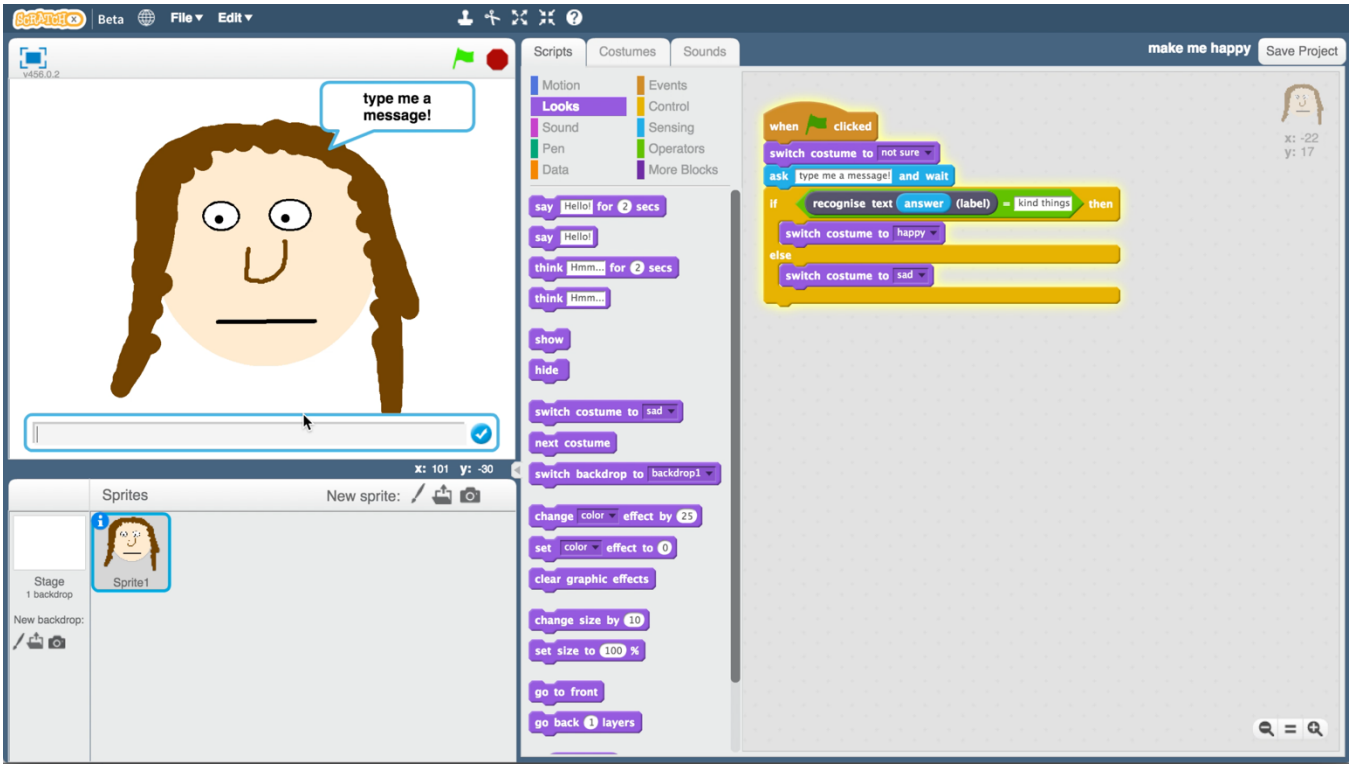
Make me happy!

In this project you will make a character that can react 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.

We won't try to program a list of rules for what is kind and what is mean.

