags` list.

- Create another custom block, this time called `Clone flags`.

We will clone the Flag sprite six times, once for each flag that will be displayed. We would like the first flag to be displayed in the top left-hand corner of the page.

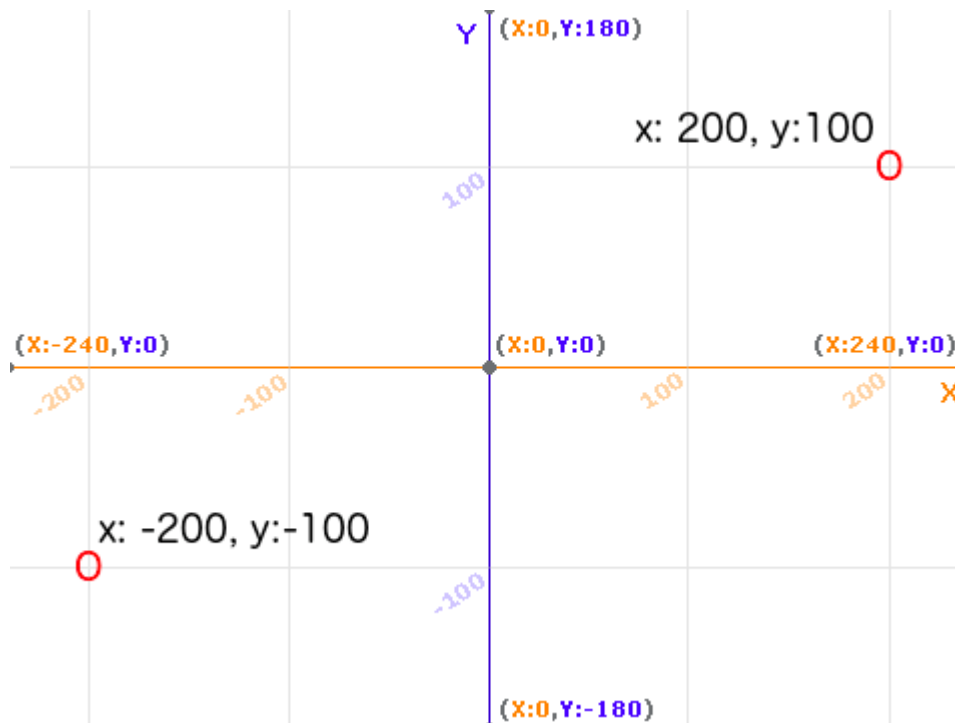
- Move your mouse to a point near the top left-hand corner of the stage. This will be where the centre of your first flag sprite appears, so don't go too close to the edge. Note down the coordinates of the point you chose.

Scratch coordinates

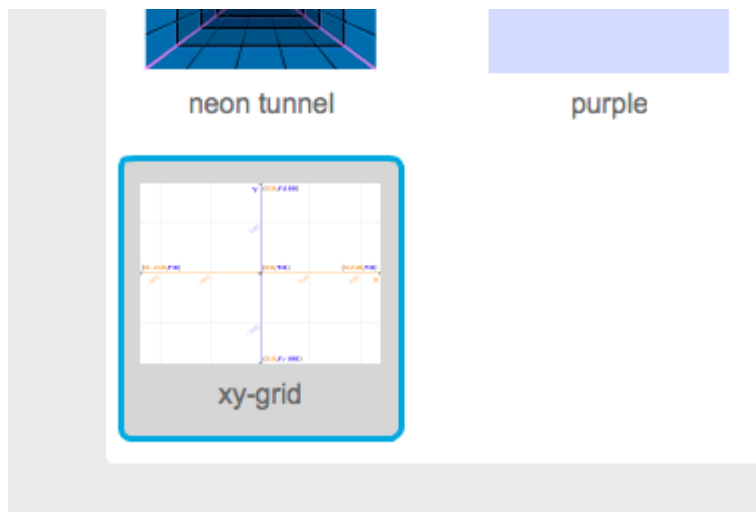
Scratch coordinates

- In Scratch, the coordinates $x:0$, $y:0$ mark the central position on the Stage.

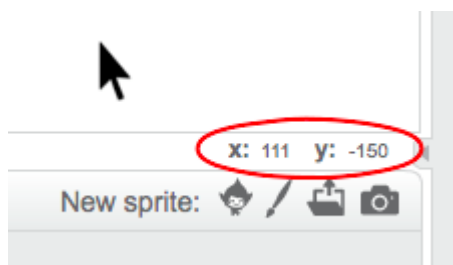
A position like $x:-200$, $y:-100$ is towards the bottom left on the Stage, and a position like $x:200$, $y:100$ is near the top right.



- You can see this for yourself by adding the **xy-grid** backdrop to your project.



- To find out coordinates of a specific position, move your mouse pointer to it and check the readings below the bottom right corner of the Stage.

