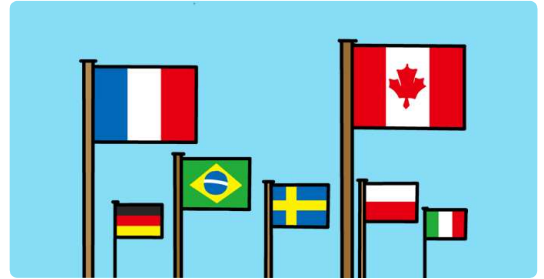




Guess the flag

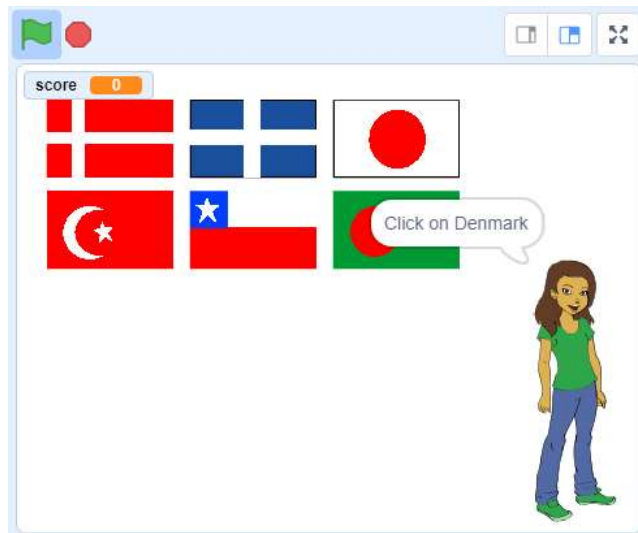
Create a flag quiz game to yourself and your friends!



Step 1 Introduction

In this resource, you will create a flag quiz to test yourself and your friends. In the quiz, 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



What you will learn

- How to broadcast a message and have other sprites respond
- How to select random items from a list



What you will need

Hardware

- A computer capable of running Scratch 3

Software

- Scratch 3 (either **online** (<https://rpf.io/scratchon>) or **offline** (<https://rpf.io/scratchoff>))

Downloads

- **Offline starter project** (<https://rpf.io/p/en/guess-the-flag-go>)



Additional notes for educators

You can **download the resources for this project** here (<https://rpf.io/p/en/guess-the-flag-go>).

You can find the **completed project** here (<https://rpf.io/p/en/guess-the-flag-get>).

Step 2 Draw the flags

Open the 'Guess the flag' Scratch starter project.



Online: open the starter project at rpf.io/guess-the-flag-on (<https://rpf.io/guess-the-flag-on>). If you have a Scratch account, you can click on **Remix** in the top right-hand corner of the online editor to save a copy of the project.

Offline: open the **starter project** (<https://rpf.io/p/en/guess-the-flag-go>) in the offline editor.

If you need to download and install the Scratch offline editor, you can find it at rpf.io/scratchoff (<https://rpf.io/scratchoff>).

Select the Costumes tab. You should see eight flags there.

Scroll to the bottom of the list of costumes, where there are two blank costumes. These costumes are there so you can add your own flags.

Click on the 'Your flag 1' costume, and change its name to the name of a country.



Draw that country's flag. Make sure your drawing is exactly the same size as the flag costume.



If you are stuck for ideas, you can find some flags on **this 'Flags of the world' web page** (<https://www.countries-ofthe-world.com/flags-of-the-world.html>).

Repeat this process for the second blank flag costume so that there are ten flag costumes in total.



Here are the flags that act as examples in the 'Guess the flag' project, but you can choose any flags you like for your game.



Step 3 Create a list of flags

Click on the Code tab. There is a list called **flags** where you store the names of the countries that your game has flag costumes for.



Add two more code blocks, one each for the other two flags you created, so there is a total of ten blocks that add all ten countries to the **flags** list.

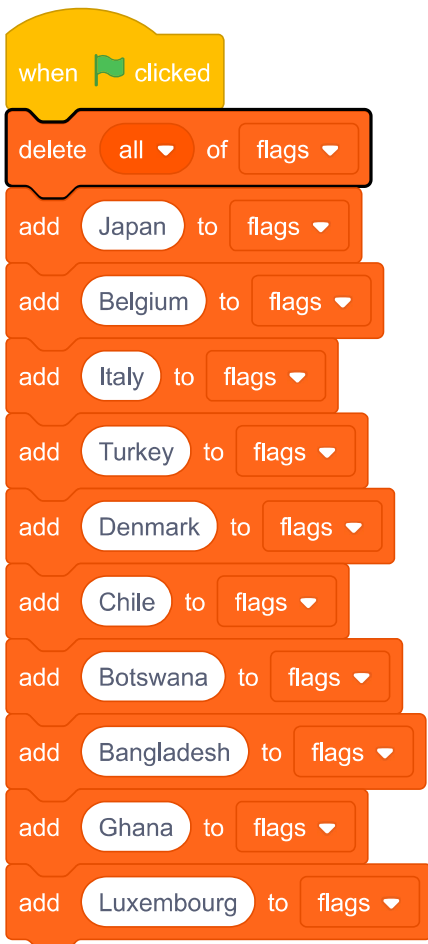


Click the green flag and check that the countries appear in the list.



