

Make me happy!

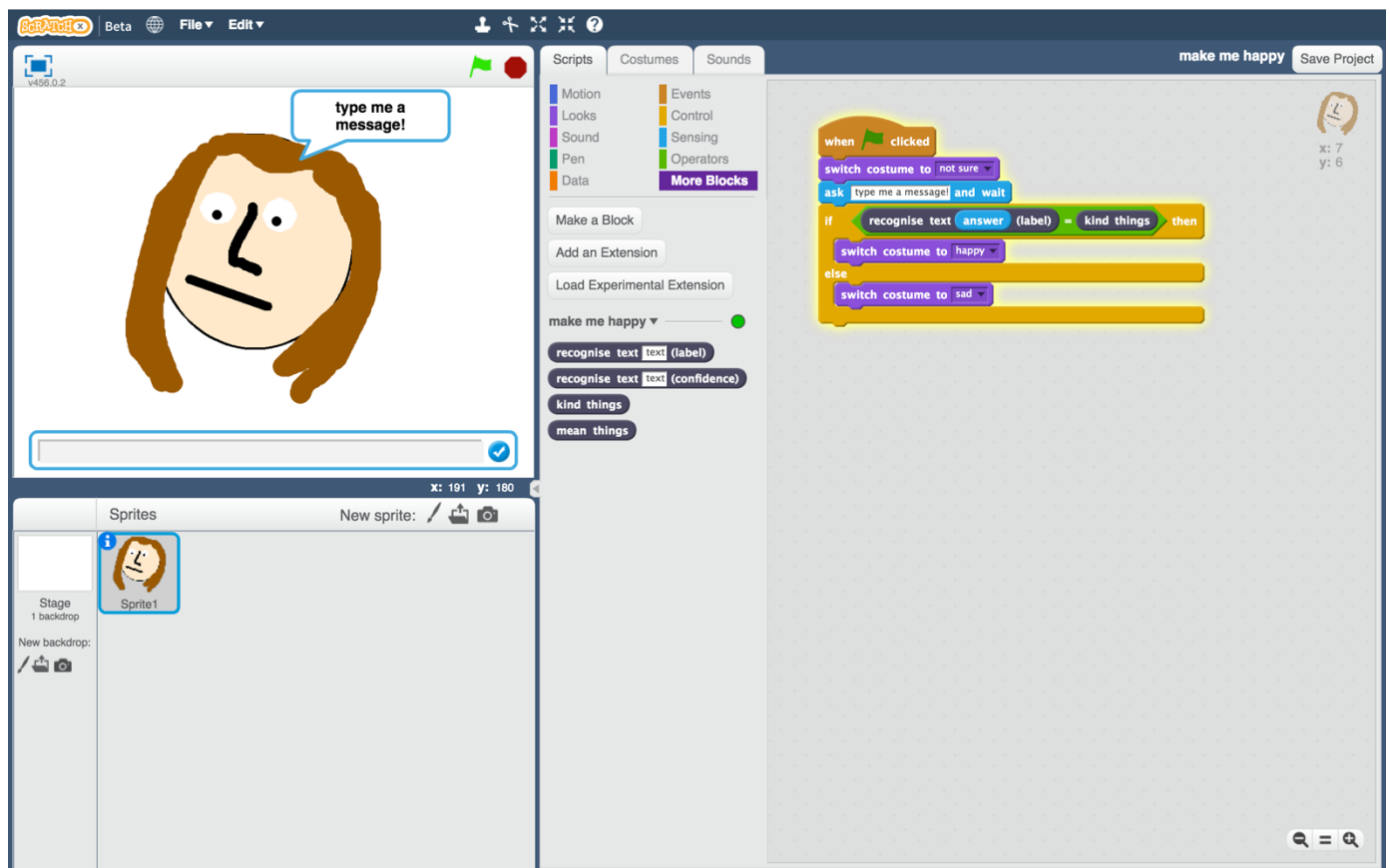
In this project you will make a character that reacts to what you say to it.

If you say kind things to it, it will look happy.

If you say mean things to it, it will look sad.

At first, you'll program a list of rules for what is kind and what is mean, and learn why that approach isn't very good.

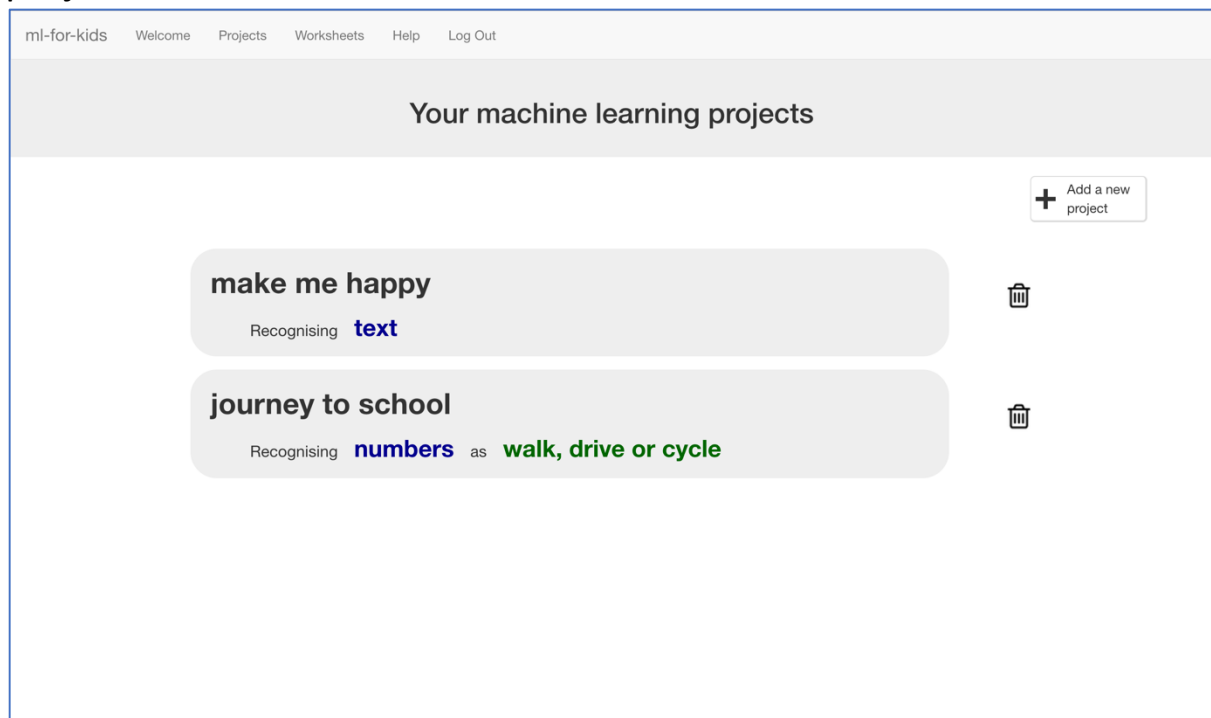
Next, you will teach the computer to recognise kind messages and mean messages by giving it examples of each.