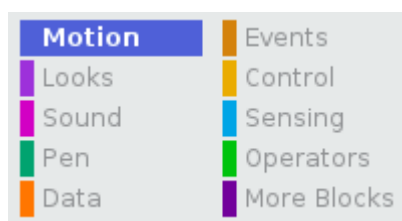


- Add a **show** block to make the sprite visible, and a **go to** block to your custom block to tell the flag sprite to begin at these coordinates.

Set a sprite's coordinates in Scratch

To set a sprite's coordinates so that it appears at a certain location on the Stage, follow the steps below.

- Click on the **Motion** menu in the **Scripts** palette.



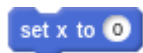
- Find the `go to x: () y: ()` block.



- Type in the `x` position and `y` position that you want your sprite to go to.



- If you only want to set the `x` or `y` position, you can use either of the following two blocks instead.



- Now add a loop that repeats six times to display the flags.
- Add code inside the loop to switch the costume to the last flag in the `Chosen flags` list and to clone the sprite. Then, delete the last flag from the list and add 110 to the `x` coordinate to move along ready to place the next flag.

I need a hint

Repeat six times:

Switch costume to the last item in chosen flags.

Clone the sprite.

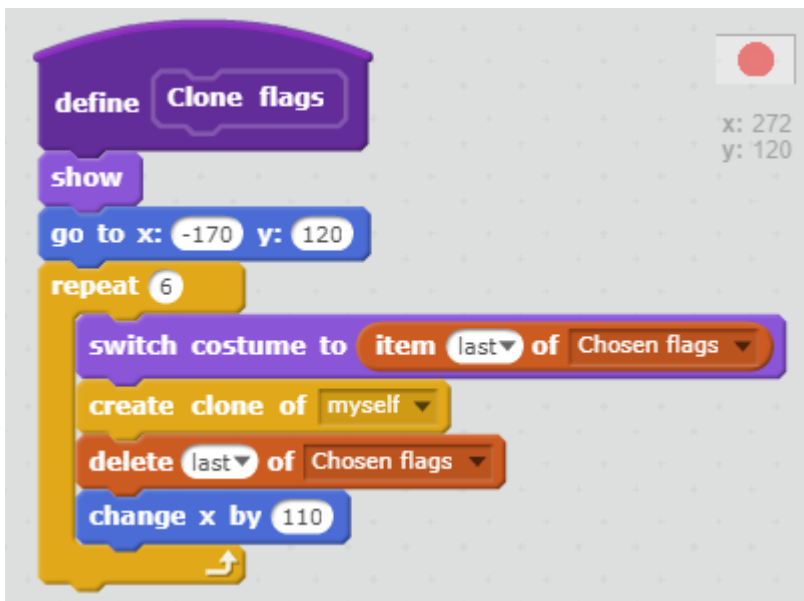
Delete the last item in chosen flags.

Move right 110.

Here are the code blocks you'll need:



This is what your code should look like:



- Run your code by pressing the green flag. You will notice that the different flags do appear, but they go off the end of the stage.



Let's make two rows of three flags.

- Add some code inside the loop to move down a row if there are three flags left in the `Chosen flags` list.

You can move down a row by using another `go to` block and keeping the `x` coordinate the same as the starting point, but decreasing the `y` coordinate to move downwards.

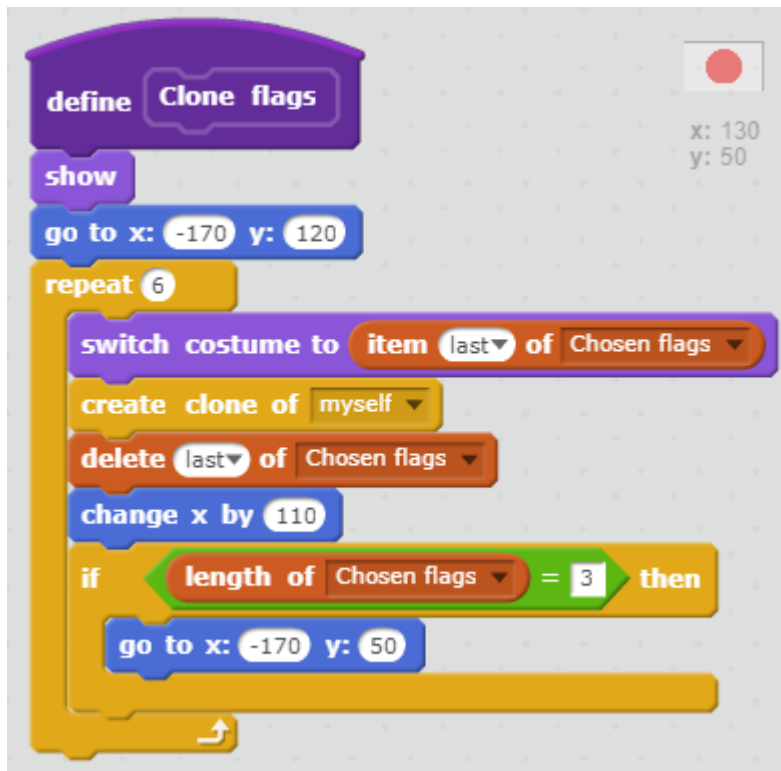
I need a hint

If the `length of chosen flags` is 3, `go to` the same `x` coordinate but a smaller `y` coordinate.

