Chatbots

Introduction

Scratch is a visual programming language suitable for children and teens to create their own interactive stories, games and animations. Projects can also be uploaded and shared with the Scratch community via the Scratch website for others to enjoy.

Chatbots are becoming increasingly popular today, you probably have one in your home already in the form of Amazon's Alexa or Google's Home. Chatbots are basically computer programs you can interact with either by talking to them or typing to get a response..

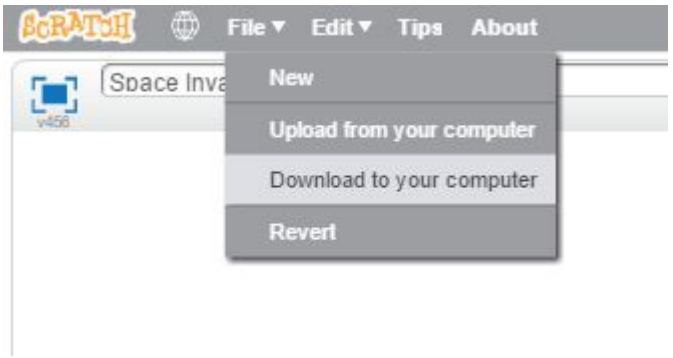
For us to start building our chatbot we must once again start thinking as a computer does. Just like our Space Invaders game all tasks and instructions need to be broken down into small chunks so they are easily followed. This guide breaks down our chat bot into easy to follow sections with suggestions at the end to help improve it and make it better.

Let's get started by first loading Scratch. You can either install Scratch on your computer or load it via your web browser by navigating to https://scratch.mit.edu/

Top Tip

Don't forget to save your work regularly. It's good practice to get in the habit of saving any progress you've made. So when you're feeling great after solving a coding problem, save it, save it, save it!

The save options may differ slightly between Scratch versions but head on up to the File option in the top left and you'll find it.

