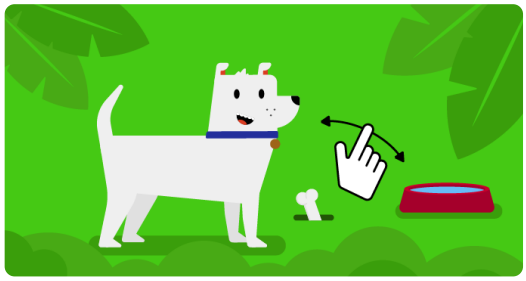
This sprite needs you

Create a virtual pet, character, or nature simulation that the user can interact with to help it.



Step 1 You will make

Create a virtual pet, character, or nature simulation that the user can interact with to help it. Your game will need to meet the project brief.

> Virtual pet games are one type of game where users interact with characters to meet the characters' needs. You might have played with a small handheld Tamagotchi or enjoyed games like Catz and Adopt me! on a computer. Can you think of any other virtual pet games?

You will:

- Make a character or simulation that is fun or interesting to interact with
- Use any combination of `variables`, `broadcast`, and `if` blocks, with the skills you have already learnt, to allow a user to meet the needs of a character
- Understand how characters in games and apps are controlled by algorithms

PROJECT BRIEF: Help me grow

You need to create a virtual pet, plant, or other simulation that the user can interact with to meet its needs. You will use variables to keep track of how your main sprite is doing. It might be happy, bored, thirsty, or sleepy.

Your simulation should:

- Use at least one `variable` to keep track of what the main sprite needs
- Have a way for the variable(s) to change automatically
- Give the user a way to improve the variables to give the main sprite what it needs
- Use `if` blocks to control when things happen
- Use `broadcast` blocks to communicate between other sprites and the main sprite

Your simulation could:

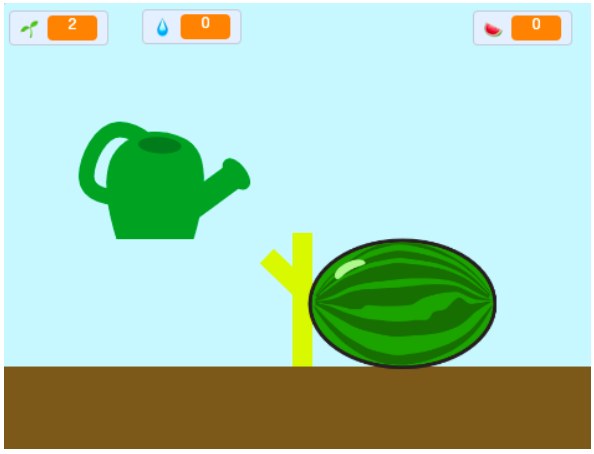
- Have a message, such as kindness or keeping crops healthy

- Alert the user when levels are too high or low
- Allow the user to chat with their sprite, or change its name

Get ideas

Think about what you will choose as a main sprite. It could be a pet that the user cares for, a person that the user helps make good decisions, a plant or crop that the user helps to grow, or a nature or fantasy object that the user needs to create the right conditions for.

See inside example projects in the 'This sprite needs you — Examples' Scratch studio: https://scratch.mit.edu/studios/29683913/



Step 2 Your idea

Plan your virtual pet, character, or nature simulation. The user will have to create the right conditions to support the main sprite's needs. Your main sprite might get bigger, gain experience, increase health, or something else.

You can plan by just thinking, tinkering, drawing or writing, or however you like!

> When basic needs like food and safety are met, we are creating the right conditions for creatures (including ourselves) to learn and grow. So the next time you're feeling cranky or distracted, check your basic needs.

Why are you making your project?

Think about the purpose of your project.

It could be:

- To teach a new skill, such as looking after an animal or keeping a plant alive
- To share a message, such as look after your mental health or improve your neighbourhood
- To entertain users, by giving them something they wouldn't usually get the chance to interact with

Who is it for?

Think about who you will make your project for (your audience).

Knowing your audience will help you design a project they will want to play again and again.

Get started

Open a new Scratch project (http://rpf.io/scratch-new). Scratch will open in another browser tab.

Working offline

To set up Scratch for offline use visit our Scratch guide (https://learning-admin.raspberrypi.org/en/projects/getting-started-scratch/1).

Use your new Scratch project, a pen and paper, or both to plan your ideas.

What will your main sprite be?

- A character such as a pet or wild animal, or a person
- A plant or crop that needs sunlight, water, and nutrients
- A feature from science or nature, such as a rainbow or a fire
- A machine, such as a submarine or a recycling machine
- Or something else

Think about your main sprite and decide:

- What objects and `variables` might be needed?
- Will the `variables` change over time, for example hunger or tiredness increasing every few seconds?
- How will the user interact with the main sprite? This could be clicking on food to feed an animal, or chatting with a character to reduce their boredom.

Give your project a title that tells the user what to expect, and makes them want to try it out.

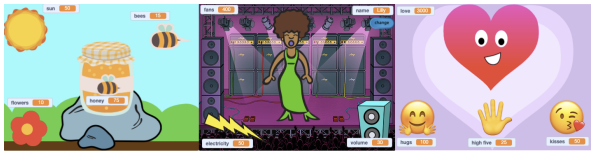
The example projects were: Bat simulator, Make a rainbow, Watermelon farmer, and Music Machine.

You can always update the title later if you think of a better one.

Save your project

Step 3 Build and test

Now it's time to make your project. Start small, and add more to your project if you have time.



Tip: Remember to test your project each time you add something. It is much easier to find and fix bugs before you make more changes.

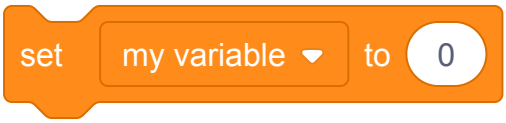
You will need to decide in what order to build your project. You could:

- Create one variable and allow the user to control it. Add animations, costumes, sound, effects, and conversation to make the sprite really come to life
- Create multiple variables with simple ways for the user to control them, and then add more effects later

Adding a sprite and then creating a `variable` is a great start.

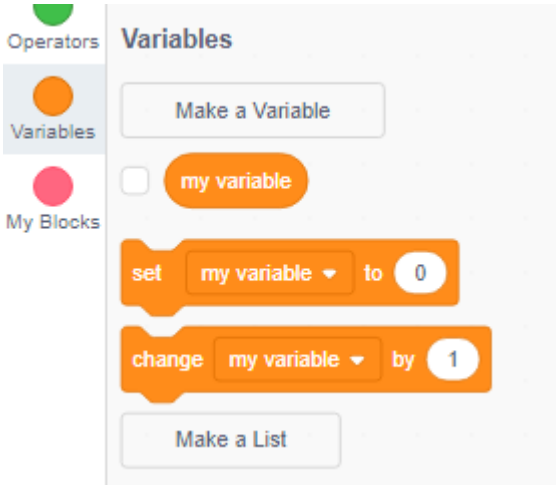
You have built up some really useful skills. Here is a reminder to help you make your project:

Using variables

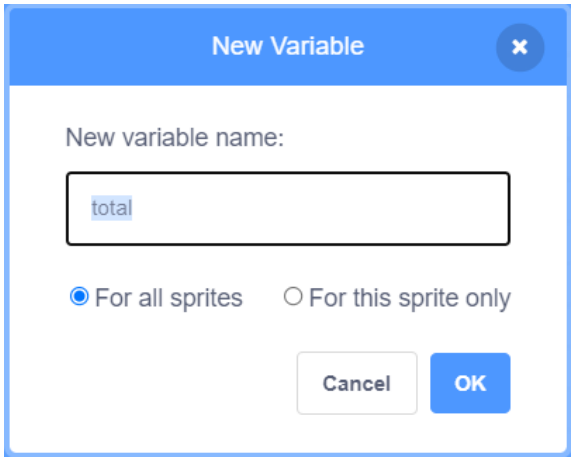


Make a variable and set a start value

Click on Variables in the Code 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.

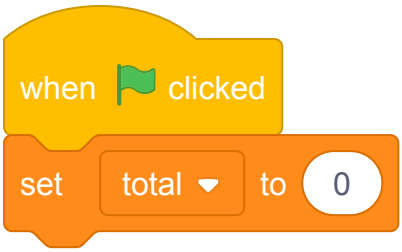


The variable will show on the Stage:

If you want to hide the variable on the Stage, uncheck the box next to the variable in the `Variables` blocks menu.

Setting a start value

If your variable should have the same starting value every time your project is run, then add a script to set it:



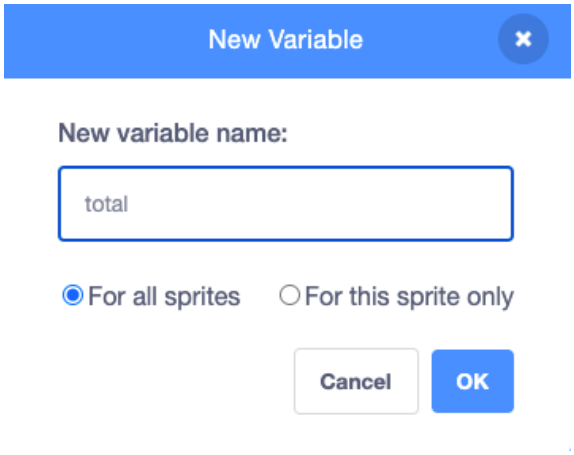
Set a variable with a button

Add a sprite to act as a button.

Tip: any sprite could be a button but there are already some button sprites in Scratch that you can use.

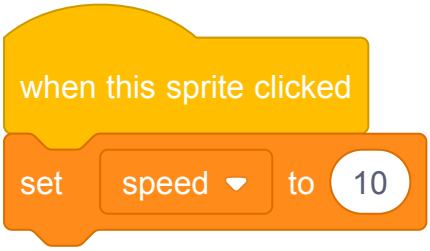
Click on the `Variables` Blocks menu and select the Make a Variable button.

Give the `variable` a name that is easy to recognise.

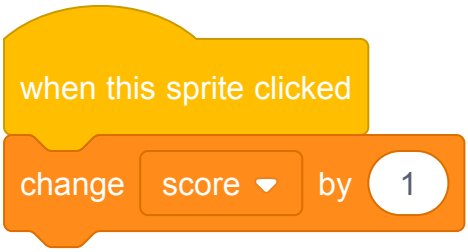


You will need to add code to your button sprite to update the `variable`. You could:

- Use the button to `set` the `variable` to a new value.



- Use the button to `change` the existing value in a `variable` by a new amount.



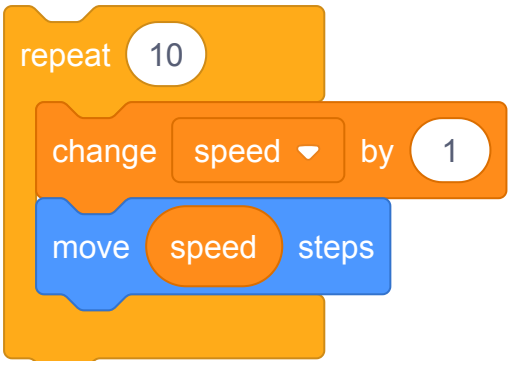
- Use the button to `ask` a question and `set` the `variable` to the `answer`.

Change a variable in a loop

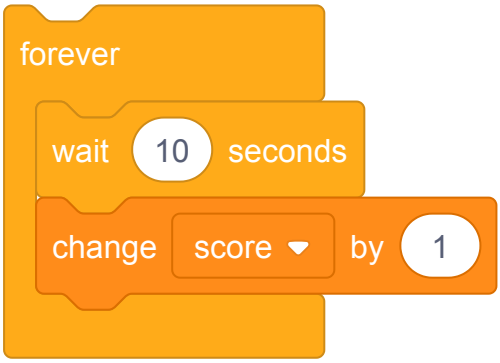
Loops can be used to `repeat` code blocks a set number of times, `repeat until` a condition is met, or run `forever`.

Inserting a `change` block into a loop will `change` your `variable` each tim the loop runs.

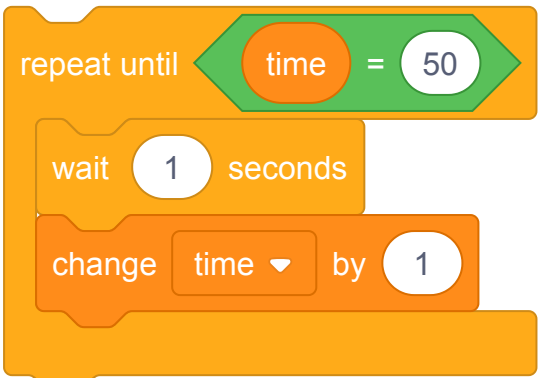
The code below would accelerate a sprite gradually:



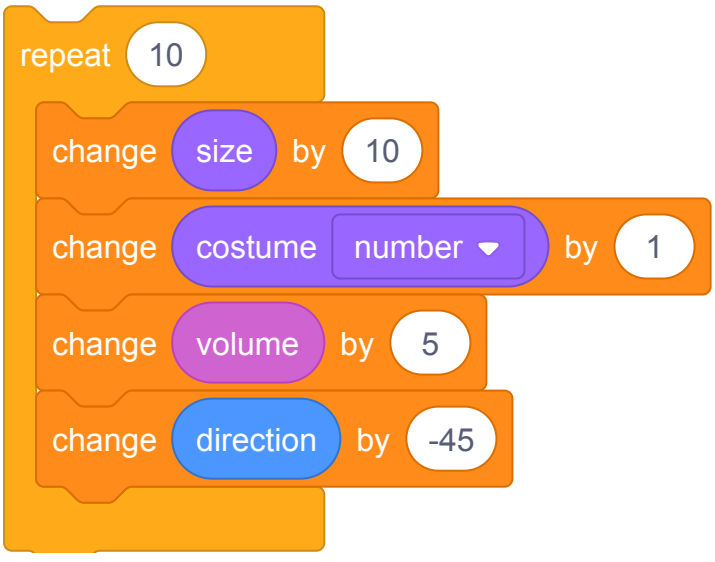
The code below would increase a player's score the longer they played the game:



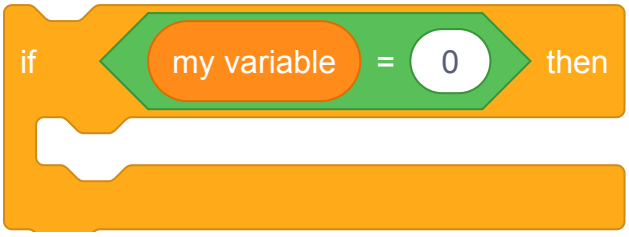
The code below would keep running the loop adding `1` to the `time` variable until `time` `=` `50`.



