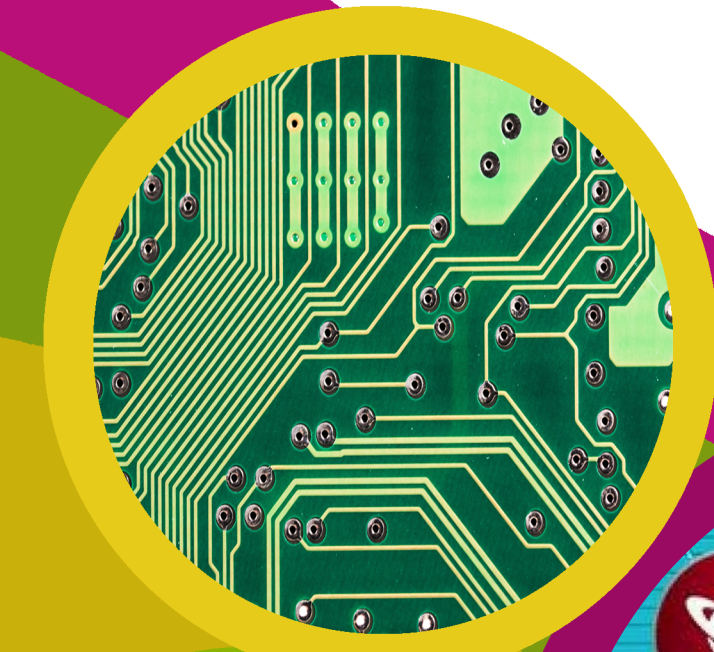


# technocamps

## Using Scratch across the CfW



## Overview

Coding can be implemented across all the Areas of Learning and Experience, reinforcing learning in the classroom and improving digital literacy in the process.

In today's world digital literacy is an essential skill for learners to develop. The technological requirements for jobs are ever increasing, and a strong start in digital skills will prepare learners and give them an advantage.

Digital Resources:

<https://tc1.me/educonf22resources>

Youtube Tutorials:

<https://tc1.me/progacrosscurriculum>

## Online Resources

## More Ideas to Program



### Health and Wellbeing

- Food Pyramid
- Pong



### Mathematics and Numeracy

- Drawing Shapes
- Estimating Pi



### Science and Technology

- States of Matter
- Water Cycle



### Languages, Literacy and Communication

- Translating Quiz
- Pronouns Quiz



### Expressive Arts

- Algorithmic Art
- Matching Art Styles



### Humanities

- Interactive Timeline
- Migration Simulation

## Extensions

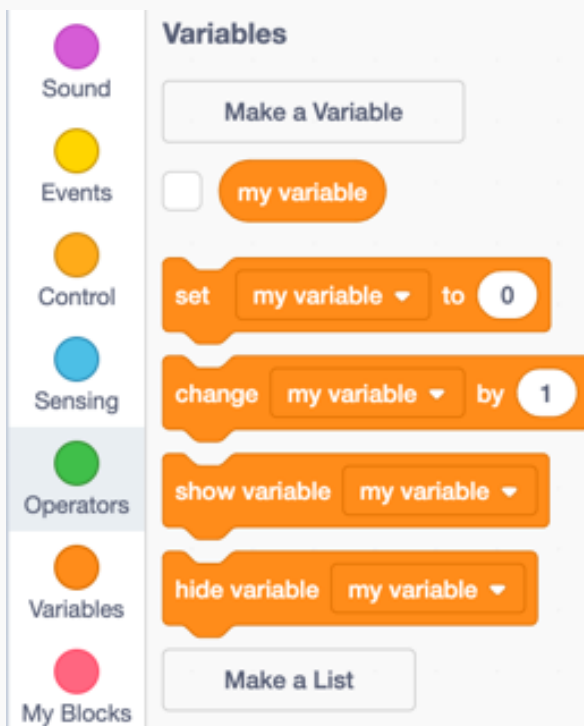


Extensions can be added into Scratch to give us additional blocks that help us accomplish specific tasks.

One of these extensions makes use of Google Translate to translate text between languages.

We can add this extension by clicking in the **bottom left corner** of Scratch and searching through the Extensions.

## Variables and Lists



We will have to create both a variable and a list for this program. These can be added under the variables tab.

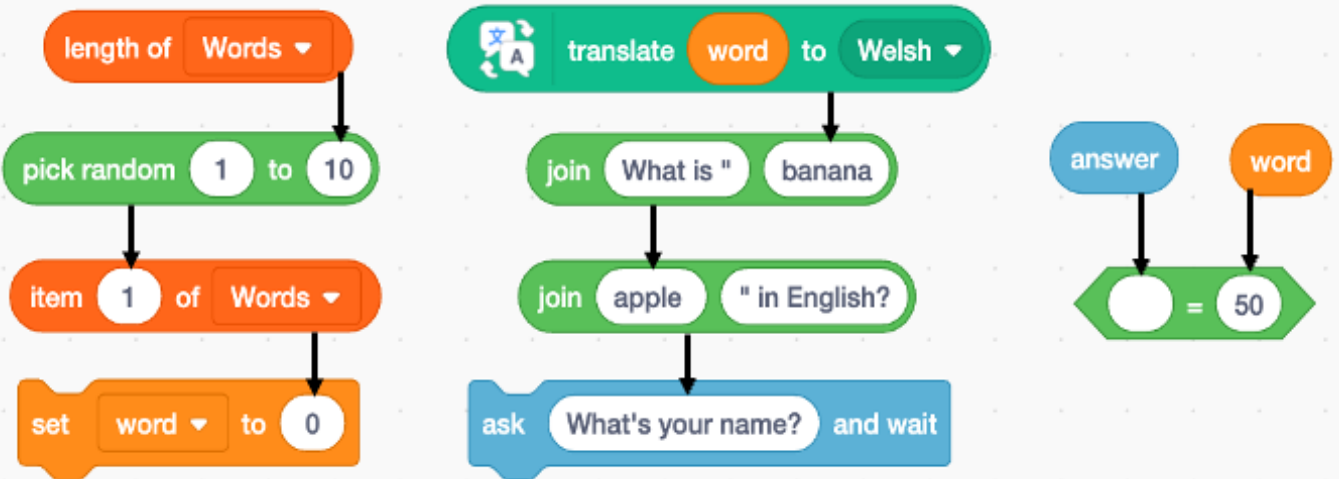
Make a variable called **word**, this will hold the word that is currently displayed on-screen.

Make a list called **words**, this will hold all the words that will appear to be translated. Words can be added to the list on the game screen, and then hidden from view in the variables tab.

## Translator Blocks



## Assembling Translator





## If Statement

The image shows three Scratch code snippets. The first is a 'when green flag clicked' block followed by a 'say Correct! for 2 seconds' block. The second is an 'if' block with a diamond-shaped condition, a 'then' block containing a 'say Wrong for 2 seconds' block, and an 'else' block. The third is a 'when green flag clicked' block followed by a 'forever' loop containing an 'if' block with a diamond-shaped condition, a 'then' block containing a 'say Correct! for 2 seconds' block, and an 'else' block containing a 'say Wrong for 2 seconds' block.

## Full Code

The full code for the game is as follows:

```
when green flag clicked
  forever loop
    set word to item pick random 1 to length of Words of Words
    ask join join What is translate word to Welsh in English? and wait
    if answer = word then
      say Correct! for 2 seconds
    else
      say Wrong for 2 seconds
```

## Drawing a Square

The image shows two sets of Scratch code blocks. The left set includes: 'when I receive Draw Square', 'move 100 steps', 'pen down', 'turn 90 degrees', and 'repeat 4'. The right set includes: 'when clicked', 'go to x: 0 y: 0', 'erase all', and 'broadcast Draw Square'.

