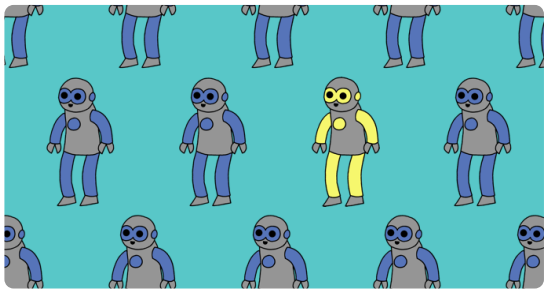
Projects

Line up

Create a game where you look for a character hidden in a crowd of other characters

Scratch



Step 1 Introduction

In this project you will make a Scratch game in which you need to find a sprite that is hidden among a huge crowd of other characters.

What you will make

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

What you will learn

- How to create custom blocks that have inputs
- How to use lists to store grid coordinates
- How to use loops to cycle over items in a list

What you will need

Hardware

- A computer capable of running Scratch 3

Software

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

Additional notes for educators

You can **find the solution for this project here** (http://rpf.io/p/en/lineup-get).

Step 2 Add costumes

Open a new Scratch project.

Online: open a new online Scratch project at **rpf.io/scratch-new** (http://rpf.io/scratch-new).

Offline: open a new project in the offline editor.

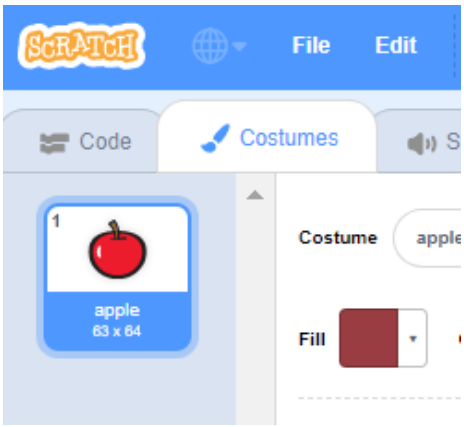
If you need to download and install the Scratch offline editor, you can find it at **rpf.io/scratchoff** (http://rpf.io/scratchoff).

Add some more costumes to the cat sprite. You need to add at least forty different costumes for your sprite.

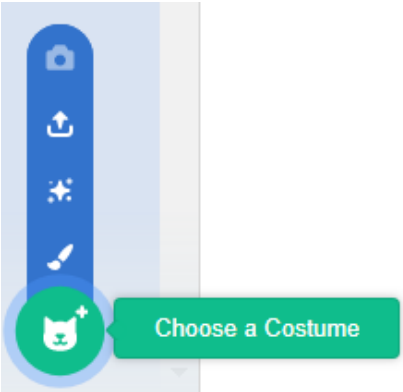
It's best to select costumes from the **People** section, but if you want to, you can choose costumes from other sections as well.

Adding new costumes in Scratch

- With your sprite selected, click on the Costumes tab

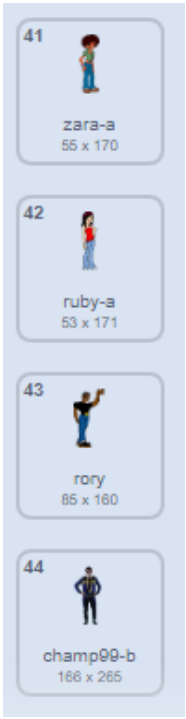


- Click **Choose a Costume** and choose one of the five options From bottom to top they are:
  - Choose costume from library
  - Paint new costume
  - Use a random (surprise) costume
  - Upload costume from file
  - New costume from camera



- If you wish to delete the imported costume, select it and click on the small cross in the top right hand corner.

Once you have your costumes, you can delete the default cat costumes if you want to.



Step 3 Create a grid

You are going to create a grid of stamped costumes:

To do this you need to know the x and y coordinates of where each stamp should be placed.

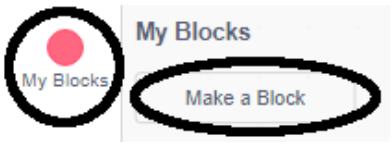
First, create a new block called `generate positions`. The block needs to have two 'number input' parameters. Call the two parameters `rows` and `columns`.

The values of these parameters will decide how many rows and columns your grid has.

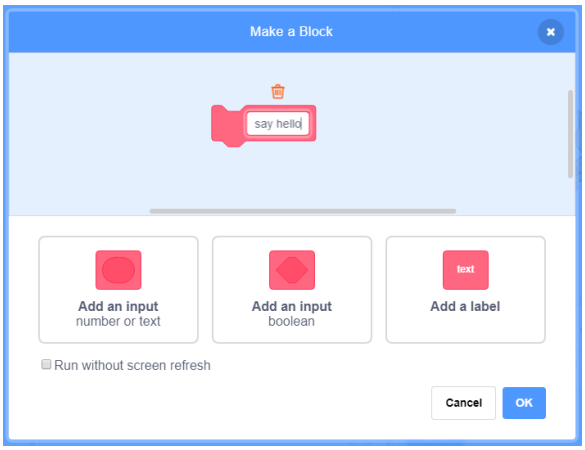
Making a block

Making a block

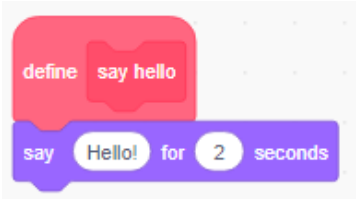
- Click on **My Blocks**, and then click **Make a Block**.



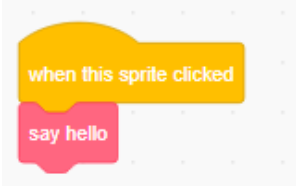
- Give your new block a name and then click **OK**.



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



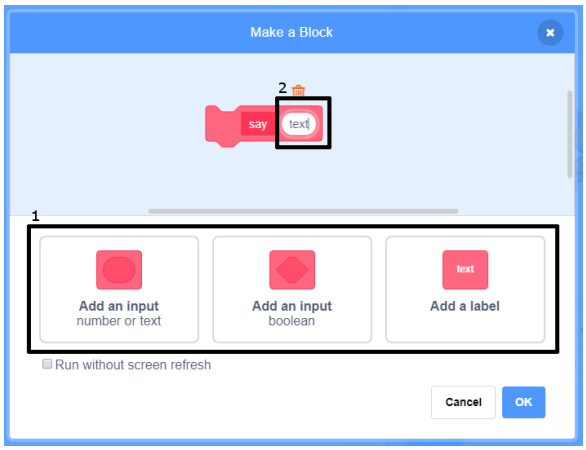
- You can then use your new block just like any normal block.



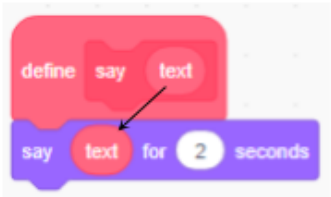
- The code attached to your new `define` block is run whenever the block is used.

Making a block with parameters

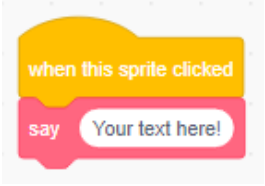
- You can also create blocks that have 'gaps' for adding data. These 'gaps' are called 'parameters'. To add parameters, first make a new block, and then click on the options below to choose the type of data you want to add. Then give your data a name, and click **OK**.



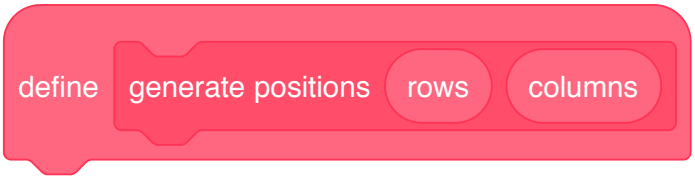
- You will see a new `define` block as usual, except that this one contains the data gap you added and which you gave a name.



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

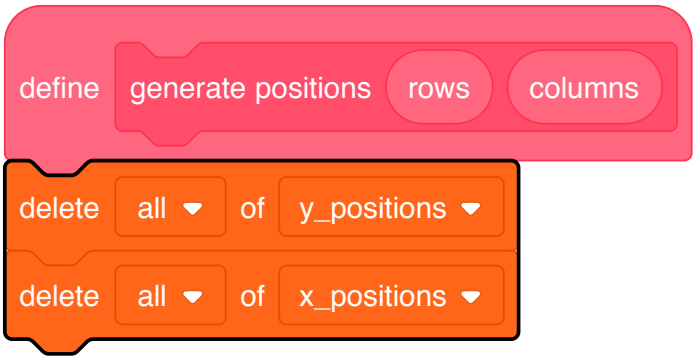


- As usual, the code attached to your new `define` block is run whenever the block is used.



Create two lists, and call one of them `x_positions` and the other `y_positions`. These lists are for storing the `x` and `y` coordinates for the stamps.