You can also use built in Scratch variables, such as `size`, `costume number`, `volume` and `direction`:



Checking conditions

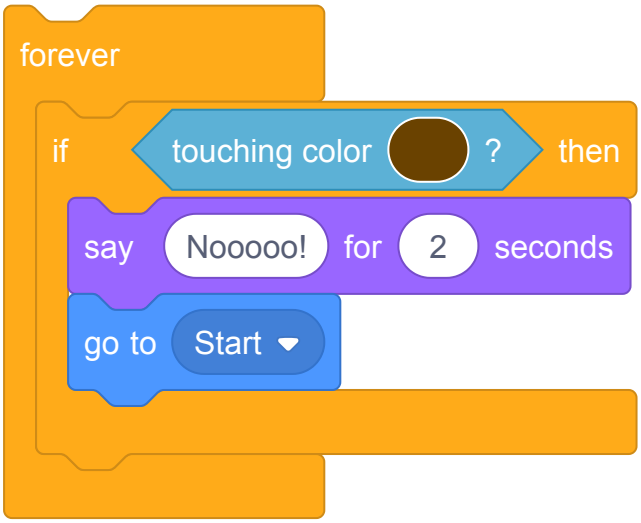


Checking a condition with if inside a forever loop

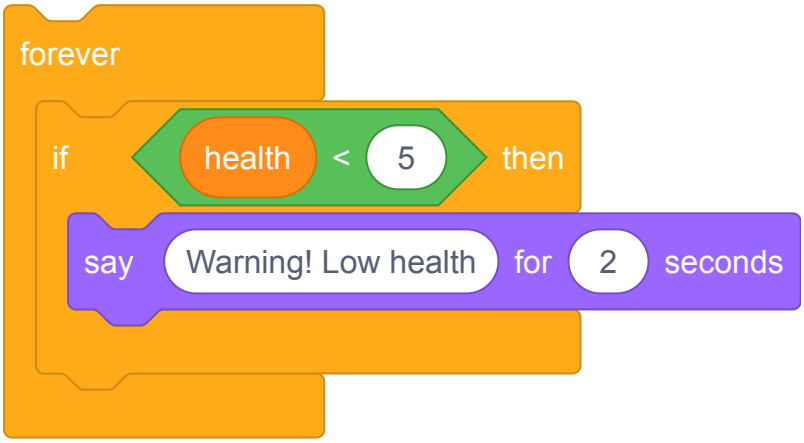
In Scratch it's often useful to run code blocks every time a condition becomes true.

You can do this by placing an `if` block inside a `forever` block. You will need to trigger the script, for example with a `when flag clicked` or `when I receieve` block.

You can check for important conditions in a game:



Or check use an `operator` with a `variable` value.

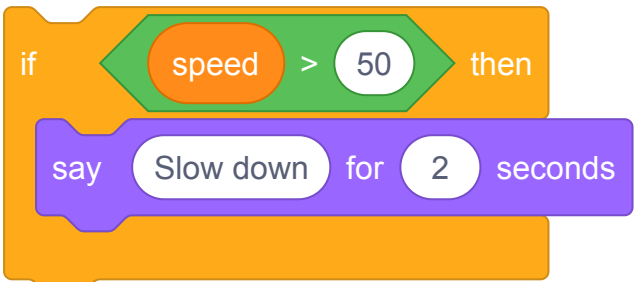


Using operators to check conditions

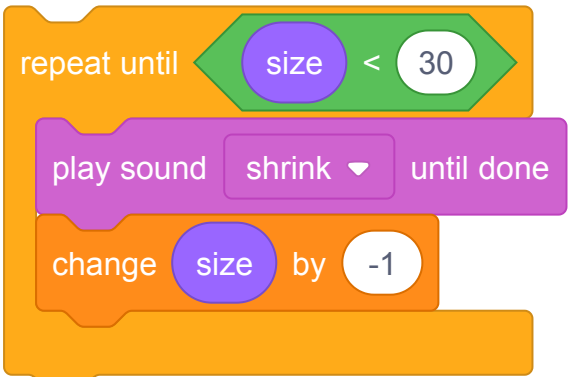
There are a number of hexagon shaped `operator` blocks that can be used to check conditions in `if`, `if...else`, `wait until`, and `repeat until` blocks to make a decision.

The `>`, `<`, or `=` operators check the relationship between two values or `variables`:

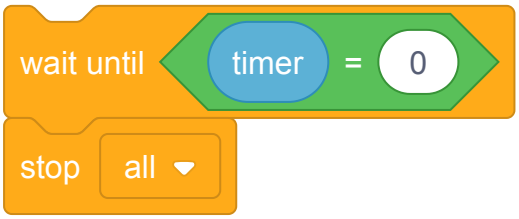
- Greater than:



- Less than:

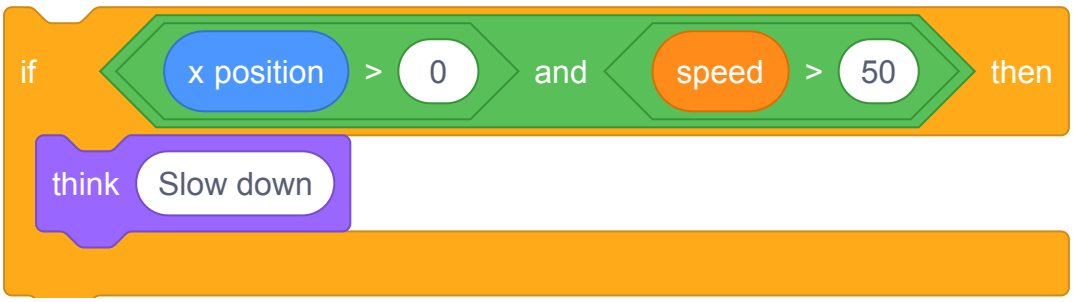


- Equal to:

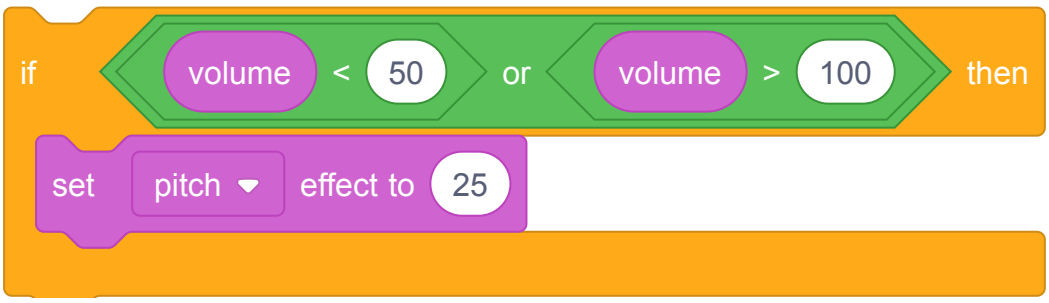


The `and`, `or`, and `not` operators make logical decisions if:

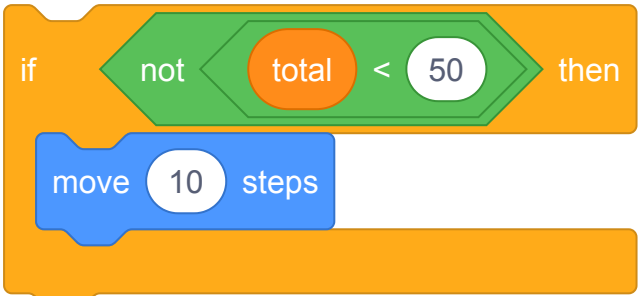
- `and` - both conditions are true:



- `or` - either condition is true:



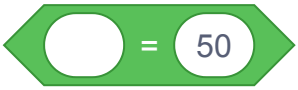
- `not` - the condition is false:



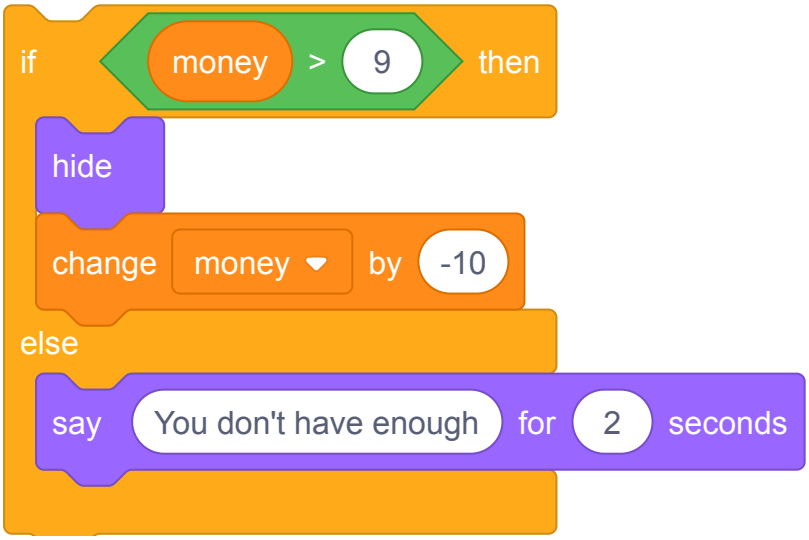
Using if...then and if...then...else blocks

The blocks inside an `if...then` block will only run if the condition in the hexagonal input is true.

There are lots of hexagonal shaped condition blocks in Scratch, including blocks in the `Sensing` and `Operators` blocks menus.



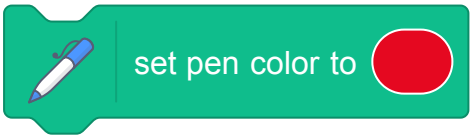
If you want to run different blocks when the condition is false then use an `if...then...else` block instead:



You can build more complex checks by 'nesting' `if...then` and `if...then...else` blocks one inside the other.

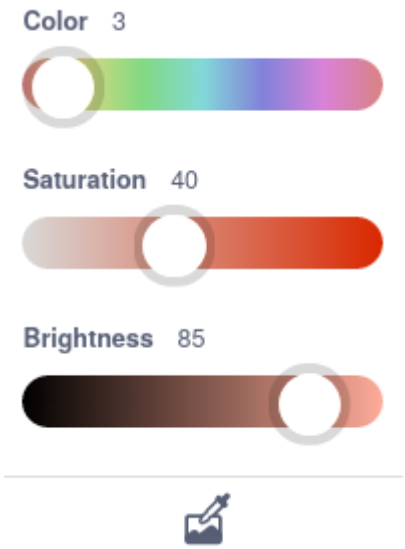
Set colour input with eyedropper

Some blocks in Scratch allow you to choose a colour.

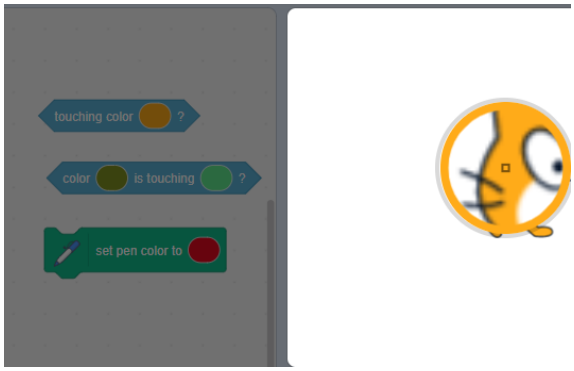


You can choose a colour to match a colour that appears on the Stage.

Click on the colour input to open the colour picker and then click on the eyedropper at the bottom.

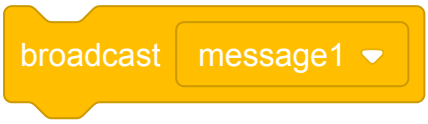


Move the mouse pointer over to the Stage and move around until you have selected the colour you want and then click (or tap) to select the colour.



The colour in the block input will change to match the colour you chose. Click in the Code area to close the colour picker.

Broadcasting and receiving messages



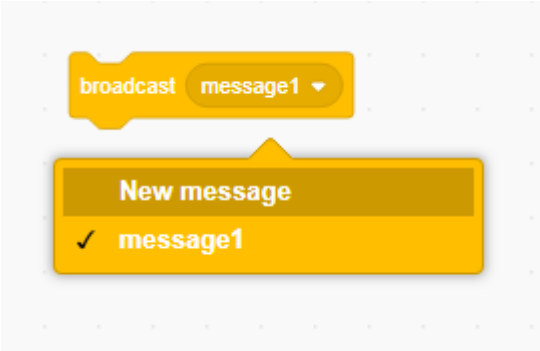
Broadcast a message in Scratch

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.