## Variable Patterns

The image shows the Scratch 'Variables' panel. It includes a 'Make a Variable' button, a variable named 'my variable', and several code blocks: 'set my variable to 0', 'change my variable by 1', 'show variable my variable', and 'hide variable my variable'. A red arrow points from the text 'Create a variable named "line length"' to the 'my variable' variable name.

Using a variable with a value that changes as we loop, we can make some unique patterns.

Create a variable named "line length".

The image shows a 'New Variable' dialog box. It has a text input field containing 'line length', radio buttons for 'For all sprites' (selected) and 'For this sprite only', and a checkbox for 'Cloud variable (stored on server)'. There are 'Cancel' and 'OK' buttons at the bottom.

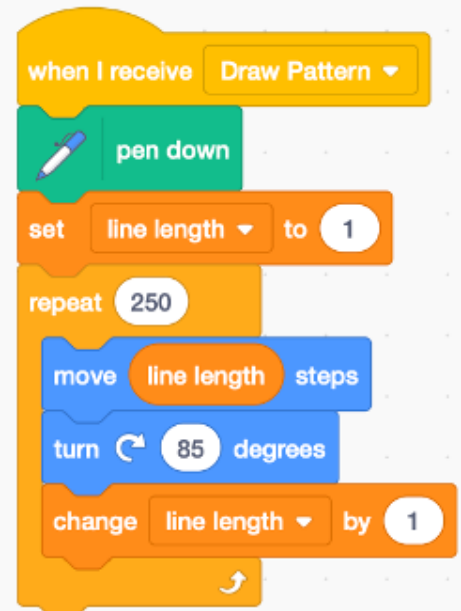
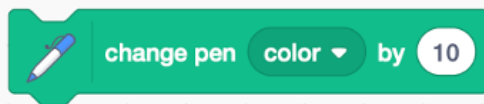
## Variable Patterns



## Variable Patterns

By editing the values inside the loop, (particularly the angle) you will see different patterns emerge.

You can also use the change colour block to add some more colour to the patterns.



## Background

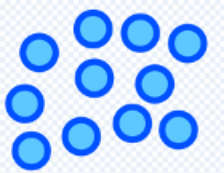
Begin by making a background that includes land, sea, mountains and a river.

These are the key components that we're going to need to create an animation of the water cycle.

**The positional values in the code below will depend on the drawing.**



## Costumes



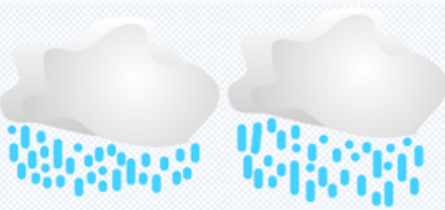
We're going to need 5 costumes to switch between in this animation.

**Water Vapour** - to demonstrate evaporation

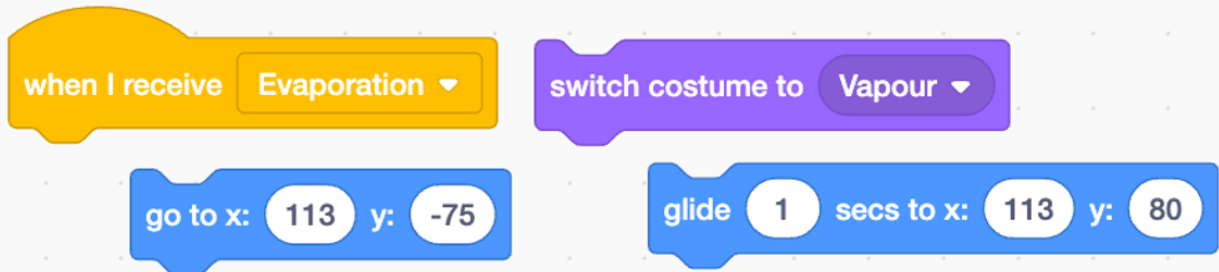
### Clouds

**2x Rainclouds** - to animate the rainfall

**River Flow** - to show the rain water returning to the sea.

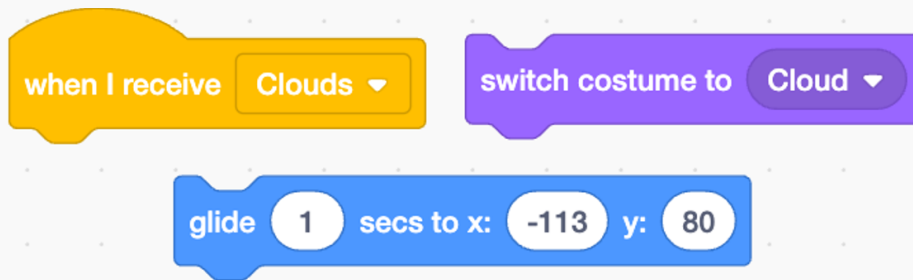


## Sprite - Evaporation



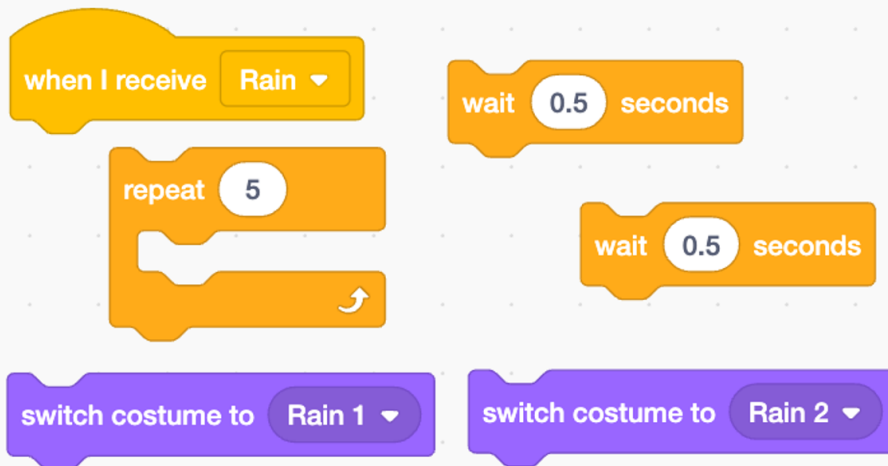
```
when I receive Evaporation
  switch costume to Vapour
  go to x: 113 y: -75
  glide 1 secs to x: 113 y: 80
```

## Sprite - Clouds



```
when I receive Clouds
  switch costume to Cloud
  glide 1 secs to x: -113 y: 80
```

## Sprite - Rain



```
when I receive Rain
  repeat 5
    wait 0.5 seconds
    switch costume to Rain 1
    wait 0.5 seconds
    switch costume to Rain 2
```



## Sprite - River Flow




```
when I receive River Flow
  switch costume to Droplets
  go to x: -77 y: 52
  glide 1 secs to x: -81 y: 34
  glide 1 secs to x: -80 y: -4
  glide 1 secs to x: -95 y: 10
  glide 1 secs to x: -60 y: -18
  glide 1 secs to x: -11 y: -40
  glide 1 secs to x: 62 y: -42
```

The script for the 'Sprite - River Flow' consists of the following blocks:

- when I receive **River Flow**
- switch costume to **Droplets**
- go to x: **-77** y: **52**
- glide **1** secs to x: **-81** y: **34**
- glide **1** secs to x: **-80** y: **-4**
- glide **1** secs to x: **-95** y: **10**
- glide **1** secs to x: **-60** y: **-18**
- glide **1** secs to x: **-11** y: **-40**
- glide **1** secs to x: **62** y: **-42**

## Background - Animating



```
when clicked
  broadcast Evaporation and wait
  broadcast Clouds and wait
  forever
    broadcast Rain and wait
    broadcast River Flow and wait
```