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



1. Go to http://machinelearningforkids.co.uk in a web browser

2. Click on "**Get started**"

3. Click on "**Log In**" and enter your username and password

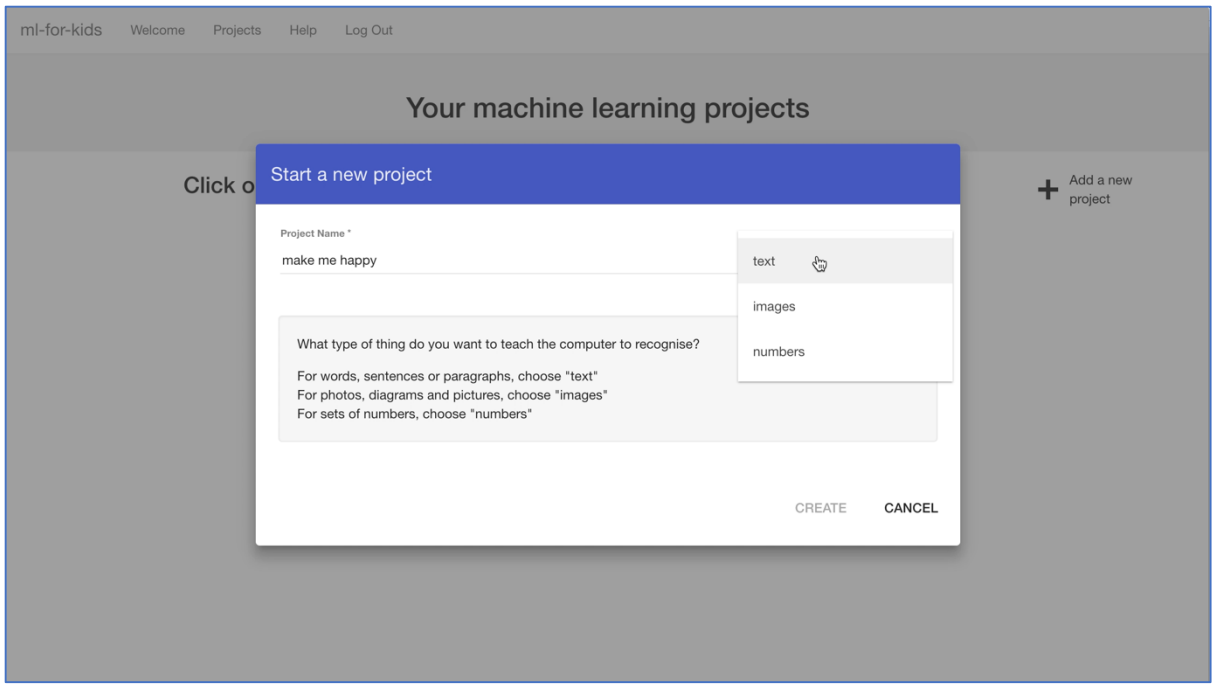
If you don't have a username, ask your teacher or group leader to create one for you.

If you can't remember your username or password, ask your teacher or group leader to reset it for you.

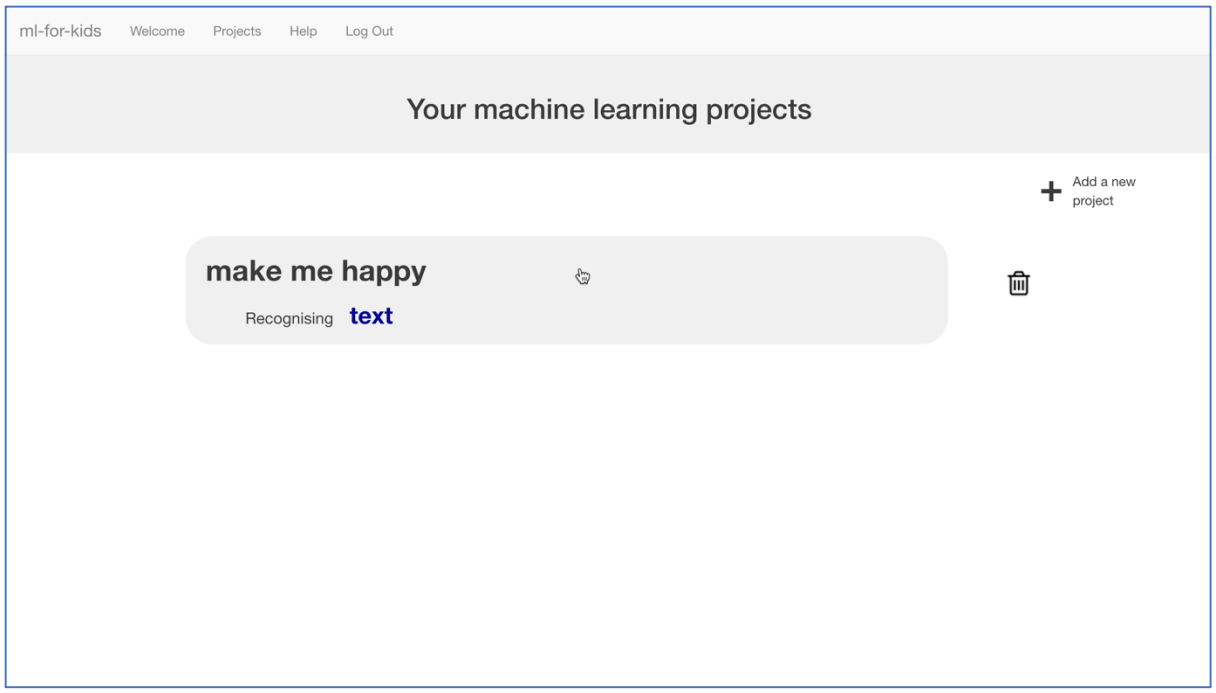
4. Click on "**Projects**" on the top menu bar