1. Go to <https://machinelearningforkids.co.uk/> in a web browser
2. Click on **“Get started”**
3. Click on **“Log In”** and type in your username and password
If you don't have a username, ask your teacher or group leader to create one for you.
If you can't remember your username or password, ask your teacher or group leader to reset it for you.
4. Click on **“Projects”** on the top menu bar
5. Click on the **“+ Add a new project”** button.
6. Name your project **“make me happy”** and set it to learn how to recognise **“text”**

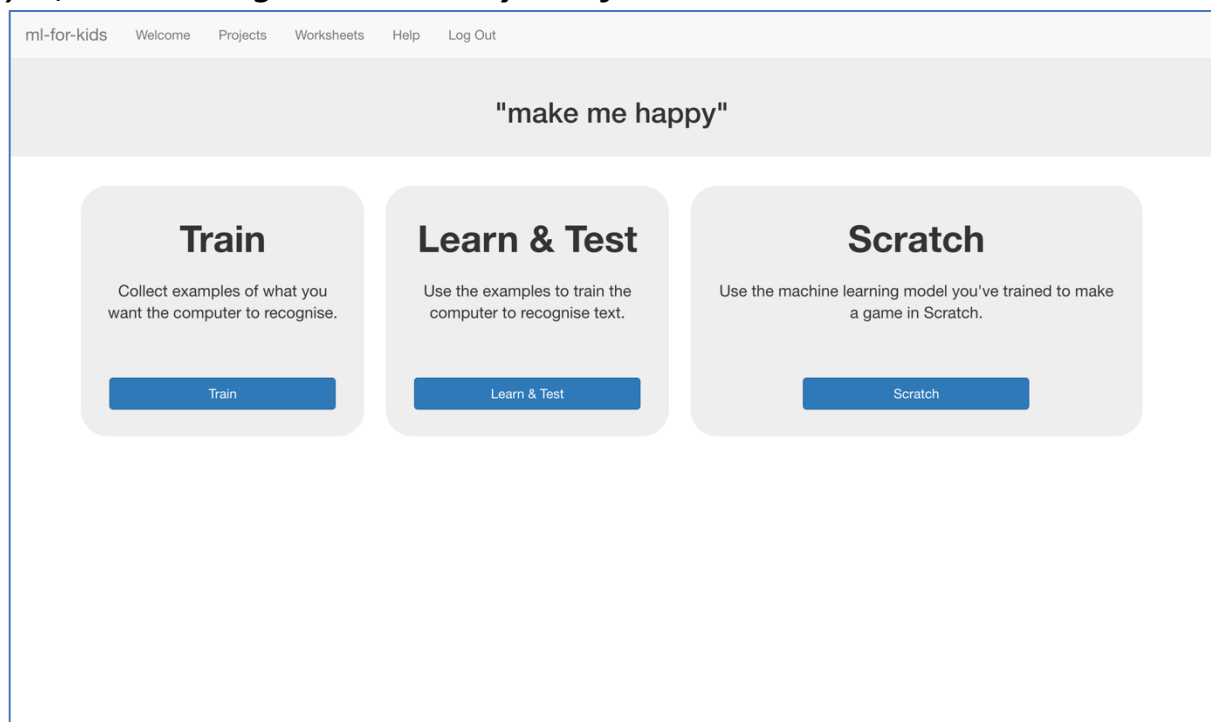
The screenshot shows the 'Start a new project' dialog box on the machinelearningforkids.co.uk website. The dialog box has a blue header with the title 'Start a new project'. Below the header, there is a text input field for 'Project Name *' with the value 'make me happy'. To the right of the input field is a dropdown menu with three options: 'text' (selected), 'images', and 'numbers'. Below the input field and dropdown menu is a text area with the question 'What type of thing do you want to teach the computer to recognise?' and three lines of instructions: 'For words, sentences or paragraphs, choose "text"', 'For photos, diagrams and pictures, choose "images"', and 'For sets of numbers, choose "numbers"'. At the bottom right of the dialog box are two buttons: 'CREATE' and 'CANCEL'. In the background, the website's navigation bar is visible with links for 'ml-for-kids', 'Welcome', 'Projects', 'Worksheets', 'Help', and 'Log Out'. The main heading 'Your machine learning projects' is also visible. On the right side of the background, there is a button with a plus sign and the text 'Add a new project', and a trash can icon.

