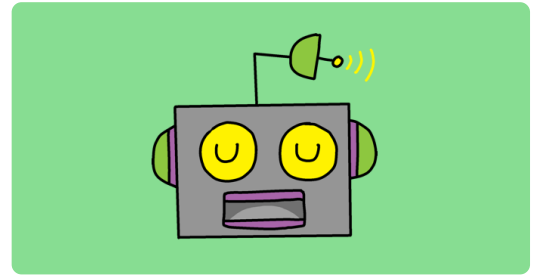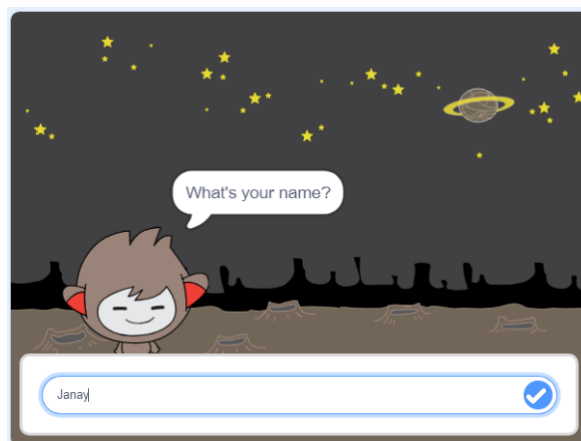# Projects

Chatbot

Learn how to program your own talking robot!

Step 1   Introduction

You are going to learn how to program a character that can talk to you! A character like that is called a chat robot, or chatbot.

What you will make

ℹ️ **What you will need**

Hardware

- Computer capable of running Scratch 3

Software

- Scratch 3 (either **online (https://rpf.io/scratchon)** or **offline (https://rpf.io/scratchoff)**)

Downloads

- **Find files to download here (http://rpf.io/p/en/chatbot-go)**.

What you will learn

- Use code to concatenate strings in Scratch
- Know that variables can be used to store user input
- Use conditional selection to respond to user input in Scratch

Additional notes for educators

You can find the **completed project here** [(http://rpf.io/p/en/chatbot-get)](http://rpf.io/p/en/chatbot-get).

Step 2   Your chatbot

Before you start creating your chatbot, you need to decide what its personality is. **Think about the following questions**:

- What is the chatbot's name?
- Where does it live?
- Is it happy? Serious? Funny? Shy? Friendly?
- What does it like and dislike?

Open the Chatbot Scratch starter project.

Online: open the starter project at **rpf.io/chatbot-on (http://rpf.io/chatbot-on)**.

If you have a Scratch account you can make a copy by clicking **Remix**.

Offline: download the starter project from **rpf.io/p/en/chatbot-go (http://rpf.io/p/en/chatbot-go)** and then open it using the offline editor.

If you need to download and install the Scratch offline editor, you can find it at **https://rpf.io/scratchoff (rpf.io/scratchoff)**.

The starter project contains four character sprites. One of these will be your chatbot.

You can either use the sprite that is already selected, or choose a different one by clicking the one you want and then selecting the **show** or **hide** eye icon from the Sprite menu below the stage.
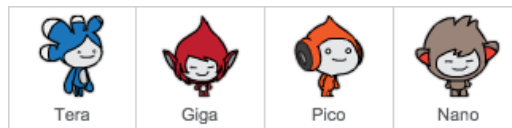
Step 3   A talking chatbot

Now that you have a chatbot with a personality, you're going to program it to talk to you.