Inside your `generate positions` block, add blocks to delete all the items from both lists, so that each time the game starts, the lists are empty.

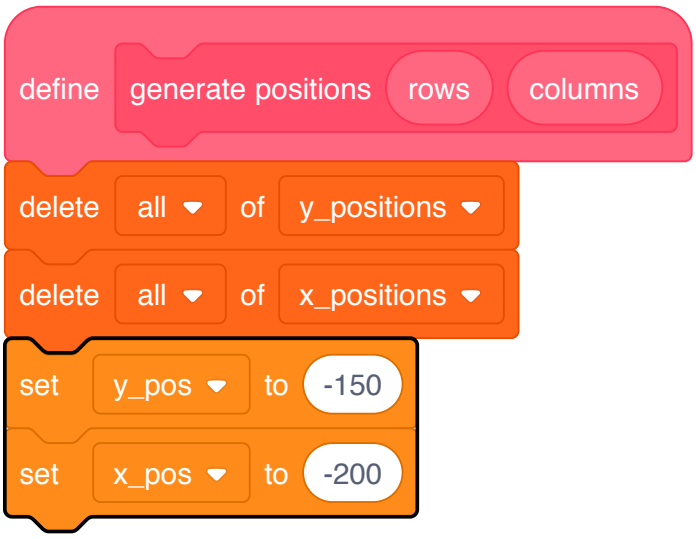


Next, create two variables, and call one of them `x_pos` and the other `y_pos`.

The `x_positions` list should contain ten numbers in total, and these should start at `-200` and go up to `200`.

For now, the `y_positions` list can just contain the number `-150` ten times, so that the grid only has one row.

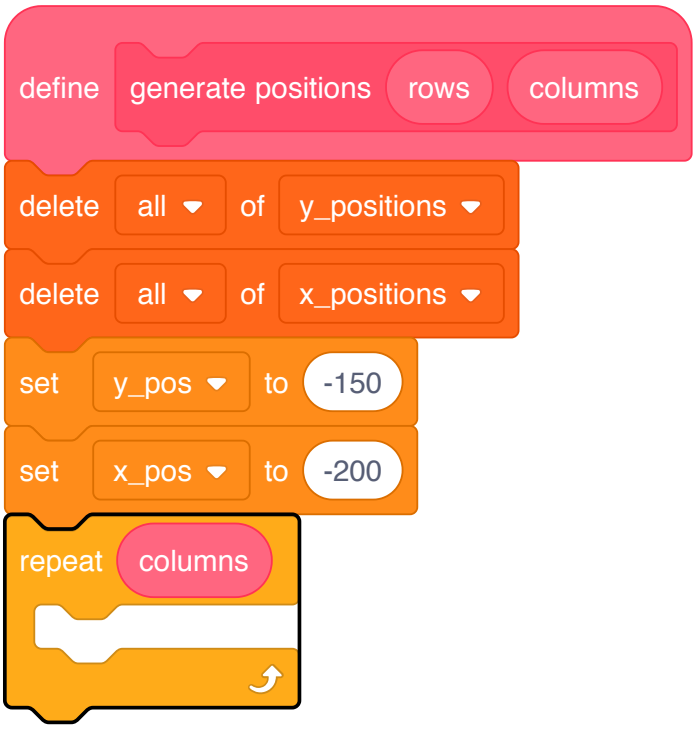
Start by adding code to the `generate positions` block to set the `y_pos` variable to `-150` and the `x_pos` variable to `-200`. This is the location of the first stamped sprite.



Next, add a `repeat` loop to put coordinates into the lists.

The `repeat` loop should run once for every column you want the grid to have.

The `generate positions` block takes `columns` as an input, so you can use `columns` for the `repeat` loop.



Within the `repeat` loop, add the values of `x_pos` and `y_pos` into the lists. Then you need to increase the value of `x_pos` by a little. How much should the value of `x_pos` increase by?

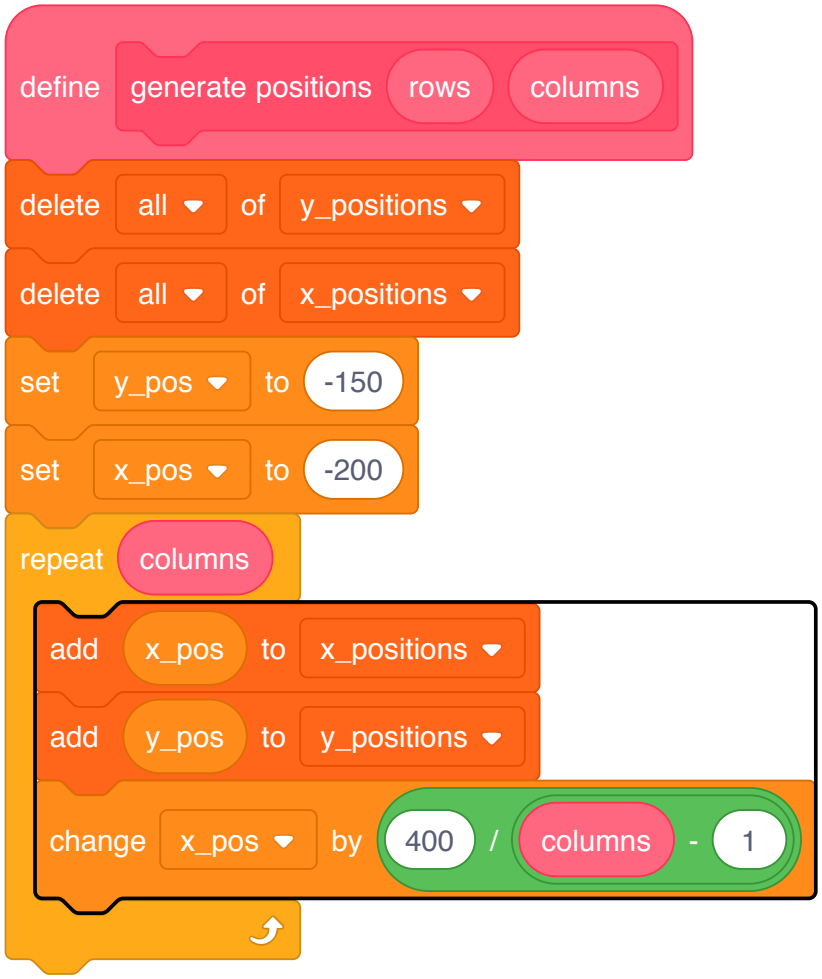
This is how to figure it out:

- `x_pos` starts out with the value `-200`
- The final time the loop `repeat` runs, `x_pos` should reach the value `200`
- That's a total increase of `400`
- The first `x_pos` value is for the first column on the grid, and how many columns there are is determined by the `columns` input

So after the first `x_pos` value is added, each time around the loop, the value of `x_pos` should increase by `400 / (columns - 1)`

Add in the code that will add all the `x_pos` and `y_pos` values into the `x_positions` and `y_positions` lists.

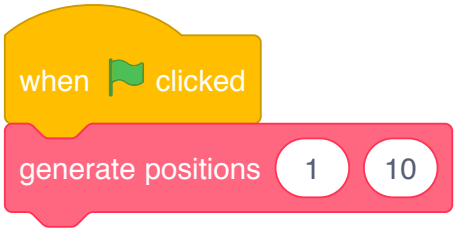
Here is the completed script for the `generate positions` block:



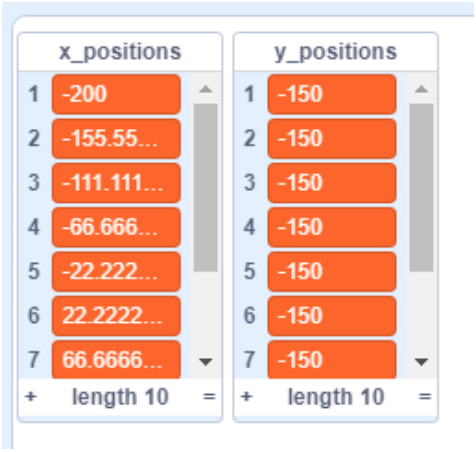
Step 4 Test the script

To test the script, you need to **call** the custom block and provide it with the number of `columns` you want in your grid.

Add this code to your sprite:



Now click on the green flag to run your code. You should see your two lists fill with values.



If your results don't look like this, then go back to the previous step, have a look at the hints, and try to fix your script.

Step 5 Stamp a row

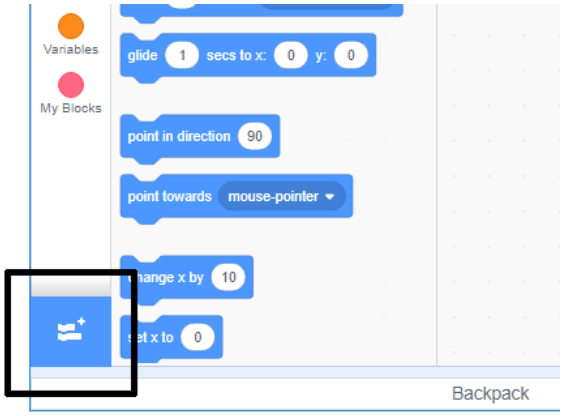
So far you have ten values in each of the two lists. Now stamp some costumes at the Stage coordinates stored in the lists.