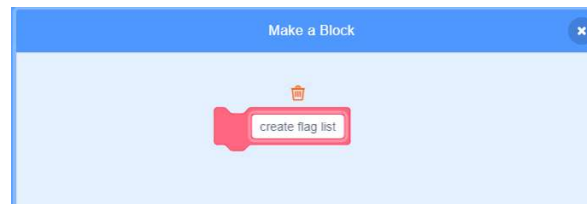
If you press the green flag more than once, the countries get added to the list again, and the result is a list of 20 countries instead of 10.

At the start of the code, add a block to **delete all** of the countries in the list before adding them. This will stop the countries from being added to the list more than once.

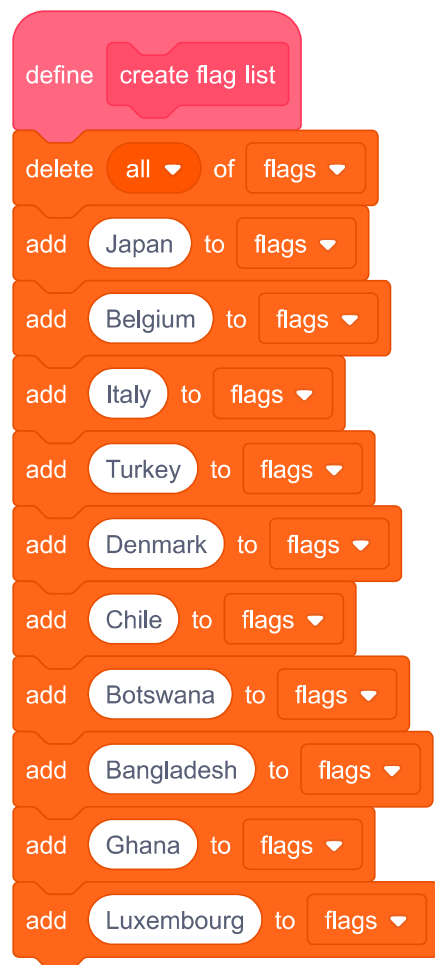


Next, make a custom block. A custom block is a special block with a name. The custom block you'll make will let you create a list of flags using only this one block instead of lots of blocks.

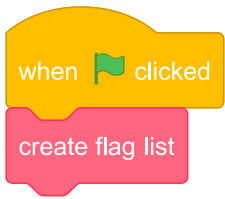
Click on **My Blocks** and then on **Make a Block**. Call your custom block `create flag list`.



Drag all the code away from below the `when flag clicked` block to below the new `create flag list` block.



Below the `when flag clicked` block, add the new `create flag list` block.



Step 4 Choose random flags

For each round of the quiz, choose six random flags from the `flags` list to be the options.

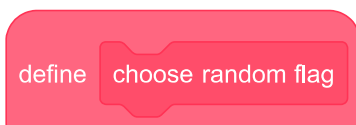
Create another list called `chosen_flags`. This list will store the six random flags.



Create a variable called `flag_number`.



Create a custom block and call it `choose random flag`.



Add code to the custom block to set the `flag_number` variable to a random number between 1 and the number of items in the `flags` list.



There is a special block in the Variables tab for finding the number of items in a list.

I need a hint

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 from the `flags` list. Then add the item text into the `chosen flags` list. Add this code to your custom block:



```
define choose random flag
  set flag number to pick random 1 to length of flags
  add item flag number of flags to chosen flags
```

Add the custom `choose random flag` block to the code that runs after the green flag is clicked.



```
when green flag clicked
  create flag list
  choose random flag
```

Test that your code works by clicking the green flag several times and checking that different countries are added to the `chosen flags` list every time. (If you have hidden the list, tick the box next to the list name to make the list visible.)



Can you see that, if you click the green flag lots of times, your `chosen flags` list quickly fills up with more than six items?

Add blocks to delete all the items from the `chosen flags` list before choosing six flags for the quiz.



```
when green flag clicked
  create flag list
  delete all of chosen flags
  repeat 6
    choose random flag
```


Test your code again by clicking the green flag several times and checking that the `chosen flags` list is filled with six countries each time.



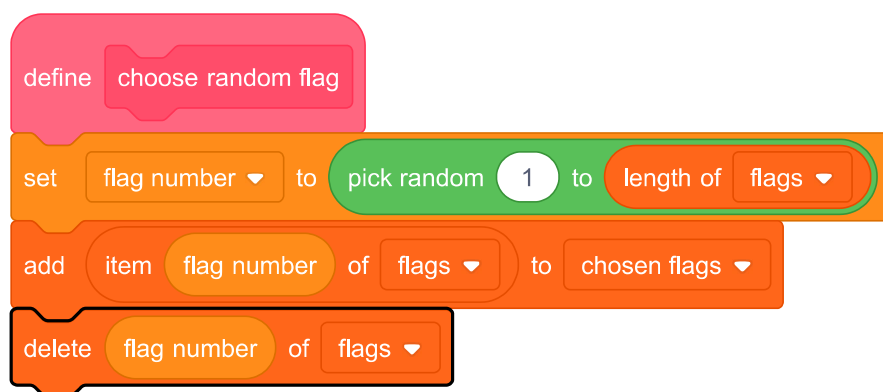
You might notice that sometimes the same country gets added to the list more than once.