Here are the code blocks you'll need:



This is what your code should look like:



- Press the green flag and check that you now have flags in two rows.

It looks like the last flag is displaying twice. In actual fact what is happening is that the original flag sprite is still visible at the end.

- Add a `hide` block at the end of your custom block's code to hide the original sprite.
- Add your new `clone flags` block to the end of the code that runs when the green flag is clicked.

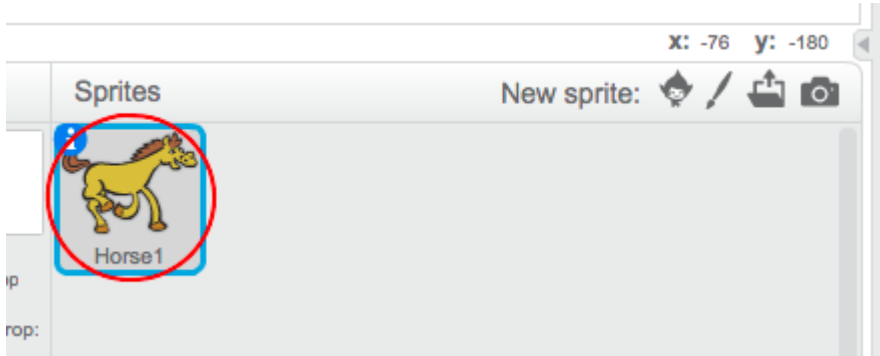
Challenge

- Can you make the flag sprites appear one by one?
- Can you make each flag make a sound (a pop, for example) when it appears?

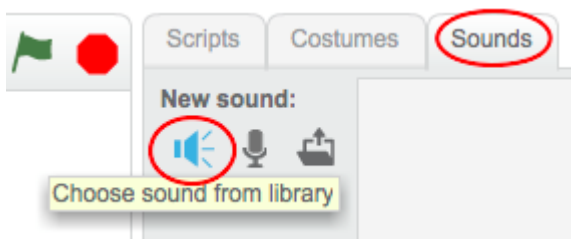
Adding a sound from the Scratch library

Add a sound to your Scratch sprite

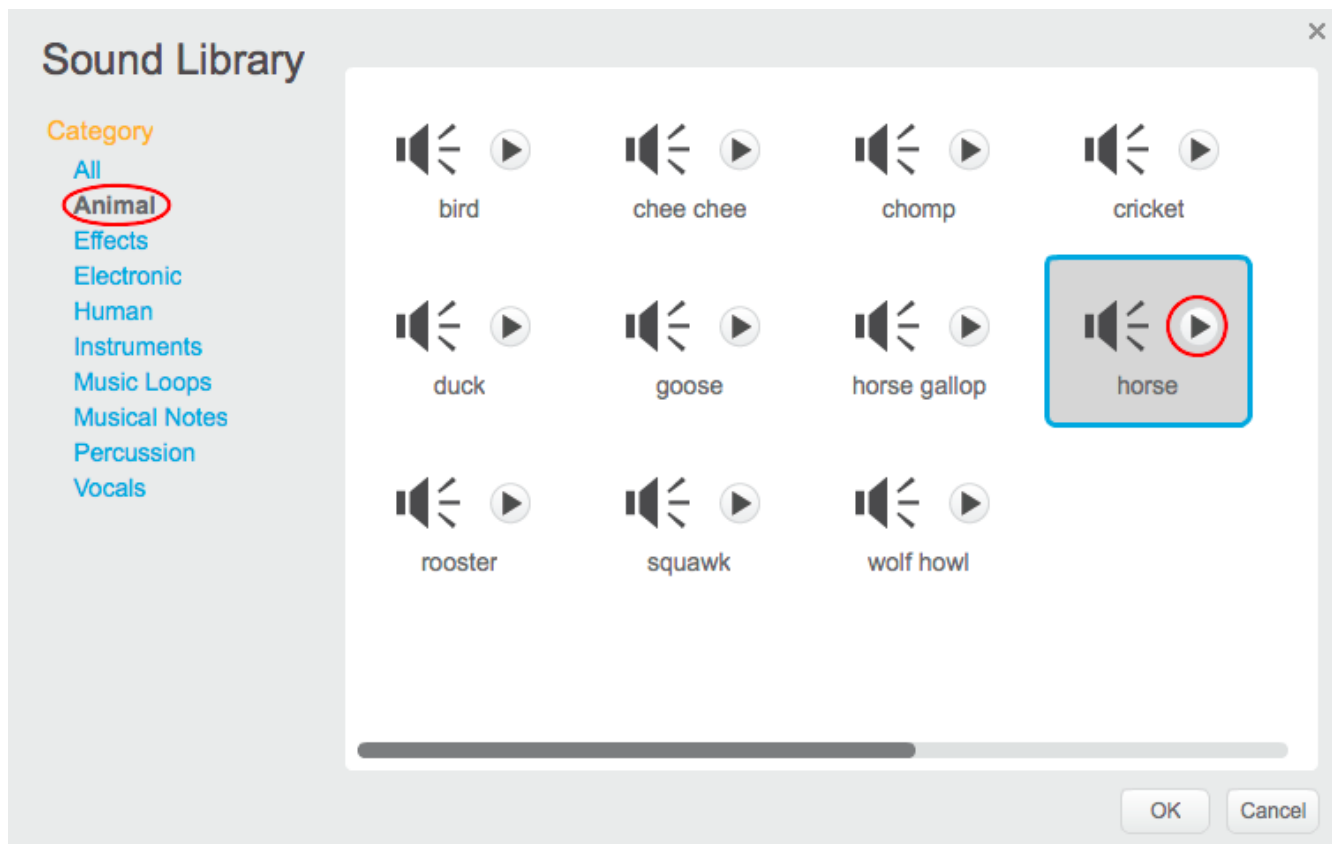
- Select the sprite you want to add the sound to.