Click on your chatbot sprite, and add this code to it so that `when it's clicked`, it `asks for your name` and then `says "What a lovely name!"`.


```
when this sprite clicked
ask  What's your name?  and wait
say  What a lovely name!  for  2  seconds
```
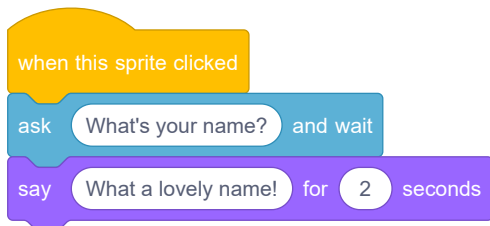
Click on your chatbot to test your code. When the chatbot ask for your name, type it into the box that appears at the bottom of the Stage, and then click on the blue mark, or press Enter.
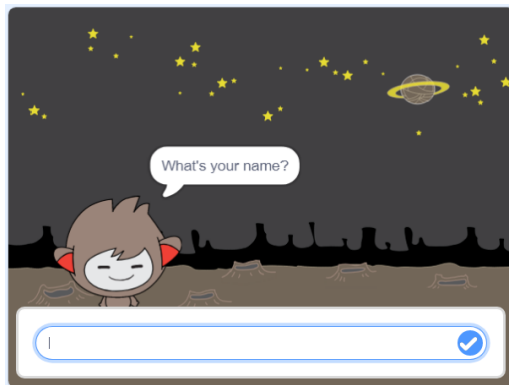
Right now, your chatbot replies "What a lovely name!" every time you answer. You can make the chatbot's reply more personal, so that the reply is different every time a different name is typed in.
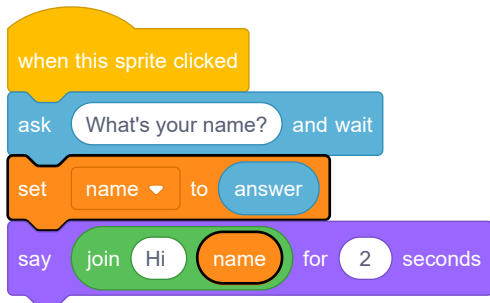
Change the chatbot sprite's code to **join** "Hi" with the **answer** to the "What's your name?" question, so that the code looks like this:

```
when this sprite clicked
ask  What's your name?  and wait
say  join  Hi  answer  for  2  seconds
```
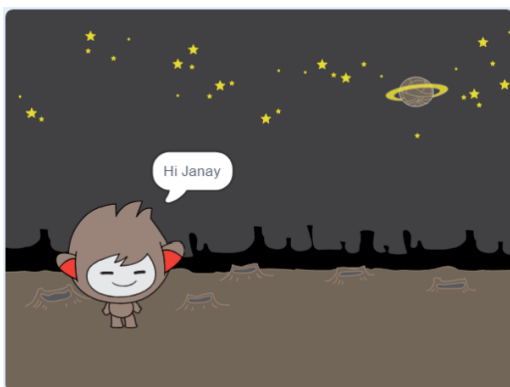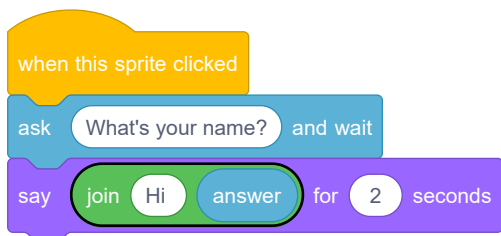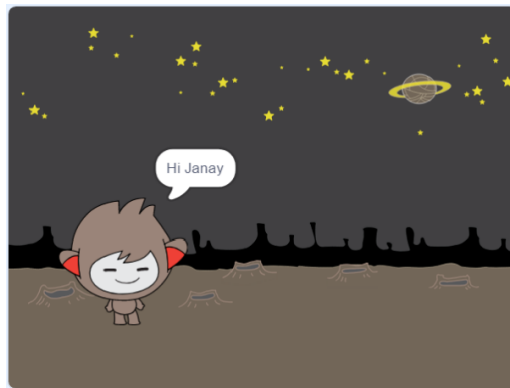
By storing the answer in a **variable**, you can use it anywhere your project.

Create a new variable called **name**.