Change your `choose random flag` block so that the same country never gets added twice to the `chosen flags` list.



Add a block to the end of your custom block code to delete the `flag number` from the `flags` list after it has been added to the `chosen flags` list.



If you want to hide the lists and variables so that they don't take up space on the Stage, go to the Data section and unselect the boxes next to the list names or variable names. If you want to show the lists and variables again, just select the boxes.

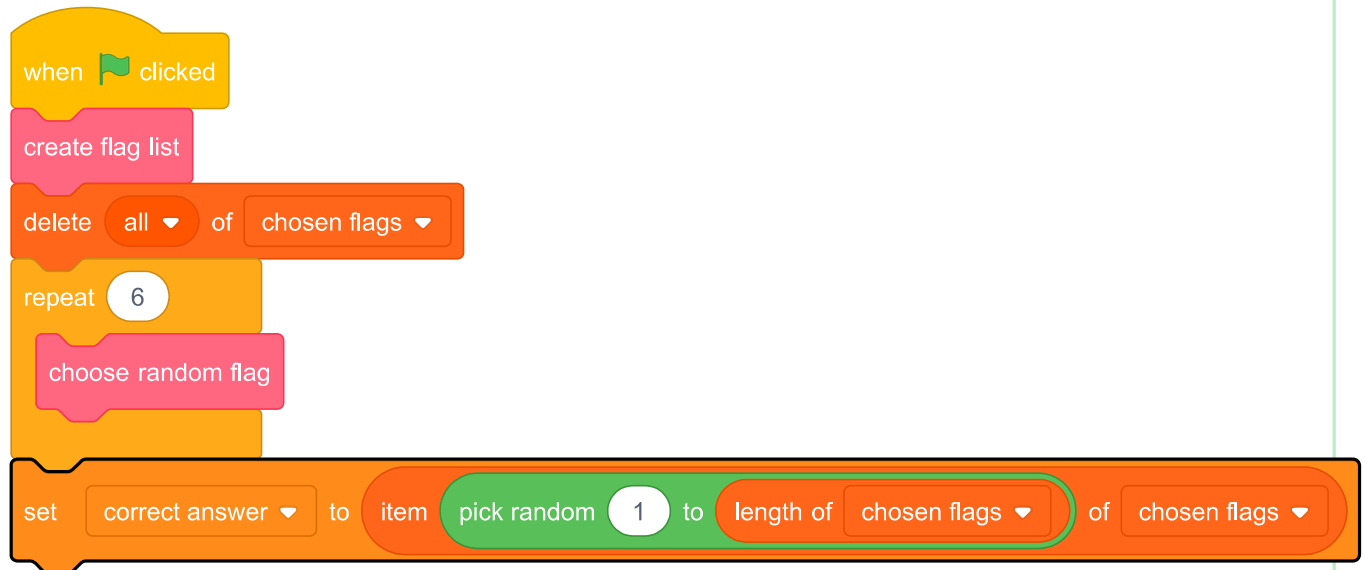
Step 5 Choose a correct answer

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

Create a new variable called `correct answer`.



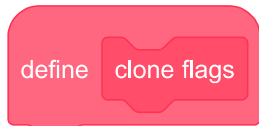
After the six flags are chosen, set the `correct answer` variable to be a random item from the `chosen flags` list.



Step 6 Show the flags

The person taking the quiz needs to see the pictures of the flags in the **chosen flags** list.

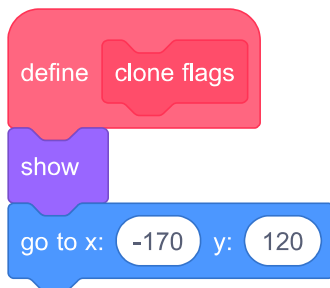
Create another custom block, and call this one **clone flags**.



This custom block will clone the Flag sprite six times, once for each flag that should be displayed.

The first flag should be displayed in the top left-hand corner of the Stage.

As part of the instructions for your **clone flags** block, make the Flag sprite visible, and add a **go to** block to tell the sprite to show at the coordinates **-170, 120** in the top left-hand corner of the Stage.