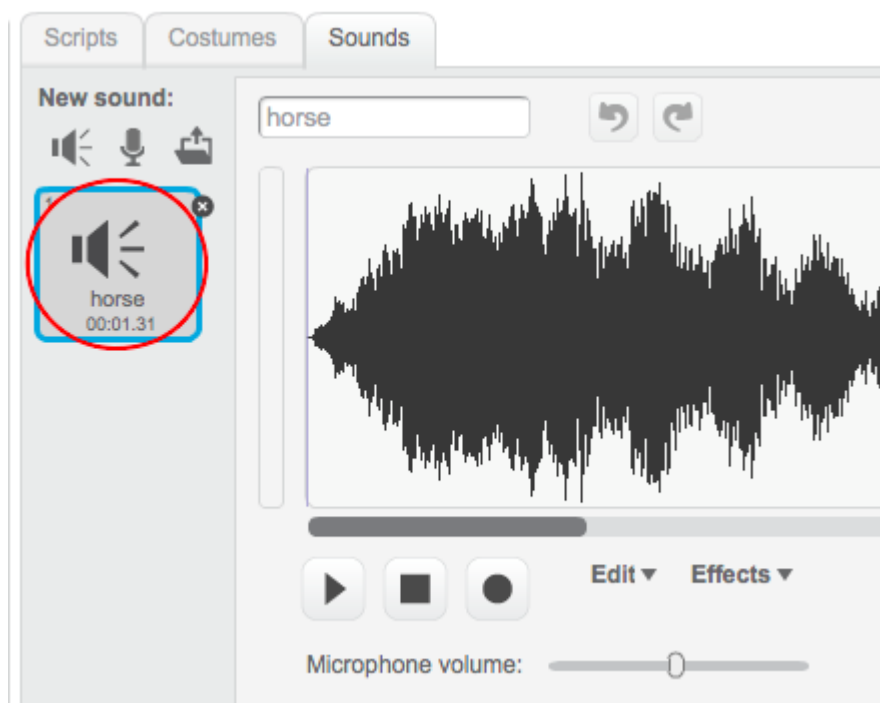
- Click the **Sounds** tab, and click **Choose sound from library**:



- Sounds are organised by category, and you can click the **Play** button to hear a sound. Choose a suitable sound and click **OK**.



- You should then see that your sprite has your chosen sound.



Ask the question

- In the **Flag** sprite, broadcast the message “Announce country” immediately after the block which clones the flags.

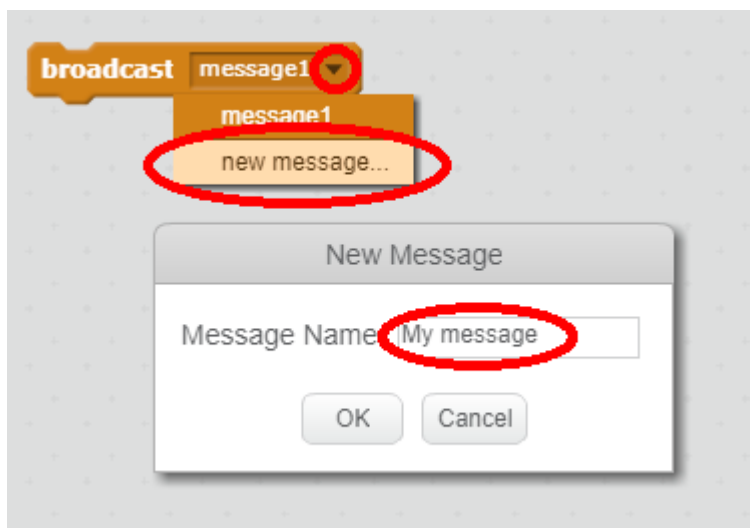
Broadcast a message in Scratch

A broadcast is a way of sending a signal from a sprite which can be heard by all sprites. Think of it like an announcement made over a loudspeaker.