7. You should now see “make me happy” show up in the list of your projects. Click on it.



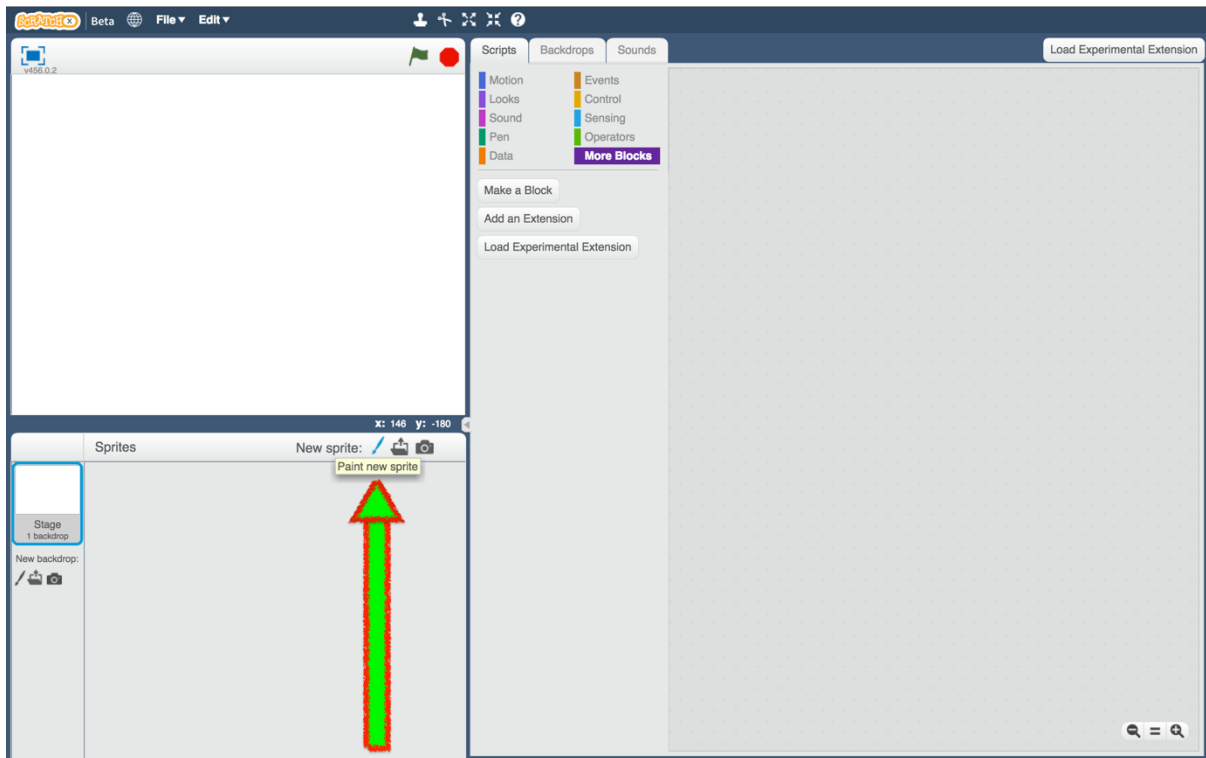
8. We'll start by getting a project ready in Scratch. Click on the **Scratch** button.

*The next page will warn you that you haven't done any machine learning yet, but clicking on **Scratch by itself** will launch Scratch.*

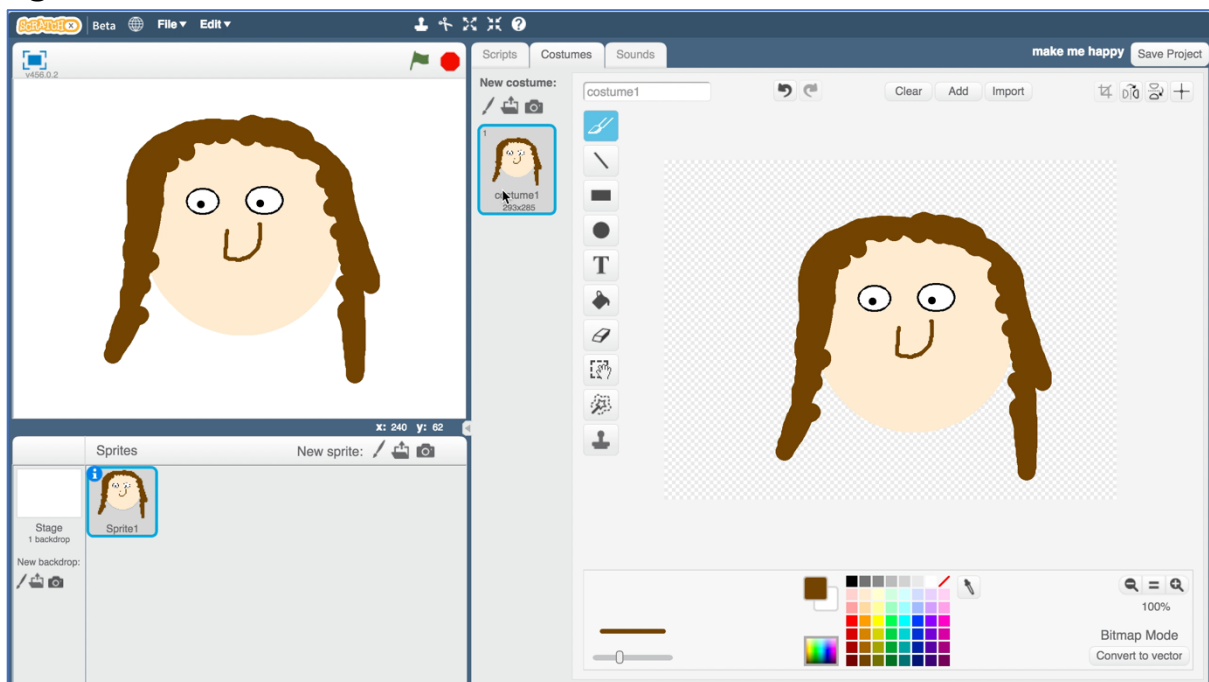


9. Create a new sprite by clicking on the paintbrush icon in the Sprites window.

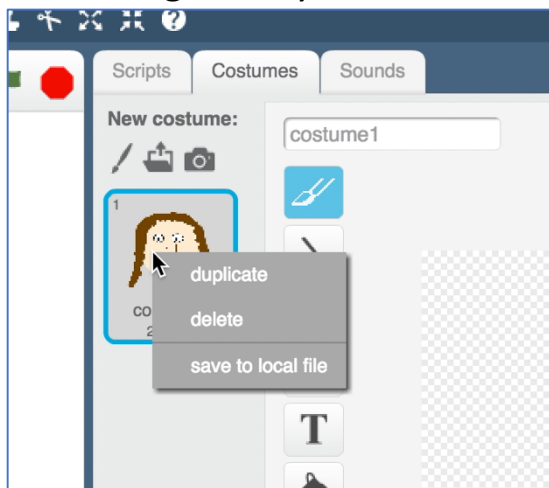
There are a few similar looking paintbrush buttons – make sure you click the one marked below.