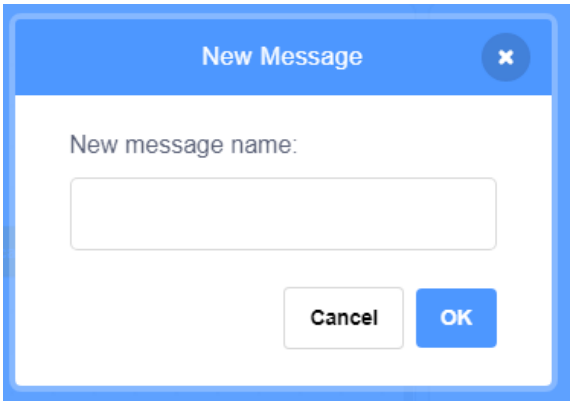
Broadcasting spells: Use the magic wand to click on the buttons and cast spells. What does each spell do to the characters? See inside (https://scratch.mit.edu/projects/518413238/editor)

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.

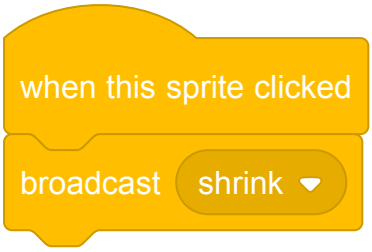


- Then type your message



Send a broadcast

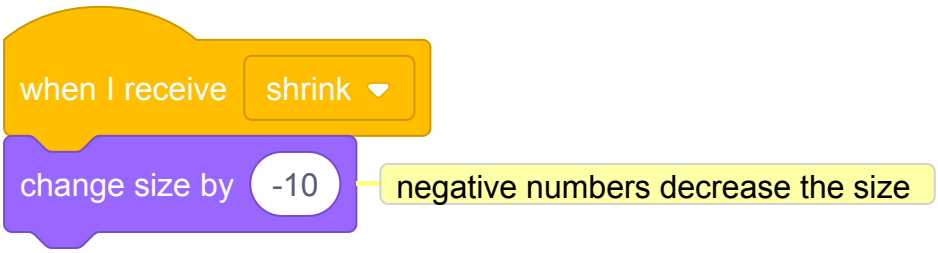
You can decide when to `broadcast` your message. For example:

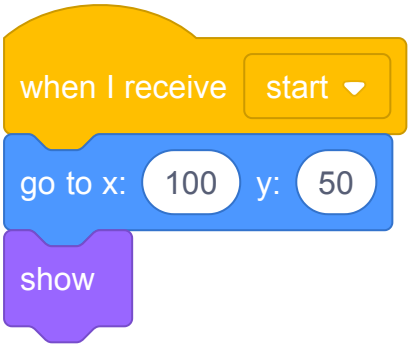


Receive a broadcast

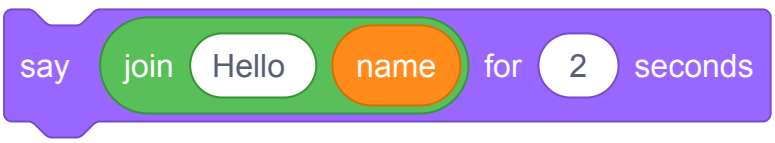
Sprite can react to a `broadcast` by using a `when I receive` block. Multiple sprites can respond when they receive the same message.

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





Working with text:



Use Emojis in text

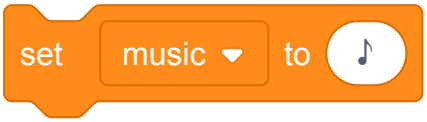
You can use the emoji keyboard to add use emojis anywhere you can type text in Scratch.

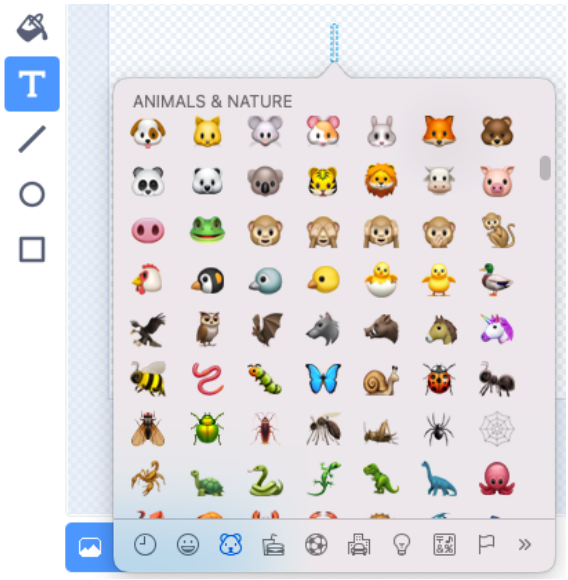
Instead of typing text use the emoji keyboard shortcut for your Operating System:

- Windows - Windows key + '.'
- MacOS - ctrl + cmd + space

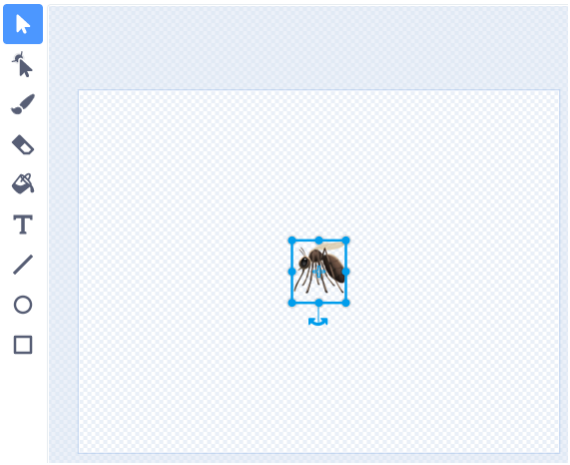
- Linux - ctrl + ‘.’

You can use emojis in the value of a variables:





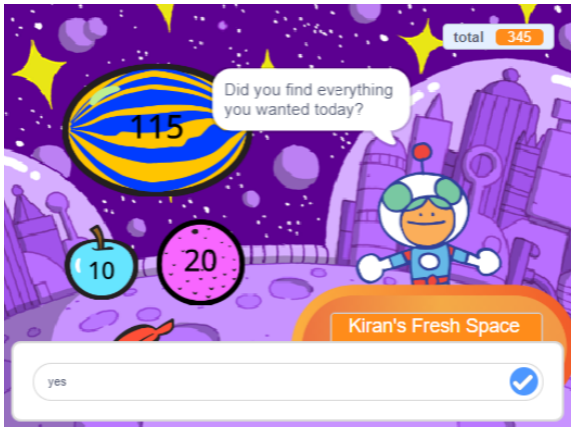
Or, in the Text tool in the Paint editor.



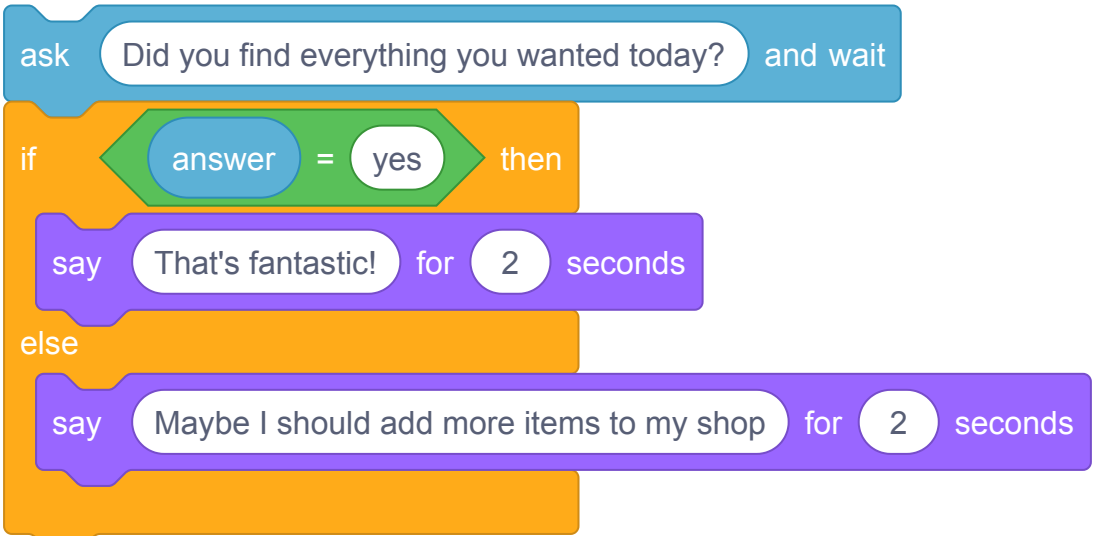
Tip: Emojis can look different on different computers so they might not look the same on a tablet and a desktop computer. Some emojis aren’t available on some computers, but most modern computers will support them.

Chat using ask and answer

You can use the `ask` and `answer` blocks from the `Sensing` blocks menu to have a conversation.



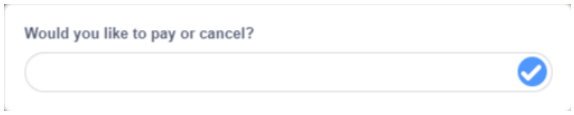
Add blocks to a script on the sprite that will `ask` a question:



Debug: Check that you have spelled the options correctly in your code and in your answer. It's okay if you use capital letters, so "Yes" and "YES" will match "yes".

Add multiple questions to create a chatbot or non-player character that you can talk to.

Tip: If you `hide` the sprite that asks a question, then question will appear inside the input box instead of as a speech bubble.

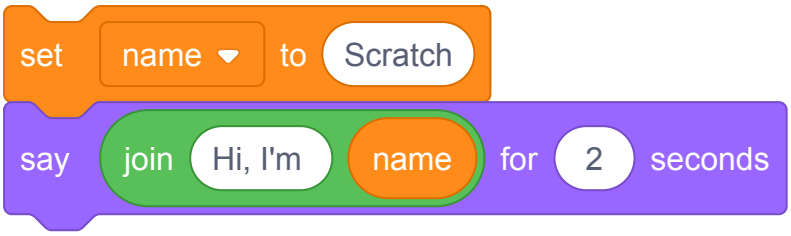


Join text and variables in Scratch

You can use the `join` block from the `Operators` block to join text and variables to make longer strings.

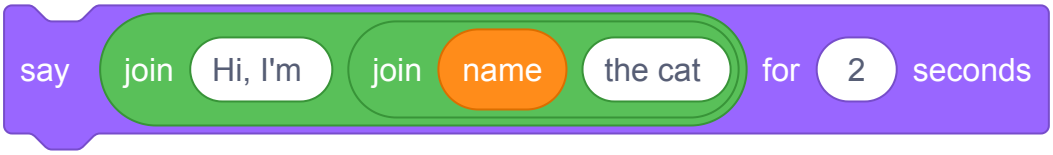
Drag a `join` block into the block where you want to use it:

Replace the text with the text you want to use, or drag in a variable:



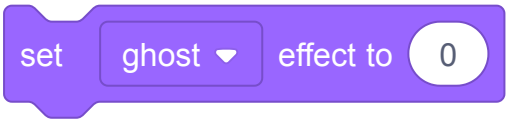
Tip: The `join` block doesn't add spaces so you will need to type them.

You can drag a `join` block inside another `join` to create longer text strings:



Notice the 'space' at the end of `Hi, I'm` and the beginning of `the cat`.

Backdrops, movement, and graphic effects

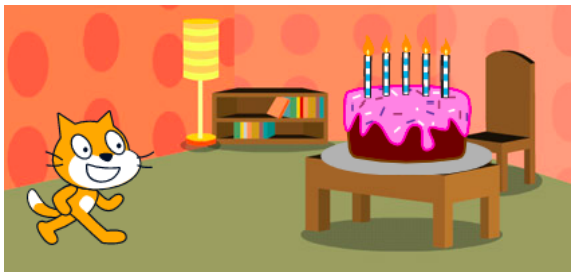


Glide to an object and back

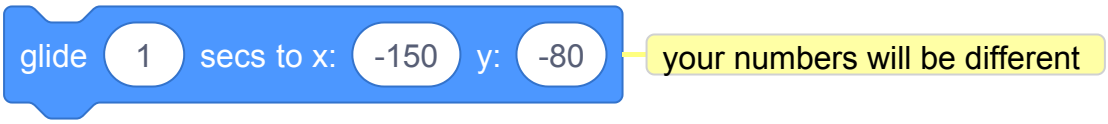
The `glide` blocks in Scratch can be used to move a sprite across the Stage.

A sprite can `glide` to a specific point (coordinates), a `random postion`, the `mouse pointer`, or to another sprite.