The script for the 'Background - Animating' consists of the following blocks:

- when **clicked**
- broadcast **Evaporation** and wait
- broadcast **Clouds** and wait
- forever loop containing:
  - broadcast **Rain** and wait
  - broadcast **River Flow** and wait

## Sprite - Full Code

```
when I receive Evaporation
  go to x: 113 y: -75
  switch costume to Vapour
  glide 1 secs to x: 113 y: 80

when I receive Clouds
  switch costume to Cloud
  glide 1 secs to x: -113 y: 80

when I receive Rain
  repeat 5
    switch costume to Rain 1
    wait 0.5 seconds
    switch costume to Rain 2
    wait 0.5 seconds

when I receive River Flow
  switch costume to Droplets
  go to x: -77 y: 52
  glide 1 secs to x: -81 y: 34
  glide 1 secs to x: -95 y: 10
  glide 1 secs to x: -80 y: -4
  glide 1 secs to x: -60 y: -18
  glide 1 secs to x: -11 y y: -40
  glide 1 secs to x: 62 y: -42
```

## Background - Full Code

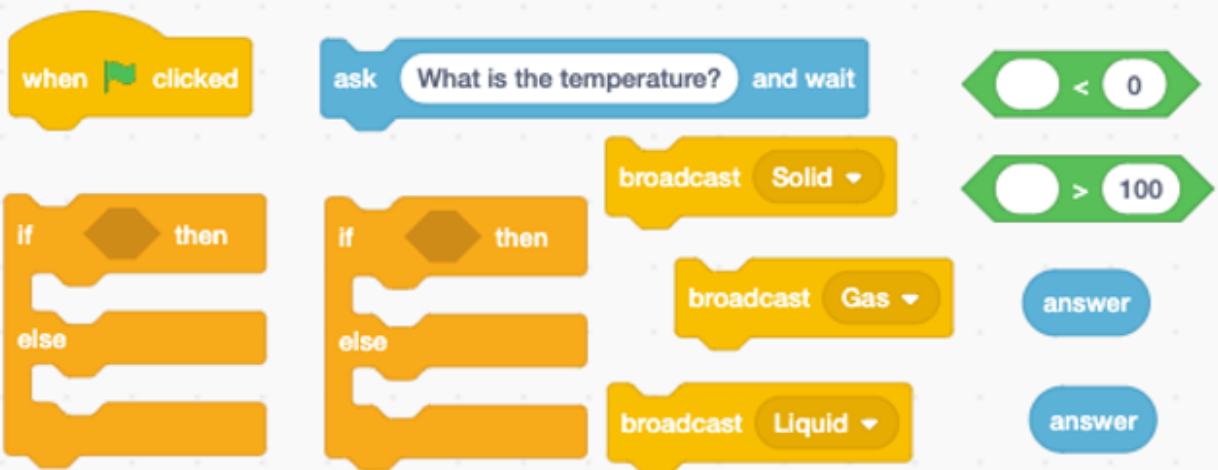
```
when clicked
  forever
    broadcast Evaporation and wait
    broadcast Clouds and wait
    broadcast Rain and wait
    broadcast River Flow and wait
```

## Sprites

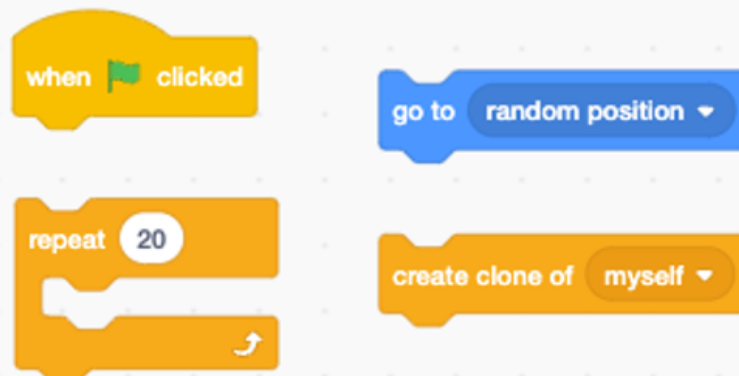


Create a simple circular sprite to represent an atom. If you prefer this could be a molecule instead. We will only require 1 sprite as it will be cloned.

## Background



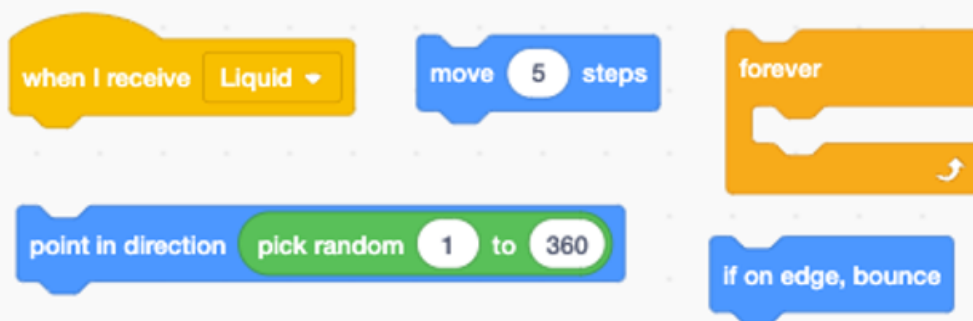
## Sprite - Clones



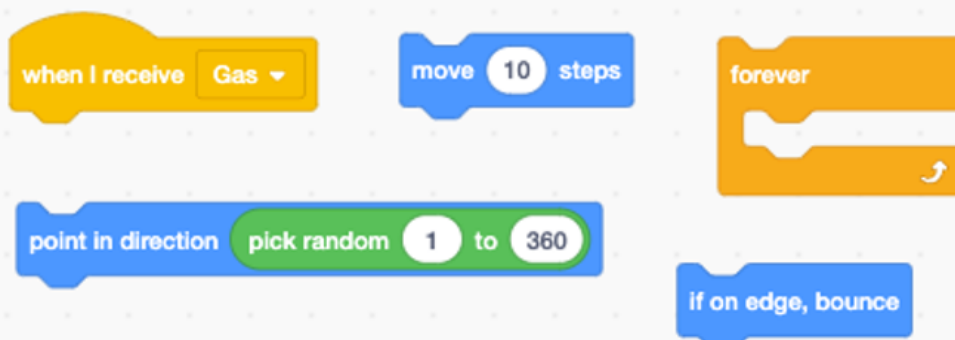
## Sprite - Solid



## Sprite - Liquid



## Sprite - Gas



## Background - Full Code

```
when clicked
ask "What is the temperature?" and wait
if answer < 0 then
broadcast Solid
else
if answer > 100 then
broadcast Gas
else
broadcast Liquid
```

## Sprite - Full Code

```
when clicked
repeat 20
go to random position
create clone of myself
```

```
when I receive Solid
forever
point in direction pick random 1 to 360
move 5 steps
wait 0.1 seconds
move -5 steps
wait 0 seconds
```

```
when I receive Liquid
point in direction pick random 1 to 360
forever
move 5 steps
if on edge, bounce
```

```
when I receive Gas
point in direction pick random 1 to 360
forever
move 10 steps
if on edge, bounce
```