Add the **Pen** extension to your project.

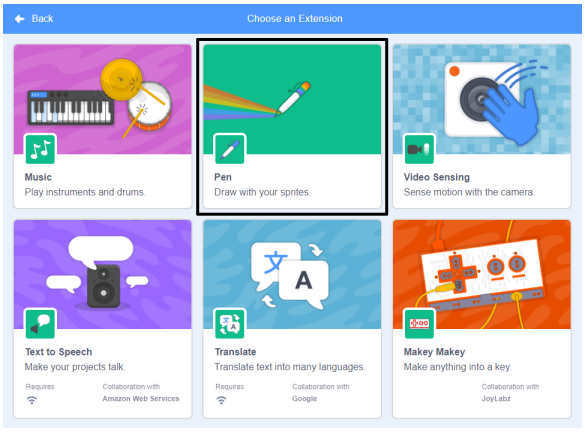
How to add the Pen extension

To use the Pen blocks in Scratch, you need add the **Pen extension**.

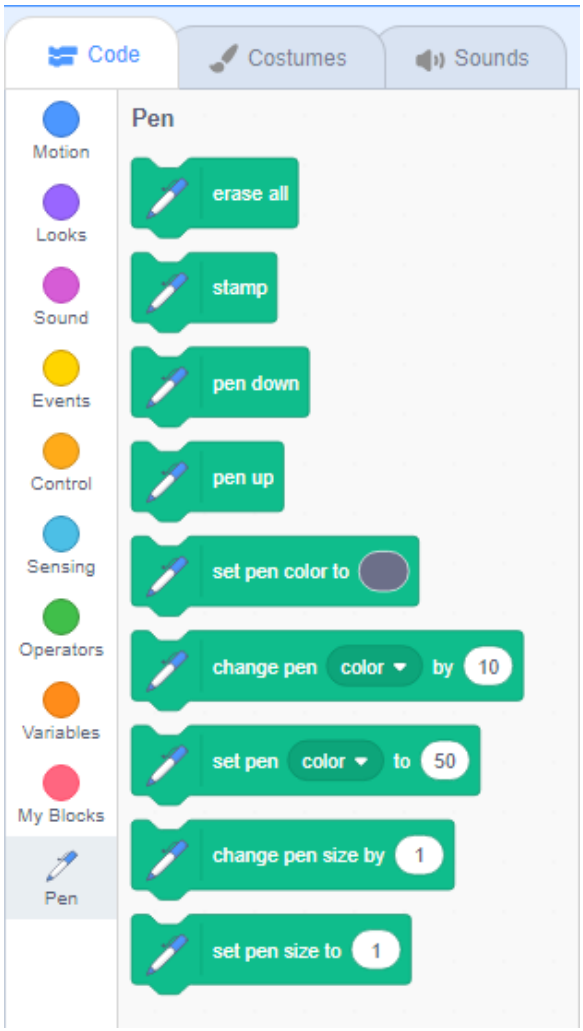
- Click on the **Add extension** button in the bottom left-hand corner.



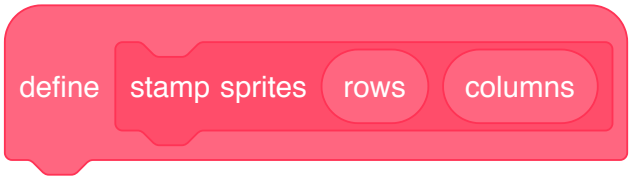
- Click on the **Pen** extension to add it.



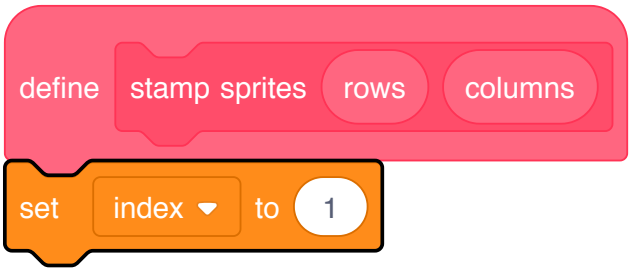
- The Pen section then appears at the bottom of the blocks menu.



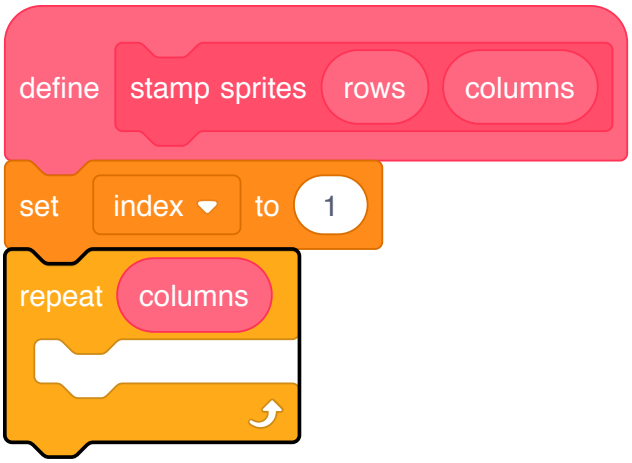
Create a new block and call it `stamp sprites`. This block needs two number inputs named `row` and `columns` just like the other custom block.



Create a new variable called `index` with which to track the position in the lists that your program is reading. To begin with, set `index` to `1` to fetch the first item of each list.



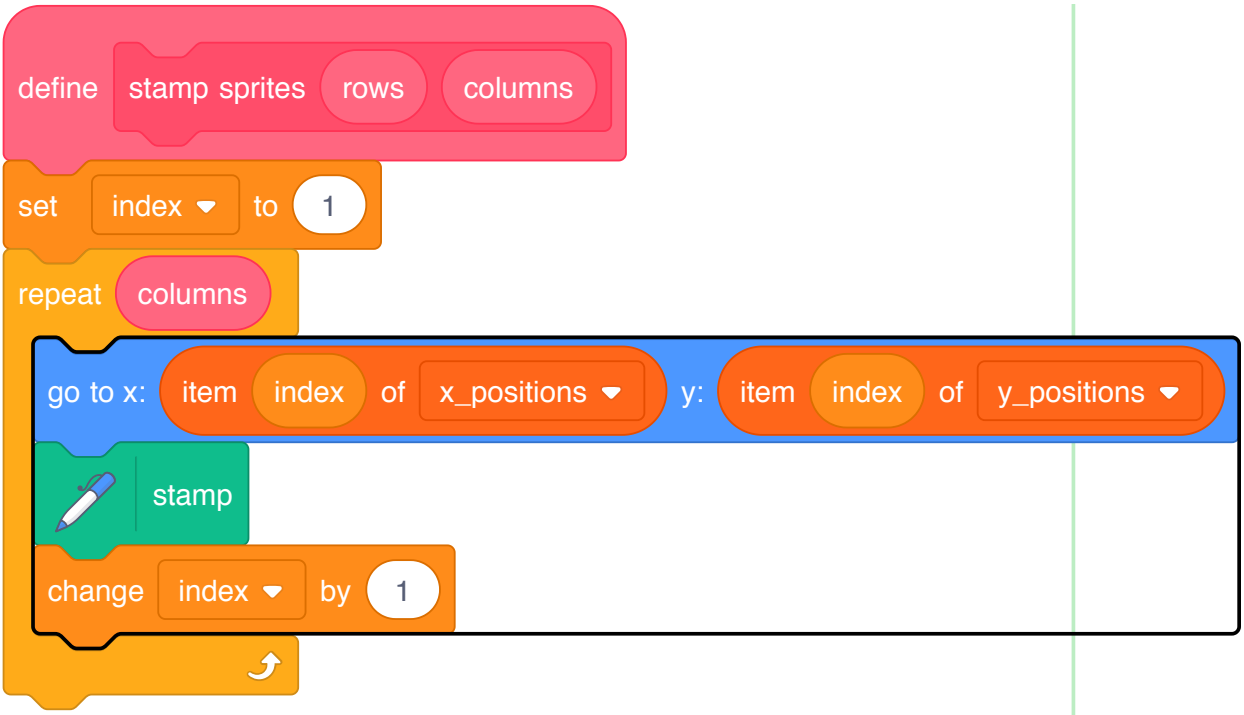
The `stamp sprites` block should stamp a sprite for each pair of coordinates in the list. To do this, the block needs a `repeat` loop that runs once for each column.