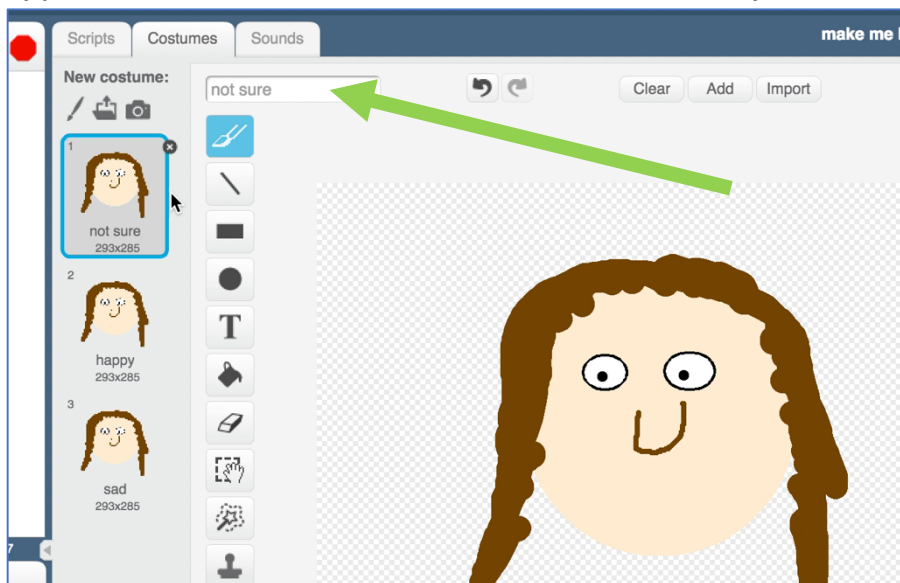
10. Draw a face, without a mouth, in the sprites editor view on the right.



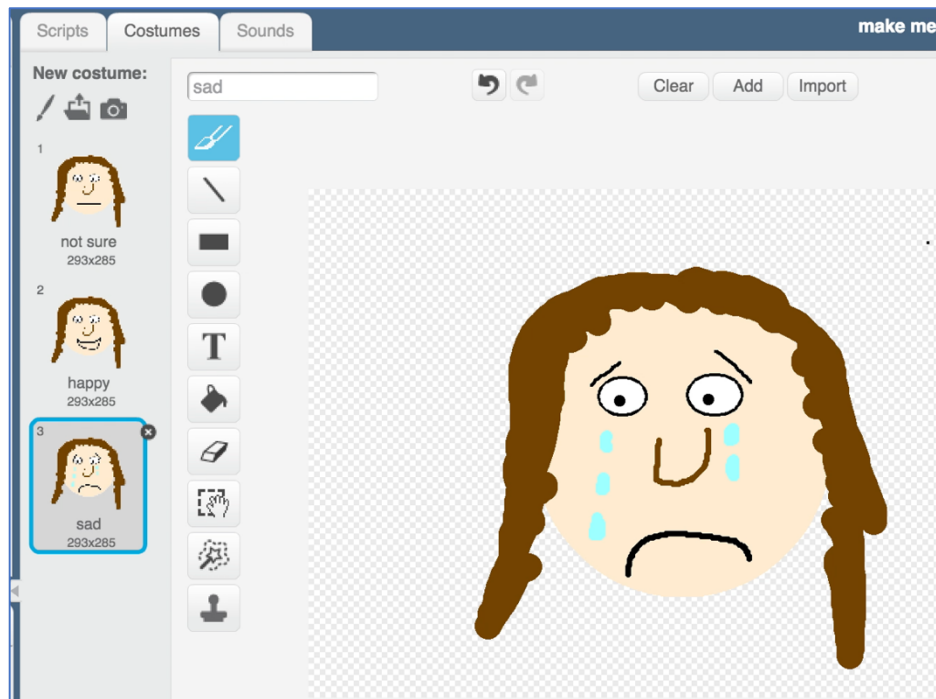
- 11.** Right-click on the costume, and click “Duplicate”.
Do that again so you have **three** copies of the costume.



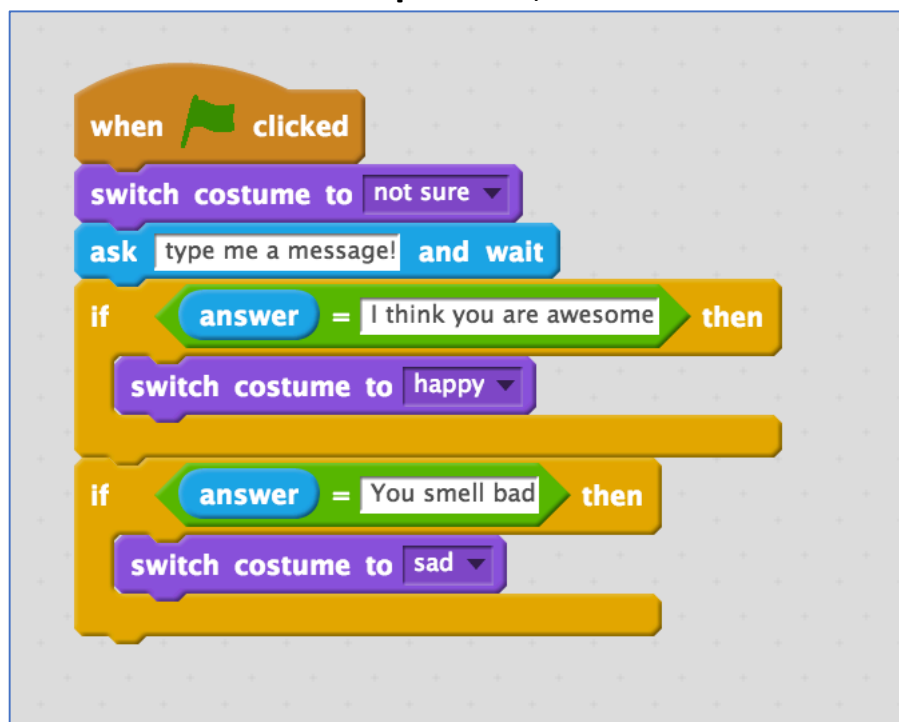
- 12.** Name the three costumes “not sure”, “happy” and “sad”
Type the names into the white box shown by the arrow below.