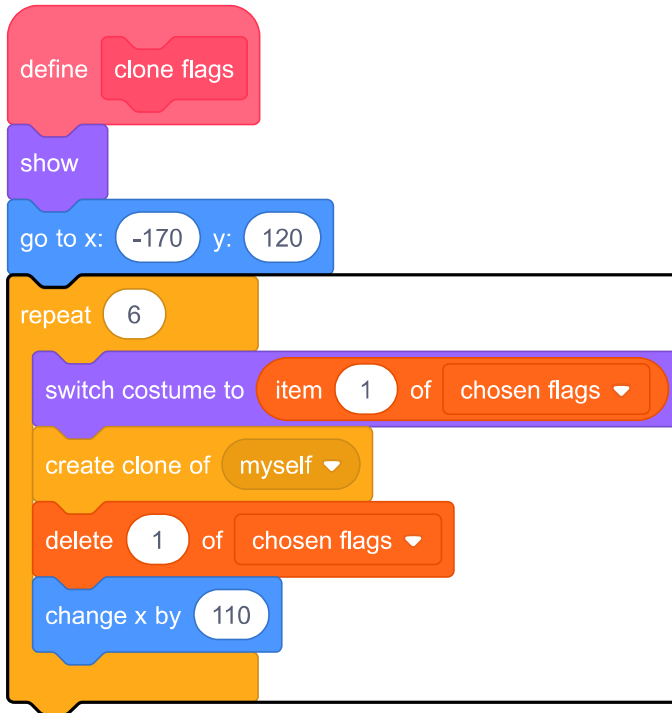
Below that code, add a loop that repeats six times.



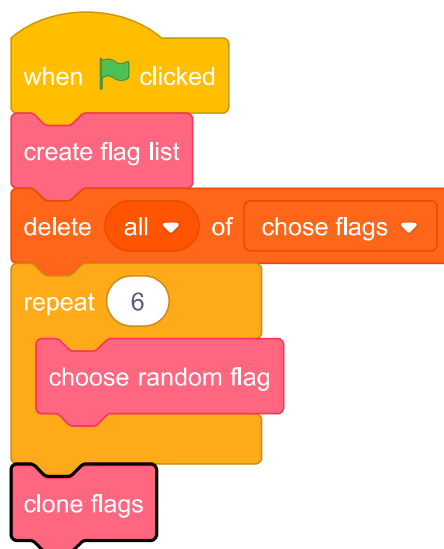
Inside the loop, add code blocks to switch the sprite's costume to the first flag in the `chosen flags` list, and to clone the sprite. Then, add code blocks to delete the first flag from the list, and to add `110` to the `x` coordinate to move the sprite to the position of the second flag.

I need a hint

This is what your code should look like:



Add your `clone flags` block to the end of the code that runs when the green flag is clicked.



Run your code. Notice that the different flags appear, but some are cut off by the edge of the Stage.

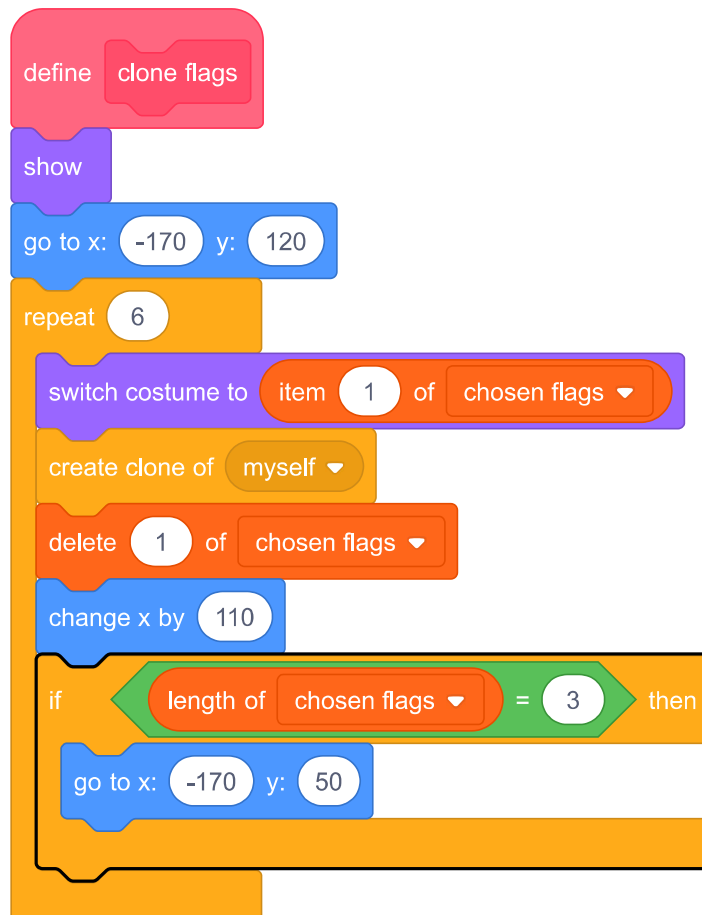


Instead of putting all six flags in one row, make two rows of three flags.

Add some code inside the **repeat** loop of the **clone flags** block to move the Flag sprite down a row if there are three flags left in the **chosen flags** list.



You can the sprite 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.



Click the green flag and check that the flags display in two rows.



It looks like the last flag is displayed twice. This is because the original Flag sprite is still visible at the end.

Add a **hide** block at the end of the code inside the **clone flags** block to hide the original sprite.



If you want to, you can try making the flag sprites appear one by one or playing a sound (a pop, for example) each time a flag appears.