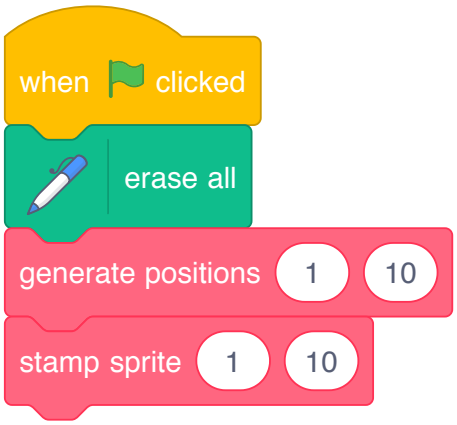
Within the `repeat` loop:

- Move the sprite to the `index` position in the `x_positions` and `y_positions` lists
- `Stamp` the sprite
- Change the `index` by `1`

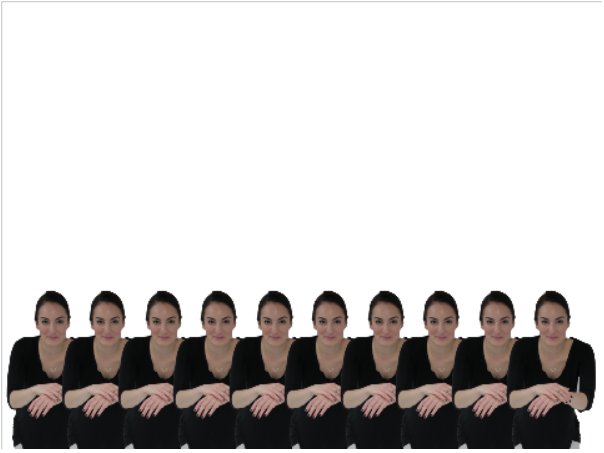
Here is the completed script for the `stamp sprites` block:



Add a `erase all` block below the `when flag clicked` block to clear the Stage each time the game starts. Then add the `stamp sprites` block at the bottom of the `when flag clicked` script so you can test your new code.



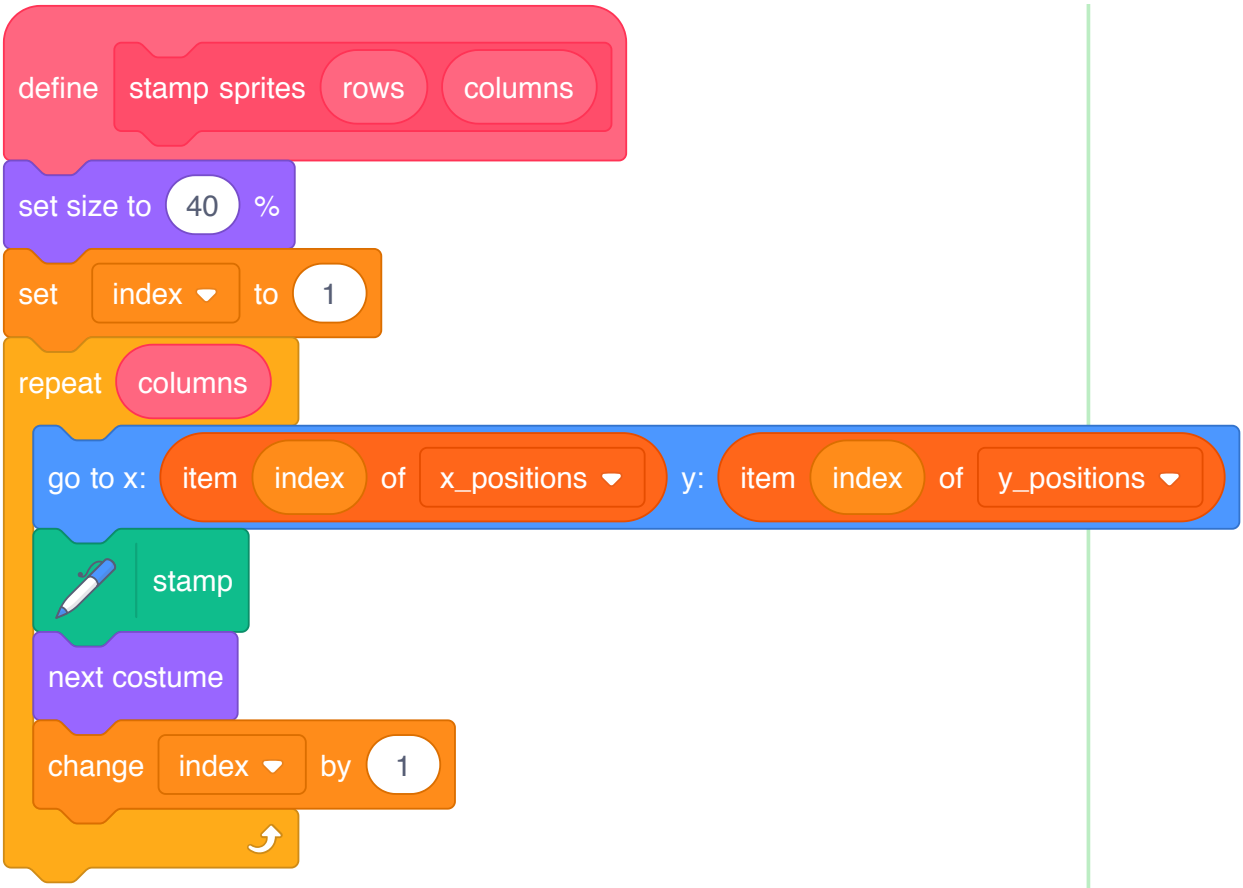
Click the green flag. You should see something like this, depending on the costumes your sprite has:



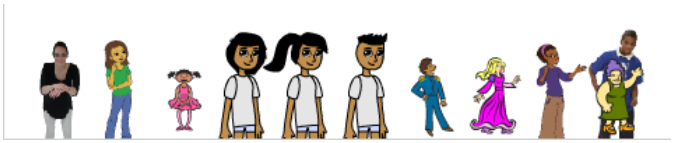
Step 6 Change the costumes

At the moment, your program stamps the same sprite costume over and over, and the size of the costume is too large.

Add code to the `stamp sprites` block to make the sprite a suitable size before the `repeat` loop starts. Add a block inside the loop to switch the `next costume` after the `stamp` block.



When you run the script now, you should see something like this:



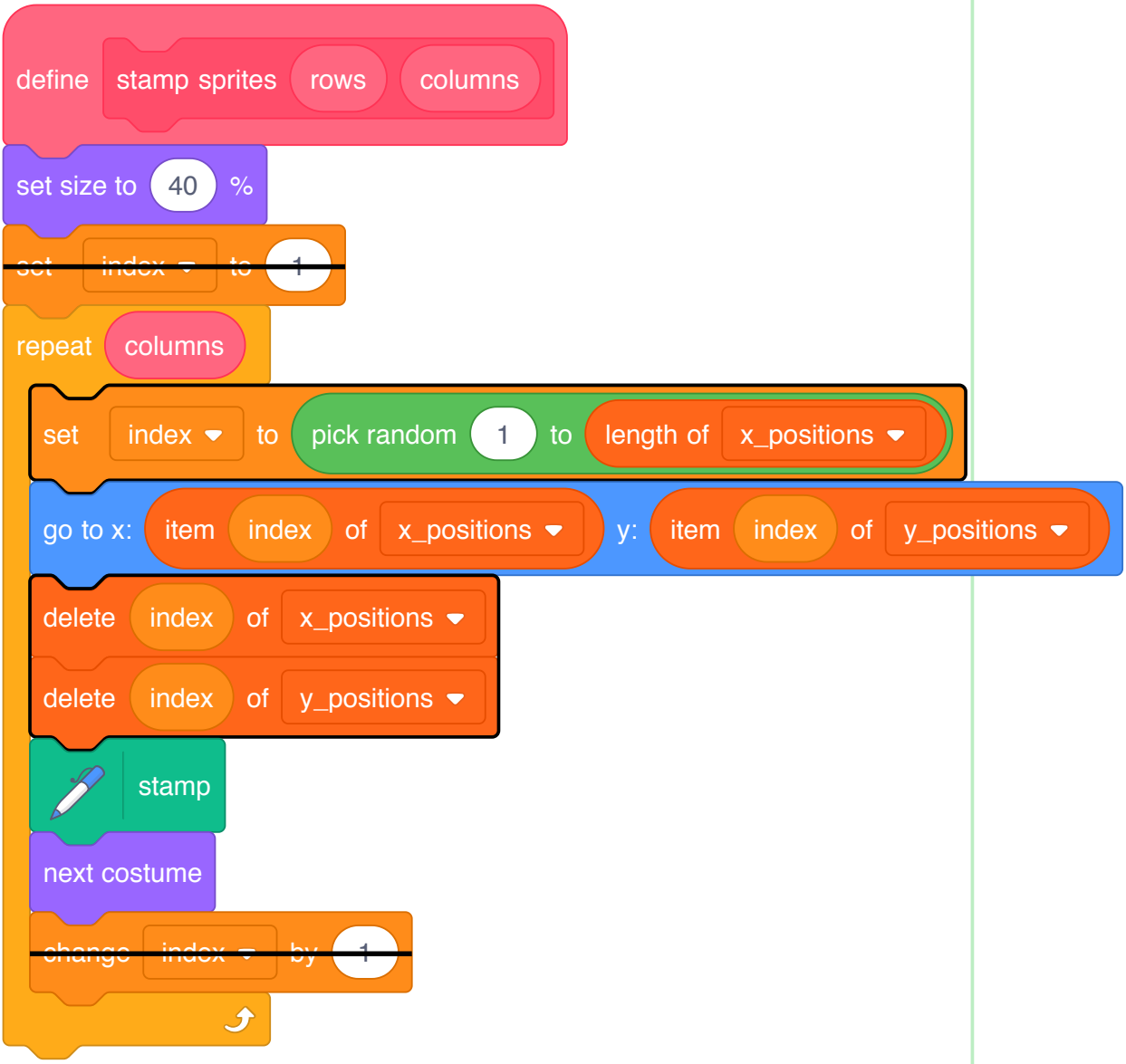
Your program cycles through all the costumes in order. So that each costume does not show up in the same place every time the program runs, you should stamp the sprite in random places on the grid.