Send a broadcast

You can send a broadcast by creating a broadcast block and giving it a name:

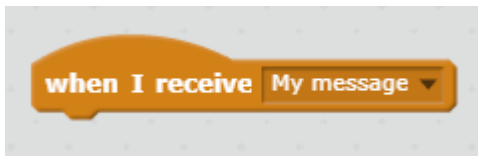
- Find the broadcast block in the **Events** tab
- Select **New Message** in the drop-down menu, then type your message.



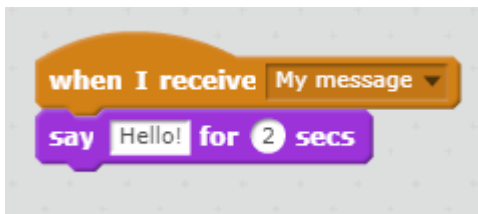
The message text can be anything you like, but it is useful to give the broadcast a sensible description. What happens when the message is received depends on the code you write.

Receive a broadcast

A sprite can react to a broadcast by using this block:



You can add blocks below this block to tell the sprite what to do when it receives the broadcast signal.



- Add a new sprite of your choice to be your quiz master. We chose the bear for our example.

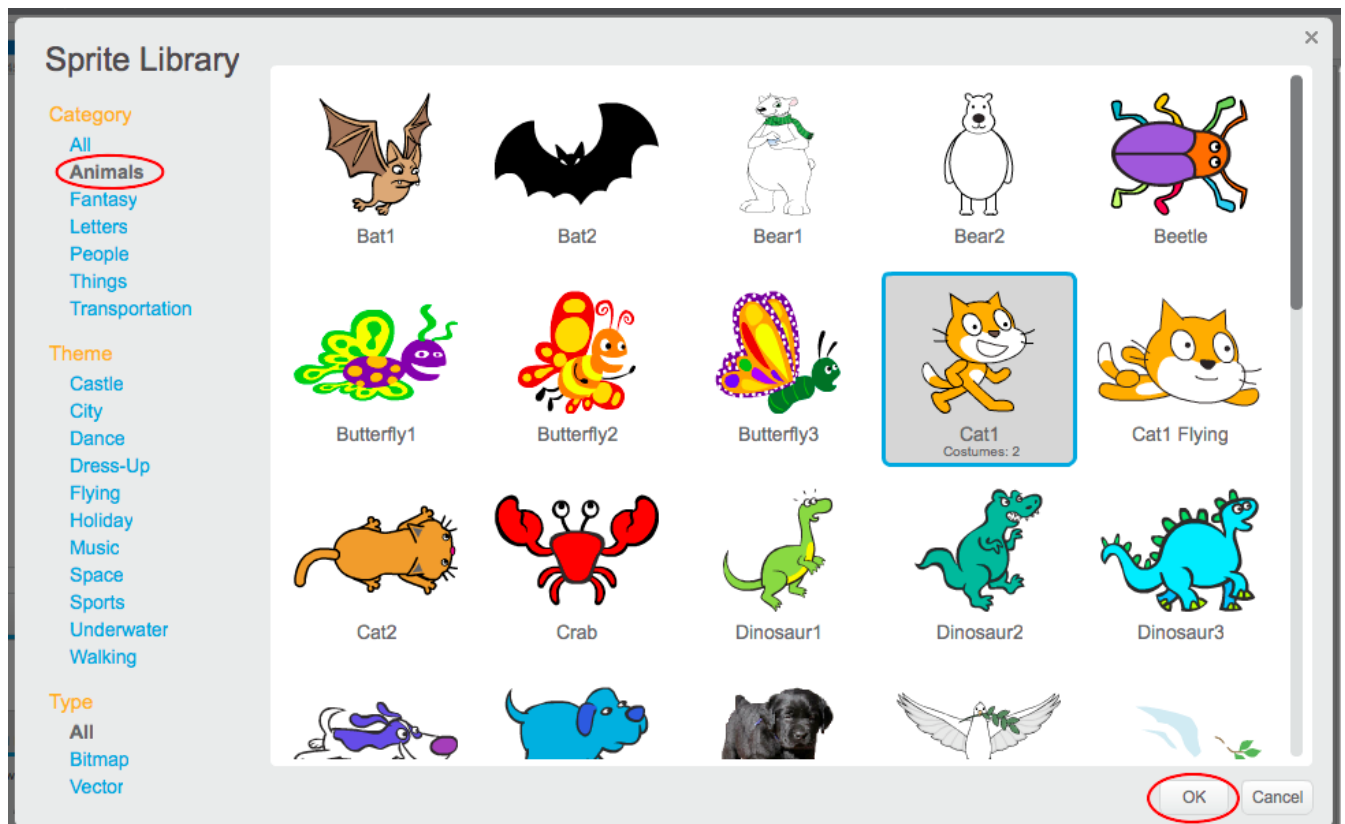


Adding a Scratch sprite from the Library

- Click **Choose sprite from library** to see the library of all Scratch sprites.