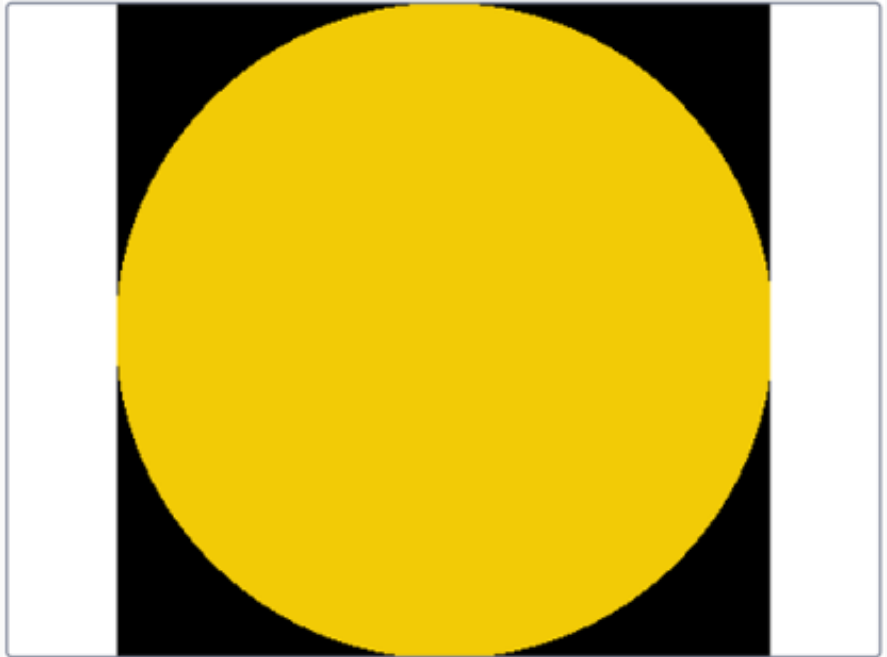


## Background

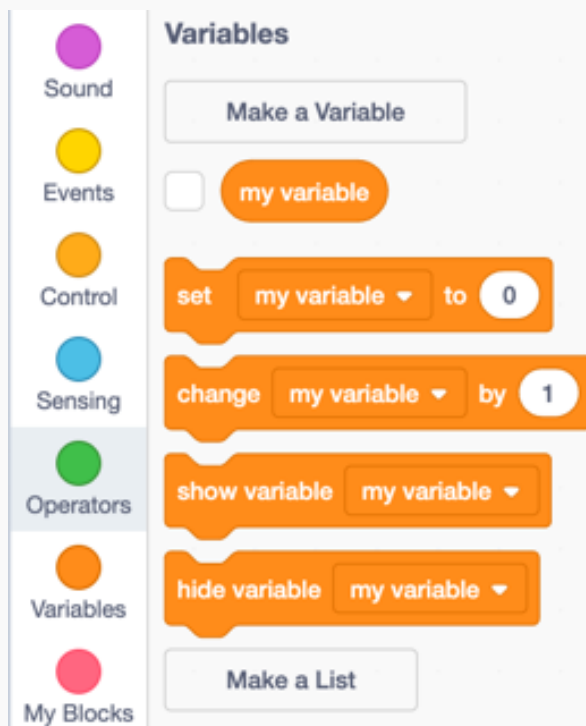
Begin by making a background consisting of a square with a circle of equal diameter inside.

Make them different colours as this is how we will calculate the ratio of areas and pi.

**The colours you choose will be used throughout the code.**



## Variables



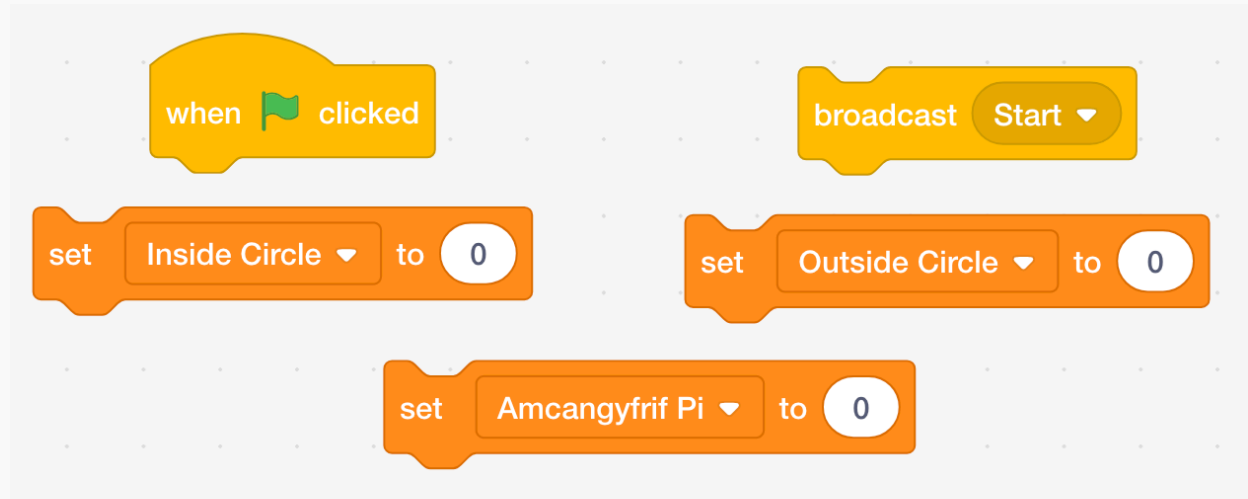
We will have to create three variables to allow us to calculate pi in this program.

**Inside Circle** - will count the number of times the sprite lands inside the circle.

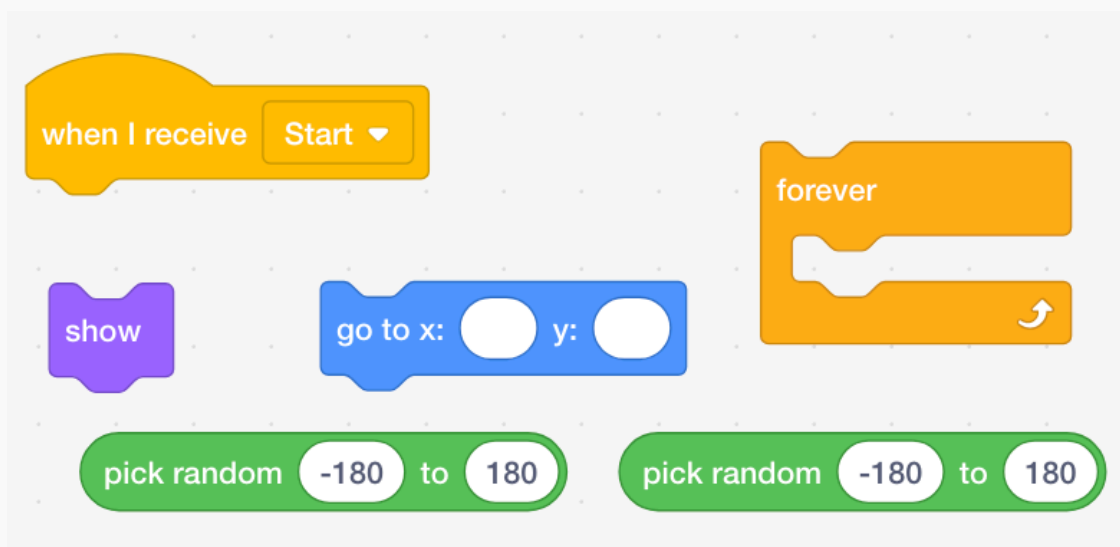
**Outside Circle** - will count the number of times the sprite lands inside the square.