To do this, you need to follow this **algorithm**:

1. `Repeat` until the list is empty
2. Set `index` to a `random` number between `1` and the length of a list
3. Move the sprite as you did before
4. Delete the item at the `index` position from the `y_positions` list
5. Delete the item at the `index` position from the `x_positions` list

Add code to stamp the sprite in random places on the grid.

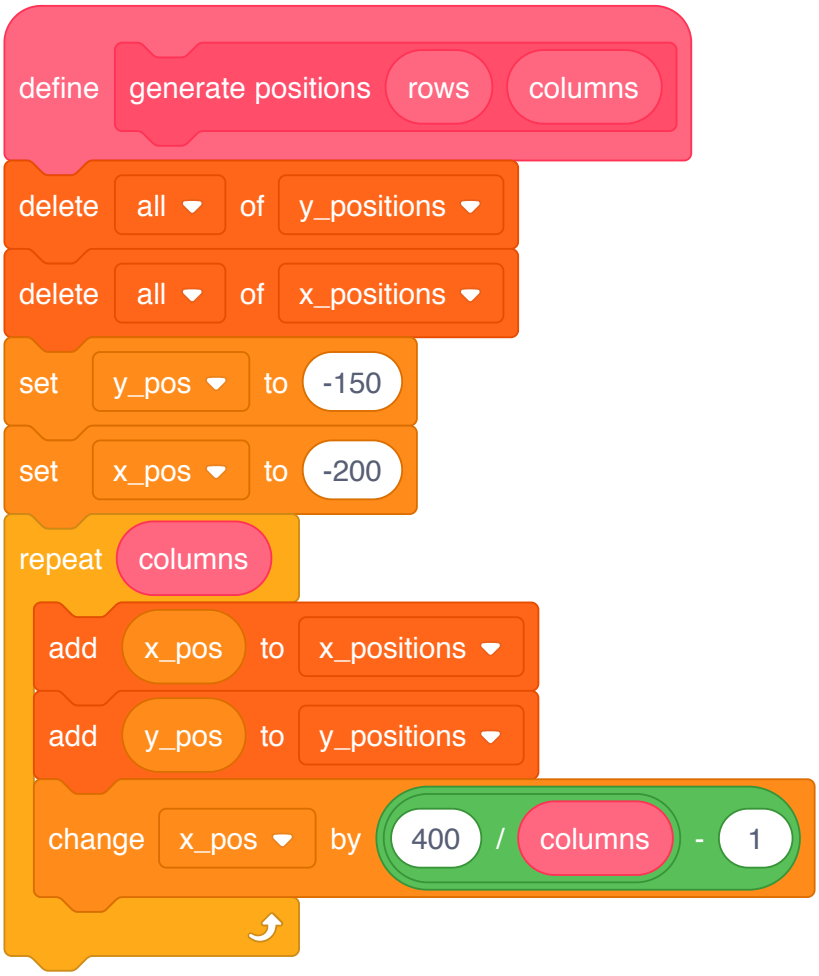
This is what your code should look like:



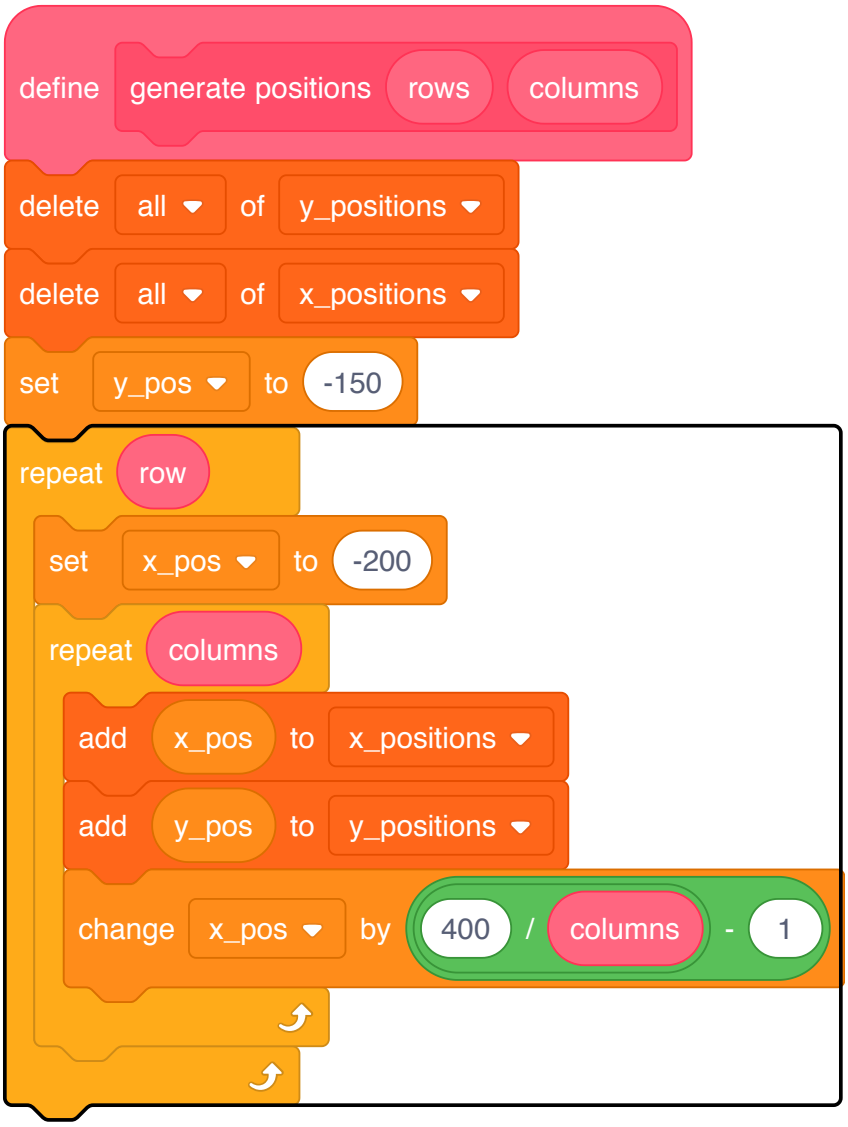
Step 7 Add rows

Now that you have the code to create a single row of stamped costumes, you should add code to create more rows.

Go to your `generate positions` block.