5. Click on the **"+ Add a new project**" button.

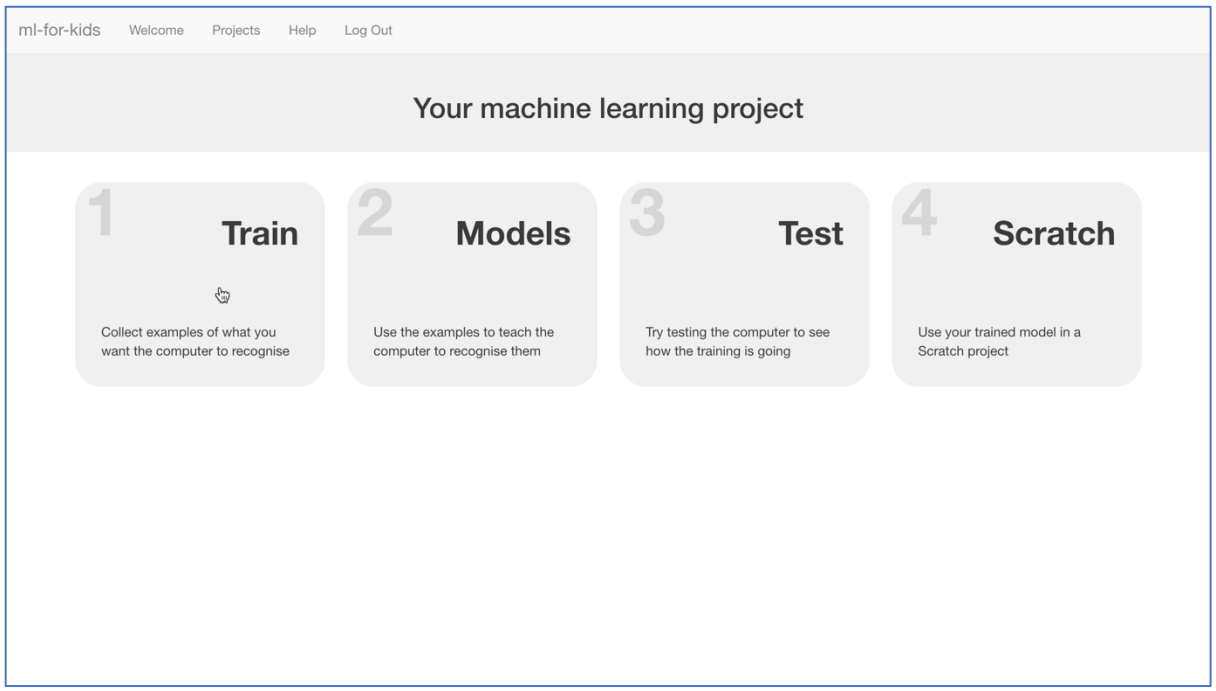
6. Call your project "make me happy" and set it to learn how to recognise "**text**"



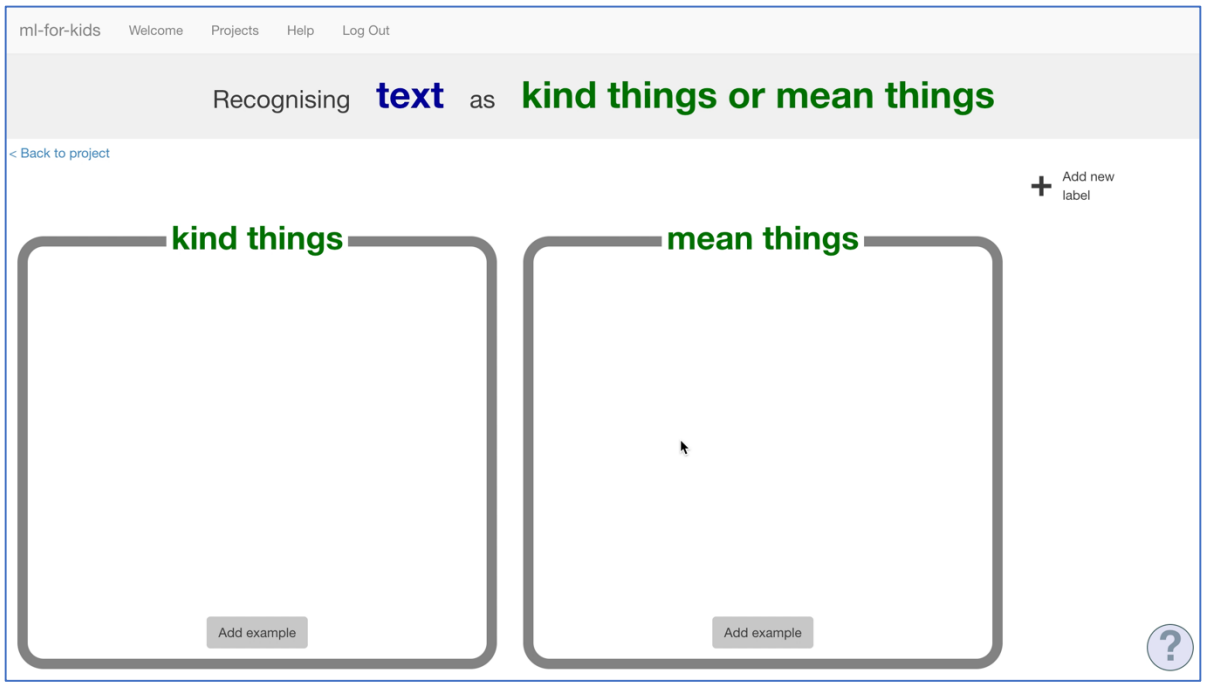
7. You should now see “make me happy” show up in the list of your projects. Click on it.