Now, change your chatbot sprites's code to set the `name` variable to `answer`:


```
when this sprite clicked
ask  What's your name?  and wait
set  name ▼  to  answer
say  join  Hi  name  for  2  seconds
```

Your code should work as before: your chatbot should say hi using the name you type in.

Test your program again. Notice that the answer you type in is stored in the `name` variable, and is also shown in the top left-hand corner of the Stage. To make it disappear from the Stage, go to the `variables` blocks section and click on the box next to `name` so that it is not marked.

Challenge: more questions

Program your chatbot to ask another question. Can you store the answer to this new question in a new variable?
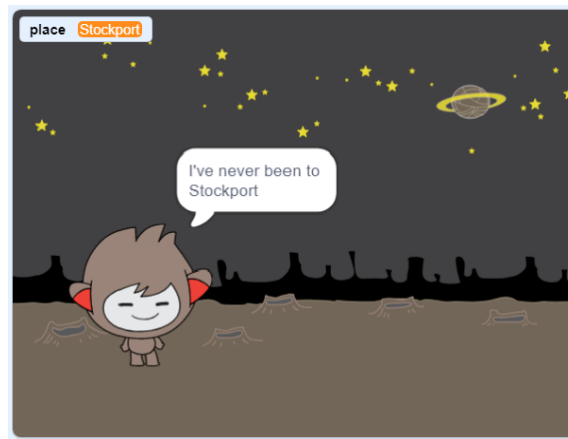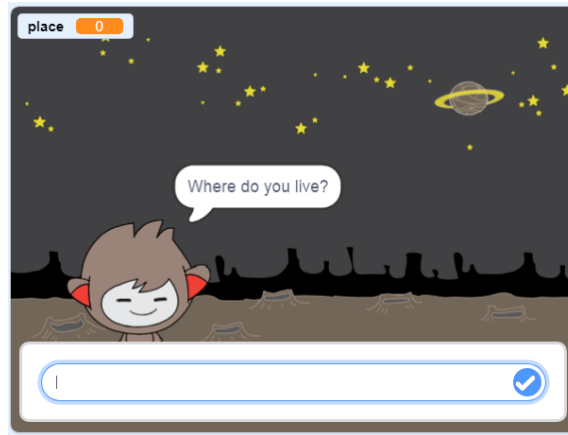
Step 4   Making decisions

You can program your chatbot to decide what to do based on the answers it receives.

First, you're going to make your chatbot ask a question that can be answered with "yes" or "no".

Change your chatbot's code. Your chatbot should ask the question "Are you OK name", using the `name` variable. Then it should reply "That's great to hear!" `if` the answer it receives is "yes", but say nothing if the answer is "no".
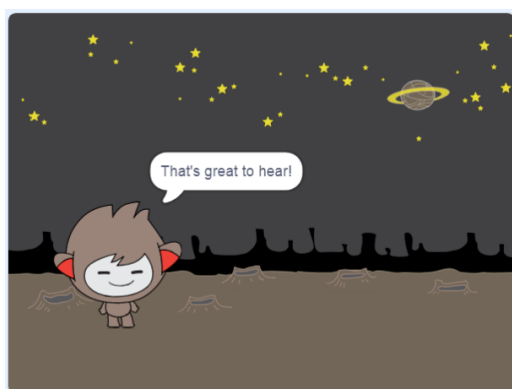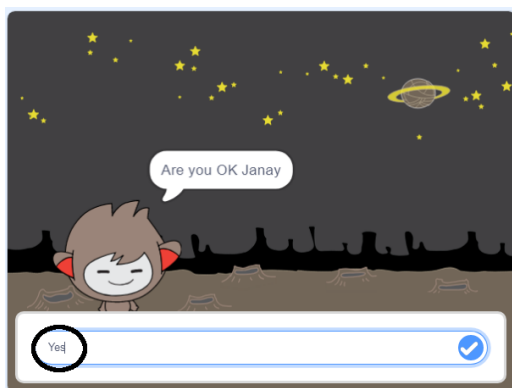

```
when this sprite clicked
ask  What's your name?  and wait
set  name ▼  to  answer
say  join  Hi  name  for  2  seconds
ask  join  Are you OK  name  and wait
if  answer = yes  then
    say  That's great to hear!  for  2  seconds
```

To test your new code properly, you should test it **twice**: once with the answer "yes", and once with the answer "no".