- You can browse sprites by category, theme, or type. Click on a sprite and click **OK** to add it to your project.



- Add some code to the bear sprite so that, when it receives the broadcast "Announce country", the bear will tell the player to click on the country name from the variable `Correct answer`.

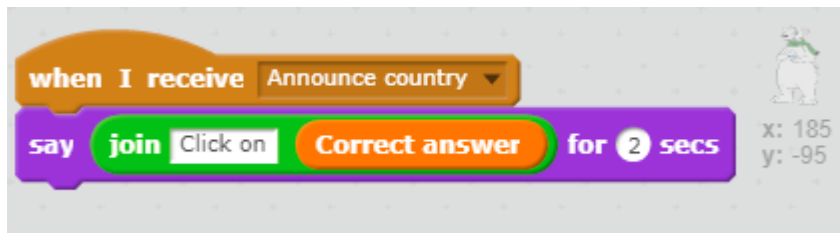
I need a hint

When I receive the broadcast, say 'Click on the correct answer'.

Here are the code blocks you'll need:



This is what your code should look like:



Check the answer

We have asked the player to click on the flag they think is correct. Now we need to check whether the flag they clicked was the right answer.

- Go back to the flag sprite code, and add a block to start a new section of code which will happen **when this sprite is clicked**.

We need to check whether the costume of the sprite that has been clicked has the same name as the correct answer.

- Add code to say 'Correct' if the costume name of this sprite is the same as the **Correct answer** variable, or to say 'Sorry, that was wrong' if it is not.

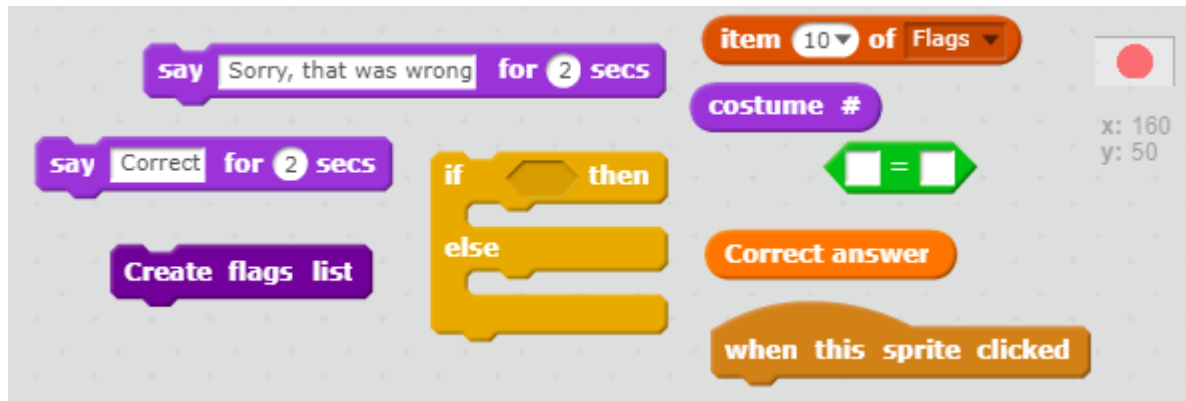
You can use this useful block here as well. This time, combine it with a **costume #** block to get the name of the current costume.



I need a hint

When this sprite is clicked, create the flags list. If the item in the Flags list with this costume # equals the correct answer, say 'Correct', or else say 'Sorry, that was wrong'.

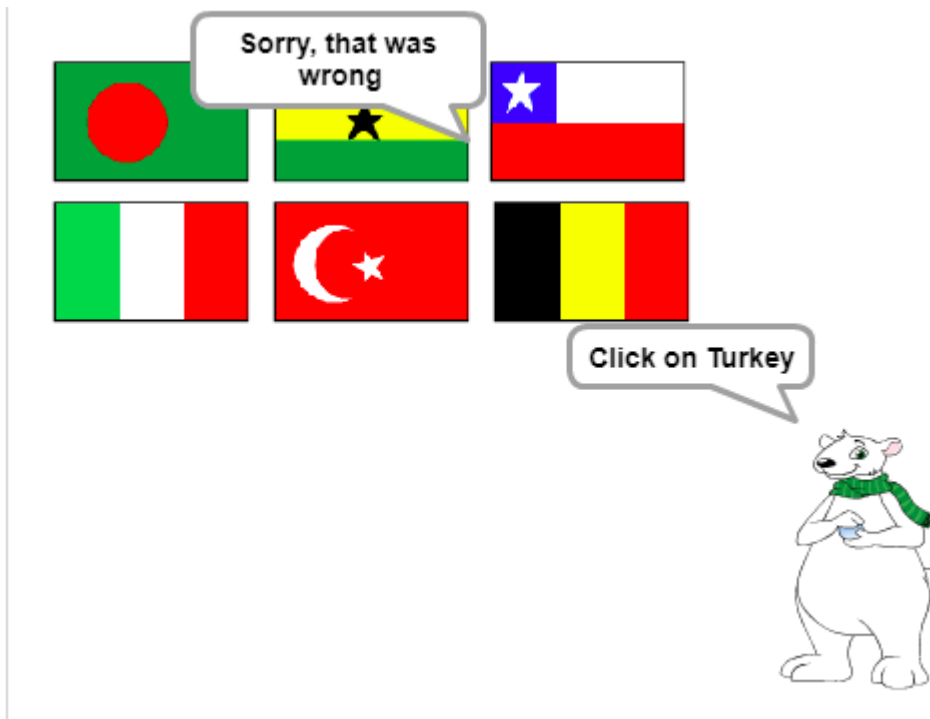
Here are the code blocks you'll need:



This is what your code should look like:



- Press the green flag and test your code by getting the answer deliberately right and wrong. Check that the right message appears depending on whether you were right or wrong.



Challenge

- Play different sounds if the player was right and if the player was wrong.
 - Use broadcasts to make the bear sprite (not the flag sprite) report whether the player was right or wrong.
-

Add a score

- Create a new variable called **Score**.
 - Add 1 to the score every time the player gets a correct answer.
 - Don't forget to reset the score to 0 when the green flag is pressed.
-

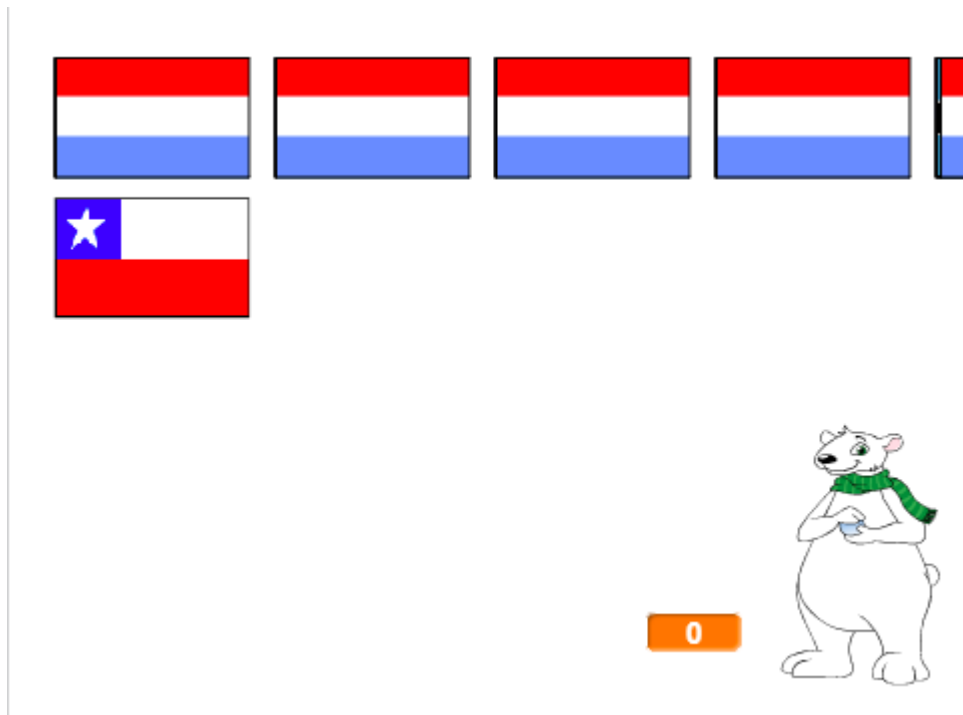
Start a new round

At the moment we only have one round to play, so the game doesn't last long. Let's set up multiple rounds.

- Create a new broadcast called 'Start the round'.
- Move all of the code which previously ran after the green flag was clicked, so that it runs when I receive 'Start the round'.
- Remove the Set score to 0 block and place it back with the when green flag clicked block, followed by the new broadcast block you just created.