Positon your sprites in their starting points then select the sprite that is going to glide:

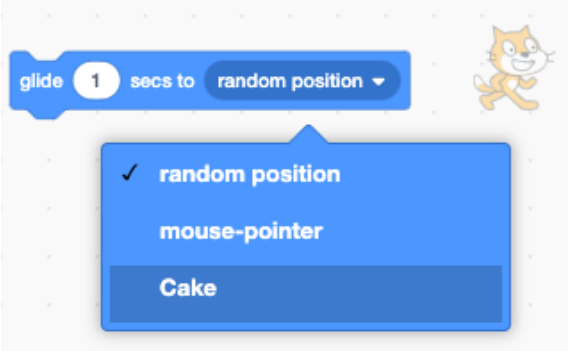


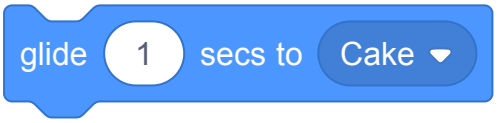
Drag a `glide (1) secs to x: y:` block into the Code area but don't attach it to any other blocks yet. This block has the coordinates of the starting point and will be used later to make the sprite return:

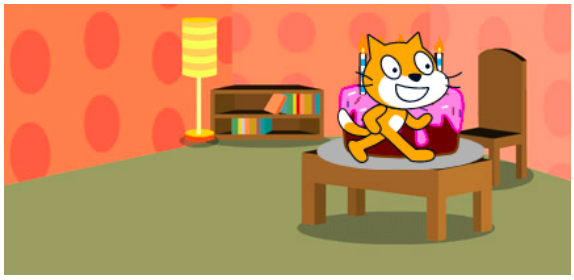


Drag a `glide (1) secs to (random position v)` block into the Code area and add it to your code at the point you want the sprite to move.

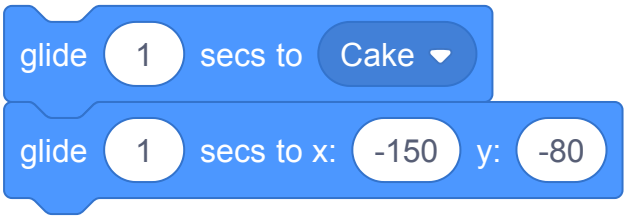
Click on the dropdown menu and select the name of the sprite you want to `glide` towards:







Finally, drag the `glide (1) secs to x: y:` block, that is already in the Code area, to your script to `glide` back to the start:

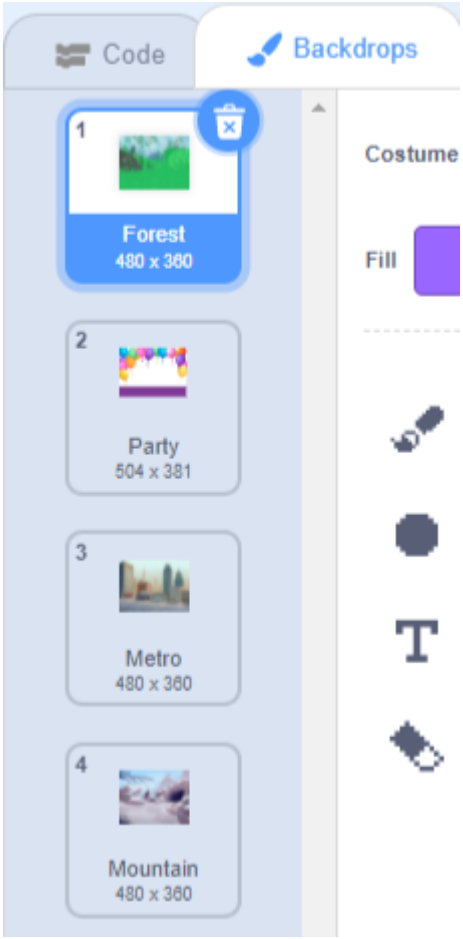


Use backdrops to create pages or levels

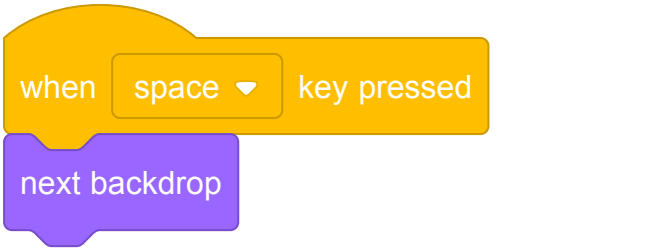
You can use backdrops in a Scratch project to create different pages or levels.

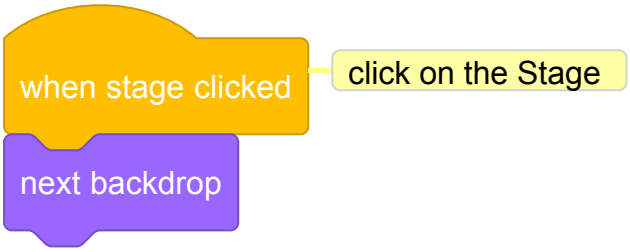
Changing backdrop tutorial: See inside (https://scratch.mit.edu/projects/498966268/editor)

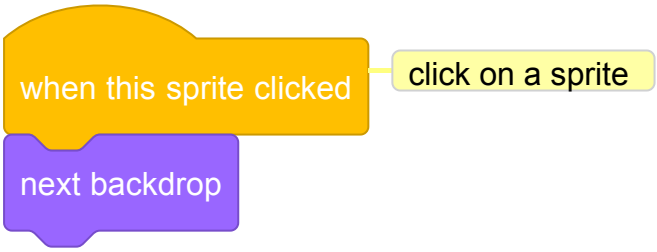
Click on the Stage pane and then the Backdrops tab to view the backdrops for your project. You can drag the backdrops to reorder them.

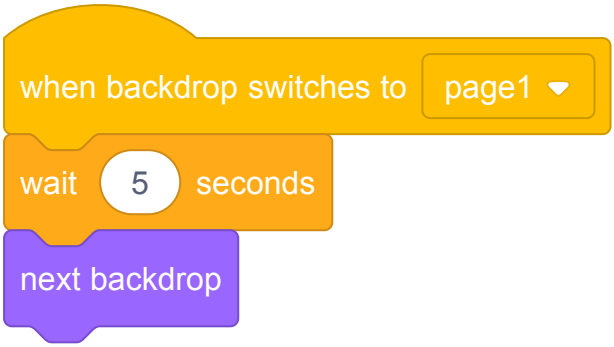


There are lots of ways to move to the `next backdrop`. Choose one that works for your project.





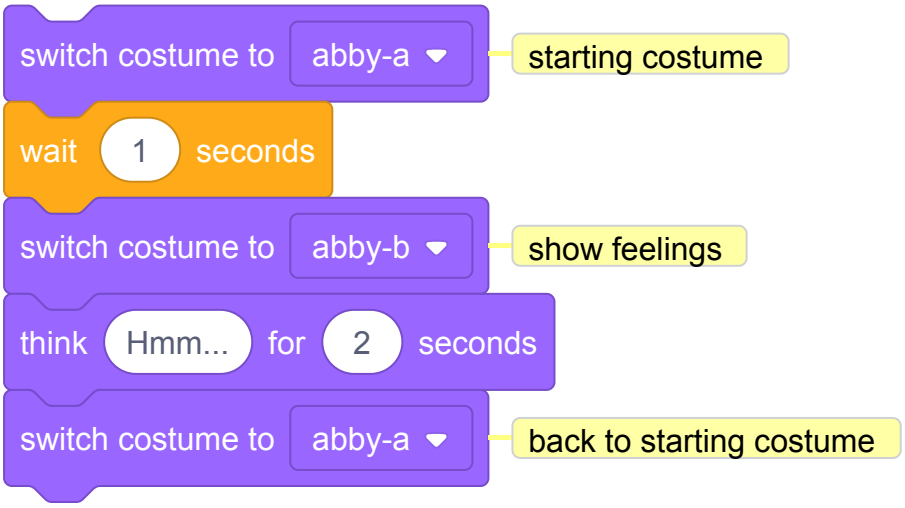




Show the mood of a sprite with costumes

Abby thinks "Hmm": See inside (https://scratch.mit.edu/projects/498767227/editor)

You can use `switch costume to` blocks before and after a `say for`, `think for`, `play sound`, or `wait` block to make your character show their feelings.

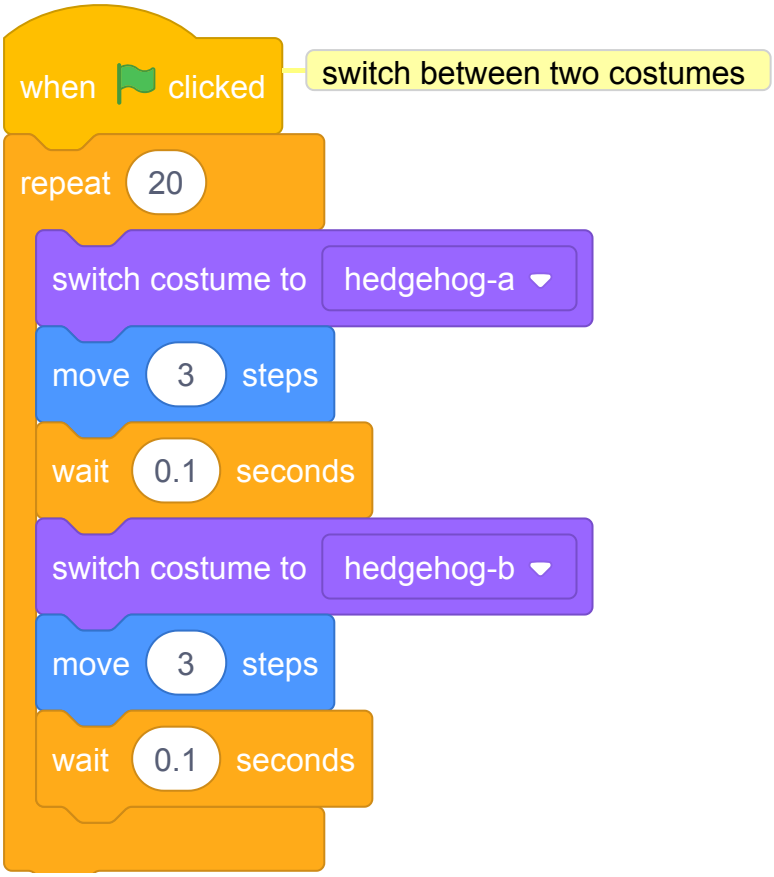


Tip: Make sure that you use a block that has a time value, not a `start sound`, `say`, or `think` block, otherwise, you will not see the costume change.

Animate sprite movement with costumes

Hedgehog walking: See inside (https://scratch.mit.edu/projects/499398615/editor)

You can use `switch costume to` and `move` blocks in a `repeat` loop to animate a moving character. Change the time in the `wait` block to change the speed.



Tip: If you want to use all the costumes that a sprite has, you can just use the `next costume` block in a loop.

Tip: Increase the number of steps in each `move` block to make the sprite go faster. Change the number in the `repeat` loop to adjust the distance.

Tip: To make the sprite `move` backwards, you can use negative numbers, for example, `move` `-3` `steps`. Or, you can use a `point in direction` `-90` block to change the sprite's direction before the sprite moves (`-90` points to the left).

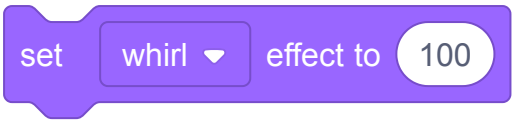
Use graphic effects

Rooster effects: See inside (https://scratch.mit.edu/projects/435730522/editor)

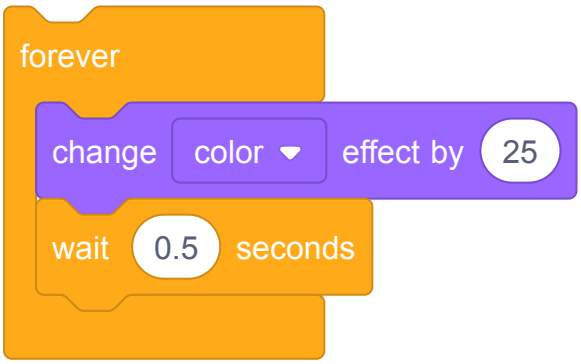
The `set color effect to` and `change color effect by` blocks both have drop-down menus in which you can choose from a range of different graphic effects that can be used to change your sprite's appearance:

- `color`: from `0` to `199` (bigger numbers will wrap around, so `200` is the same as `0`)
- `fisheye`: `0` means no effect, bigger numbers cause a bigger 'fisheye' effect, and negative numbers cause a reverse 'fisheye' effect
- `whirl`: `0` means no effect, big numbers make a big whirl to the left, and big negative numbers make a big whirl to the right
- `pixelate`: `0` means no effect, and bigger numbers create more pixels
- `mosaic`: `0` means no effect, and bigger or negative numbers affect the number of copies
- `brightness`: `0` means no effect, numbers up to `100` make the sprite lighter, and negative numbers down to `-100` make the sprite darker
- `ghost`: `0` means no effect, and numbers up to `100` make the sprite more transparent

Try to `set` the different effect values to see what each one does. Explore how different effect changes make your sprite look.

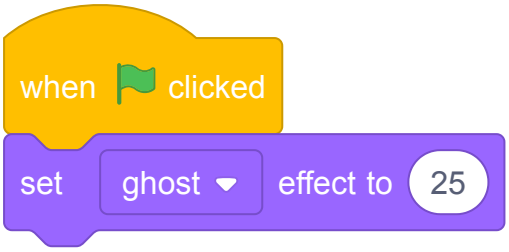


Tip: A `color effect` of `225` is the same as a `color effect` of `25`, so you can keep changing the colour. For other graphic effects, no other changes will be made after you reach the maximum or minimum number for the effect.



Use the `clear graphic effects` block to start again. Clicking on the green flag also clears all graphic effects.

To set a graphic effect for a sprite when the project is started, place a `set graphic effect to` block under a `when green flag clicked` block:



Tip: You can also `set` and `change` graphic effects for the Stage.