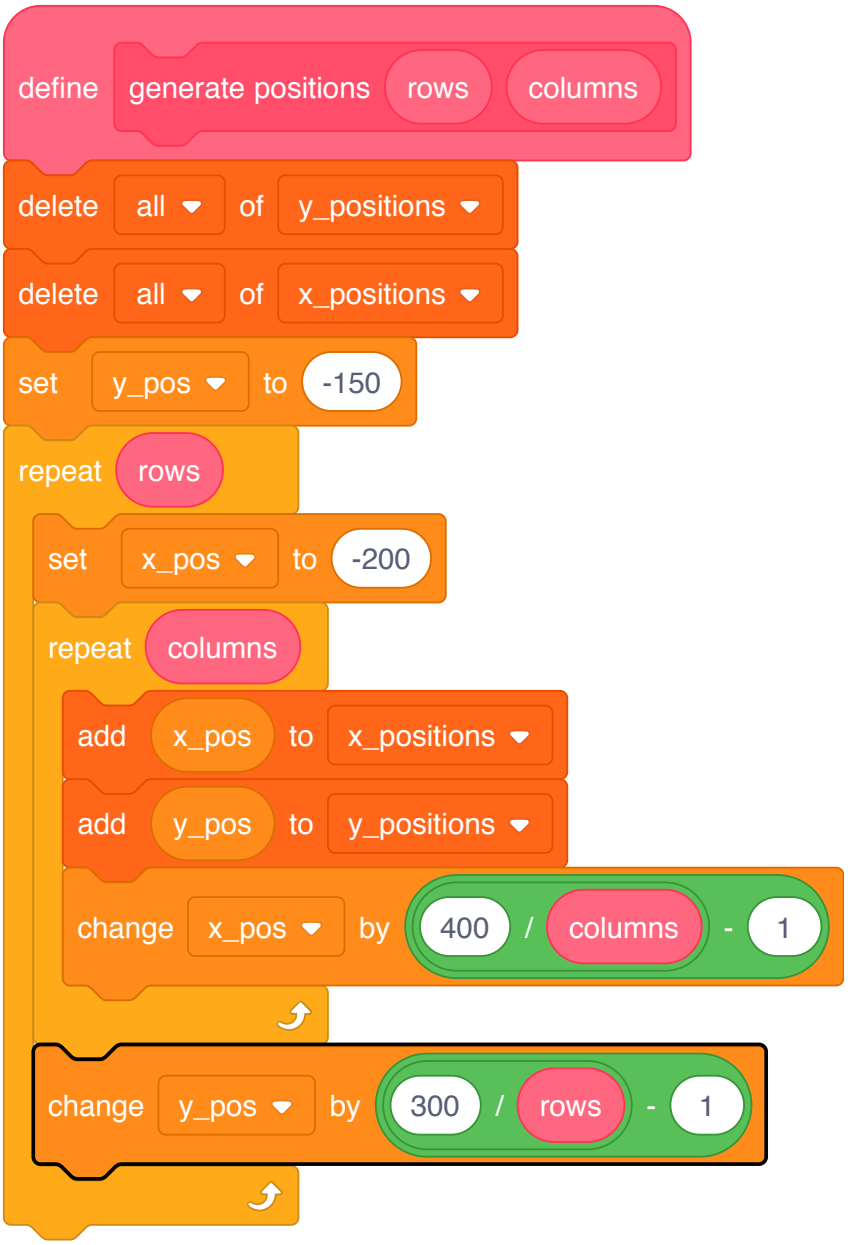
Add another `repeat` loop that runs the number of times you give to the `generate positions` block as the `rows` input. Place the `repeat` loop into your script as shown here:



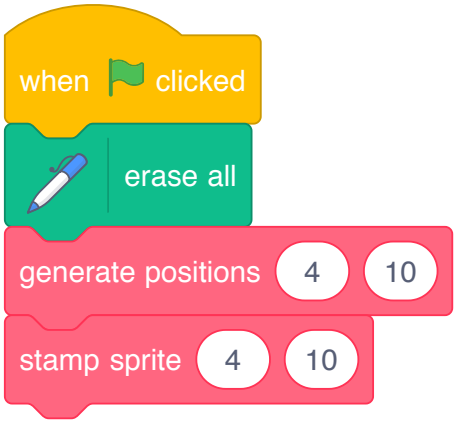
Next you need to increase the value of `y_pos` each time the `repeat (rows)` loop runs.

You do this in a similar manner to how you increase the value of `x_pos` in the `repeat (columns)` loop.

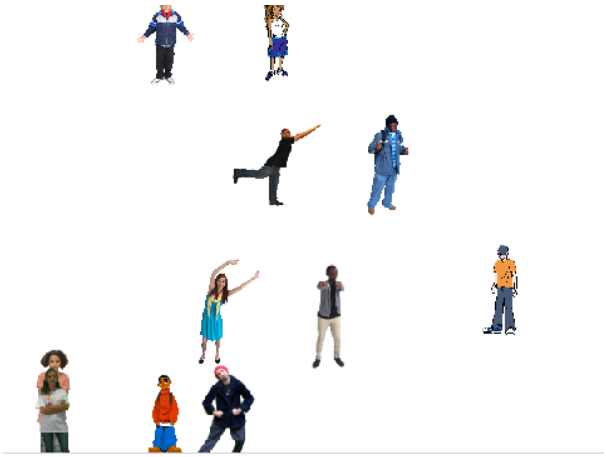
At the end of the code inside the `repeat (rows)` loop, `y_pos` should increase up to `150`, which is `300` away from its starting value of `-150`. This needs to happen for each row of stamps.



Make sure you give the number of `rows` as an input to your blocks.



Run your code now.

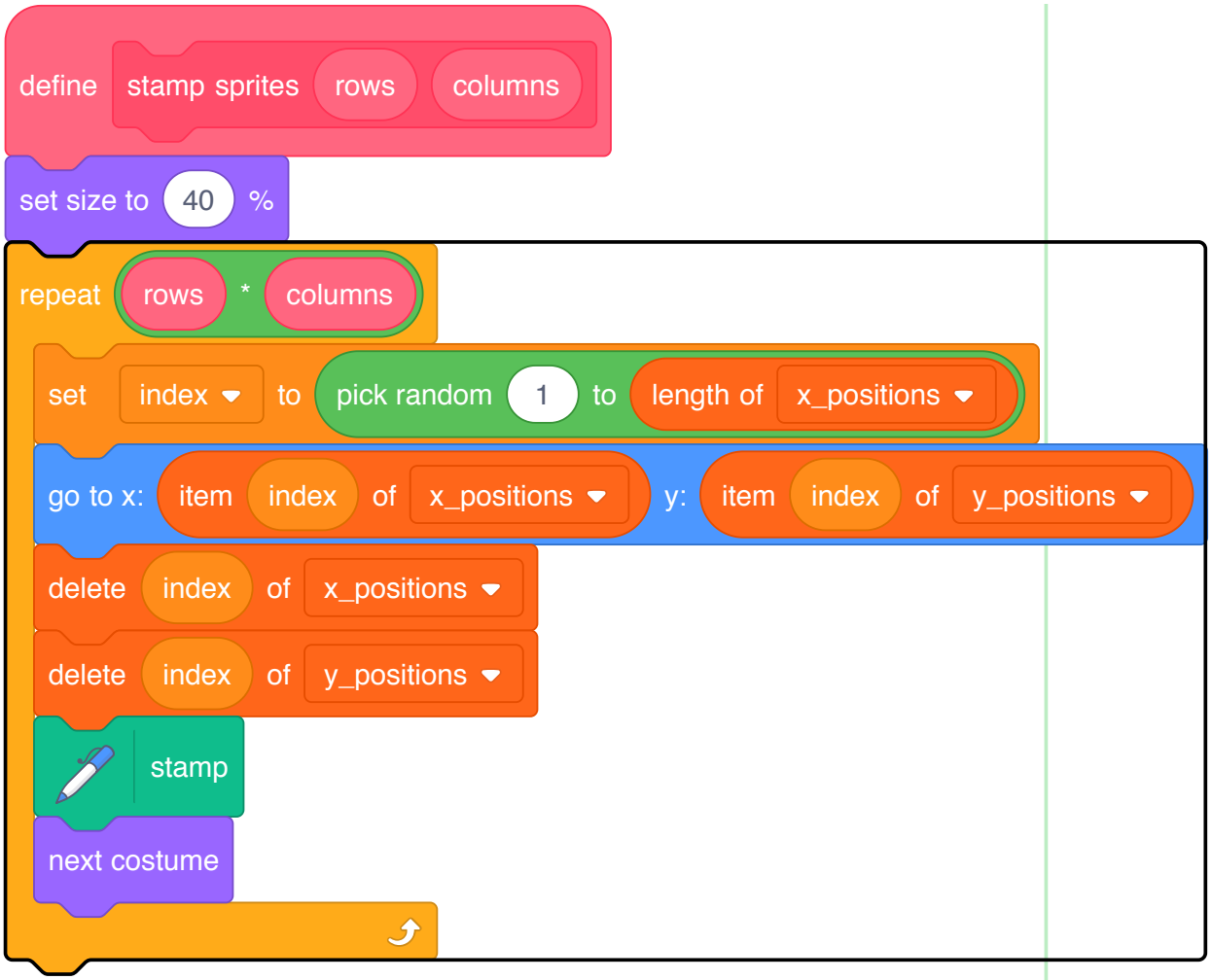


You won't get a neat grid of stamps.

This is because, right now, the `stamp sprite` block only runs for the total number of columns.

Change your `stamp sprites` script so that it `repeats` enough times to stamp the complete grid of sprites.