**Pi Estimation** - will be the ratio of times inside and outside the circle, which is equal to pi.

## Background - Starting Conditions



## Sprite - Random Movement

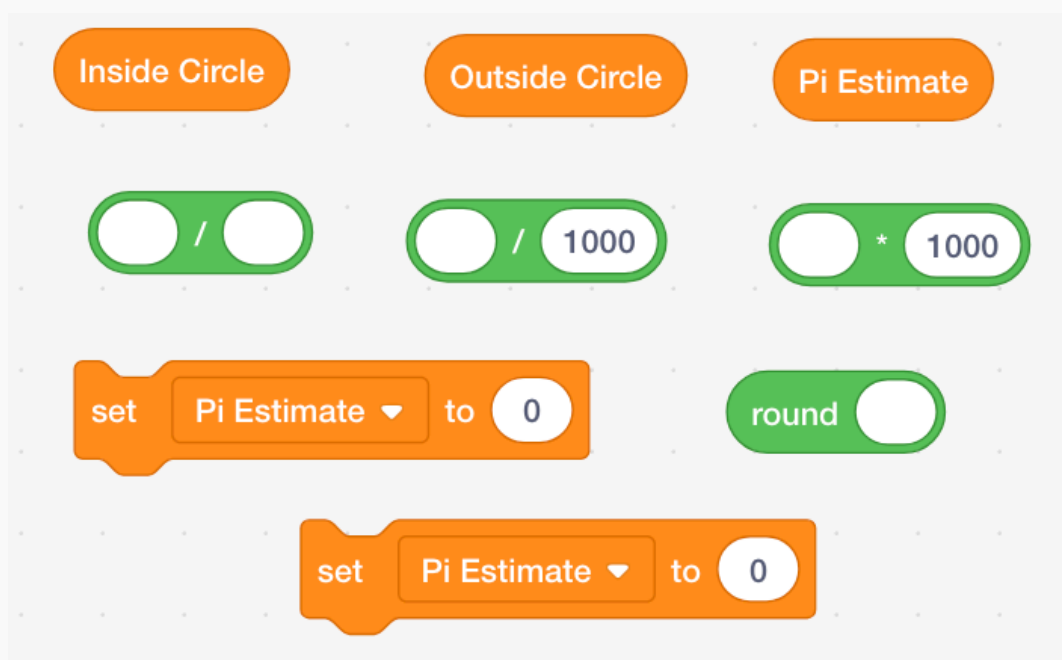


## Sprite - Inside or Outside?

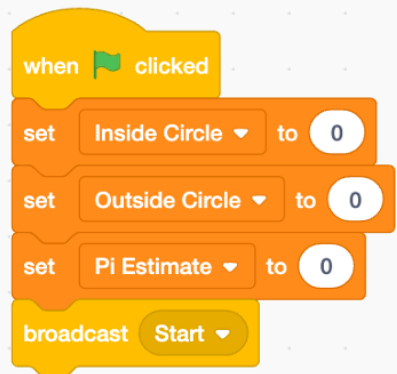


Use the colour chooser to select the colours from your background.

## Sprite - Calculating the Ratio

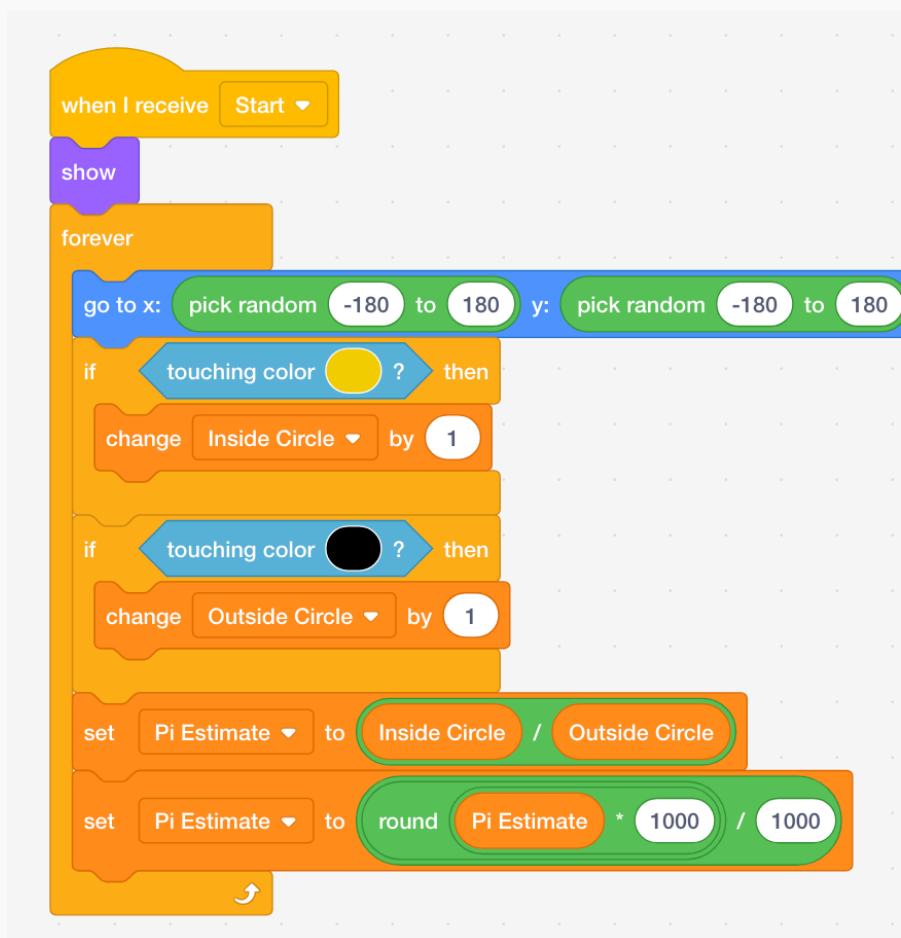


## Background - Full Code



```
when green flag clicked
  set Inside Circle to 0
  set Outside Circle to 0
  set Pi Estimate to 0
  broadcast Start
```

## Sprite - Full Code



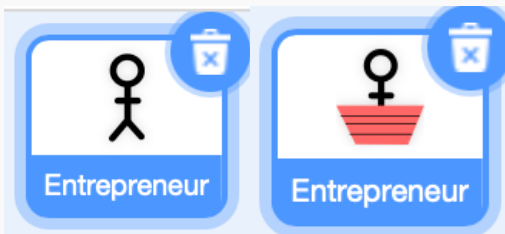
```
when I receive Start
  show
  forever
    go to x: pick random -180 to 180 y: pick random -180 to 180
    if touching color yellow then
      change Inside Circle by 1
    if touching color black then
      change Outside Circle by 1
    set Pi Estimate to Inside Circle / Outside Circle
    set Pi Estimate to round Pi Estimate * 1000 / 1000
```

## Background

Begin by making a background that looks like a map for your people to migrate across.



## Costumes

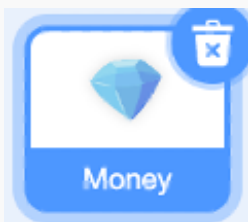


We're going to need 3 different sprites for this simulation.



**Entrepreneur** - These will be our people migrating across the map, they will need a separate boat costume!

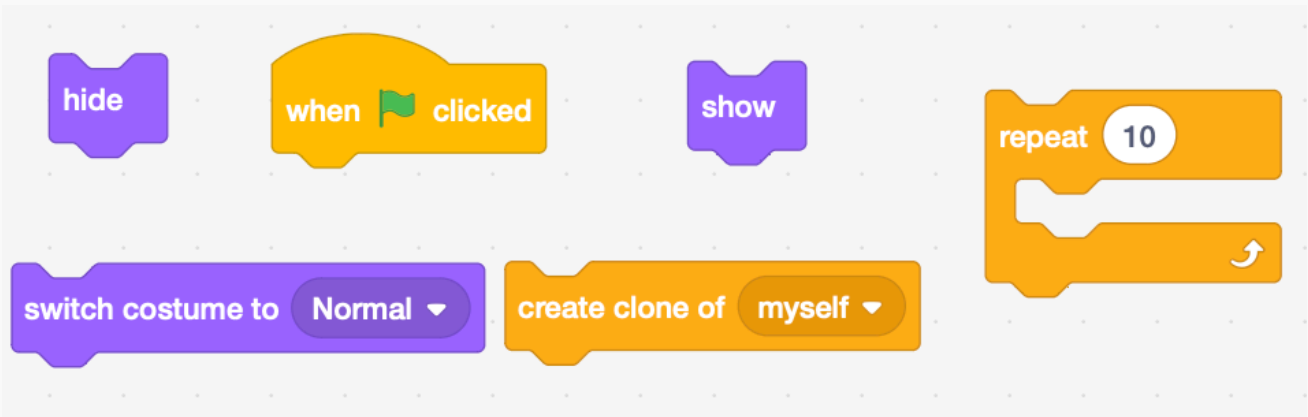
**War** - Our entrepreneurs will want to avoid any wars.