At the moment, your chatbot doesn't doesn't say anything to the answer "no".
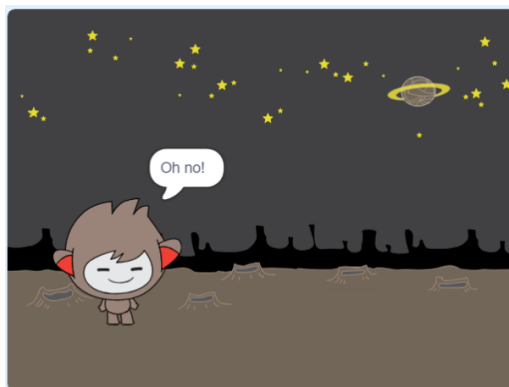
Change your chatbot's code so that it replies "Oh no!" if it receives "no" as the answer to "Are you OK name".

Replace the `if, then` block with an `if, then, else` block, and include code so the chatbot can `say "Oh no!"`.


```
when this sprite clicked
ask  What's your name?  and wait
set  name ▾  to  answer
say  join  Hi  name  for  2  seconds
ask  join  Are you OK  name  and wait
if   answer  =  yes  then
    say  That's great to hear!  for  2  seconds
else
    say  Oh no!  for  2  seconds
```
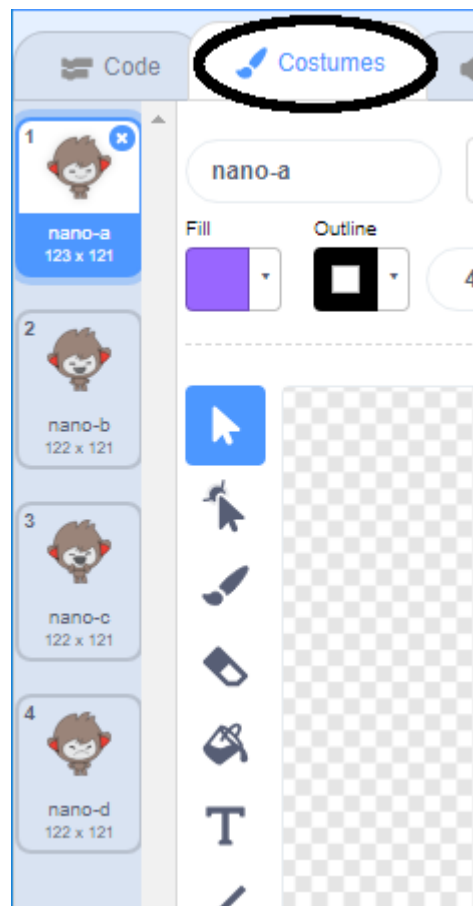
Test your code. You should get a different response when you answer "no" and when you answer "yes": your chatbot should reply with "That's great to hear!" when you answer "yes" (which is not case-sensitive), and reply with "Oh no!" when you answer **anything else**.
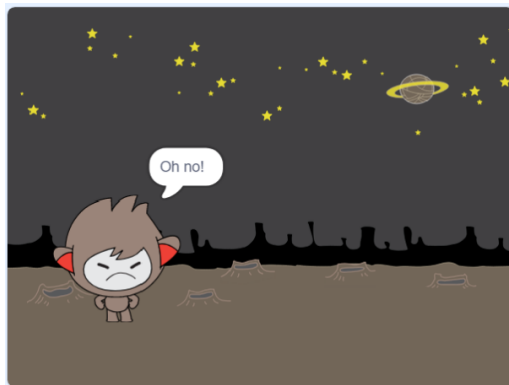

You can put any code inside an `if, then, else` block, not just code to make your chatbot speak!