- 13.** Draw a mouth on each of the costumes.
The “not sure” face should be a straight line.
The “happy” face should have a smile.
The “sad” face should look sad.

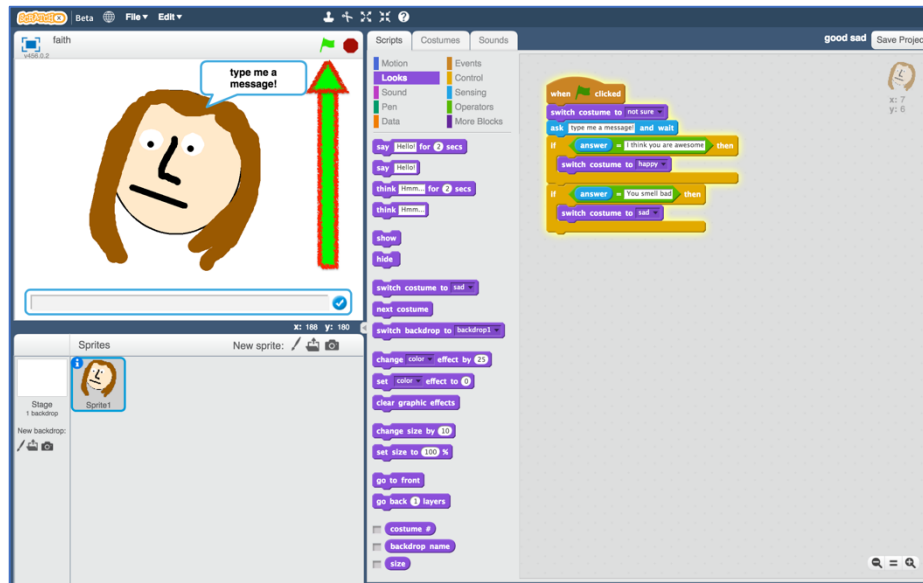


- 14.** Click on the “**Scripts**” tab, and enter the following script.



- 15.** Save your project.
Click on **File** -> **Save** to save the project to a file.

16. Click on the **green flag** to test.



17. Type in a message and watch it react!

Type "I think you are awesome" and press enter. The character smiles. Click on the green flag again and type "You smell bad". The character should cry.

Type anything else, and the character's face won't change.

What have we done so far?

You've created a character that should react to what people type, and programmed it using a simple rules-based approach.

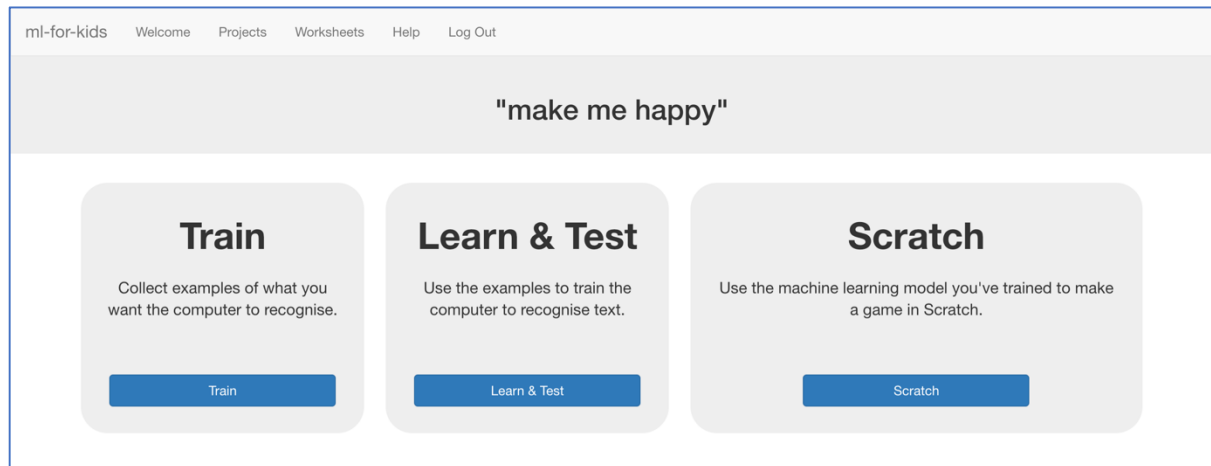
If you want it to react to other messages, you will need to add extra **if** blocks.

The problem with is that you need to predict exactly what messages the character will receive. Making a list of every possible message would take forever!