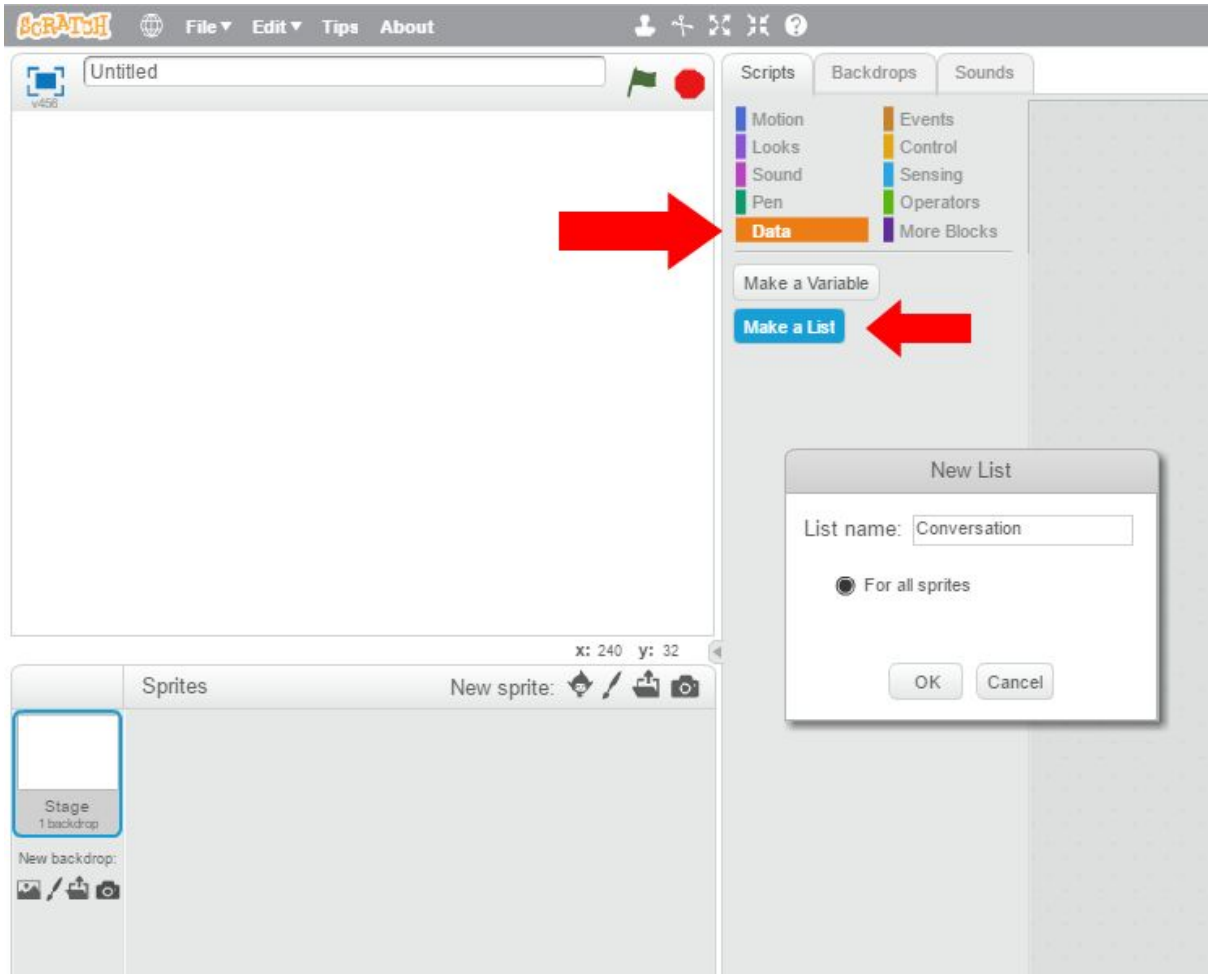


Let's get started

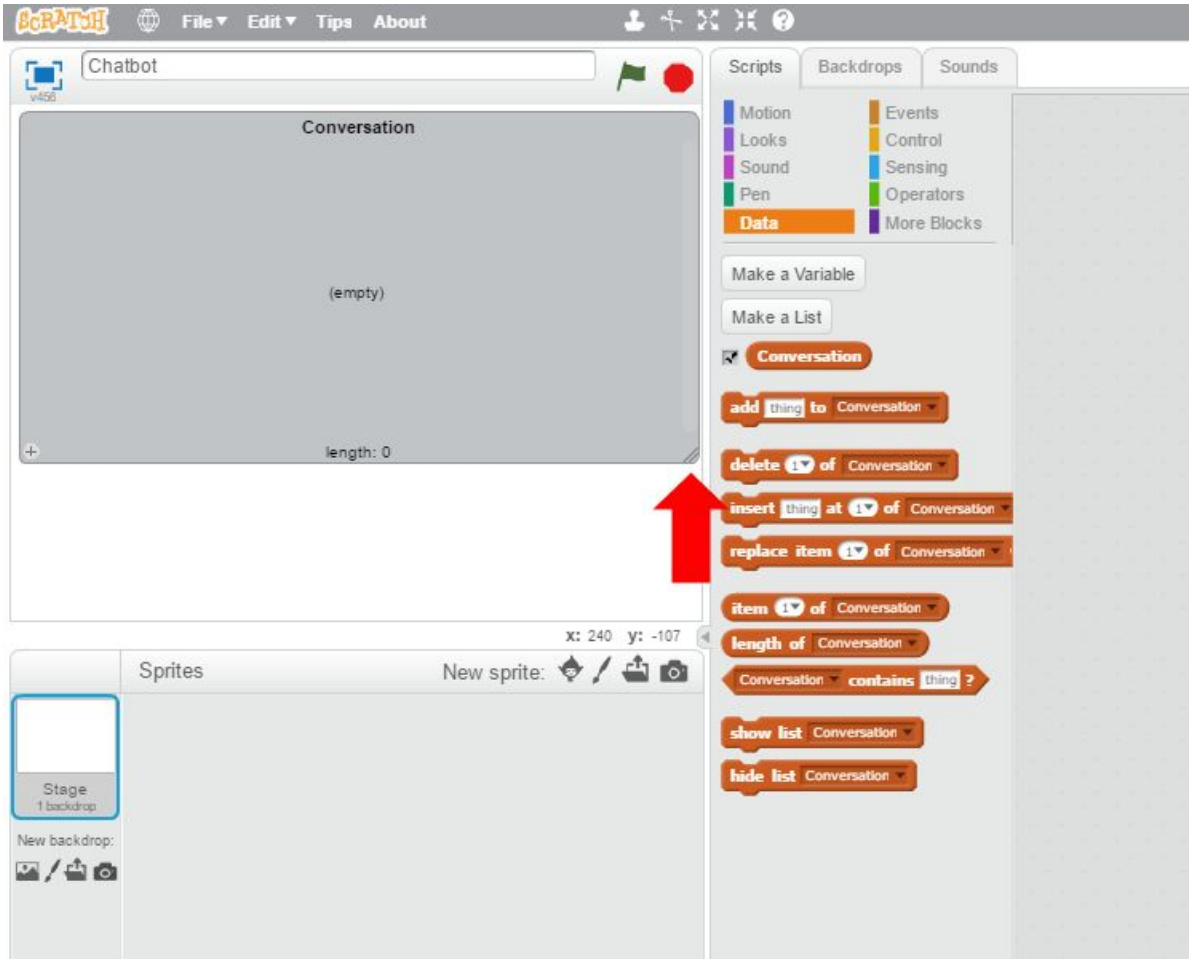
Step 1:

We are going to interact with our chatbot by typing sentences for it to understand. Then our code will try to understand the words we've given it and build a suitable response back, starting a conversation with us and appearing intelligent.

For this first step we need to produce an area on the screen that will display the conversation. This will show all the sentences we've written to it but also display the responses given to us by our chatbot. To do this we first need to make a list, which can be found in the Data section. We'll call it "Conversation" and have it set "For all sprites" as shown in the image below.