8. The first step is to collect some examples. Click on the **Train** button.



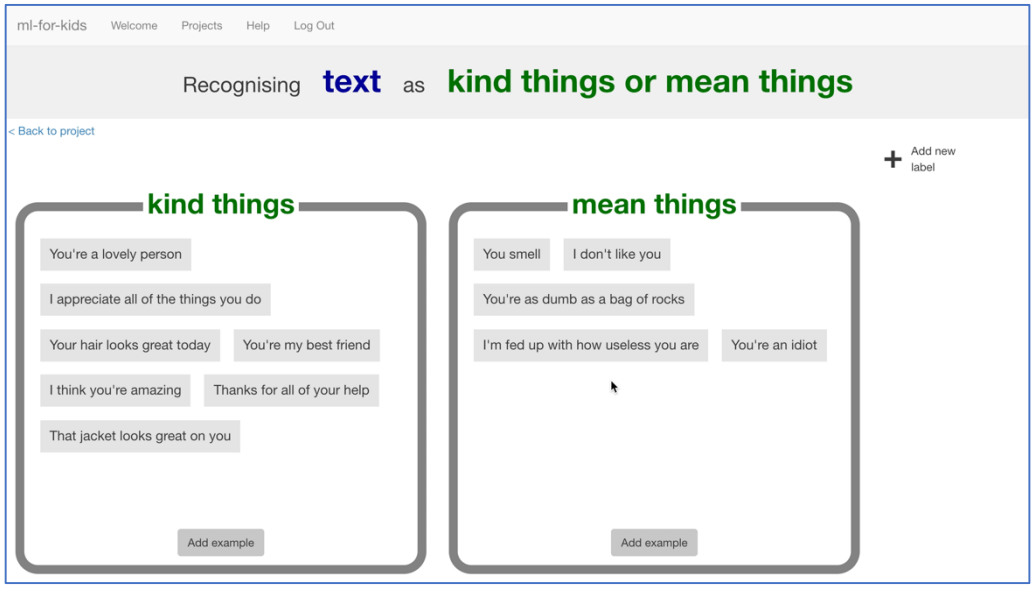
9. Click on **"+ Add new label**" and call it "kind things".
Do that again, and create a second bucket called "mean things".