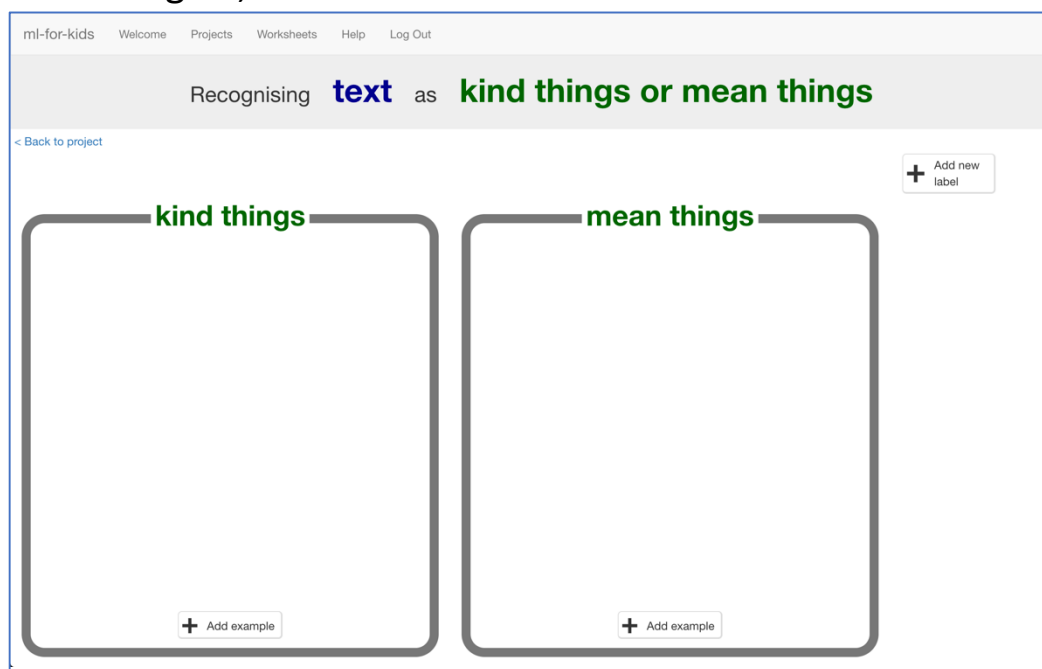
Next, we'll try a better approach – teaching the computer to recognise messages for itself.

18. Close the Scratch window.

19. We need examples to train the computer. Click the “< Back to project” link. Then click the **Train** button.



20. Click on “+ Add new label” and call it “kind things”. Do that again, and create a second bucket called “mean things”.



21. Click the “Add example” button in the “kind things” bucket, and type in a kind message.

22. Click on the “Add example” button in the “mean things” bucket, and type in a mean message.

23. Repeat steps 21 and 22 until you've written at least **ten** examples of each.

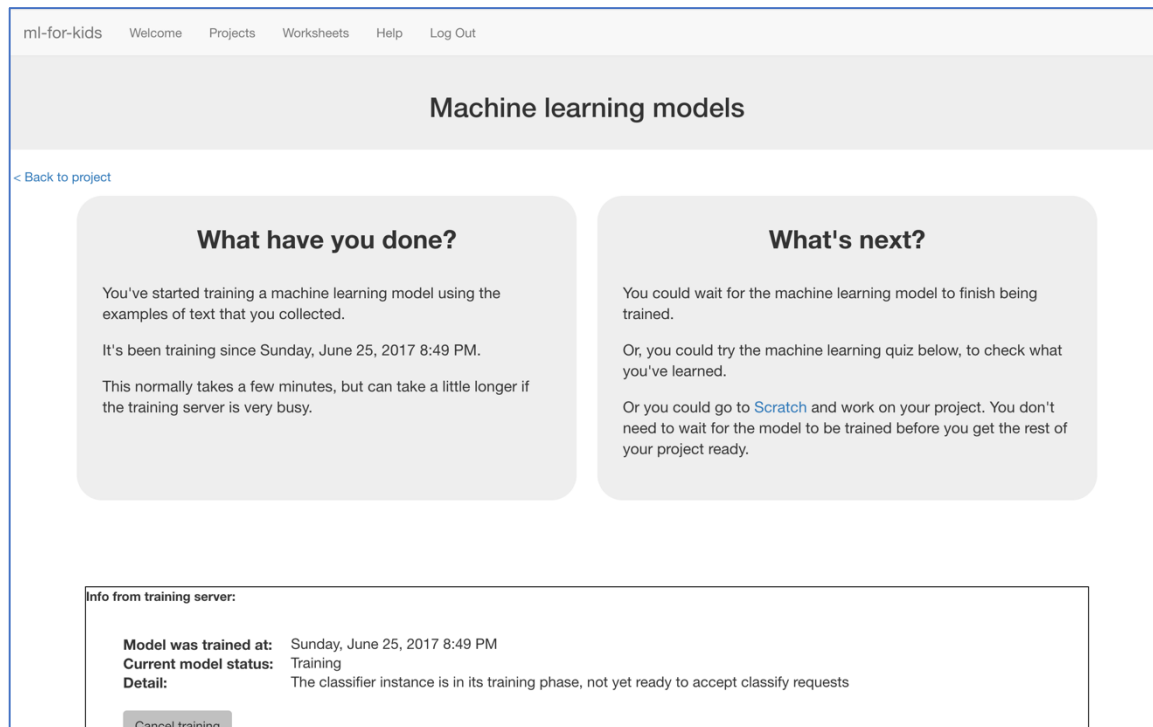
The screenshot shows a web interface titled "Recognising text as kind things or mean things". At the top, there's a navigation bar with links: "ml-for-kids", "Welcome", "Projects", "Worksheets", "Help", and "Log Out". Below the navigation bar, there's a header area with the title "Recognising text as kind things or mean things". On the left, there's a link "< Back to project". On the right, there's a button "+ Add new label". The main content area is divided into two columns. The left column is titled "kind things" and contains several text boxes with examples: "You're a lovely person", "I appreciate all of the things you do", "Your hair looks great today", "You're my best friend", "I think you're amazing", "Thanks for all of your help", and "That jacket looks great on you". At the bottom of this column is a button "+ Add example". The right column is titled "mean things" and contains several text boxes with examples: "You smell", "I don't like you", "You're as dumb as a bag of rocks", "I'm fed up with how useless you are", "You're an idiot", and "You smell bad". At the bottom of this column is a button "+ Add example".

24. Click on the “< Back to project” link.
Then click on the “Learn & Test” button.

25. Click on the “Train new machine learning model” button.
As long as you've collected enough examples, the computer should start to learn how to recognise messages from the examples you've given to it.