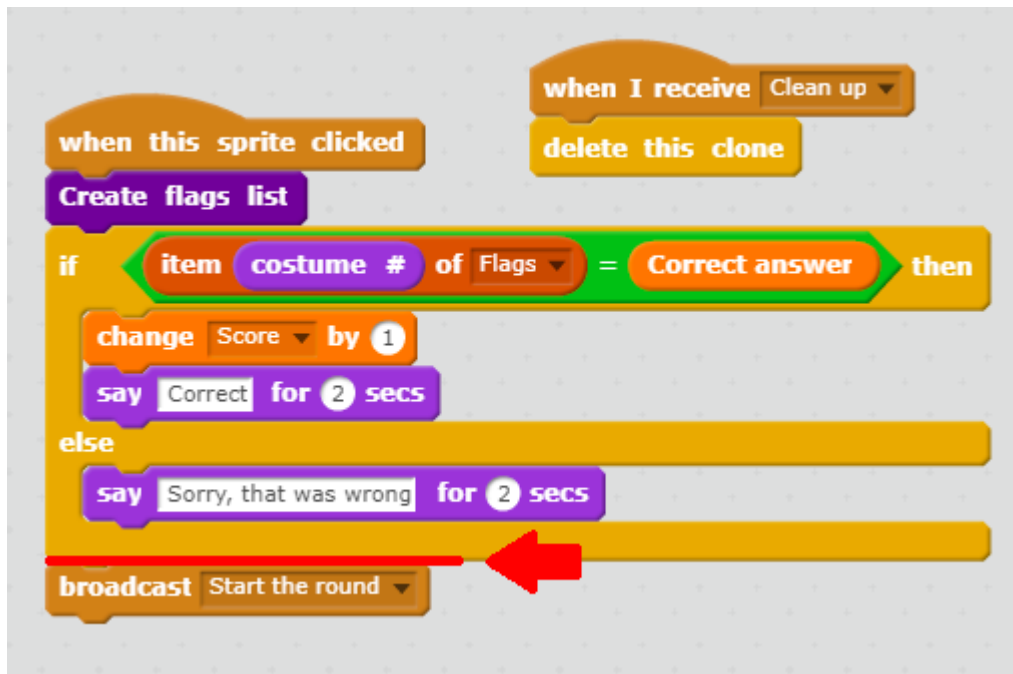


- After the code where you check the answer, add another copy of the broadcast block to start a new round after an answer has been given.
- Press the green flag to test your code. Click on a flag to play a round. You will notice that the next round does not get set up properly.



This is because we need to clear up the cloned flags we created before beginning another round.

- Create another new broadcast called 'Clean up'.
- Set the flag sprite to **Delete this clone** when it receives the 'Clean up' broadcast.
- Place the block that broadcasts 'Clean up' just before you begin a new round after an answer has been given.



- Test your code again and check that you can play multiple rounds successfully, and that your score continues to increase as you get answers correct.

Challenge: improve the game

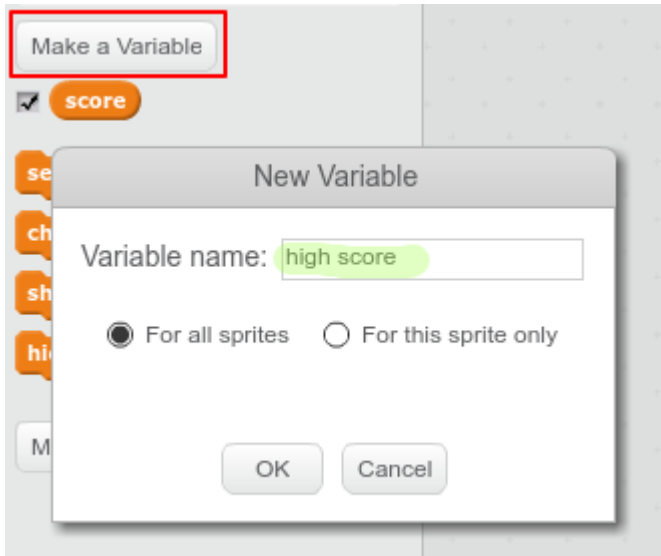
- Add a fixed number of rounds and then tell the player their percentage of correct answers.
- Store the player's high score.

Create a high score in Scratch

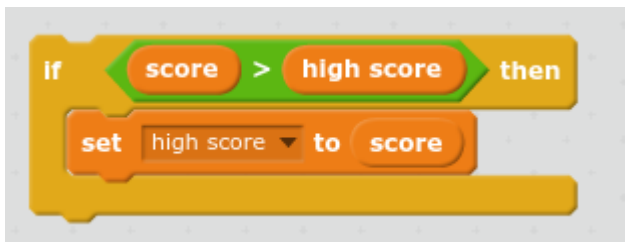
It's fun to keep track of a high score in a game.

Let's say you have a variable called `score`, which gets set to zero at the beginning of each game.

Add another variable called `high score`.



At the end of the game (or whenever you want to update the high score), you'll need to check whether you have a new `high score`.



- Add a timer for each round to force the player to give their answer quickly.
- Add more flags to make the game more challenging.
- If the player clicks on an incorrect flag, remove that flag and allow them another chance to click on a different flag instead of starting a next round.
- Make the game into a two-player game where players take turns to guess, and their scores are recorded separately.

Published by the Raspberry Pi Foundation – www.raspberrypi.org

Licensed under Creative Commons "*Attribution-ShareAlike 4.0 International* (CC BY-SA 4.0)"
Full project source code available at <https://github.com/RaspberryPiLearning/guess-the-flag>