10. Click on the "**Add example**" button in the "kind things" bucket, and type in a kind message.

11. Click on the "**Add example**" button in the "mean things" bucket, and type in a mean message.

12. Repeat steps 10 and 11 until you've collected at least ten examples of each.

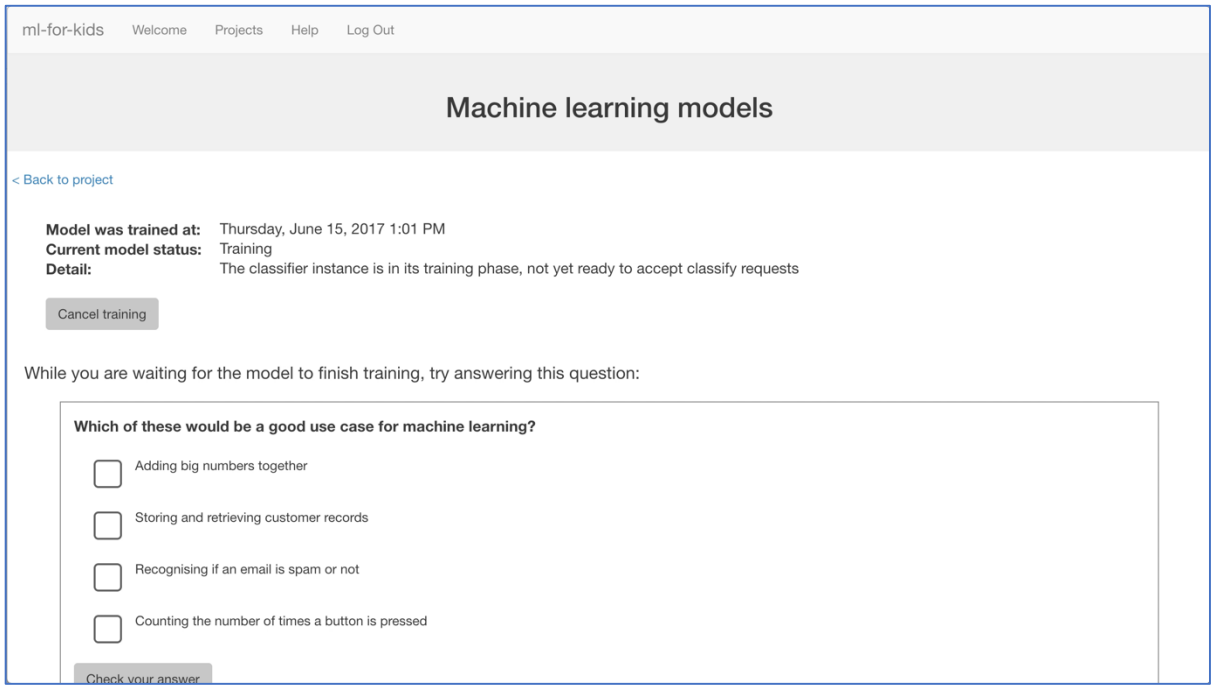


13. Click on the **"< Back to project**" button, then click on the "**Models**" button.

14. 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.

This might take a few minutes.

The "Current model status" will be "Available" once this has finished.



15. Click on the **"< Back to project**" button, then click on the "**Test**" button. Try testing your training 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.

Try testing 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 12, and add some more examples.
Make sure you repeat step 14 to train with the new examples though!

What have we done so far?

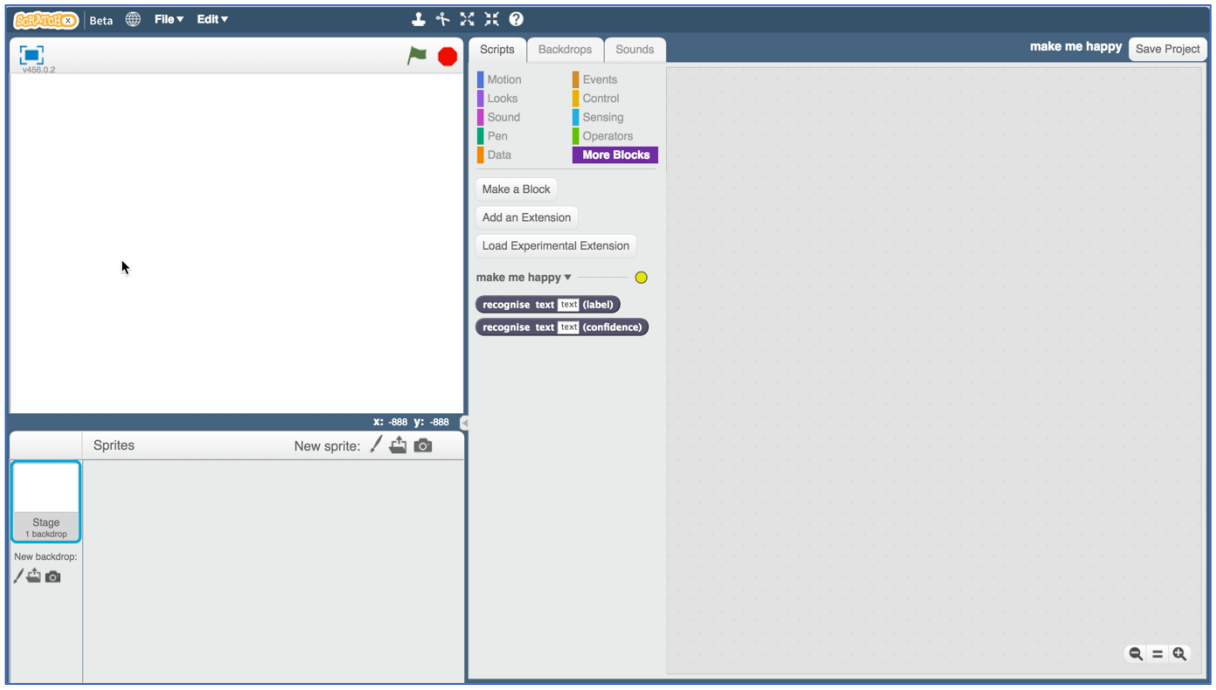
You've started to train a computer to recognise text as being kind or mean. Instead of trying to write rules to be able to do this, you are doing it by collecting examples. These examples are being used to train a machine learning "model".