Step 7 Ask the question

Let's ask the player to name the flag for a particular country.

In the flag sprite, **broadcast the message** 'announce country' immediately after the block that clones the flags.



```
when green flag clicked
  create flag list
  delete all of chosen flags
  repeat (6)
    choose random flag
  set correct answer to item pick random 1 to length of chosen flags of chosen flags
  clone flags
  broadcast announce country
```



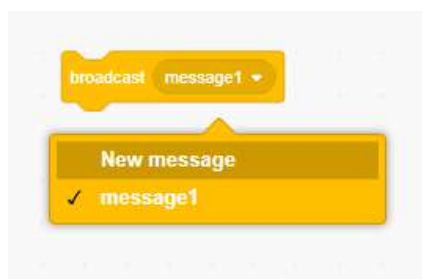
Broadcast a message in Scratch

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

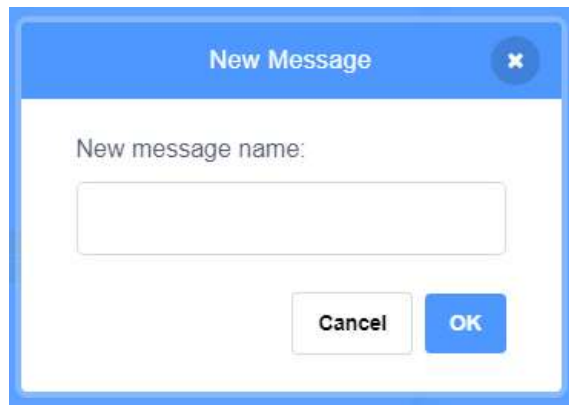
Broadcasting spells: Use the magic wand to click on the buttons and cast spells. What does each spell do to the characters? **See inside** (<https://scratch.mit.edu/projects/518413238/editor>)

You can create a message to be **broadcast**. The message text can be anything you like, but it is useful to give it a sensible description.

- Find the **broadcast** block under **Events**
- Select **New Message** in the drop-down menu.

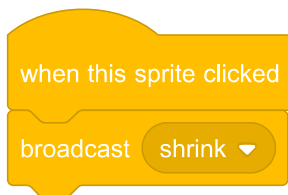


- Then type your message



Send a broadcast

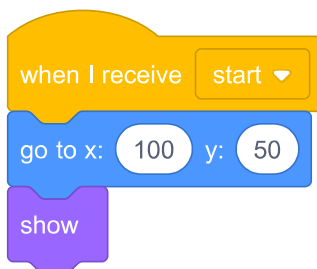
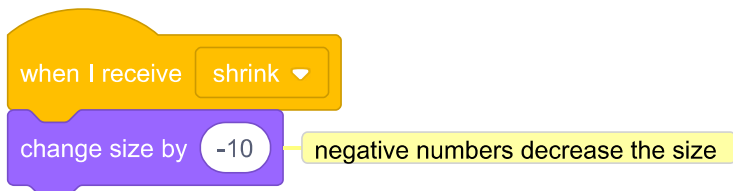
You can decide when to **broadcast** your message. For example:



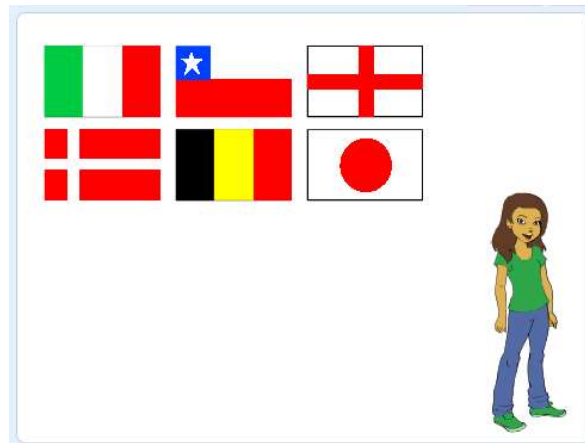
Receive a broadcast

Sprite can react to a **broadcast** by using a **when I receive** block. Multiple sprites can respond when they receive the same message.

You can add blocks below a **when I receive** block to tell the sprite(s) what to do when they receives the message.



Add a new sprite of your choice to be your quiz master. The quiz master in the example is the sprite called Abby.

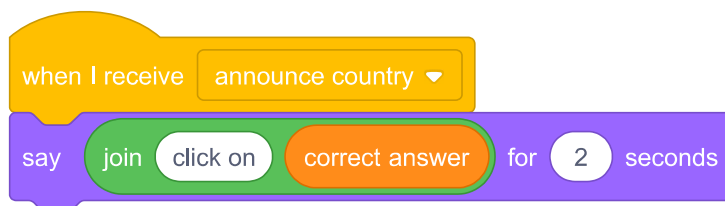


Add some code to the quiz master sprite so that, when the sprite receives the `announce country` broadcast, it tells the player to click on the country name that is stored in the variable `correct answer`.



I need a hint

This is what your code should look like:



Step 8 Check the answer

Your sprite now asks the player to click on the correct flag. Then the game needs to check whether the flag that was clicked is the right answer.