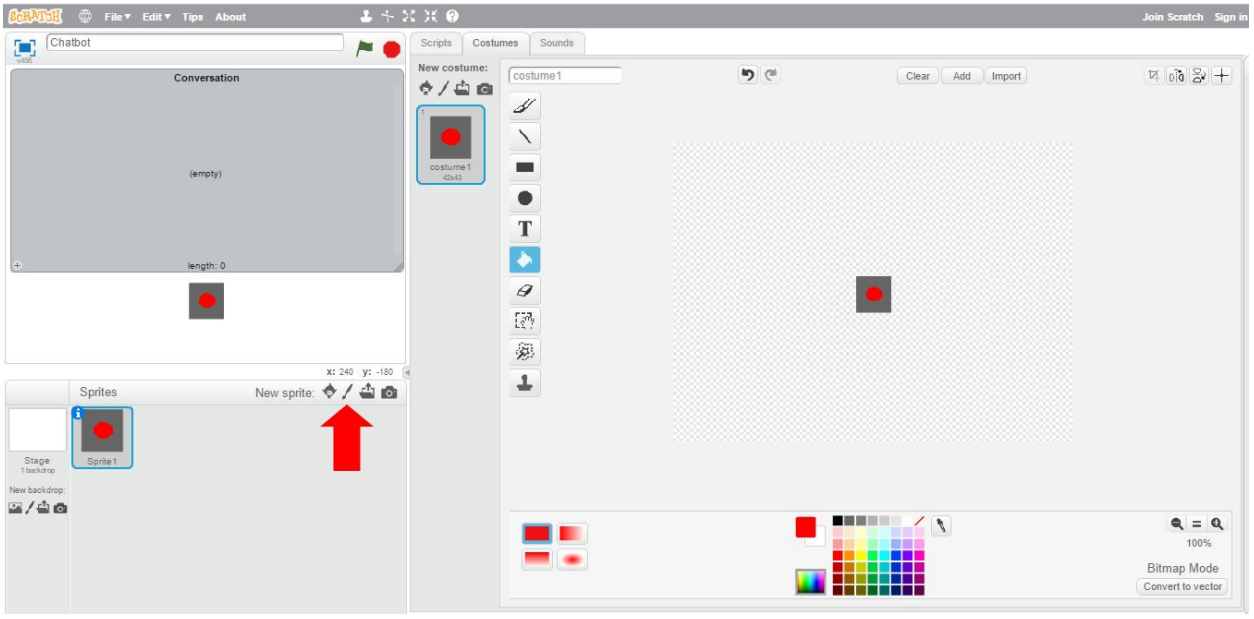


Resize it to fill part of the screen. This can be done by dragging the corners of the Conversation box. Make it a good size so that we can read the conversation taking place. Don't worry about making it too small, we can always adjust the size later on if required.

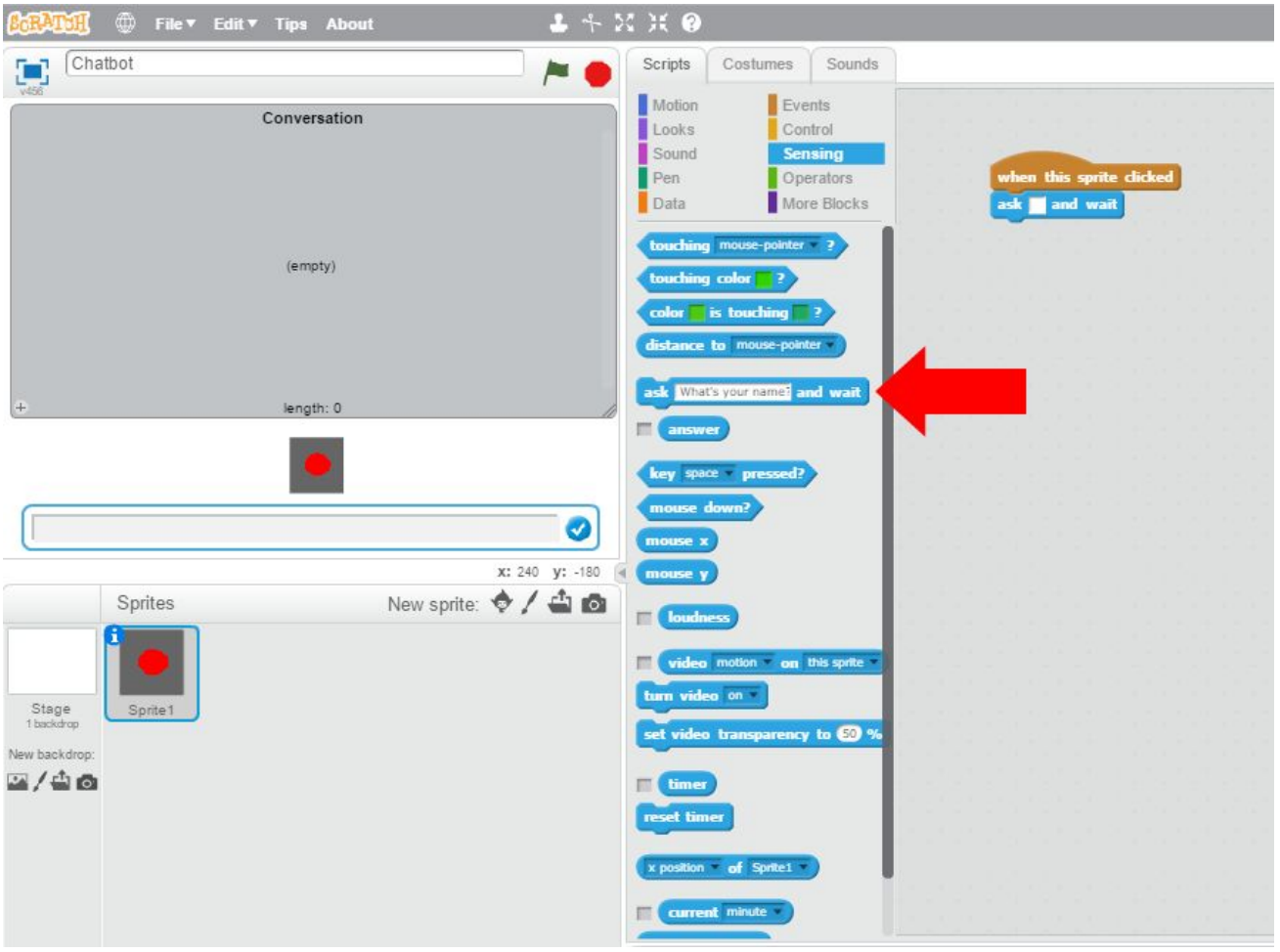


Step 2:

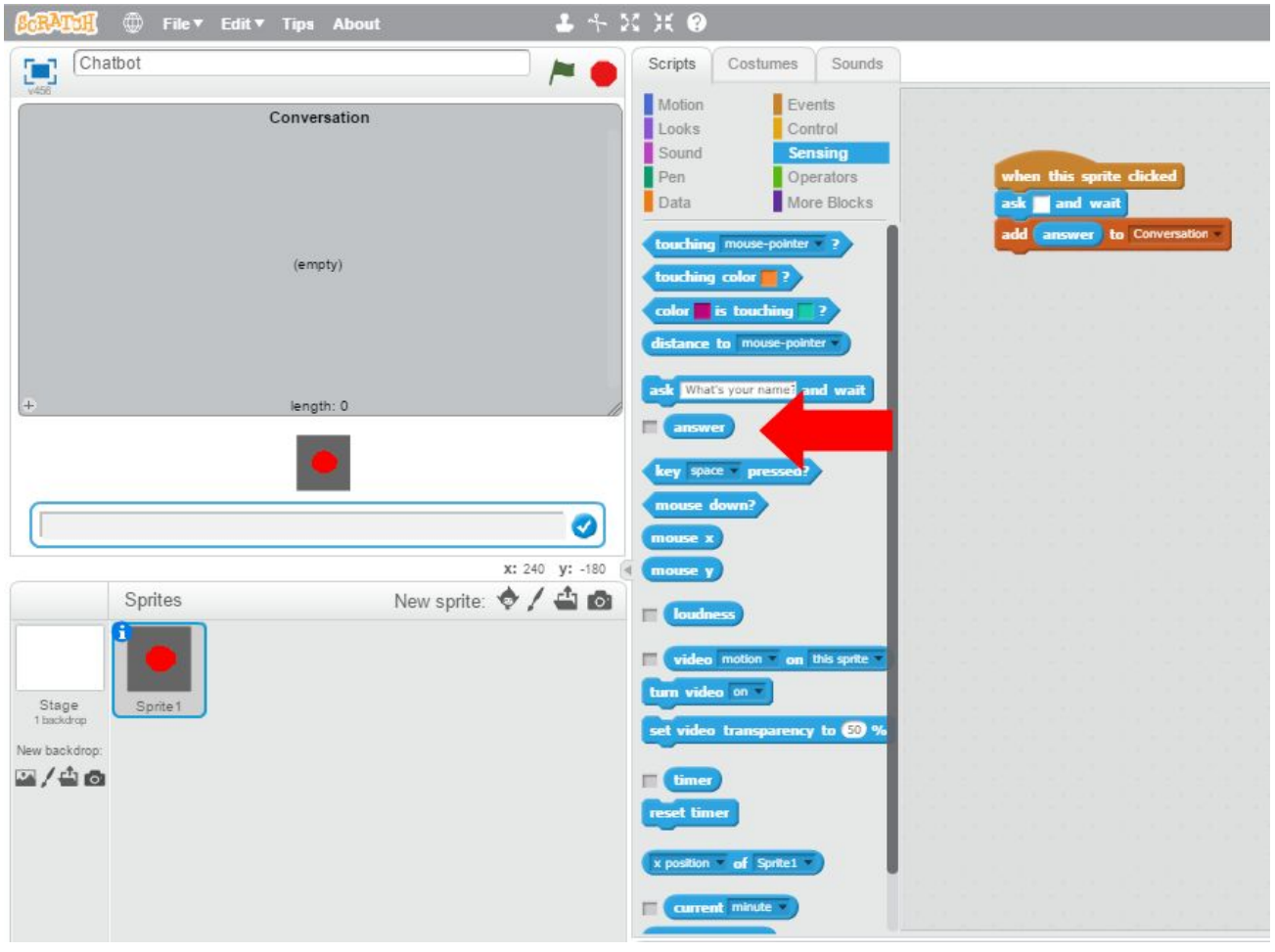
We now need a button on the screen to indicate when pressed that we are wanting to write something that is to be submitted to our chatbot for processing. Click on the New Sprite option indicated below and draw out a simple sprite to use for our button. Position it in a sensible place on the screen.



We now need to build the logic behind our button. To do this we need the “When this sprite is clicked” block from the Events section followed by a prompt to ask something found in the Sensing section as shown below. This tells the chatbot to present us with a text box to write our sentence in and wait until we submit it.



When our statement has been submitted the first thing we do is add it to the conversation box we set up earlier.



Step 3:

We are now ready to start processing the sentence we've submitted to the chatbot. The very first thing we need to do is take our statement and chop it up to make a list of all the words. For example:

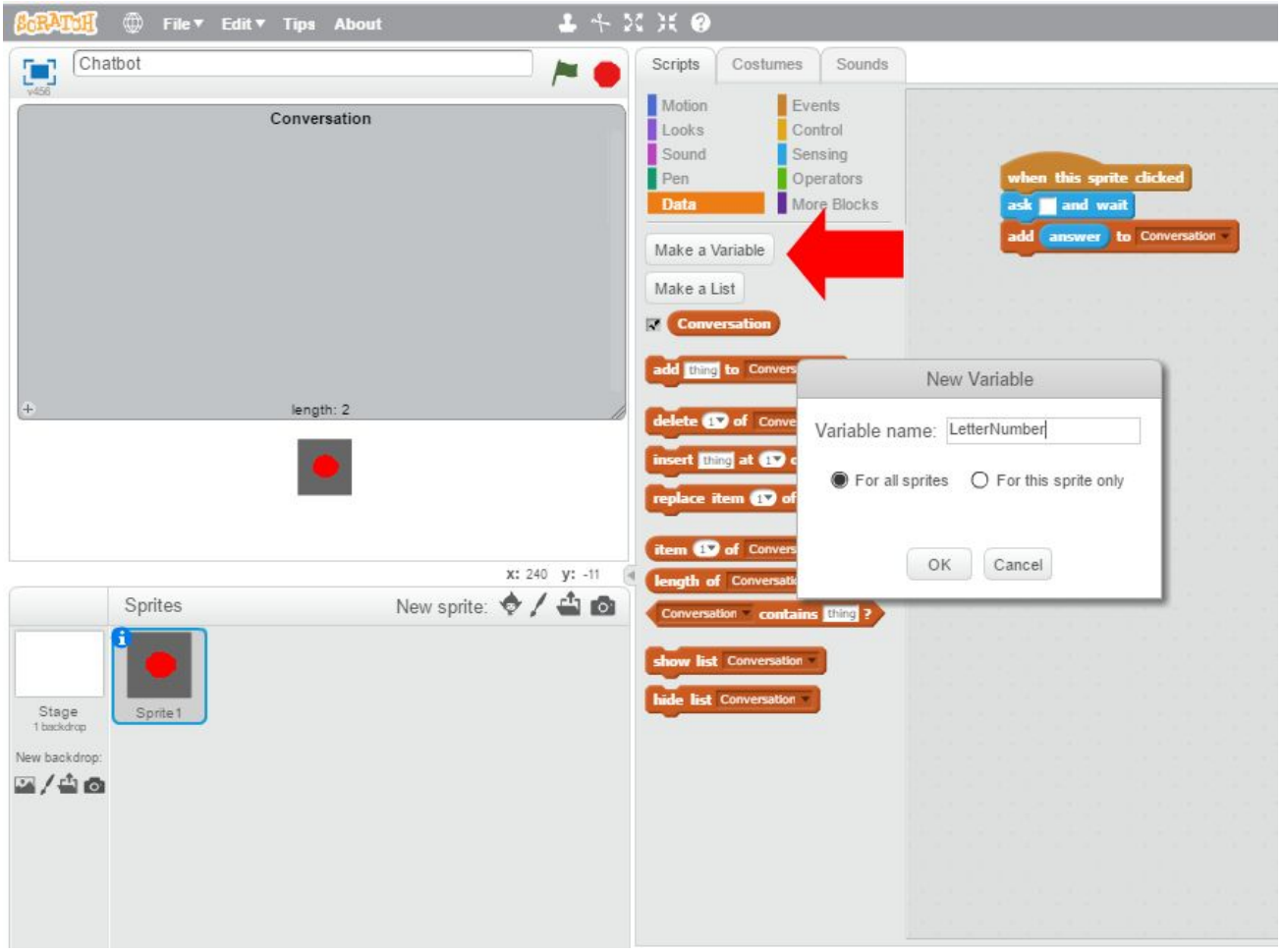
"Do you like playing football?"

Becomes:

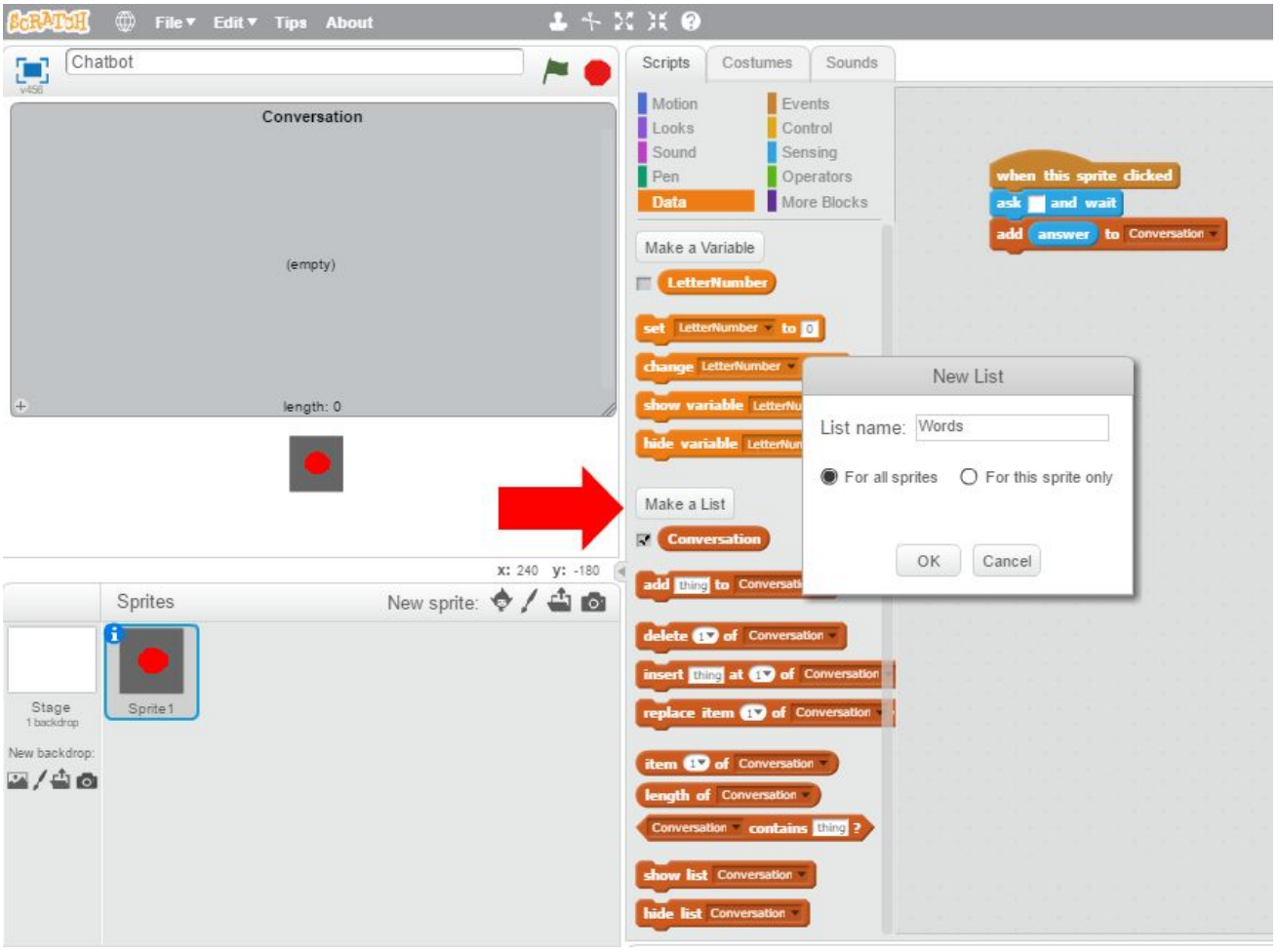
Do
You
Like
Playing
Football

Follow the screenshots below to set this up.

First we need to make a new variable from within the Data section that will allow us to record the number of words in our sentence and will help us to know whether we've processed all the words or not.

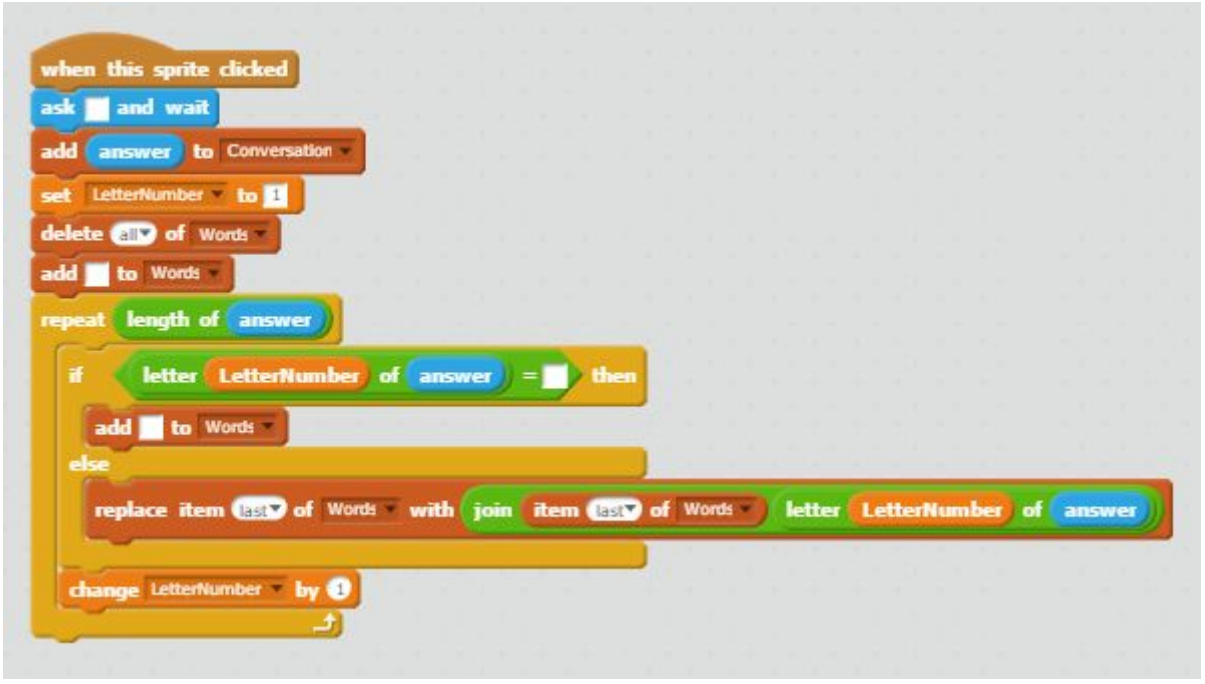


Make a new list from the Data section and call it Words. This is where we'll be storing all the chopped up single words from our sentence.



Don’t forget that when we next come to write a new sentence, we want the previous statement removed. Add these additional blocks from the Data section to reset everything from the previous conversation.