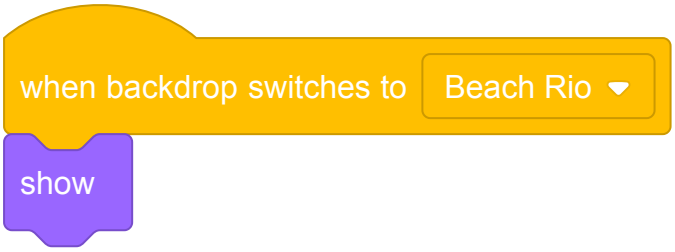
Show and hide sprites on different backdrops

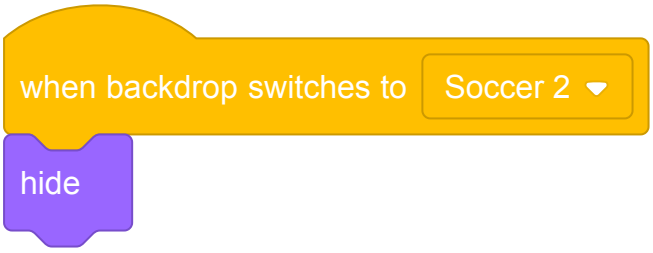
When you are creating a book, animation, presentation, or game with levels, some sprites should only show on some backdrops.

Show and hide sprites on different backdrops: See inside (https://scratch.mit.edu/projects/499876704/editor)
Click on the Stage or press the **Space** key to switch to the next backdrop.

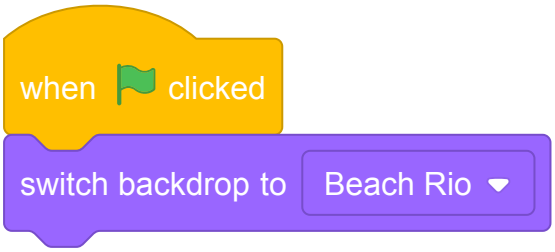
Use `show` and `hide` blocks with the `when backdrop switches to` block to make sprites only show on the backdrops where they belong.

The Beachball sprite:





Set the `backdrop` `when green flag clicked` to make sure that sprites show or hide correctly on the first backdrop:

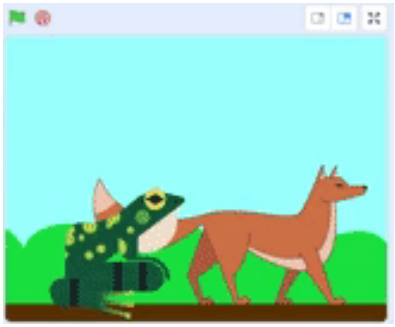


Tip: You can also make sprites `show` and `hide` `when green flag clicked`.

Position sprites with layers

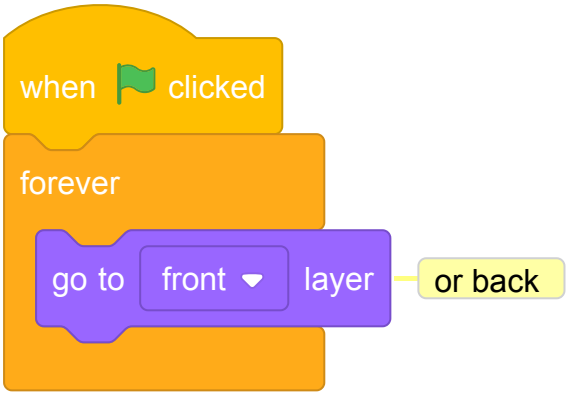
There are two ways to position your sprites into layers.

You can drag a sprite on the Stage to move it to the front layer:

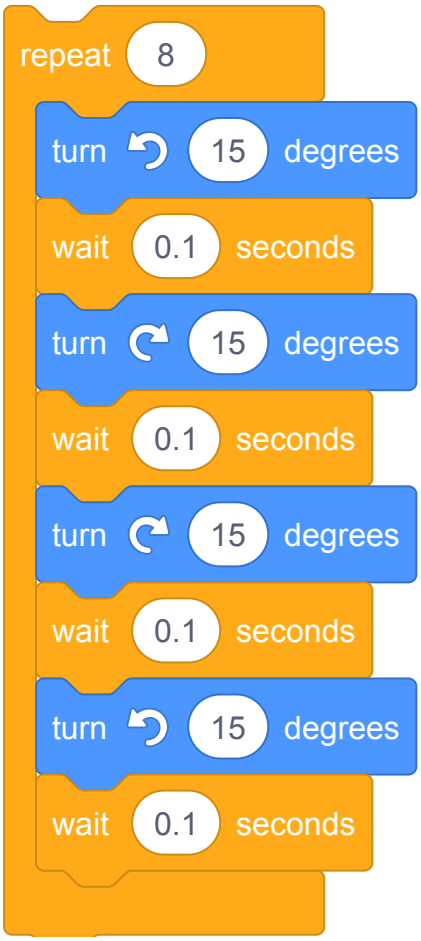


Alternatively, you can use a `go to front layer` or `go to back layer` block to position a sprite.

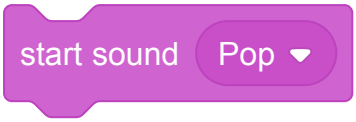
If you want a sprite to always stay at the `front` or `back`, use a `forever` loop to make the sprite move back to the correct layer if you accidentally move it:



Jiggle a sprite

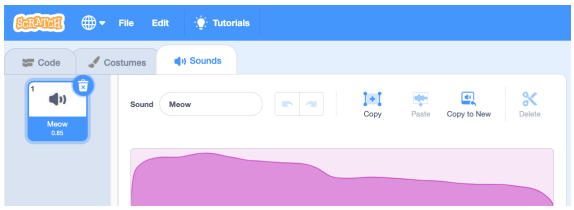


Sound:

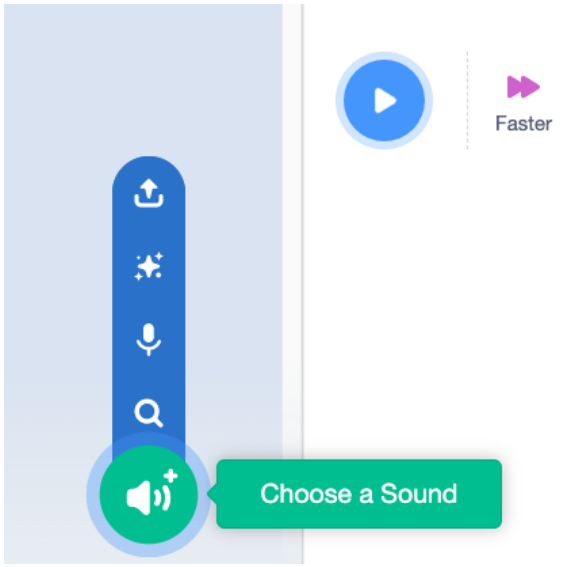


Add a sound to a sprite or the Stage

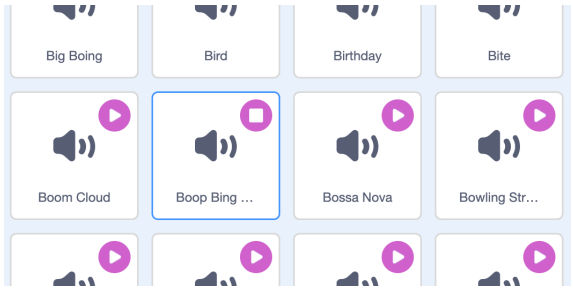
Select the sprite that you want to have the new sound, then select the Sounds tab. Each sprite starts with a default sound:



Scratch has a library of sounds that you can add to your sprites. Click on the Choose a Sound icon to open the Sound Library:

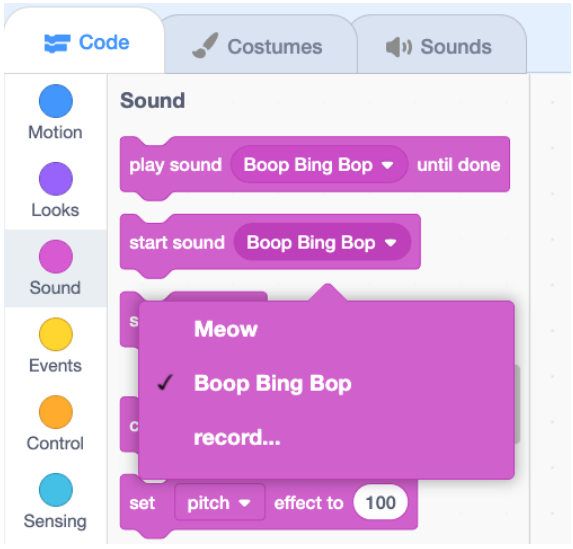


To play a sound, hold your mouse cursor (or your finger, if you are using a tablet) over the Play icon:



Click on any sound to add it to your sprite. You will be taken straight back to the Sounds tab and you will be able to see the sound that you have just added:

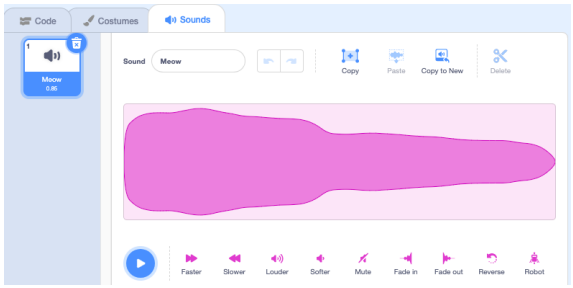
If you switch to the Code tab and look at the Sound blocks menu, you will be able to select the new sound:



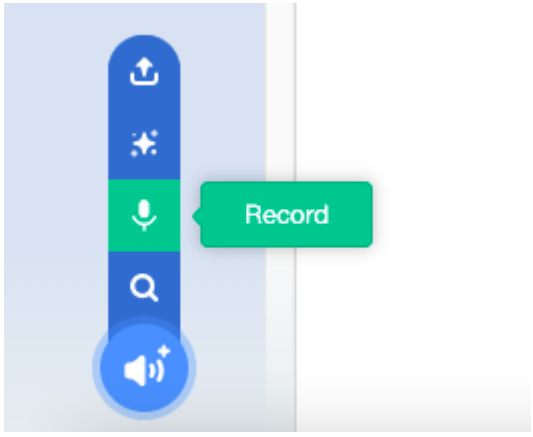
Tip: You can also add sounds to the Stage.

Record a sound

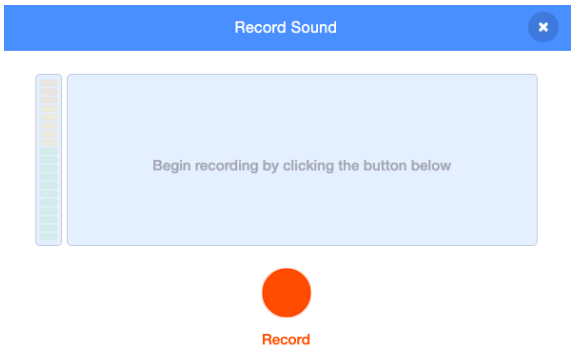
Select the sprite that you want to have the new recorded sound, then select the Sounds tab:



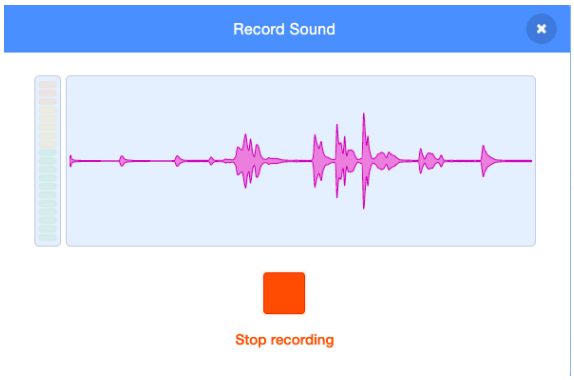
Go to the Choose a Sound menu and select the Record option:



When you are ready, click the Record button to start recording your sound:

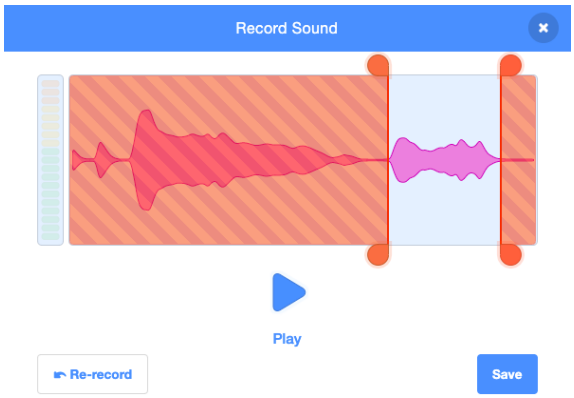


Click the Stop recording button to stop recording your sound:

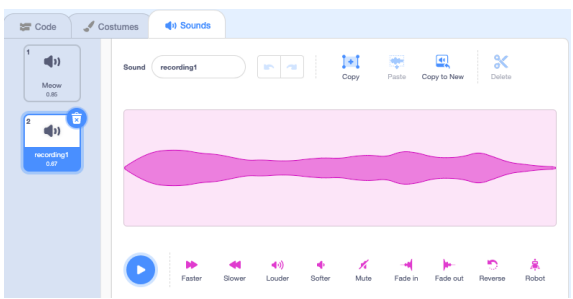


Your new recording will be shown. You can Re-record your sound if you are not happy with it.