The screenshot shows a web interface titled "Machine learning models". At the top, there's a navigation bar with links: "ml-for-kids", "Welcome", "Projects", "Worksheets", "Help", and "Log Out". Below the navigation bar, there's a header area with the title "Machine learning models". On the left, there's a link "< Back to project". The main content area is divided into two columns. The left column is titled "What have you done?" and contains the text: "You've collected examples of text for a computer to use to recognise when text is kind things or mean things." and "You've collected:" followed by a list: "7 examples of kind things," and "6 examples of mean things". The right column is titled "What's next?" and contains the text: "Ready to start the computer's training?" and "Click the button below to start training a machine learning model using the examples you've collected so far." followed by "(Or go back to the Train page if you want to collect some more examples first.)". At the bottom of the page, there's a box titled "Info from training server:" containing a button "Train new machine learning model".

26. Wait for the training to complete. This might take a few minutes. *While waiting, try to complete the machine-learning multi-choice quiz at the bottom of the page.*



27. Once the training has completed, a Test box will be displayed. Try testing your machine learning model to see what the computer has learned. Type something kind, and press enter. It should be recognised as kind. Type something mean, and press enter. It should be recognised as mean.

Test it with examples that you haven't shown the computer before. If you're not happy with how the computer recognises the messages, go back to step 21, and add some more examples. Make sure you repeat step 25 to train with the new examples though!

[ml-for-kids](#) [Welcome](#) [Projects](#) [Worksheets](#) [Help](#) [Log Out](#)

Machine learning models

[< Back to project](#)

What have you done?

You've trained a machine learning model to recognise when text is kind things or mean things.

You created the model on Sunday, June 25, 2017 8:49 PM.

You've collected:

- 7 examples of kind things,
- 6 examples of mean things

What's next?

Try testing the machine learning model below. Enter an example of text below, that you didn't include in the examples you used to train it. It will tell you what it recognises it as, and how confident it is in that.

If the computer seems to have learned to recognise things correctly, then you can go to [Scratch](#) and use what the computer has learned to make a game!

If the computer is getting too many things wrong, you might want to go back to the [Train](#) page and collect some more examples. Once you've done that, click on the button below to train a new machine learning model and see what different the extra examples will make!

Try putting in some text to see how it is recognised based on your training.

You're a horrid person and I really don't like you

Test

Recognised as **mean things**
with 66% confidence

Info from training server:

Model was trained at:

Sunday, June 25, 2017 8:49 PM

Current model status:

Available

Detail:

The classifier instance is now available and is ready to take classifier requests.

Delete this model

What have we done so far?

You've started to train a computer to recognise text as being kind or mean. Instead of trying to write rules to be able to do this, you are doing it by collecting examples. These examples are being used to train a machine learning "model".

This is called "supervised learning" because of the way you are supervising the computer's training.

The computer will learn from patterns in the examples you've given it, such as the choice of words, and the way sentences are structured. These will be used to be able to recognise new messages.

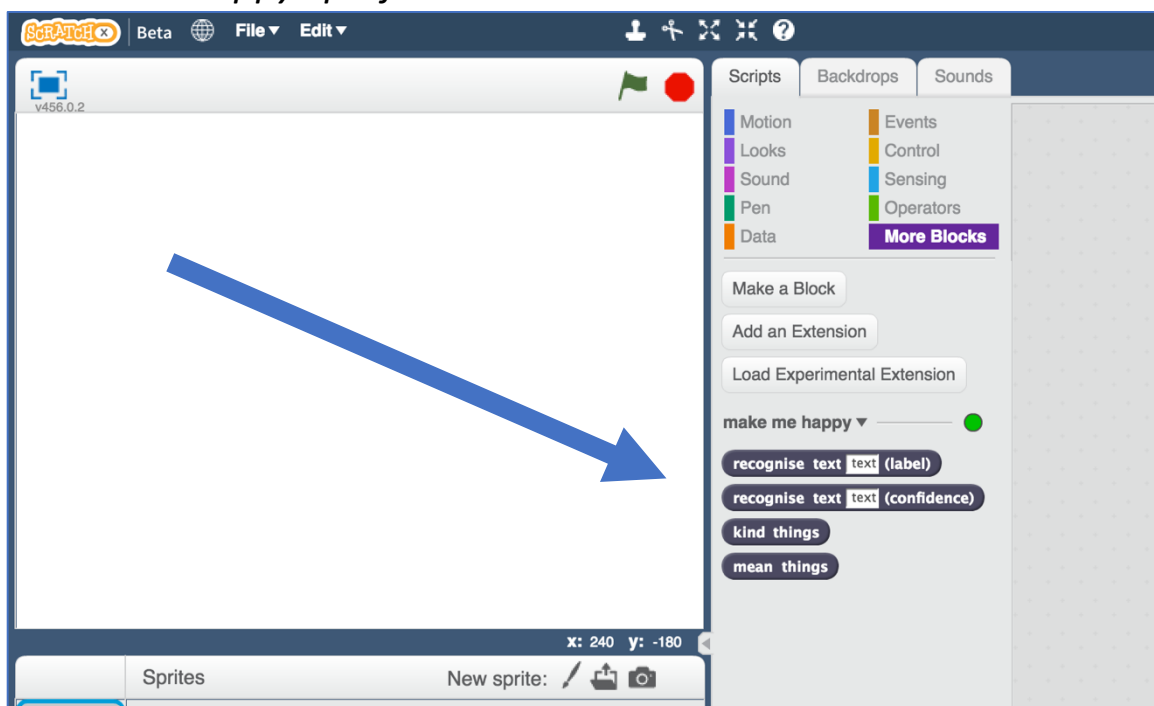
28. Click on the “< Back to project” link, then back to the “Scratch” button.

This page has instructions on how to use the new blocks in Scratch from your project.

Keep the page open if you need to check back on how to use them.

29. Click on the “Open in Scratch” button at the bottom to launch the Scratch editor.

You should see four new blocks in the “More blocks” section from your “make me happy” project.



30. Load the Scratch project you saved before.

*Click on **File** -> **Load Project***

Tips

More examples!

The more examples you give it, the better the computer should get at recognising whether a message is kind or mean.

Try and be even

Try and come up with roughly the same number of examples for kind and mean.