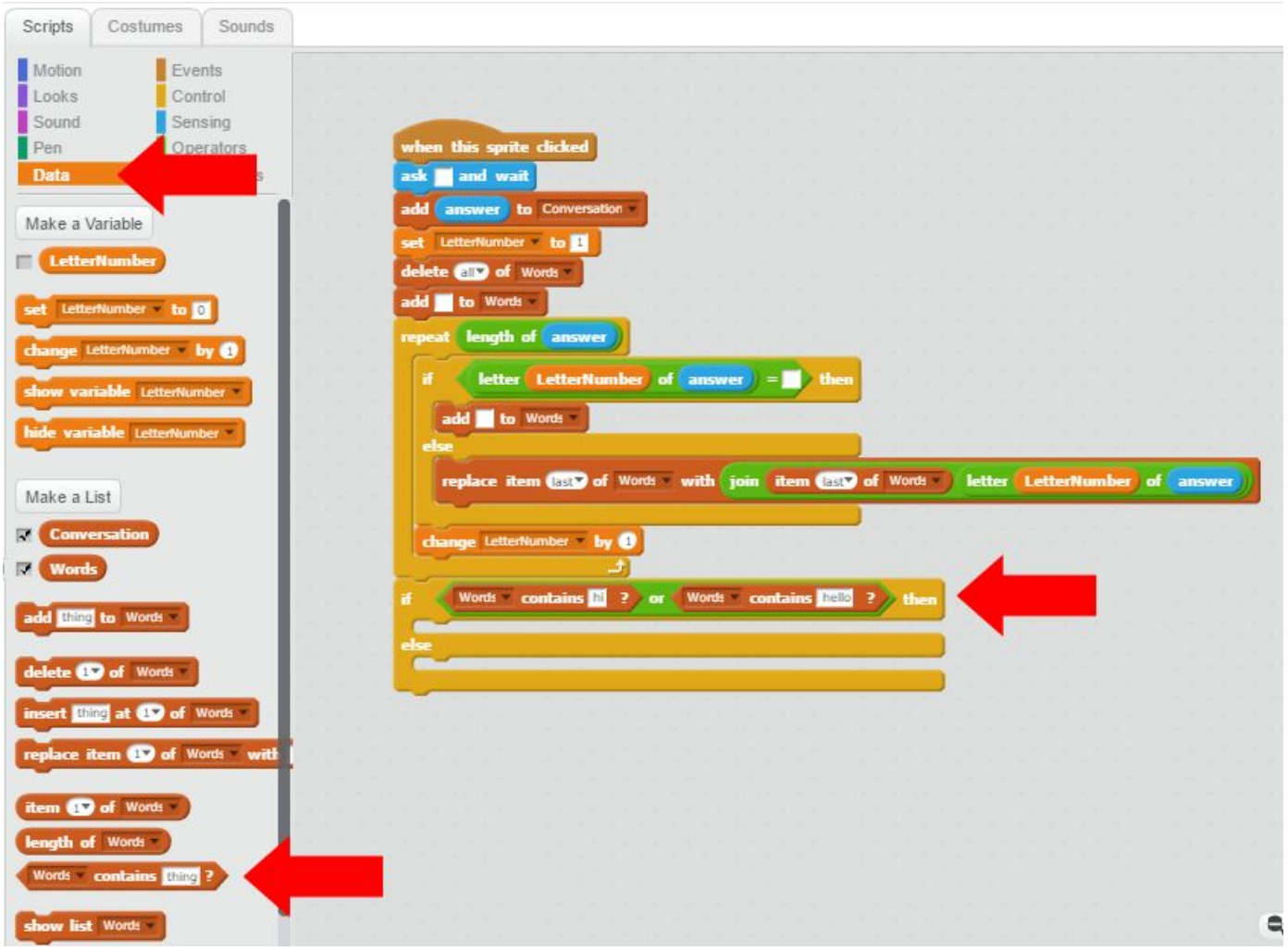
The finished blocks should look like this:



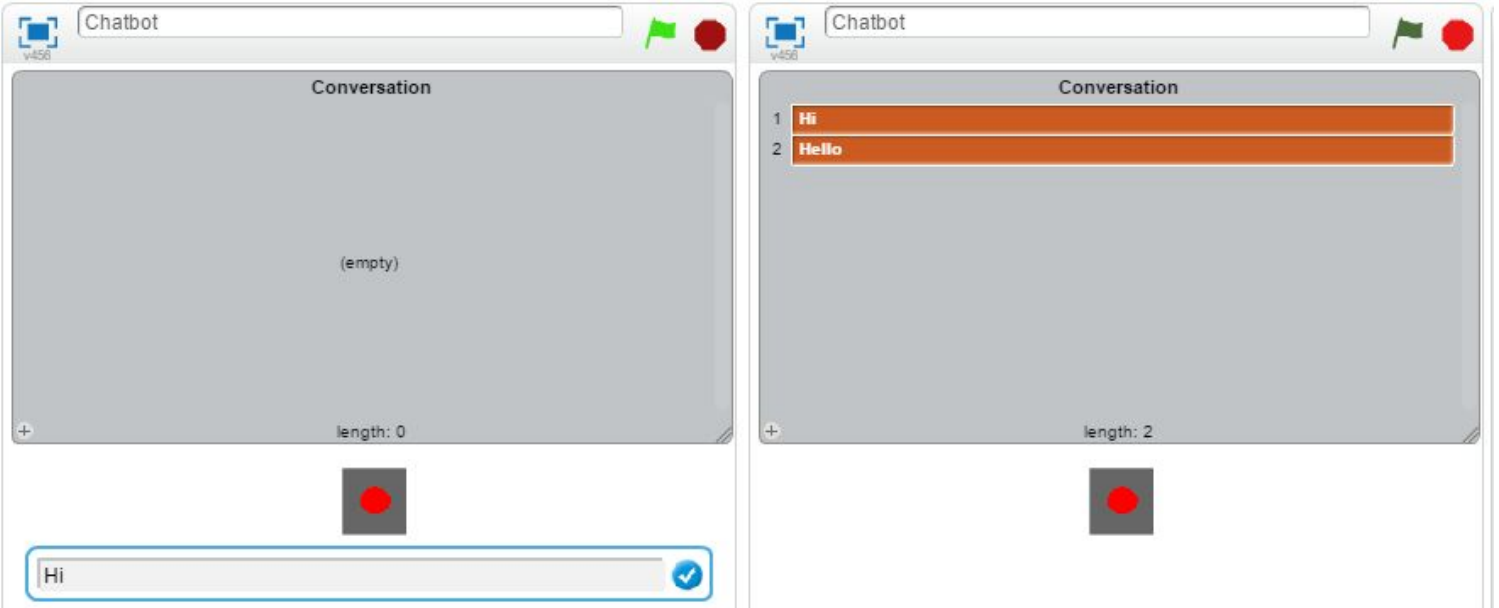
Step 4:

Now the fun part. The step can be repeated over and over again to add more intelligence to the chatbot. The more logic we add to it the cleverer it appears to become. Let's start with something basic to begin with, a simple greeting.