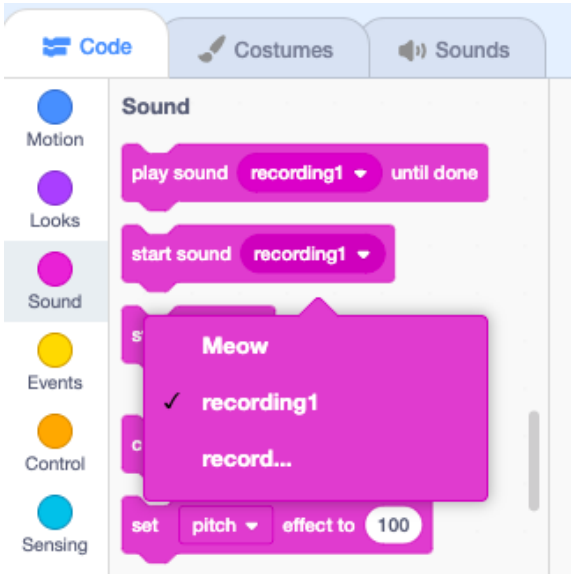
Drag the orange circles to crop your sound; the part of the sound with a blue background (between the orange circles) will be the part that is kept:



When you are happy with your recording, click the Save button. You will be taken straight back to the Sounds tab and you will be able to see the sound that you have just added:



If you switch to the Code tab and look at the Sound blocks menu, you will be able to select the new sound:

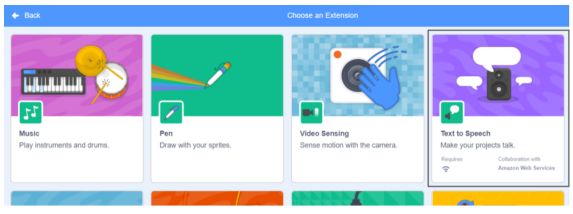


Make a sprite talk with the Text to Speech extension

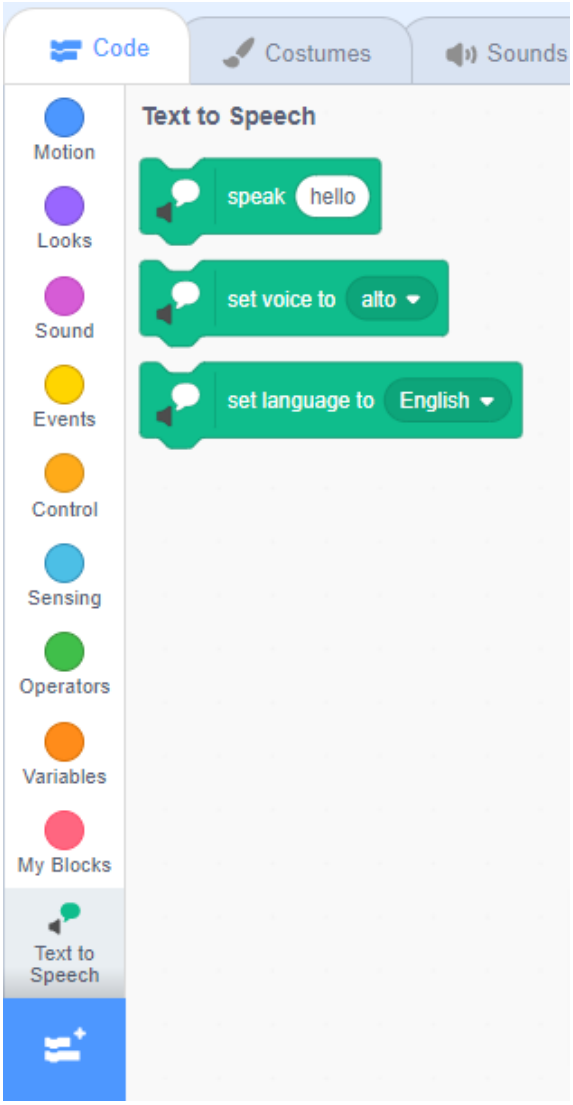
Pico and Giga talk with the speech extension: See inside [(https://scratch.mit.edu/projects/499373708/editor)](https://scratch.mit.edu/projects/499373708/editor)

Click on Add Extension:

Choose Text to Speech:



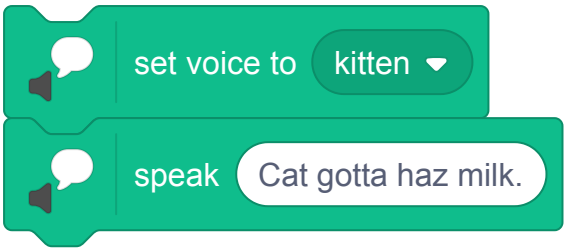
You will get a new `Text to Speech` blocks menu:



You can use the blocks in the `Text to Speech` blocks menu to make your sprites talk out loud.

You can make a sprite talk out loud when clicked:

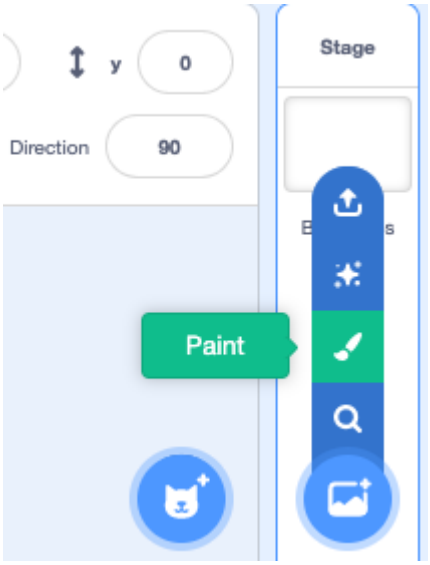
You can even give your sprite a kitten voice!



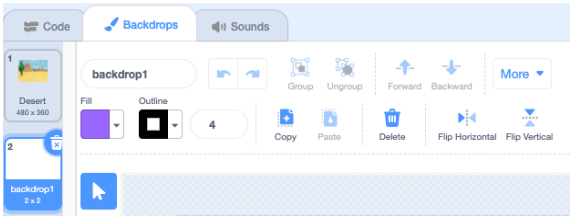
Paint editor — backdrops and costumes

Create a new backdrop in the Paint editor

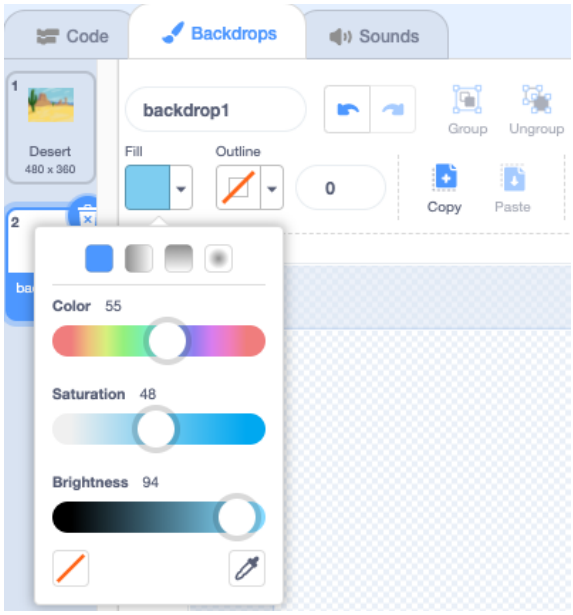
Go to the Choose a Backdrop menu and click on Paint:

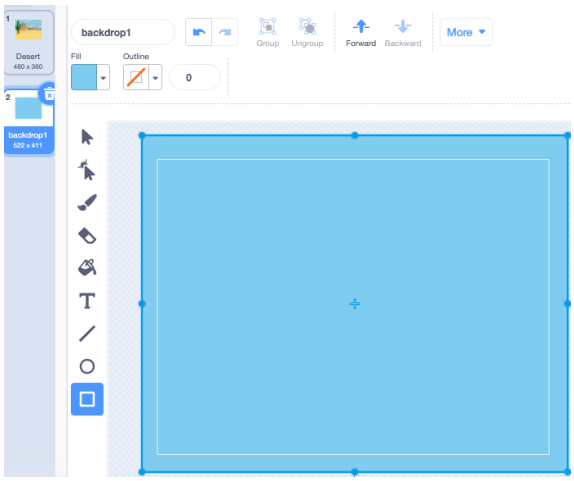


You will be taken to the Paint editor, where the new backdrop will be highlighted in the list. If you have other backdrops in your project, you will also see them in the list.

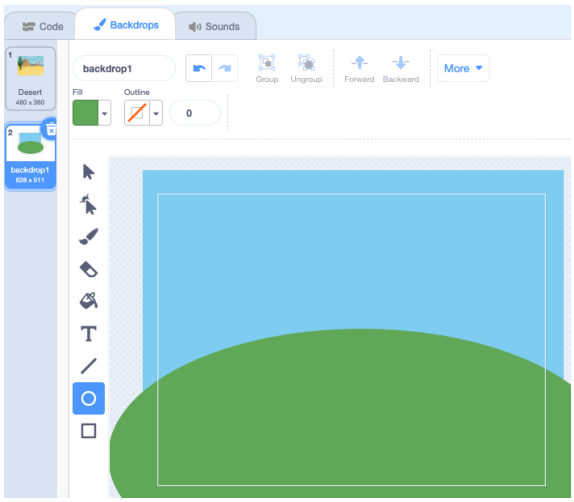


To set the main colour of the backdrop, click on the Rectangle tool, then use the Fill colour chooser to select a colour, then drag the shape over the full backdrop canvas:

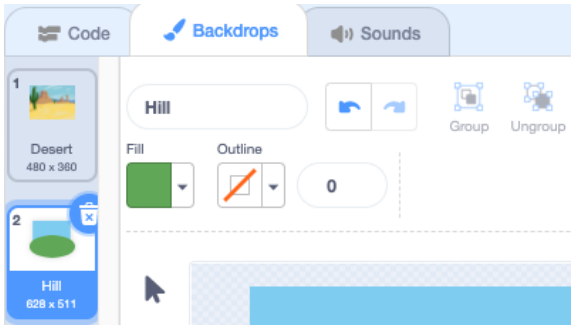




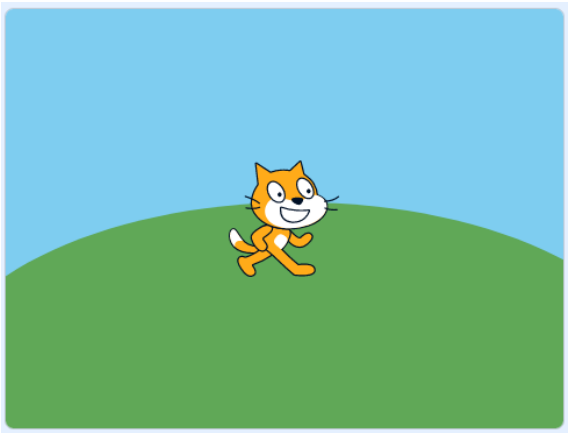
If you want to add more details to your backdrop, you can use the Rectangle tool, Circle tool, or Brush tool, or a combination of all three!



When you have finished, make sure that you give your new backdrop a name that makes sense:



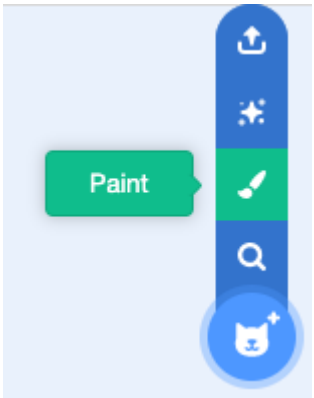
Your new backdrop will be shown on the Stage and will be available to use in Looks blocks.

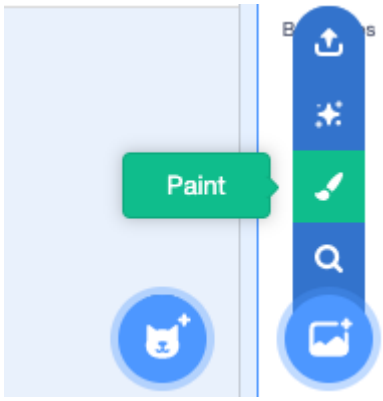


Create backdrops and sprites using shapes

You can create backdrops and costumes for sprites in the Paint editor, using just shapes.

Go to the Choose a Sprite or Choose a Backdrop menu and choose the Paint option:





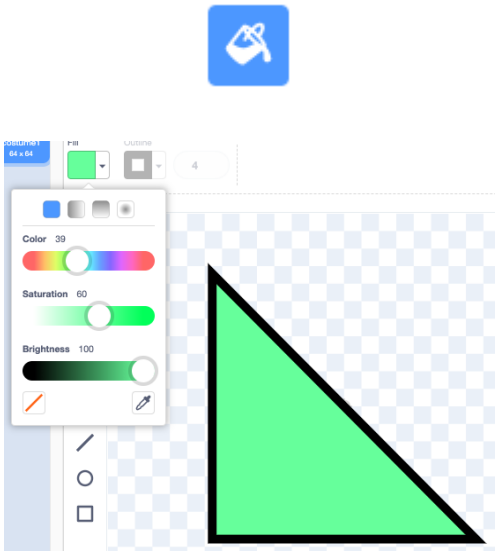
Select which tool(s) to use to make the shape(s) that you want:

- Circle: Click on the Circle tool to draw a circle. Press and hold the `Shift` key on your keyboard to draw a perfect circle.

- Rectangle: Click on the Rectangle tool to draw a rectangle. Press and hold the `Shift` key to draw a square.