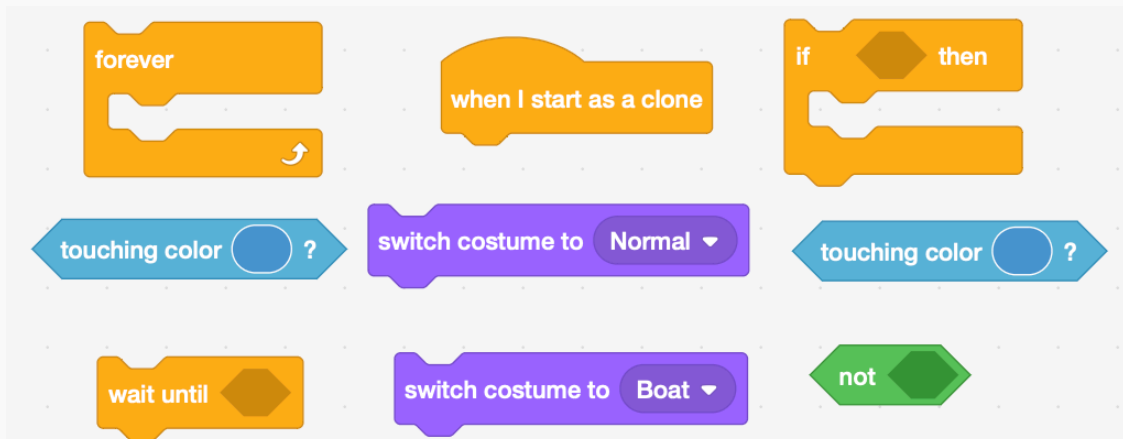
**Money** - Our entrepreneurs will be seeking out wealthy cities.



## Entrepreneur - Cloning



## Entrepreneur - Switching to Boat



## War

```
set drag mode draggable
```

```
when flag clicked
```

```
go to random position
```

## Entrepreneur - Movement Close to War

```
when I start as a clone
```

```
go to random position
```

```
forever
```

```
point in direction pick random 1 to 360
```

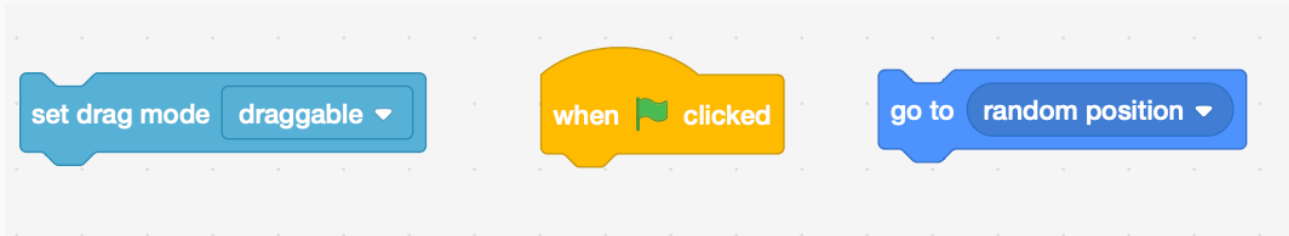
```
move 1 steps
```

```
if distance to War < 50 then
```

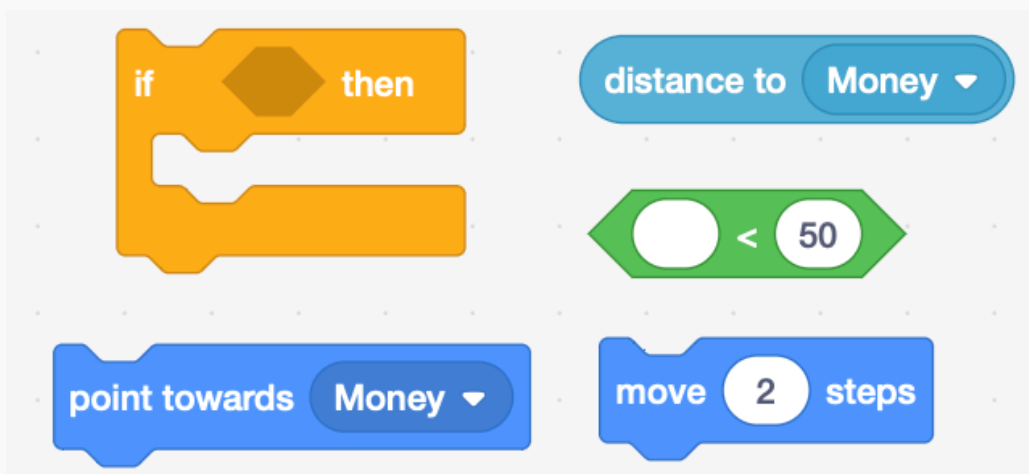
```
point in direction pick random 1 to 360
```

```
move 10 steps
```