If you click your chatbot's **Costumes** tab, you'll see that there is more than one costume.

Code | Costumes

1 nano-a
123 x 121

2 nano-b
122 x 121

3 nano-c
122 x 121

4 nano-d
122 x 121

nano-a

Fill   Outline

4

Change your chatbot's code so that the chatbot switches costumes when you type in your answer.

Change the code inside the `if, then, else` block to `switch costume`.

Test and save your code. You should see your chatbot's face change depending on your answer.

Have you noticed that, after your chatbot's costume has changed, it stays like that and doesn't change back to what it was at the beginning?

You can try this out: run your code and answer "no" so that your chatbot's face changes to an unhappy look. Then run your code again and notice that your chatbot does not change back to looking happy before it asks your name.
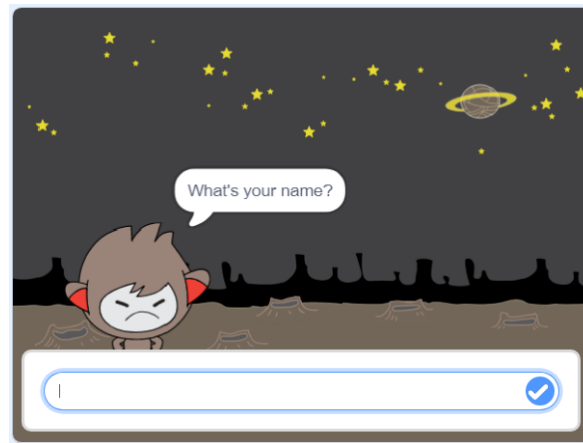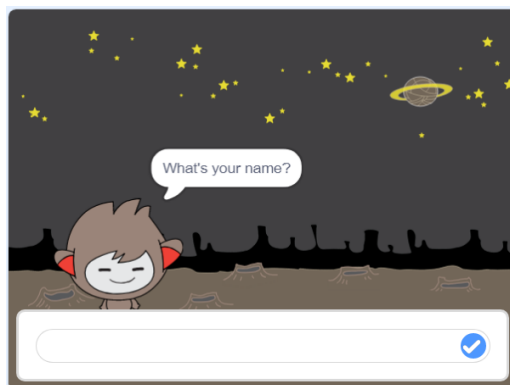

To fix this problem, add to the chatbot's code to `switch costume` at the start `when the sprite is clicked`.


```
when this sprite clicked
switch costume to ( nano-a ▼ )
ask ( What's your name? ) and wait
```
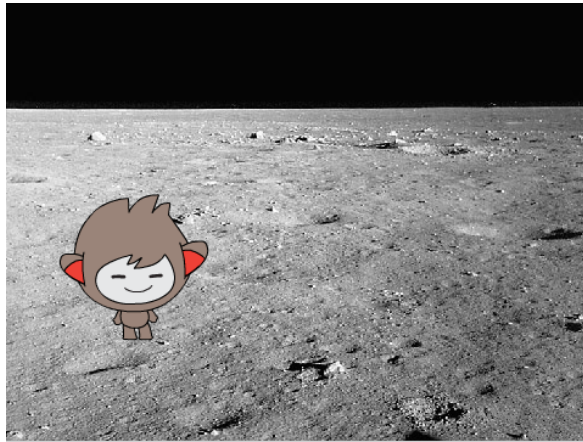
Challenge: more decisions

Program your chatbot to ask another question that can be answered with "yes" or "no". Can you make your chatbot respond differently depending on which answer it receives?
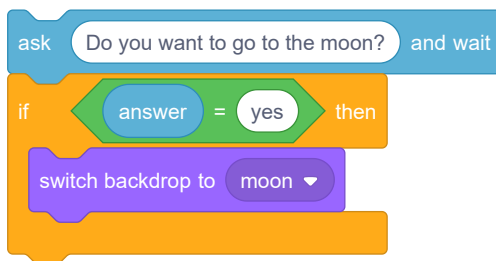
Step 5   Changing location

You can also program your chatbot to change its location!