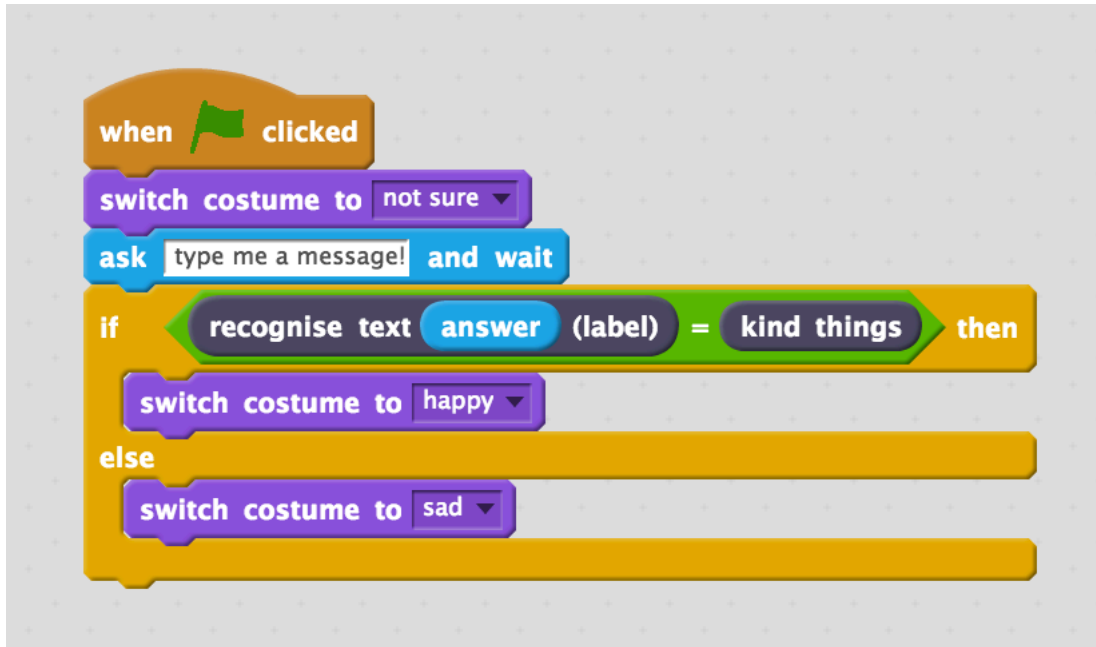
If you have a lot of examples for one type, and not the other, the computer might learn that type is more likely, so you'll affect the way that it learns to recognise messages.

Mix things up with your examples

Try to come up with lots of different types of examples.

For example, make sure that you include some long examples and some very short ones.

- 31.** Click on the “**Scripts**” tab, and update the script to use your machine learning model instead of the rules you made before.
The “recognise text ... (label)” block is a new block added by your project.
If you give it some text, it will return either “kind things” or “mean things” based on the training you’ve given to the computer.
You can use this to choose the costume to switch to.



- 32.** Click on the **green flag** to test again.

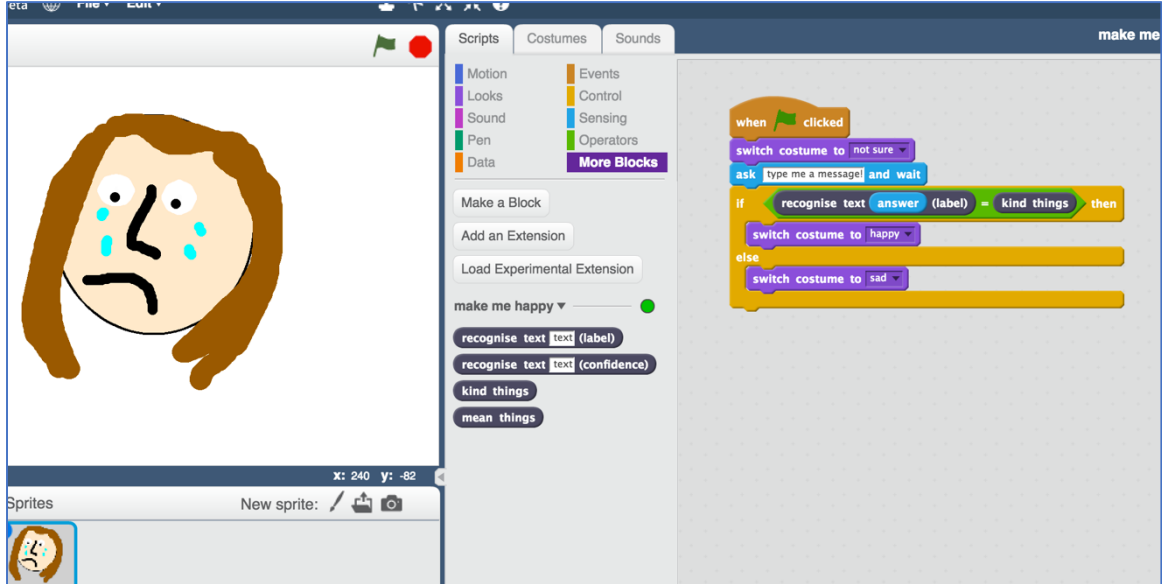


33. Test your project

Type a kind message and press enter. The character should smile.

Click the green flag again. Type a mean and unkind message and press enter. The character should look sad.

This should work for messages that you didn't include in your training.



34. Save your project.

What have we done so far?

You've modified your Scratch character to use machine learning instead of your earlier rules-based approach.

Training the computer to be able to recognise messages for itself should be much quicker than trying to make a list of every possible message.

The more examples you give it, the better it should get at recognising messages correctly.

Ideas and Extensions

Now that you've finished, why not give one of these ideas a try?

Or come up with one of your own?

Write a reply

Instead of just changing the way they look, make your character reply, based on what it recognises in the message!

Try a different character

Instead of a person's face, why not try something different, like an animal?

It could react in different ways, instead of smiling.

For example, you could make a dog that wags their tail if you say something kind to it!

Recognising random messages

Try adding a third bucket to recognise messages that aren't particularly kind or mean – like "What is the time?".

Different emotions

Instead of kind and mean, could you train the character to recognise other types of message?