Go back to the Flag sprite code, and add a block to start a new section of code that will run **when this sprite is clicked**.



Then your quiz needs to check whether the costume name of the Flag sprite that was clicked is the same as the correct answer.

Add code to say 'Correct' if the costume name of the Flag sprite is the same as the **correct answer** variable, or to say 'Sorry, that was wrong' if the name and the variable are not the same.



You can use this useful block here as well.

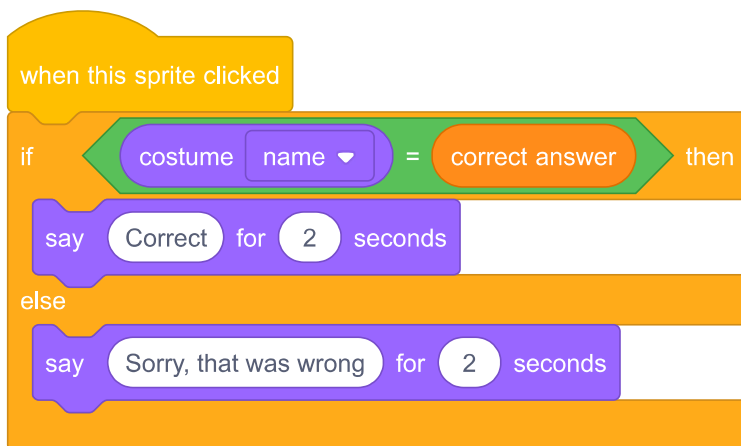


This time, combine it with a **costume name** block to get the name of the current Flag sprite costume.

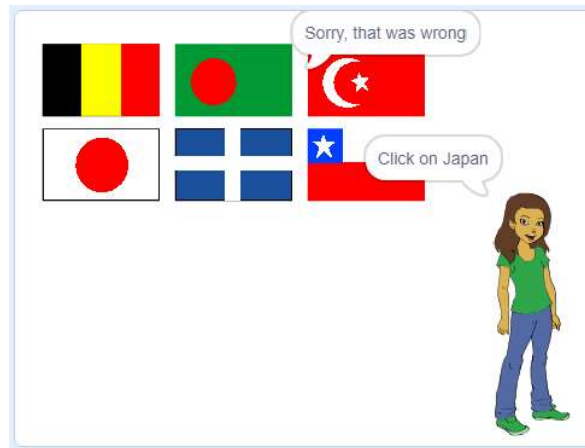


I need a hint

This is what your code should look like:

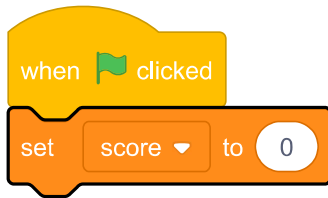


Press the green flag and test your code twice: once by picking the correct flag, and once by picking an incorrect one. Check that the right message appears depending on whether you give the right or wrong answer.

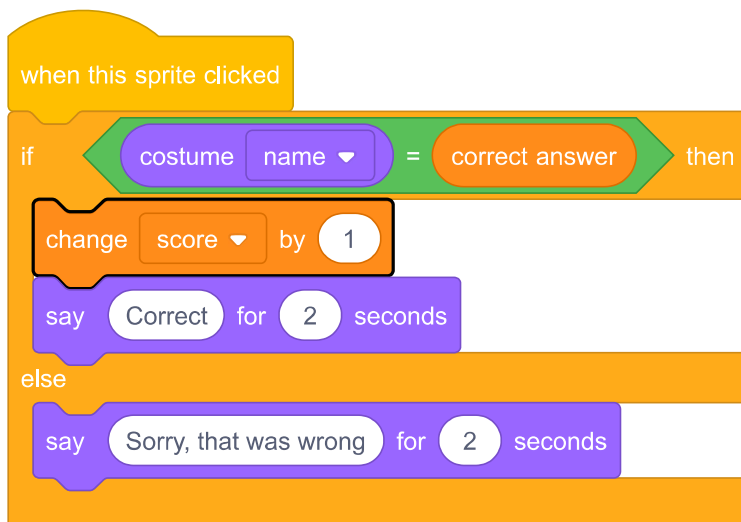


Step 9 Add a score

Create a new variable called **score** and set the score to **0** when the green flag is clicked.



Add **1** to the score every time the player gives a correct answer.