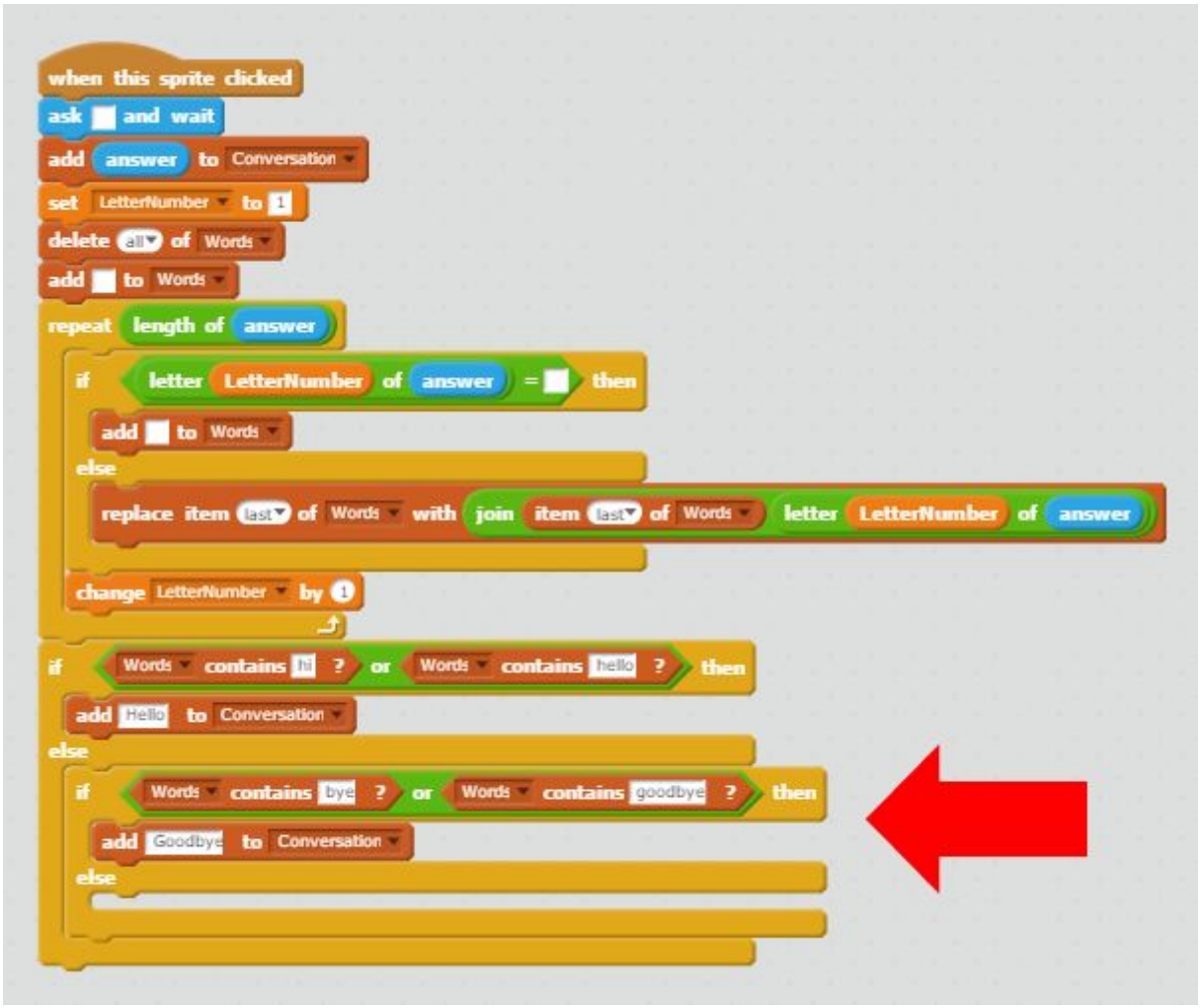
We first start off with an IF/ELSE block. Within the IF criteria we take our list of chopped up words and check it to see if contains anything that resembles a greeting, for example "Hi" or "Hello". If our list contains any of those words then any blocks contained in the IF statement will be used, if not then the program moves on running the blocks in the ELSE section instead.



Give it a test. On entering the word ‘Hi’ into the input box we should get a response of ‘Hello’ back from our chatbot.



From now on everything works off the IF/ELSE block and the more we add, the more our chatbot is able to understand. Let's add a response when it's time to say goodbye. This is done in exactly the same way as the greeting but the program is looking for "Bye" or "Goodbye" in our list of chopped up words.



Ideas to Try:

1. Now that we can get our chatbot to respond, how can we get it to respond with a response randomly selected from a list? This way, not even the user will know what the chatbot is going to say. This is where the fun really begins.

Hint:

- Create a variable called "Response".
- Create a list of responses and use the "Item Random of…" block for our new list of responses.

- Assign our Response variable the randomly selected item from the list
- Add the variable to the conversation as we have been doing before.

2. How can we get it to randomly generate a silly name for itself, based on three variables all joined together?

Hint:

- Create three lists.
- Select one random item from each list, join them together and display it as a response in the conversation window.

3. If the chatbot isn't able to provide a suitable response to our statement, how can we get it to reply saying something along the lines of "I'm sorry, ask me again later."?

Hint:

- Look for the last ELSE block.