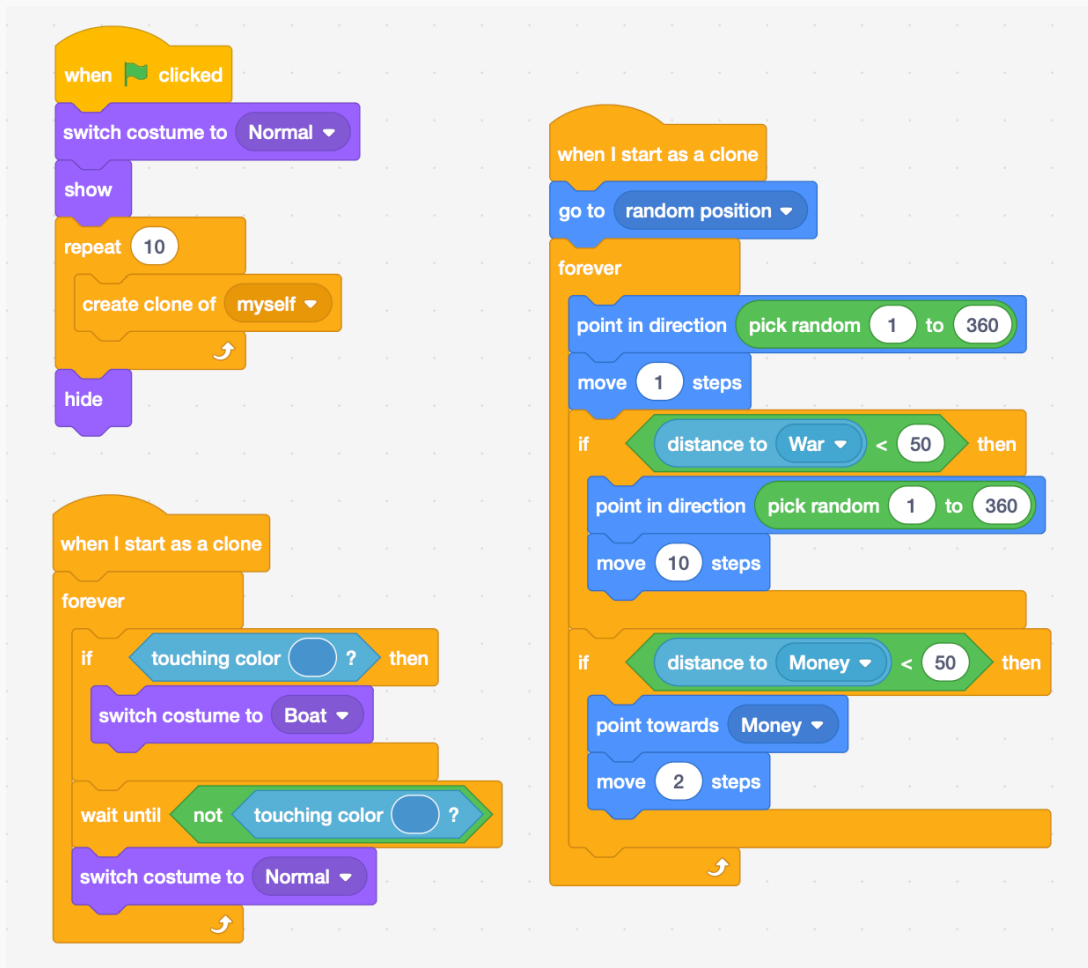
## Money



## Entrepreneur - Move Toward Money



## Entrepreneur - Full Code



```
when green flag clicked
  switch costume to Normal
  show
  repeat 10
    create clone of myself
  hide

when I start as a clone
  forever
    if touching color [blue] ? then
      switch costume to Boat
    wait until not touching color [blue] ?
    switch costume to Normal

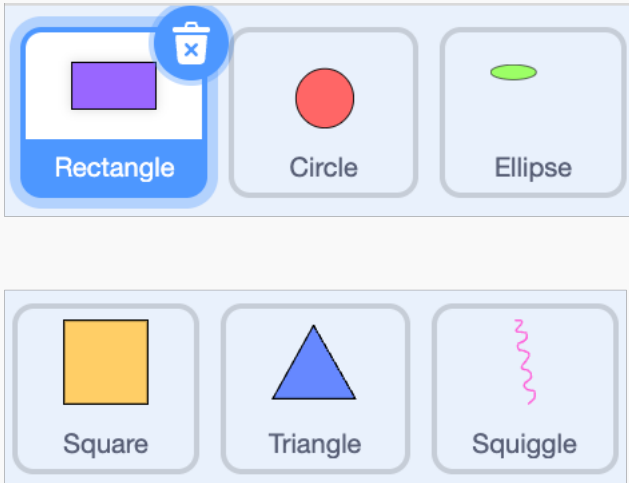
when I start as a clone
  go to random position
  forever
    point in direction pick random 1 to 360
    move 1 steps
    if distance to War < 50 then
      point in direction pick random 1 to 360
      move 10 steps
    if distance to Money < 50 then
      point towards Money
      move 2 steps
```

## Money and War - Full Code



```
when green flag clicked
  go to random position
  set drag mode draggable
```

## Sprites

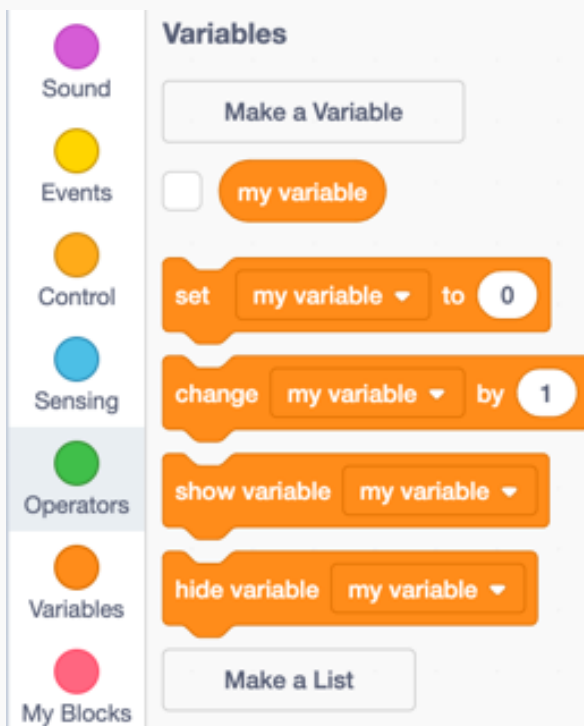


Create some sprites of various shapes.

The size and colour of these will be altered by the algorithm to add diversity, so only distinctly different shapes are needed.

The same code will be copied into each shape.

## Variables



Make four new variables to begin. These will be a counter and user inputs to create the art, name them appropriately:

**count, answer 1,  
answer 2 and answer 3**

**These could be Age, Bedtime and  
Family.**



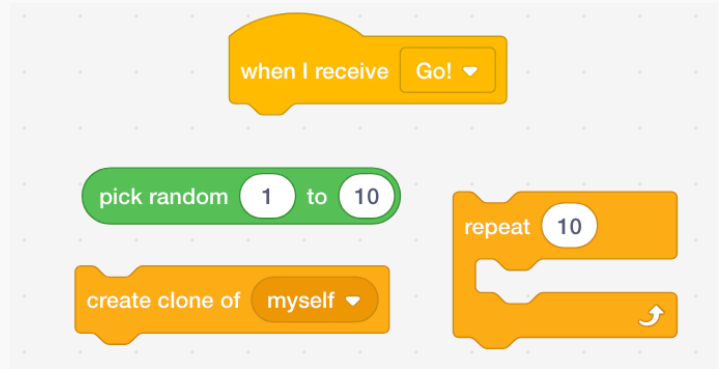
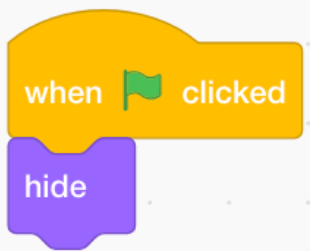
## Questions - Stage



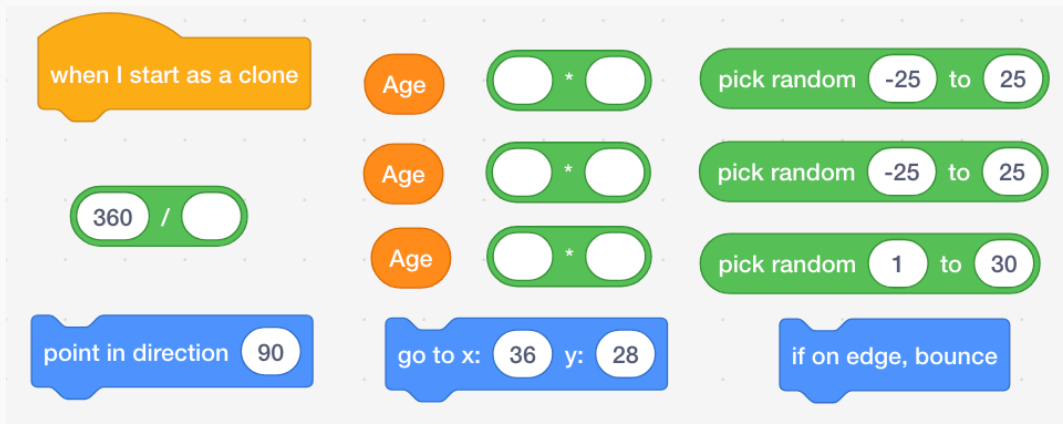
## Answers - Stage



## Starting Conditions - Sprite(s)



## Position - Sprite(s)



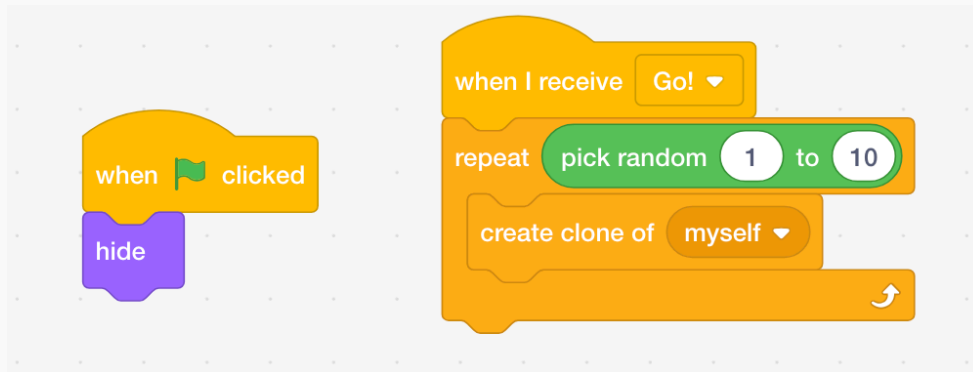
## Colour and Size - Sprite(s)