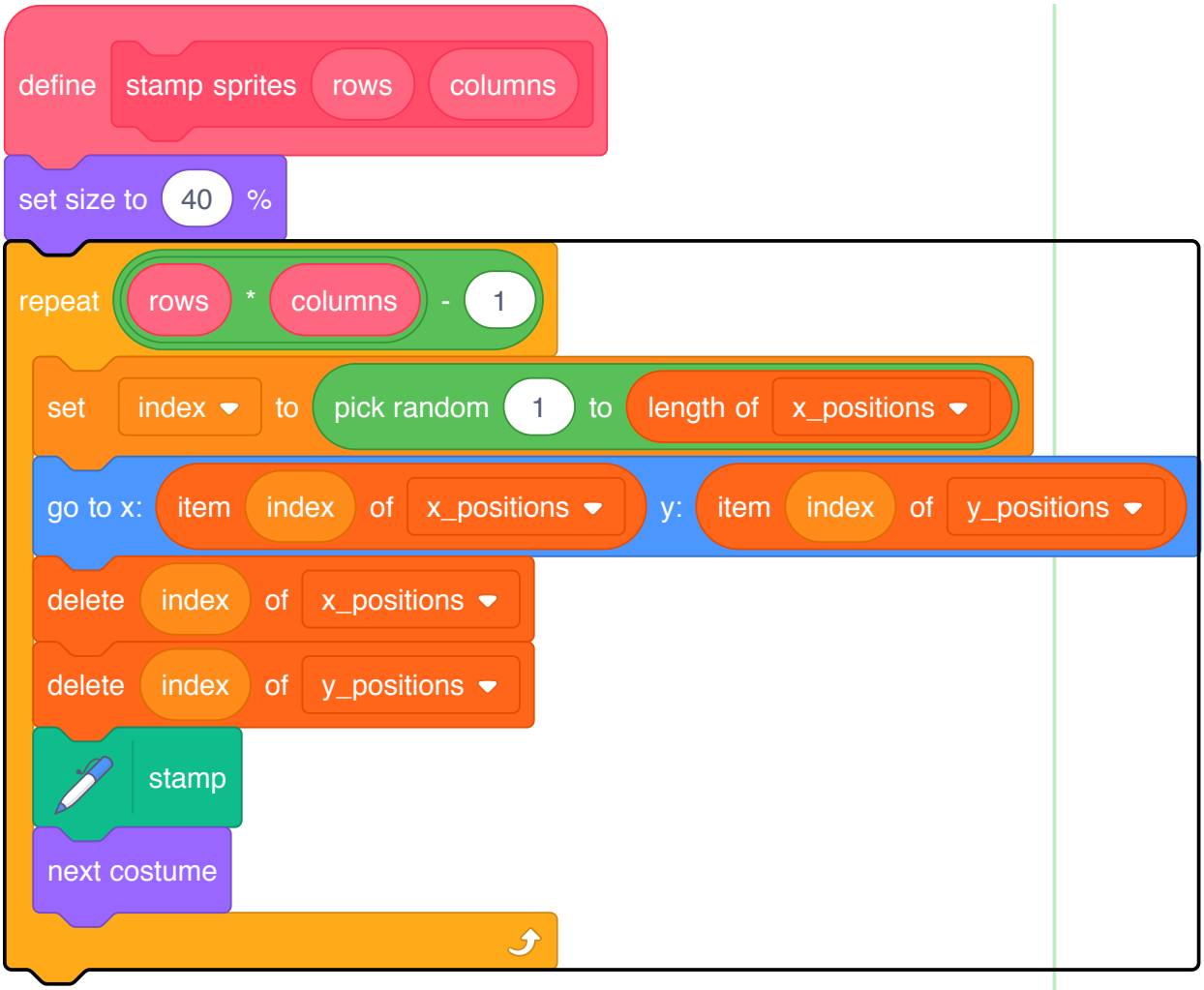
Here's the completed `stamp sprites` script:



Step 8 Hide your sprite

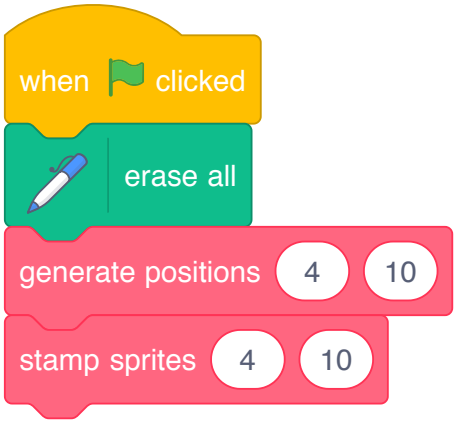
Now it's time to hide your sprite among the crowd of stamps. At the moment the sprite overlaps one of the stamps.

So this doesn't happen, make your stamp loop run one time less: `(rows * columns) - 1`



If you run the script now, you can see that your sprite still overlaps with a stamp and there is a hole in your grid. And in the `x_positions` and `y_positions` lists, there is one coordinate position left.

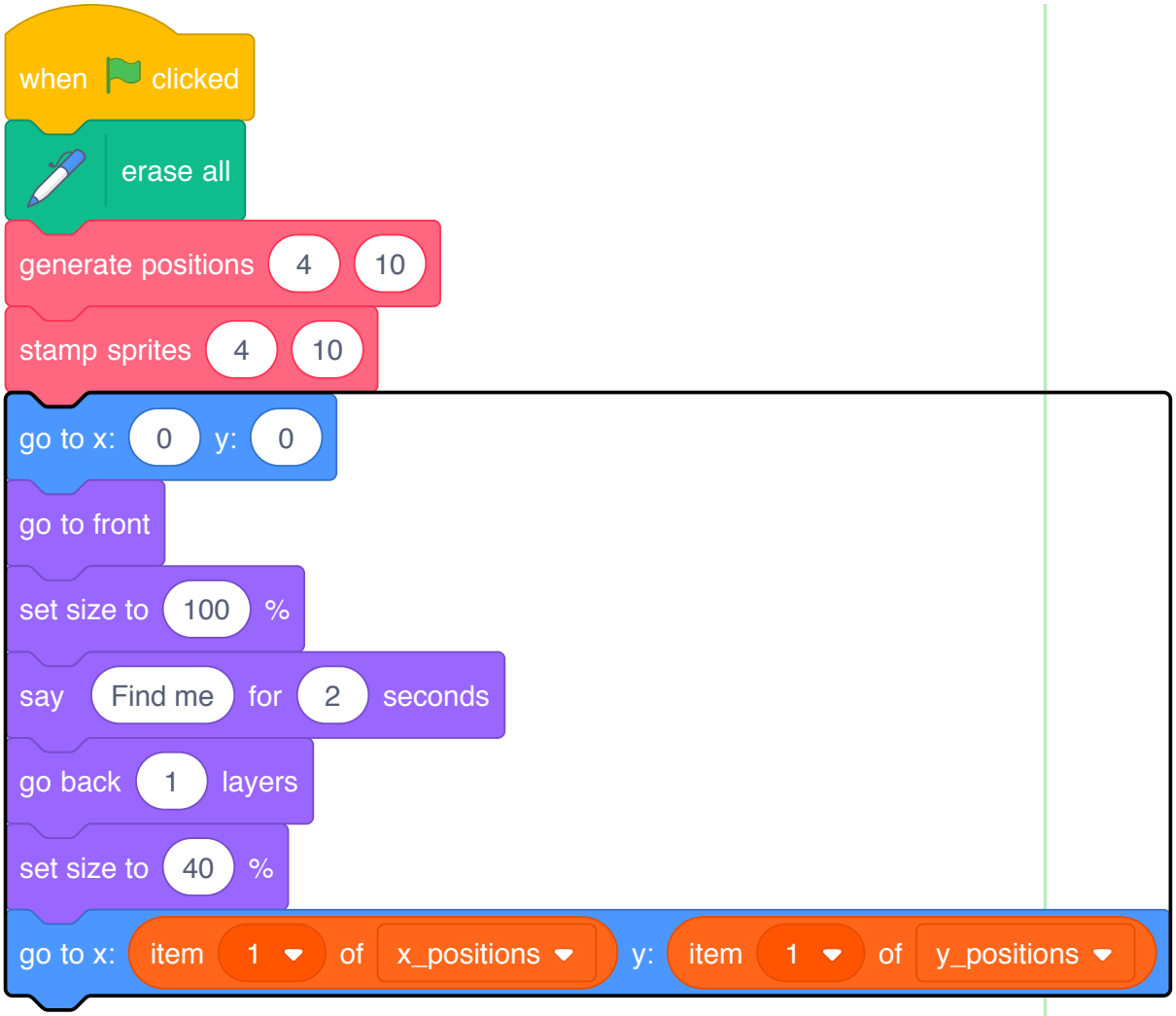
To finish this part your game, go to the `when flag clicked` section of the scripts.



At the start of the game, the sprite should appear at a large size and say “Find me”. Then the sprite should hide itself among the stamps in the empty space you have left for it.

See if you can figure out how to do this, and use the hints below if you need help.

Here is the completed `when flag clicked` script:



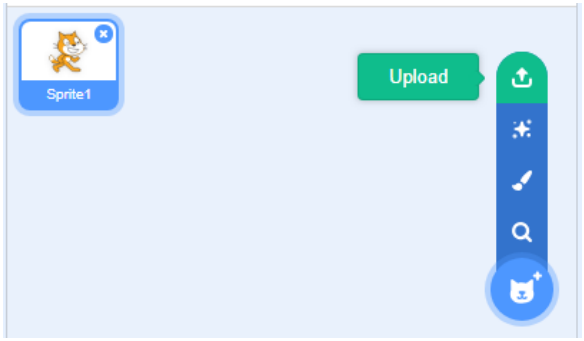
Step 9 Finish the game

To finish the game, **find and download an image of a stage curtain** (https://www.google.co.uk/search?q=stage+curtain&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjKg901k8_VAhXSL1AKHe1HDMIQ_AUICigB&biw=1362&bih=584).