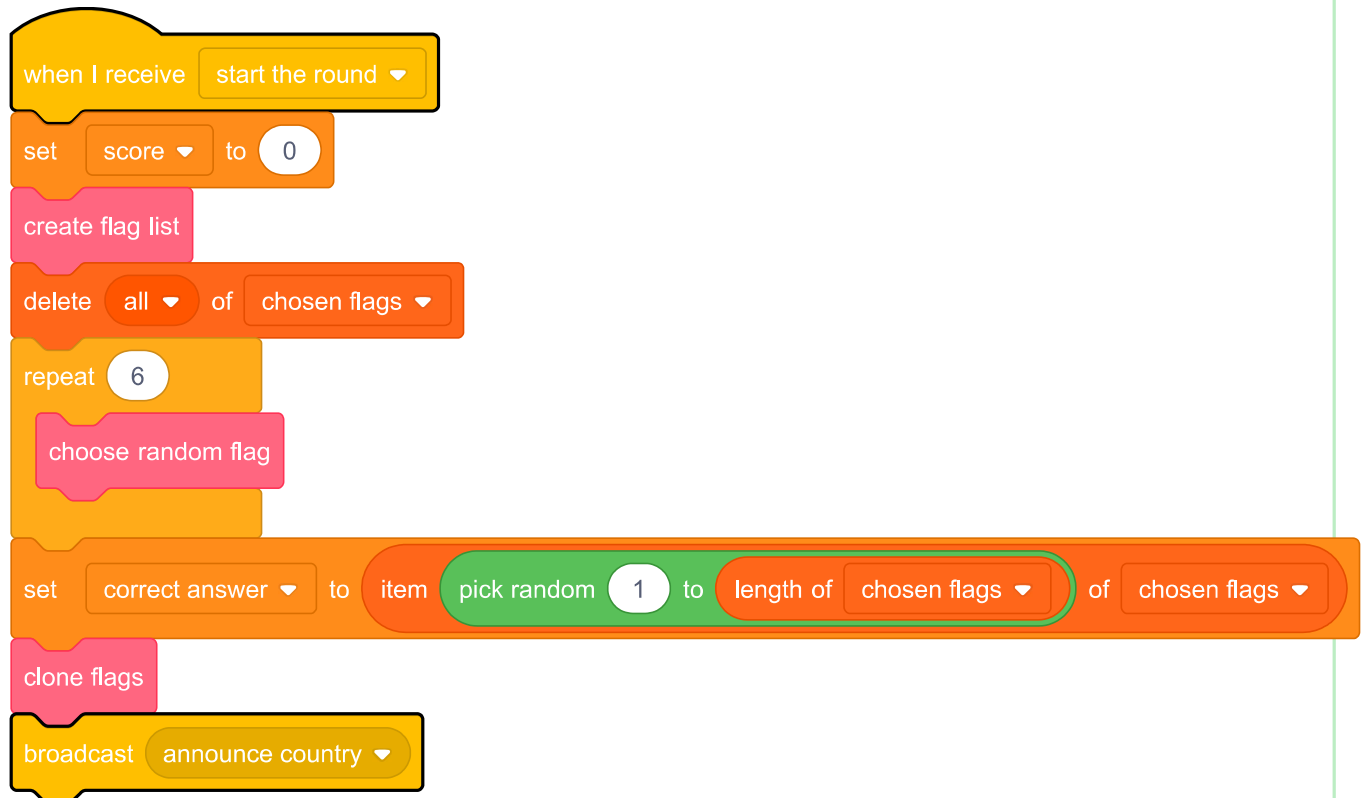
Step 10 Start a new round

At the moment there is only one round in the quiz, so the quiz doesn't last long. You are going to set up multiple rounds.

Create a new **broadcast** that sends the message 'Start the round'.



Add a **when I receive 'Start the round'** block, and then move all of the code from below the **when green flag clicked** block to below this new block.



Remove the `set score to 0` block and place it back below the `when green flag clicked` block. Then add the new `broadcast` block below both of them.



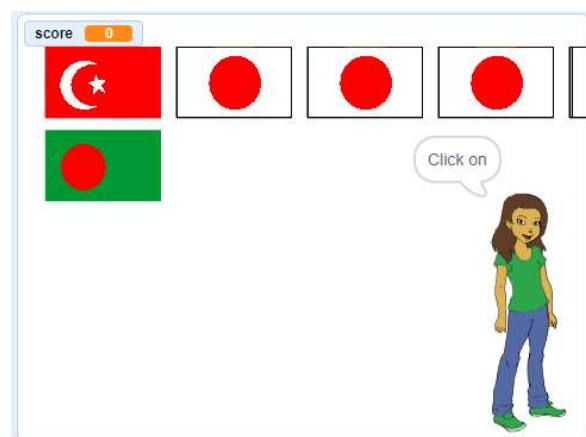
```
when green flag clicked
  set score to 0
  broadcast start the round
```

After the code that checks whether the answer is correct, add another `broadcast` block so that a new round can start once a question is answered.



```
when this sprite clicked
  if costume name = correct answer then
    change score by 1
    say Correct for 2 seconds
  else
    say Sorry, that was wrong for 2 seconds
  broadcast start the round
```

Click the green flag to test your code. Click on one of the flags to play a round. Do you notice that the next round does not get set up properly?

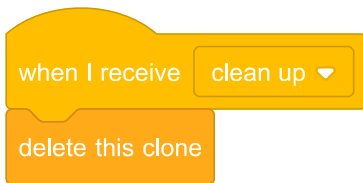


This is because before the game starts another round, the game needs to first clear up the cloned flags.