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.

16. Click on the **"< Back to project**" button, then the "**Scratch**" button.

17. Click on the "**Open in Scratch**" button to launch the Scratch editor.
You should see two new blocks in the "More blocks" section from your "make me happy" project.



Tips

You don't have to wait for the training to finish to work on your Scratch project

If you use Scratch before your model has finished training, it will just make random guesses for kind/mean. That means it will get it wrong a lot, but that shouldn't stop you working on your character or it's script blocks.

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

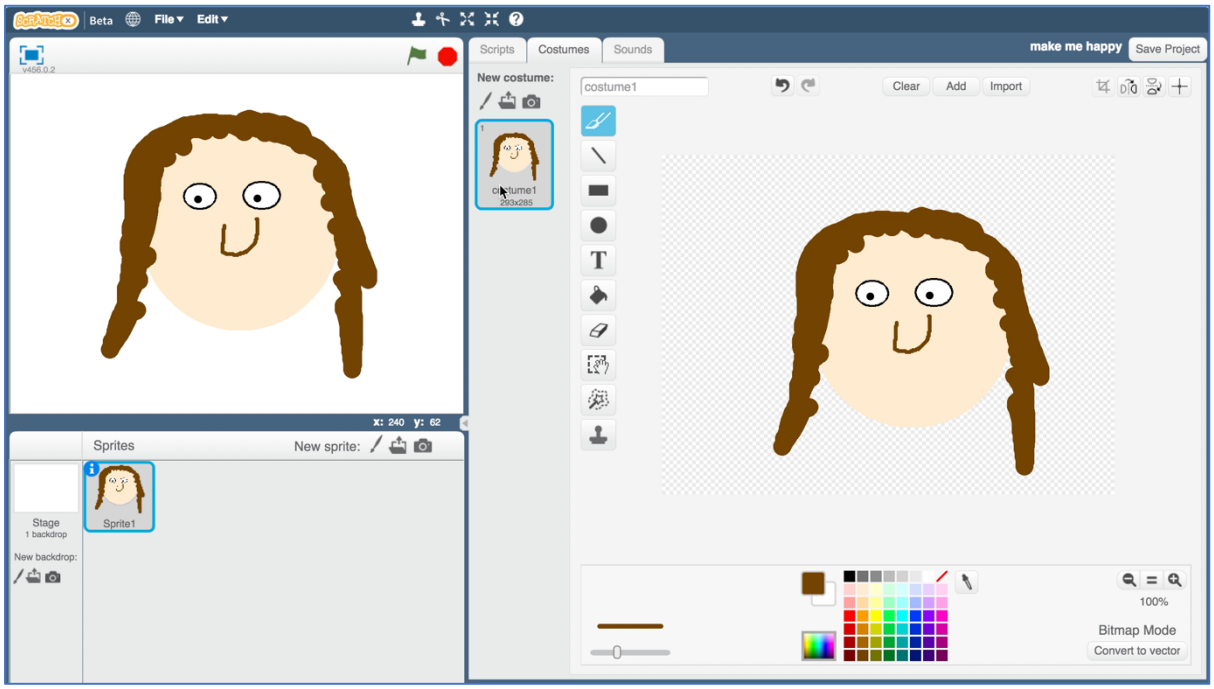
Try and come up with roughly the same number of examples for kind and mean.

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

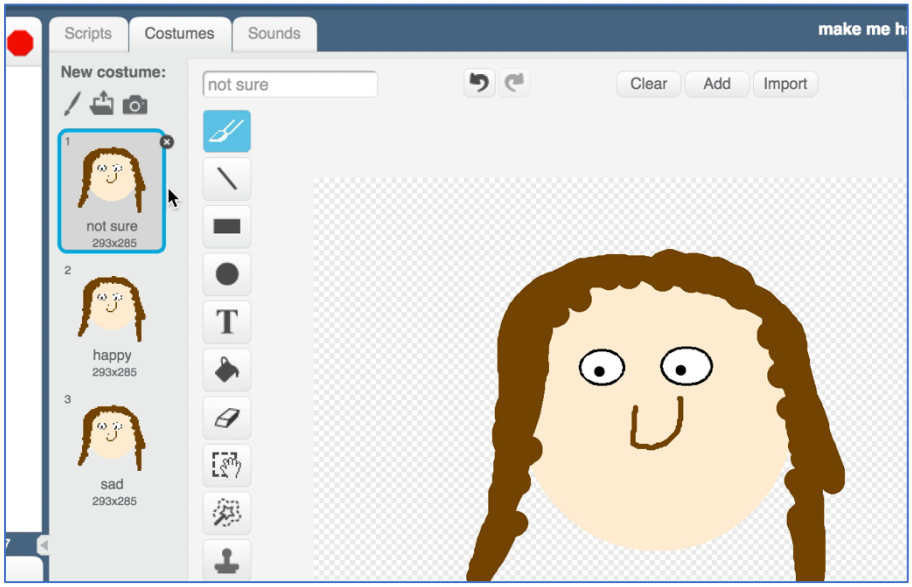
Try to come up with lots of different types of examples.
For example, make sure that you include some long examples and some very short ones.