Can you program your chatbot to ask "Do you want to go to the moon", and then change the backdrop when the answer is "yes"?

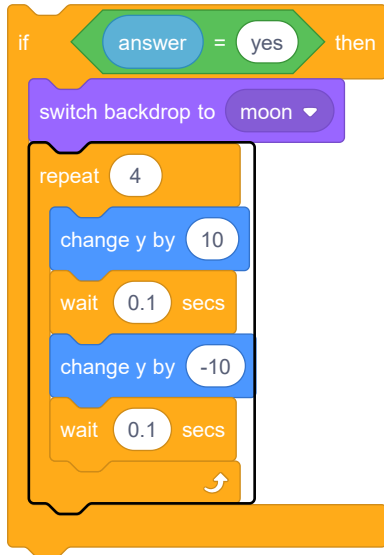This is what your code should look like:

Now you need to make sure that your chatbot starts in the right location when you click on it to talk to it. Add this block to the top of your chatbot code:

Test your program, and answer "yes" when the chatbot asks if you want to go to the moon. You should see that the chatbot's location changes.

You can also add the following code inside the new 🎁 block to make the chatbot jump up and down four times if you answer "yes":
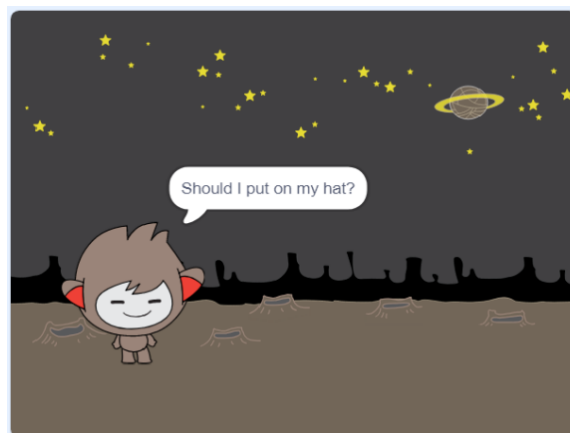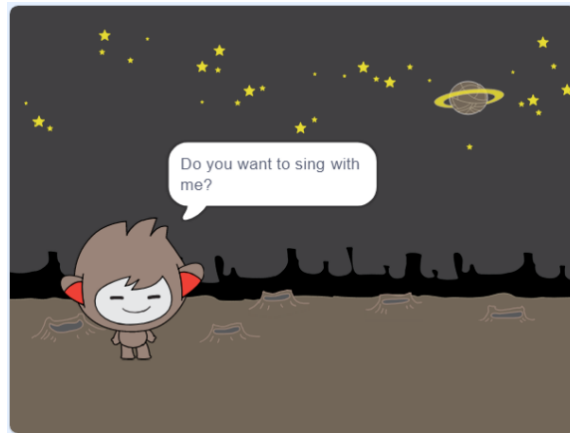
```
if ⟨ answer = yes ⟩ then
    switch backdrop to moon ▼
    repeat 4
        change y by 10
        wait 0.1 secs
        change y by -10
        wait 0.1 secs
```

Challenge: finish your chatbot

Use what you've learned to finish creating your interactive chatbot. Here are some ideas:

Once you've finished your chatbot, get your friends to have a conversation with it! Do they like your character? Can they spot any problems?

Step 6   What next?

Try the **Paint box** **(https://projects.raspberrypi.org/en/projects/paint-box?utm_source=pathway&utm_me dium=whatnext&utm_campaign=projects)** project, where you will create your own painting program!

You will click on the green flag to start, and you'll use the mouse to move the pencil and hold down the left mouse button to draw. Clicking on a colour will change pencil colours, and clicking on the eraser will change to the eraser!