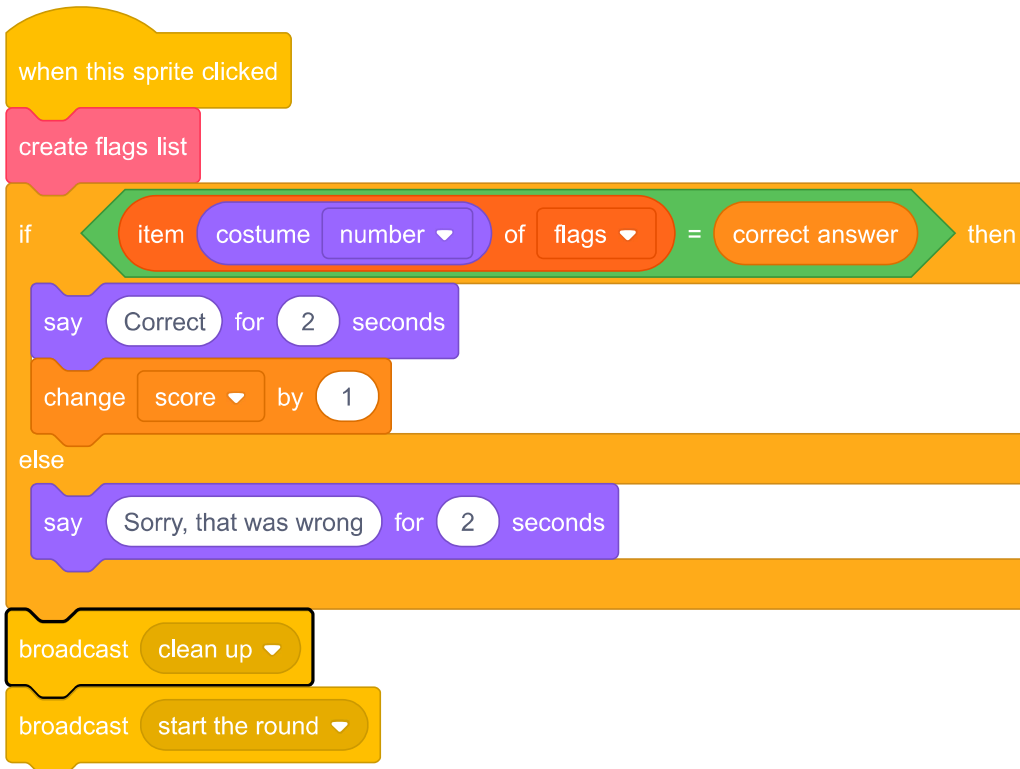
Create another new **broadcast** called 'clean up'.



Set the Flag sprite to **delete this clone** when it receives the **clean up** broadcast.



Place the **clean up** broadcast block just above where the game starts a new round after an answer has been given.



Test your code again and check that you can play multiple rounds, and that your score increases as you get answers correct.



Make sure you hide the **correct answer** variable so the player can't see it!



Challenge: improve the game

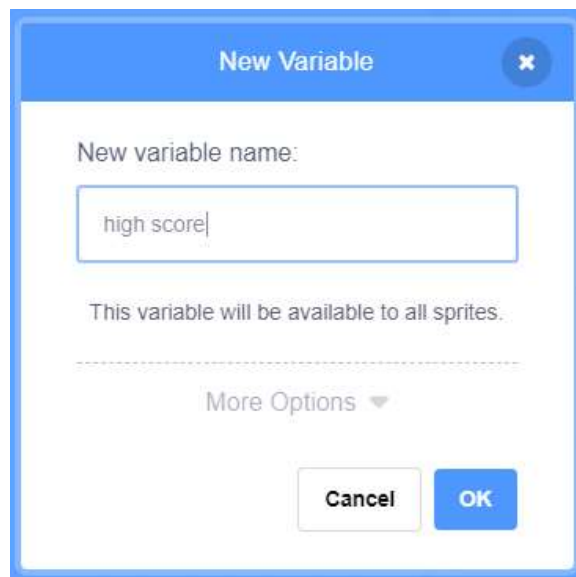
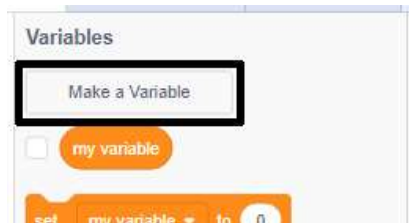
- Add a fixed number of rounds and then tell the player the percentage of rounds in which they gave the correct answer.
- Store the high score.

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, have the game remove that flag and give the player another chance to answer correctly before the next round begins.
- Make the game into a two-player quiz where players take turns to guess, and their scores are recorded separately.

Step 11 What next?

Try the **Line-up** (https://projects.raspberrypi.org/en/projects/lineup?utm_source=pathway&utm_medium=whatnext&utm_campaign=projects) project, where you will make a game of hide-and-seek and learn how to stamp sprites and use coordinates to position them in rows.

You have to find the right sprite amongst all these characters before your time runs out.



Published by Raspberry Pi Foundation (<https://www.raspberrypi.org>) under a Creative Commons license (<https://creativecommons.org/licenses/by-sa/4.0/>).

View project & license on GitHub (<https://github.com/RaspberryPiLearning/guess-the-flag>).