18. Click on the paintbrush icon in the "**Sprites**" panel to create a new sprite. Draw a face, without a mouth, in the sprites editor view on the right.



19. Right-click on the costume, and click "Duplicate". Do that again so you have three copies of the costume.

20. Rename the three costumes "not sure", "happy" and "sad"

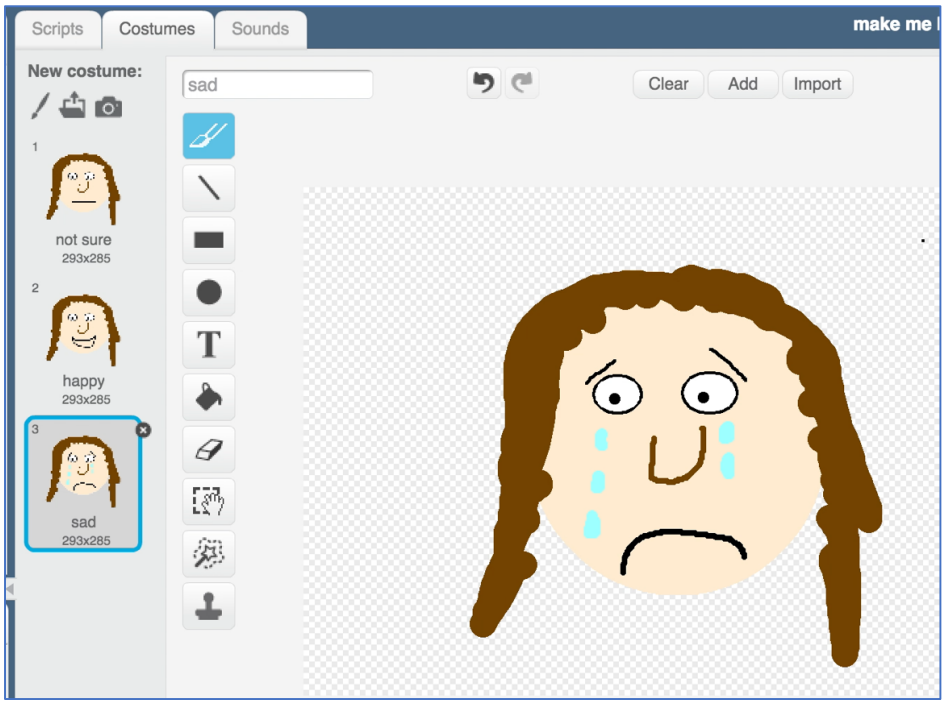


21. Draw a mouth on each of the costumes.

The “not sure” face should be a straight line.

The “happy” face should have a smile.

The “sad” face should look sad.