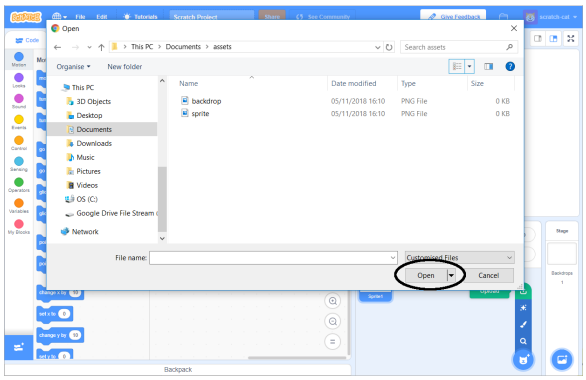
Import this image as a sprite.

Adding a sprite from a file

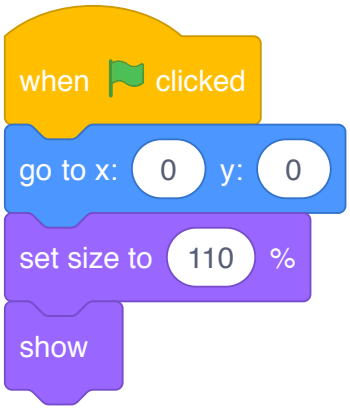
- Click **Upload** from the sprite menu to open up a file browser.



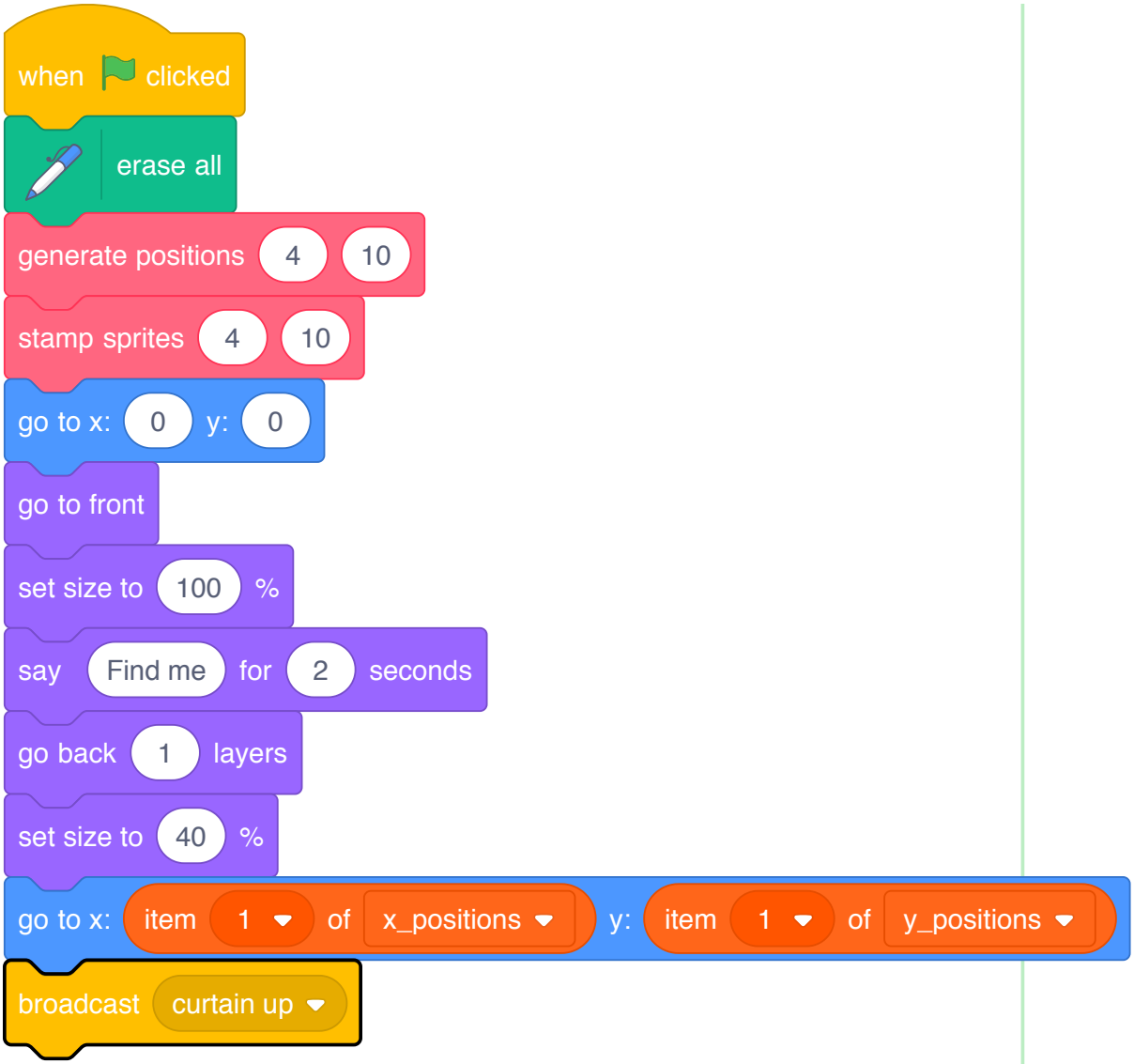
- Choose a file and click on **Open** when you are done.



Position the new curtain sprite at `x:0 y:0`, and then change its size so that it fills the screen. Make sure it is visible.



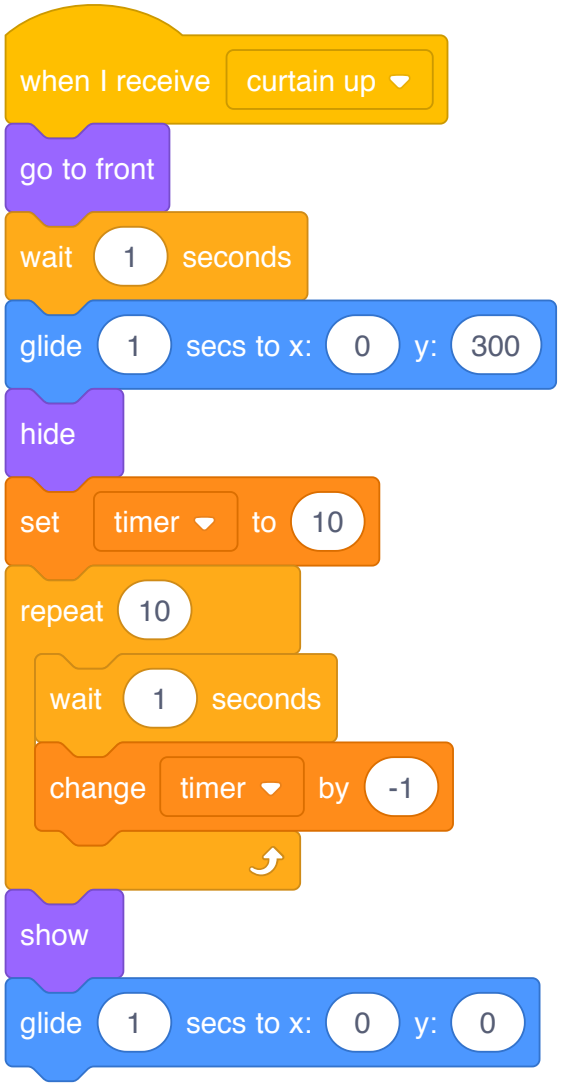
Then, in the scripts for your character sprite, add a `broadcast` with the message 'curtain up' to the end of the `when flag clicked` script.



When the curtain sprite receives the `broadcast`, the sprite needs to move upwards for 10 seconds so that it looks like the curtain is raised to reveal the stamps. Then the curtain should drop again, so the curtain sprite needs to move downwards.

Try to do this by yourself, and use the hints if you need help.

This is the completed script:



The very last part is to let the player know if they've won.

In the scripts for the the character sprite, add code so that, when the sprite is clicked, the sprite says `You've found me`, and all the scripts in the game stop.

Challenge!

Challenge: improve your game

Here are some ideas for how to make your game more interesting:

- Can you alter your scripts so that you can use even more costumes?
- How about choosing a background and then making the sprites nearer the top of the screen appear smaller, so they seem further away?
- Can you make the game run for several rounds and provide you with a score based on how long it takes you to complete five rounds?

Step 10 What next?

Try the **Flappy parrot** (https://projects.raspberrypi.org/en/projects/flappy-parrot?utm_source=pathway&utm_medium=whatnext&utm_campaign=projects) project, in which you create another game.

You will press the `space` bar to make the parrot flap its wings, and score one point for every pipe that you manage to get the parrot past.