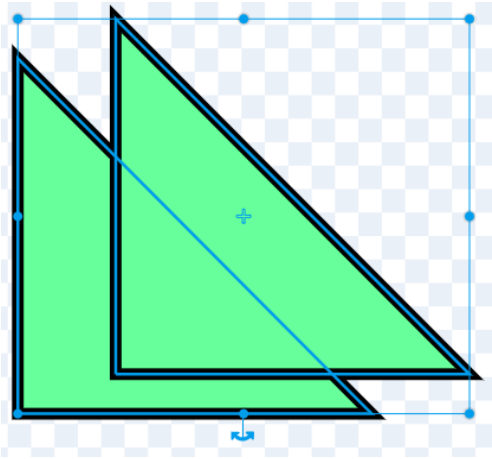
- Triangle: Use the Rectangle tool to draw a rectangle or a square. Click on the Reshape tool and select the corner that you want to remove. Click on the Delete tool to turn your shape into a triangle.

You can use the Fill tool to change the colour of a shape:



You may need to use the Forward and Backward tools to move your shapes forward or backward so that they are positioned correctly within your image:

You can select all the shapes and Group them together so that you can adjust them or move them as one shape:



Here is an example of a sprite created with the Circle and Rectangle tools:

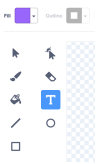
Pig: See inside [(https://scratch.mit.edu/projects/495903163/editor)](https://scratch.mit.edu/projects/495903163/editor)

Remember to name the costumes and backdrops that you create in the Paint editor.

Use the Text tool

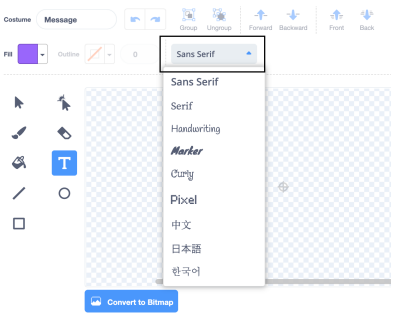
Click on the Fill colour chooser and select the colour that you want for your text:

Select the Text tool:

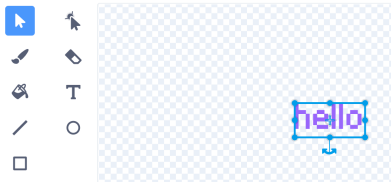


Click on the Paint editor and start typing.

Click on the Font drop-down menu and select the font that you want:



If you want to resize your text, click on the Select (Arrow) tool and select the text, then drag the corner handles to resize the text:

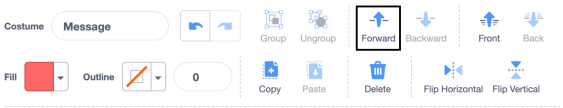


If you want to change the Fill colour of your text, use the Fill colour chooser to select a colour, then select the Fill (Bucket) tool and hold your mouse cursor over the text (or if you are using a tablet, tap the text). The text will automatically change colour. Click on the text to make the change:

If you want each word in a message to be a different colour, size, and font, follow the process above for each word in your message.

Position your text and/or group of words with the crosshair in the centre of the Paint editor:

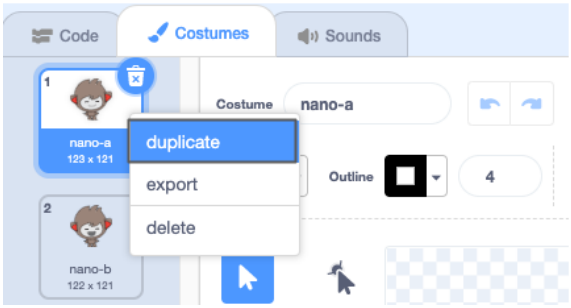
To layer your words, use the Forward and Backward tools:

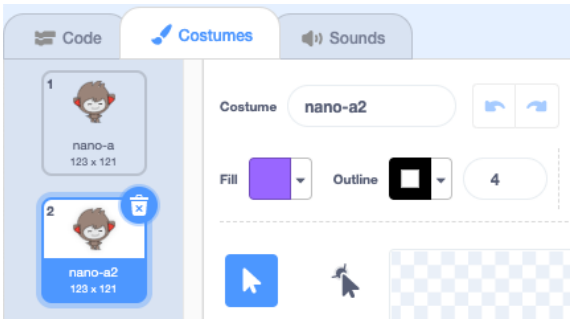


Copy parts between sprite costumes

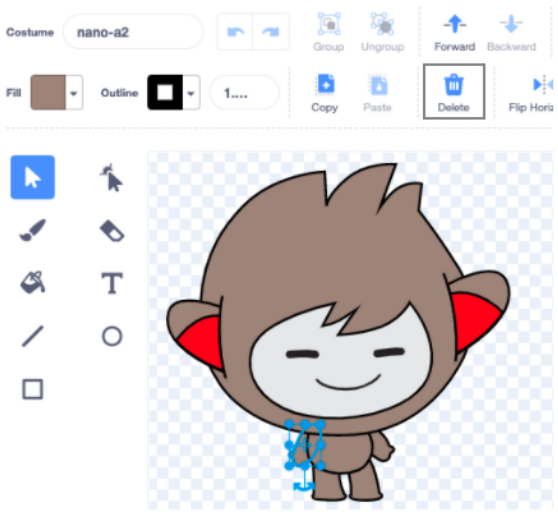
Click on the Costumes tab for your sprite.

Tip: Duplicate the costume that you want to edit — so that you can still use the original costume if you need to. To do this, right-click (or on a tablet, tap and hold) on the costume and choose duplicate. You will now have a copy of the costume:

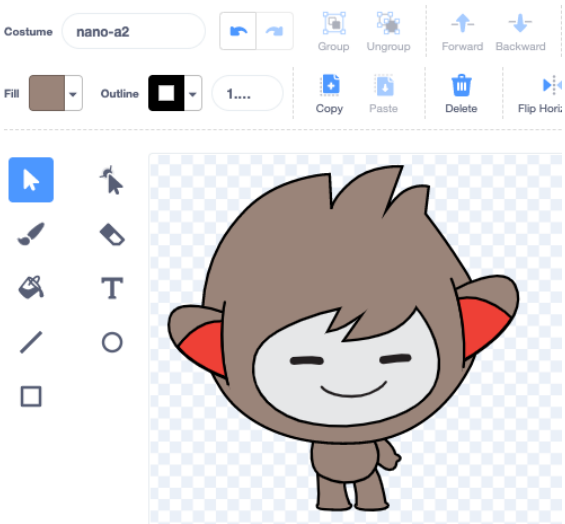




To remove any part of the costume that you do not need any more, click on the part to select it, then click on Delete:

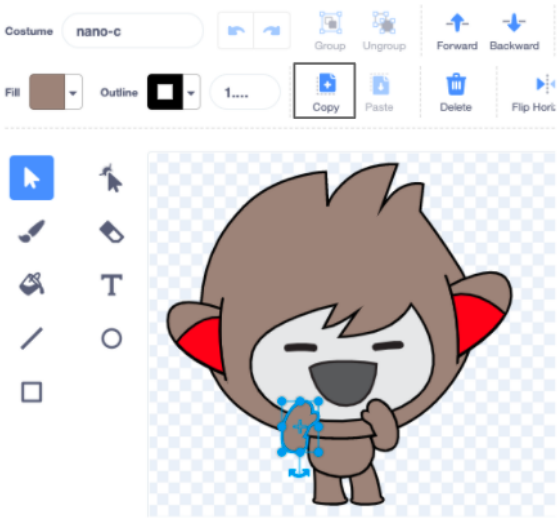


The duplicated costume with parts removed should look something like this:

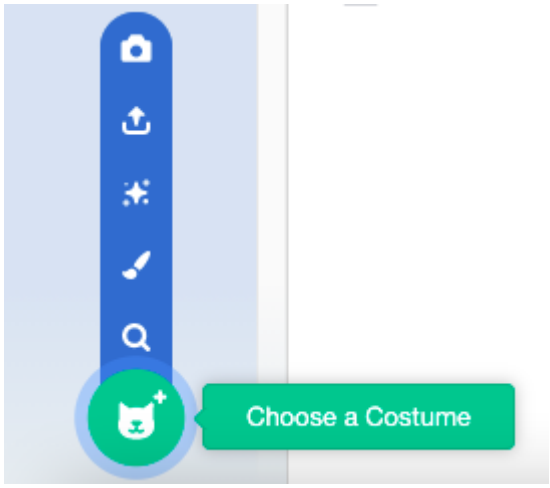


Tip: If you make a mistake in the Paint editor, you can click on Undo:

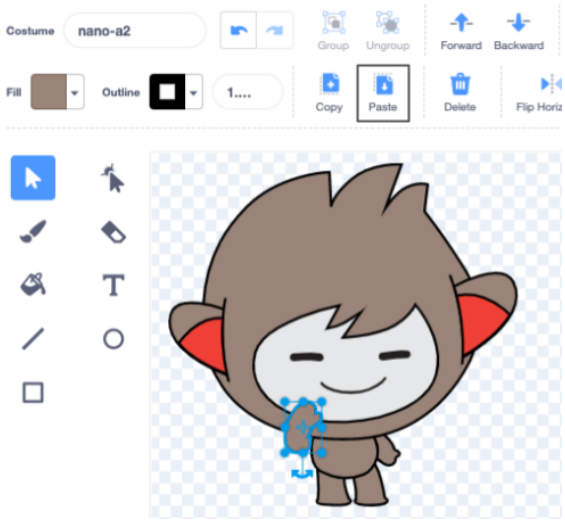
Go to the costume with the part that you want to add and click on the part that you need, then click on Copy:



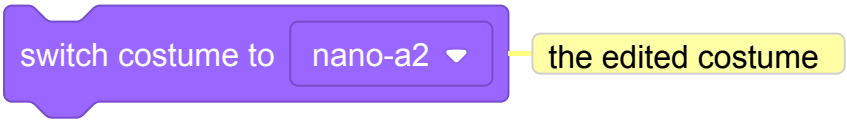
If you want to add a part from a costume that is not already shown in the Costumes tab, you need to add the costume to your sprite first. Click on the Choose a Costume icon, then find the costume that you want and click on it to add it to your sprite:



When you have copied the part that you need, go back to the duplicated costume and click on Paste. The duplicated costume should now look something like this:

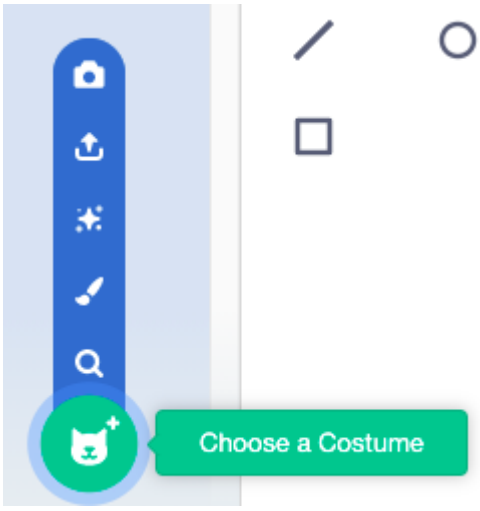


Now, switch to the Code tab. You will be able to use the new costume in your code blocks:



Add a costume to a sprite

Click on the Costumes tab and then on Choose a Costume to add any costume to the sprite from the Costume Library:



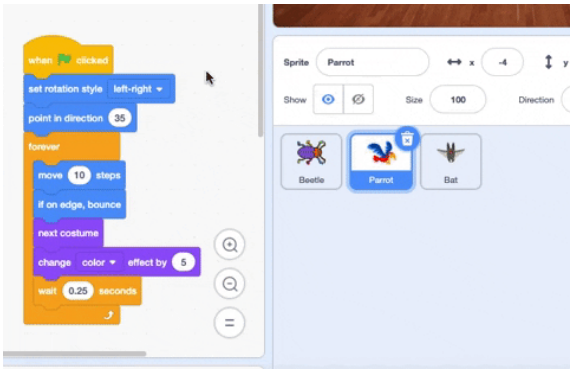
You will need to position and resize the added costume in the Paint editor, to match the sprite's other costumes.

Tip: If you position a sprite on the Stage and then change its costume, the sprite might appear to 'jump' or change size. You will need to position and resize the costumes in the Paint editor so that they all appear in the right position on the Stage.

Scratch editor

Copy code from one sprite to another

You can copy code from one sprite to another sprite in the Sprite list:



Both sprites will have the code blocks that you have copied. If you are moving the code from one sprite to the other, you can delete the code from the first sprite after you have copied it to the other sprite.

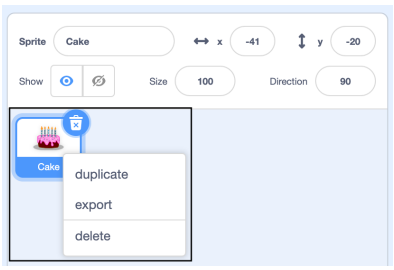
Run a Scratch project in full-screen mode

To run your project in full-screen mode in Scratch, go to the area above the Stage and click on the icon with four arrows that point outwards. This is the Full Screen Control icon:

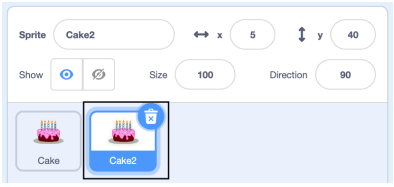
To exit full-screen mode, click on the Full Screen Control icon again. It will have four arrows that point inwards.