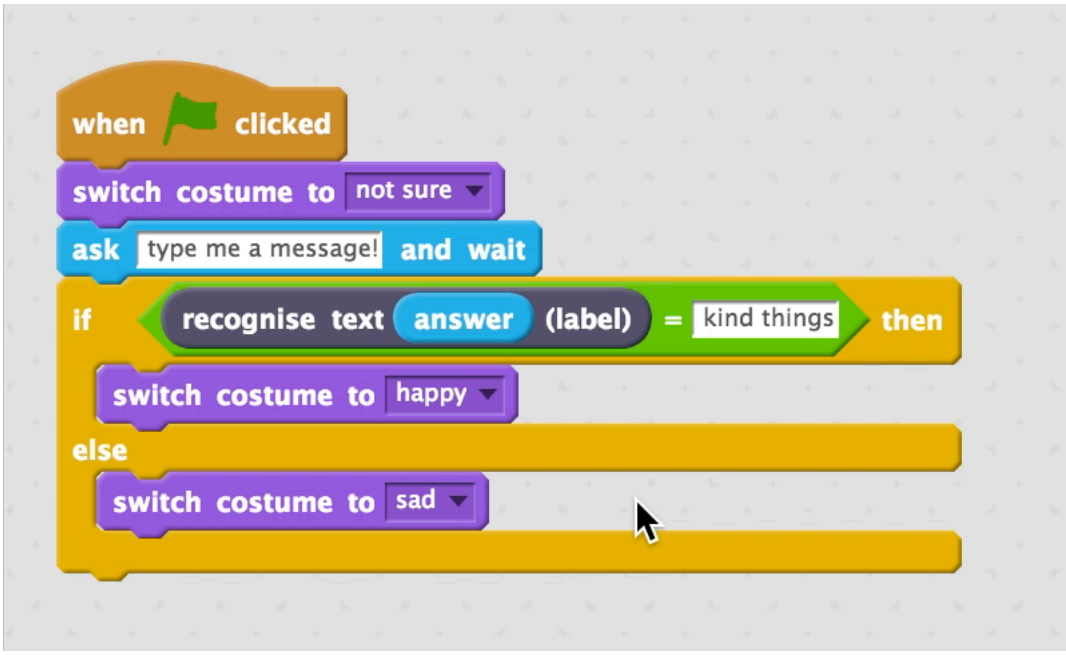


22. Click on the “**Scripts**” tab, and enter the following script.

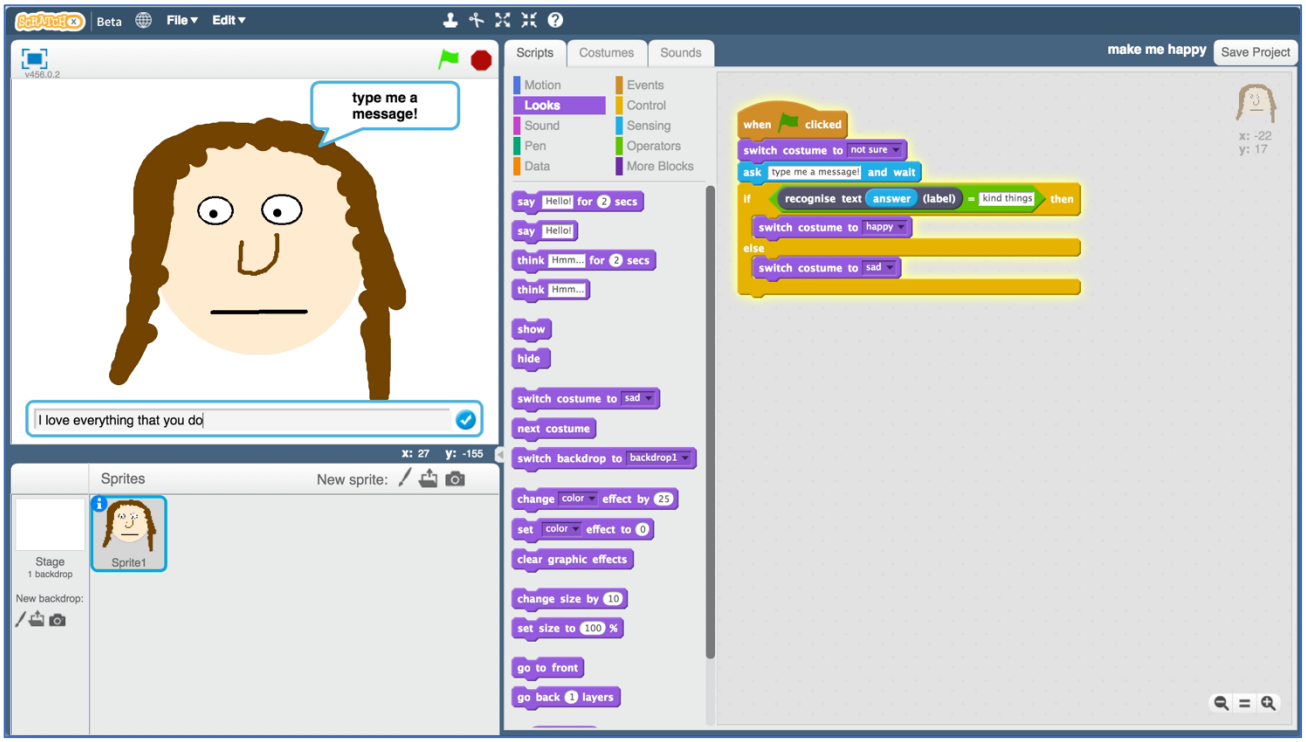
The “recognise text … (label)” block is a new block added by your project.

If you give it a text message, 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.



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



24. Type it a message and watch it react!

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

Try testing 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 12, and add some more examples.
Make sure you repeat step 14 to train with the new examples though!

Ideas and Extensions

If 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?