A collection of Scratch code blocks for color and size manipulation. On the left, there is a purple 'change color effect by 25' block. Below it are two orange 'Favourite' and 'Family' buttons, each followed by a green 'multiply' block with a white circle containing an asterisk. On the right, there is a purple 'change size by 10' block. Below it are two green 'pick random' blocks: one with values '-10 to 10' and another with values '-50 to 50'.

## Layers - Sprite(s)

A collection of Scratch code blocks for layer management. On the left is an orange 'if then' block. In the center are two green blocks: 'mod 2' and '= 0'. On the right is an orange 'Count' button, a purple 'show' block, and a purple 'go to back layer' block.

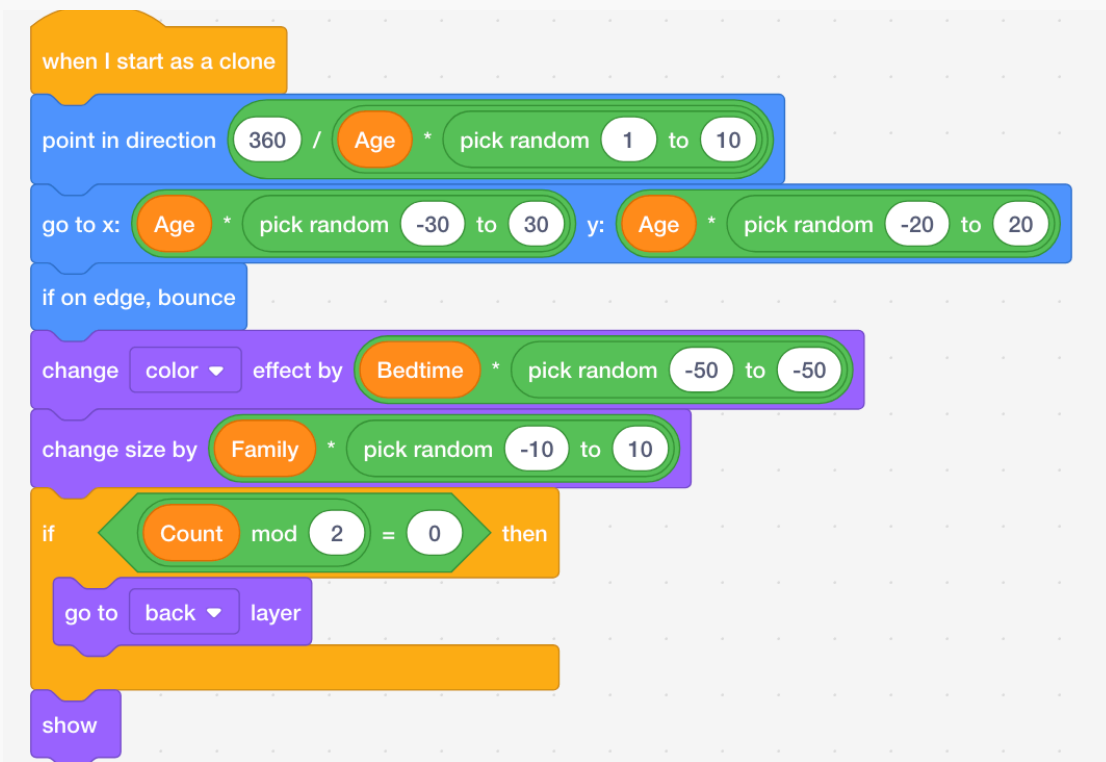
## Sprite(s) - Full Code



```
when clicked
hide

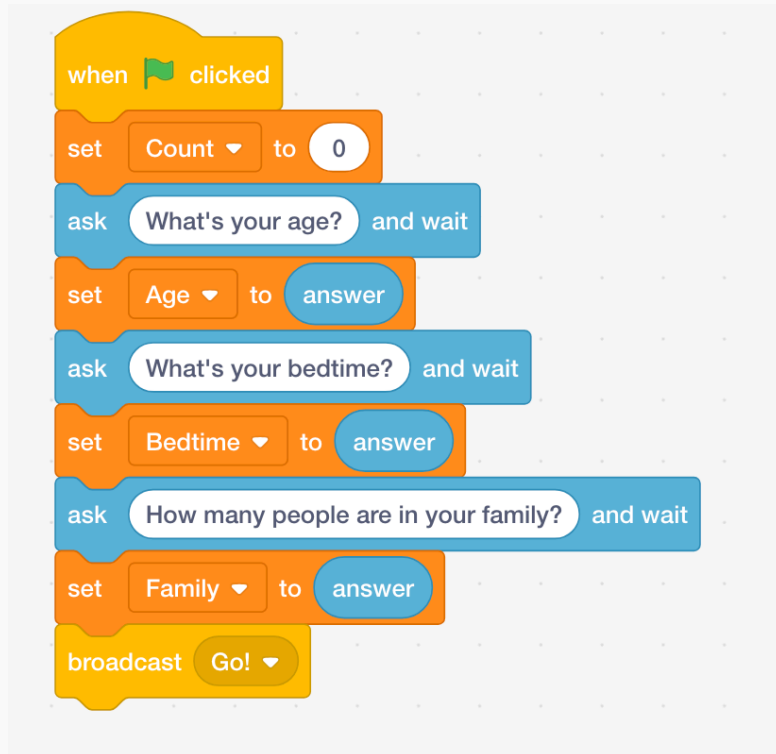
when I receive Go!
repeat (pick random 1 to 10)
  create clone of myself
```

## Sprite(s) - Clones - Full Code



```
when I start as a clone
  point in direction (360 / Age * pick random 1 to 10)
  go to x: (Age * pick random -30 to 30) y: (Age * pick random -20 to 20)
  if on edge, bounce
  change color effect by (Bedtime * pick random -50 to -50)
  change size by (Family * pick random -10 to 10)
  if (Count mod 2 = 0) then
    go to back layer
  show
```

## Stage - Questions - Full Code



```
when green flag clicked
  set Count to 0
  ask "What's your age?" and wait
  set Age to answer
  ask "What's your bedtime?" and wait
  set Bedtime to answer
  ask "How many people are in your family?" and wait
  set Family to answer
  broadcast "Go!"
```

The image shows a Scratch script on a grid background. The script starts with a yellow 'when green flag clicked' block. It is followed by an orange 'set Count to 0' block. Then there are three blue 'ask' blocks with 'and wait' text, asking for 'age', 'bedtime', and 'family size'. Each 'ask' block is followed by an orange 'set' block that assigns the 'answer' to a variable: 'Age', 'Bedtime', and 'Family' respectively. The script ends with a yellow 'broadcast "Go!"' block.



**technocamps**



@Technocamps



Find us on  
**Facebook**