Duplicate a sprite

Right-click (or on a tablet, tap and hold) on your first sprite in the Sprite list below the Stage:

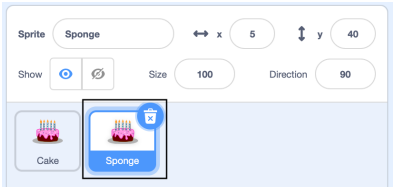


Select duplicate. This will create a copy of your first sprite, with the suffix “2”:



Rename your sprite:

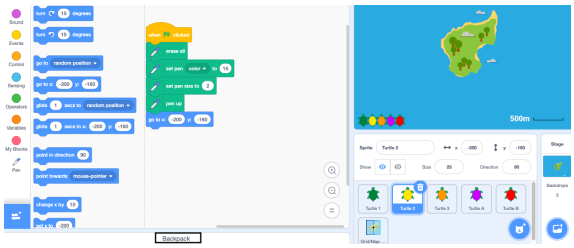
Your sprite's name will change in the Sprite list:



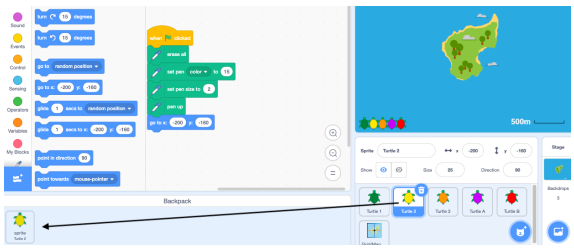
Your second sprite has exactly the same code as your first sprite. Do not run the program until you have started to change the second sprite's code — you might not see the second sprite because it might be positioned underneath the first sprite.

Using the Scratch Backpack

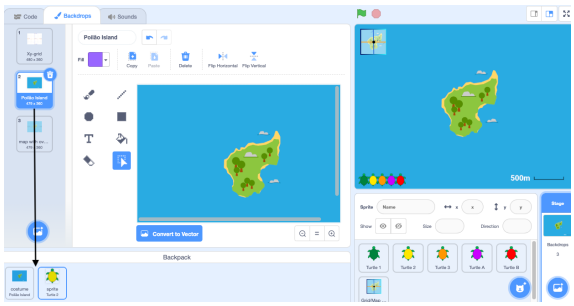
- You can use your Scratch Backpack to store costumes, sprites, sounds, and scripts that you want to copy between projects.
- You can only access your own Backpack, and you must be logged in to your Scratch account to use it.
- To open your Backpack, click on the Backpack tab at the bottom of the screen.



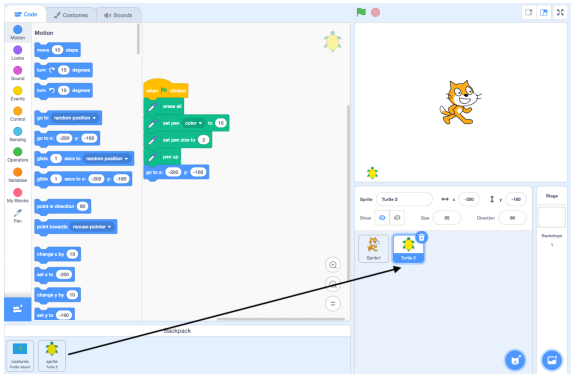
- To add a sprite to your Backpack, drag the sprite from the Sprite list to the Backpack. This will store the full sprite in your Backpack, including all of its costumes, sounds, and scripts.



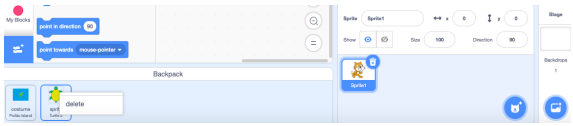
- To add a backdrop to your Backpack, select the Stage pane and click on the Backdrops tab, then choose the backdrop that you want and drag it to your Backpack.



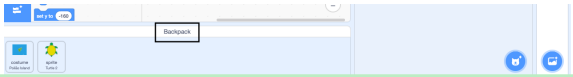
- To use an item in your Backpack in another project, open the project and drag the item from the Backpack to the correct pane or tab.



- To delete an item in your Backpack, find the item in the Backpack tab, then right-click (or on a tablet, tap and hold) on the item and select delete.



- You can hide your Backpack when you are not using it. To do this, click on the Backpack tab at the bottom of the screen.



Test: Show someone else your project and get their feedback. Do you want to make any changes to your game?

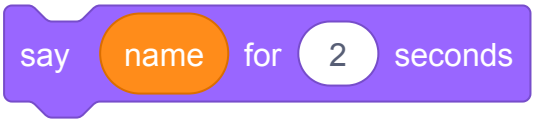
Debug: You might find some bugs in your project that you need to fix. Here are some common bugs.

Variables aren't updating correctly

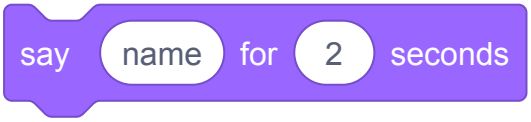
One common mistake is to confuse the `change` and `set` blocks.

- `set` replaces the value of a variable with a new value.
- `change` adds a number to a variable. If you `change by` a positive number, the variable value gets bigger. If you `change by` a negative number, then variable value gets smaller.

Another common problem is typing the name of a variable instead of dragging the variable from the `Variables` blocks menu. Variables should be orange:



not:

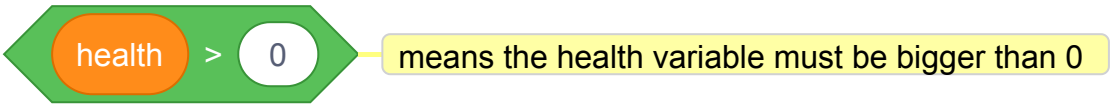


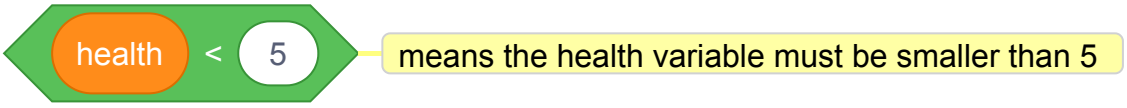
A change only happens once instead of forever

Make sure you have placed code blocks that need to keep running inside a `forever` block. It's really common to forget to do this!

Number comparisons conditions aren't working

Are you sure you have used the `>` (greater than) and `<` (less than) operators the right way around?





Tip: The number that must be bigger goes on the wider (bigger) side of the operator symbol.

Nothing happens when I broadcast a message

Make sure you have a matching `when I receive` block that does something when you `broadcast` a message. Check that the message names match.

You might find a bug not listed here. Can you figure out how to fix it?

If you get stuck, try reading your code out loud or explaining the problem to a friend. You might spot the problem.

We love hearing about your bugs and how you fixed them. Use the feedback button at the bottom of this page if you found a different bug in your project.

Save your project

Step 4 Checklist

Did you meet the project brief? Think about your project and go through the checklist below. Check off the ones that apply to your project.

Does your project:

- [ ] Use at least one `variable` to keep track of what the main sprite needs
- [ ] Have a way for the variable(s) to change automatically
- [ ] Give the user a way to improve the variables to give the main sprite what it needs
- [ ] Use `if` blocks to control when things happen
- [ ] Use `broadcast` blocks to communicate between other sprites and the main sprite

Your simulation could also:

- [ ] Have an educational message, such as kindness or keeping crops healthy
- [ ] Alert the user when levels are too high or low
- [ ] Allow the user to chat with their sprite, or change its name

Reflect on how you made your simulation to help you in your future projects:

- [ ] How did you get your ideas?
- [ ] What cool new thing(s) did you learn?

Now you are the creator of a simulation!

Take a moment to celebrate what you have made.

> Computer simulations can allow us to experience situations that we couldn't in real life. There are computer simulations that allow surgeons to practice operations. There are simulations that allow space engineers to predict what will happen when they launch a rocket, and make changes to the variables to see what happens. And there are game simulations that allow you to take care of a pet, person, family, or job.

Where will you take your new powers? What simulation will you make next?

Step 5 Upgrade your project

If you have time, you can upgrade your project. You might have ideas to add more already, or you might want to go back to the first step and look at other projects again for more inspiration.

You could:

- Add more `variables` that represent things your main sprite wants or needs.
- Check more conditions using `if` statements, and make your sprite do different things depending on whether the condition is true or false
- Add more sprites that can be used to interact with your main sprite, such as buttons or toys.
- Add animations using `graphic effects` or costumes
- Add more `sound effects`

Get inspiration

Each example project in the Introduction (.) has a 'See Inside' link for you to open the project in Scratch, look at the code to get ideas, and see how they work.

Take a look at some This sprite needs you projects created by community members in the Raspberry Pi Foundation's studio of This sprite needs you - Community projects See inside (https://scratch.mit.edu/studios/29722869/).

Small improvements

When you make a Scratch project, there are lots of ways to make sure you've done a really good job.

> Marginal gains are small upgrades that don't make much difference on their own, but when added together, they make a big improvement to the quality of your work and speed you can make more changes.

Can you use marginal gains to change one thing now that will help you in future? Things to do:

- Name all sprites, costumes, sounds, variables, and messages sensibly. It's much easier to understand projects that use helpful names
- Fill in the Project Page with a good title, instructions, and notes, so that users know how to use your project, and where you got ideas from
- Add short comments that explain what your code is doing
- Make sure your scripts are tidy in the Code tab. You can right-click and choose Clean up Blocks or arrange the scripts in a way that makes sense to you. Try and put similar scripts together.

Save your project

Step 6 Share

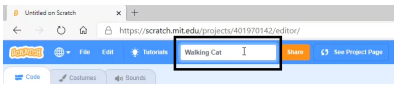
Now is the time to share your simulation so that users can try it.

If you have a Scratch account, you can share your project through Scratch. Check with an adult first. You can send a link to people you know, and the whole Scratch community will be able to find your project to try it out. Make sure you fill in the Instructions section, so that users know what they need to do to help your main sprite.

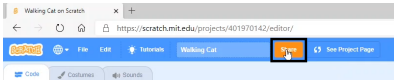
Share your Scratch project

Please make sure that you do not share any personal information about yourself when sharing your Scratch projects.

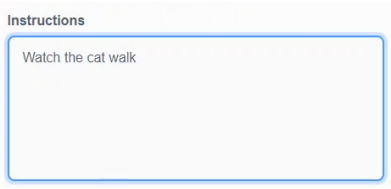
- Give your Scratch project a name.



- Click the Share button to make the project public.



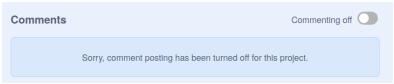
- If you like, you can add instructions in the Instructions box, to tell other people how to use your project.



- You can also fill in the Notes and Credits box: if you have made an original project, you can write some short comments, or if you have remixed a project, you can credit the original creator.

- Click the Copy Link button to get the link to your project. You can send this link to other people by email or text, or on social media.

Scratch provides the ability to comment on your own and other people's projects. If you do not want to allow people to comment on your project, you should turn off commenting. To turn off commenting, set the slider above the Comments box to Commenting off.

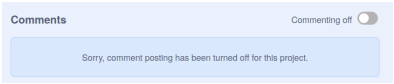


Why not invite your friends to create a project? Let them know how you had fun.

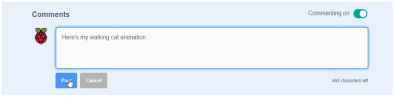
If you have a Scratch account, and have shared your project, keep an eye out for comments that others may make about your creation. Feedback is a great way to reflect on your work. Report and ignore any comments that are unhelpful or irrelevant.

Comments and feedback in Scratch

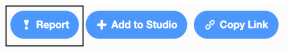
Scratch provides the ability to comment on your own and other people's projects. If you do not want to allow people to comment on your project, you should turn off commenting. To turn off commenting, go to the Project Page and set the slider above the Comments box to Commenting off:



If you are happy and feel safe to allow people to write comments on your project, you can leave the first comment:



If you think a comment or project is mean, insulting, too violent, or otherwise inappropriate, click the Report button to let the Scratch Team know about it. To report a comment, click the Report button above the comment. To report a project, click the Report button on the Project Page:



Read the Scratch Community Guidelines (https://scratch.mit.edu/community_guidelines) so that you know how you and others can maintain a friendly and creative community.

Inspire others with your project by submitting it to our Scratch studio!

To share your project with others who make the 'This sprite needs you' project, please complete this form. (https://form.raspberrypi.org/f/community-project-submissions)

Step 7 What next?

You have reached the end of the More Scratch (https://projects.raspberrypi.org/en/pathways/more-scratch) path! Try using Scratch to make more cool stuff.

To learn more skills, so you can bring all your ideas to life in Scratch, move on to our Further Scratch (https://projects.raspberrypi.org/en/pathways/further-scratch) path!

If you need a reminder of what you have learned, you can go to our 'Getting started with Scratch' guide (https://projects.raspberrypi.org/en/projects/getting-started-scratch).

Take part in Coolest Projects

Check out Coolest Projects (https://coolestprojects.org/), the world's leading technology showcase for young people! On the Coolest Projects website (https://coolestprojects.org/), you can find out when project registration is open, and get ready to register your project!

If you have made a Scratch project, you can register your project in the Scratch category when Coolest Projects project registration is open. Your project doesn't have to be finished — prototypes and works in progress are welcome too! When you have entered your project, your creation will be showcased in the Coolest Projects online gallery, for people all over the world to see! Join other young people in celebrating and recognising each other's achievements as a community.

You can use the Coolest Projects How to make a project (https://coolestprojects.org/2020/03/31/how-to-make-a-project-workbook-and-additional-resources/) resources to help you think of original project ideas.

You can also explore our other Scratch projects (https://projects.raspberrypi.org/en/projects?software%5B%5D=scratch&curriculum%5B%5D=%201) and try them out.