

### Text to speech translator (Skill level: easy)

Use Scratch 3.0 and the blocks of 'Text to speech' and 'Translate'. To add these blocks, click on the blue 'Add extension' box on the bottom left of the Scratch screen. In this program students can explore text to speech blocks and translating into different languages.



You can either have students build the program from scratch (pun intended!) or if they find the task challenging, students can look at the completed code first to learn how it works. It can be found here: <https://scratch.mit.edu/projects/338489965/editor/>

1. The first step is to create a simple program that uses an 'Ask' block to ask the user what they want to translate. Test the program to make sure the answer is displayed on screen by the sprite.

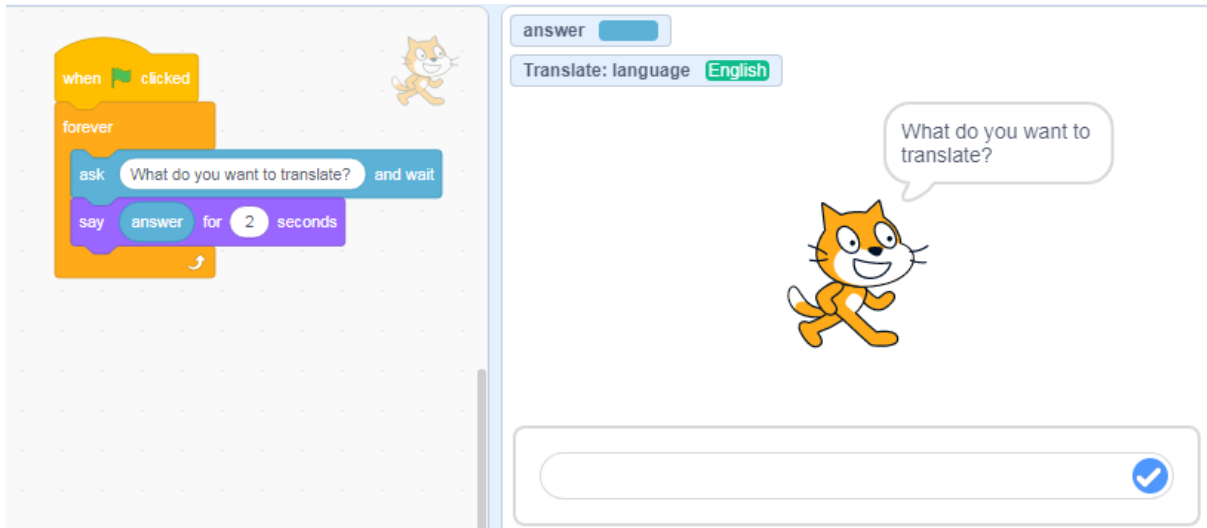
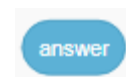
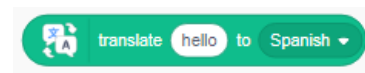


Image: Scratch program showing Sprite 1 coding blocks to input text

2. The next step is to add the translate block to replace the answer.
3. Change the language to your choice of language.
4. The answer block replaces the 'hello' text in the translate block. The text on screen will now be in the language chosen.



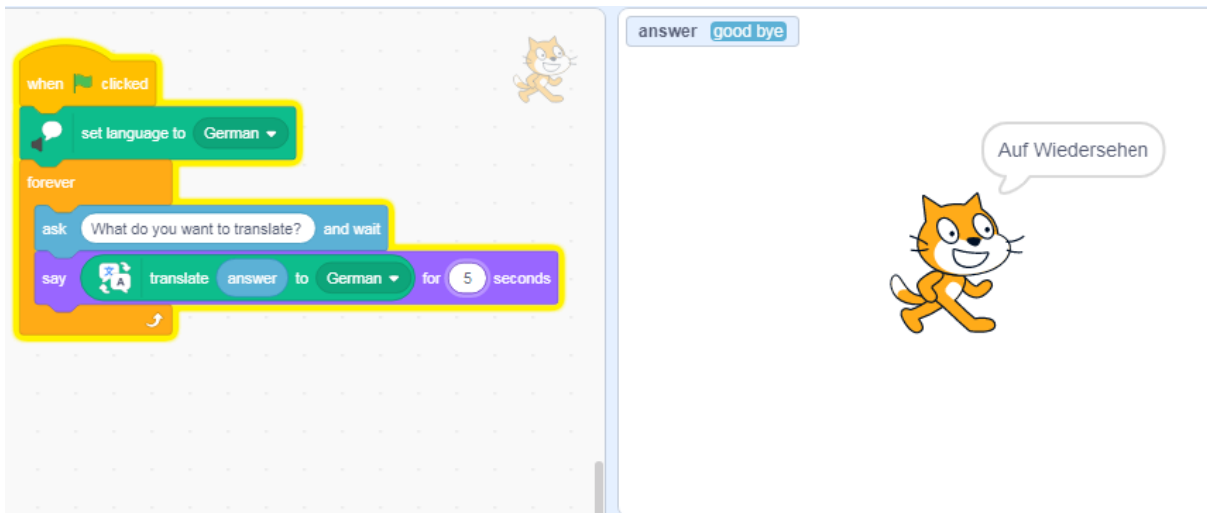
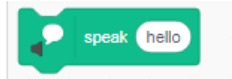


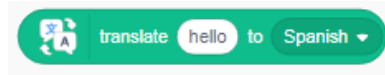
Image: Scratch program showing Sprite 1 coding blocks to translate input text

5. The final step is to get the answer spoken in the translated language.

- Add the speak block.



- Replace the hello text with the translate block.



- Replace the hello text of that block with the answer block.



So now when you type in the input bar, the text (answer) is translated in text on screen, and then spoken in the selected language.

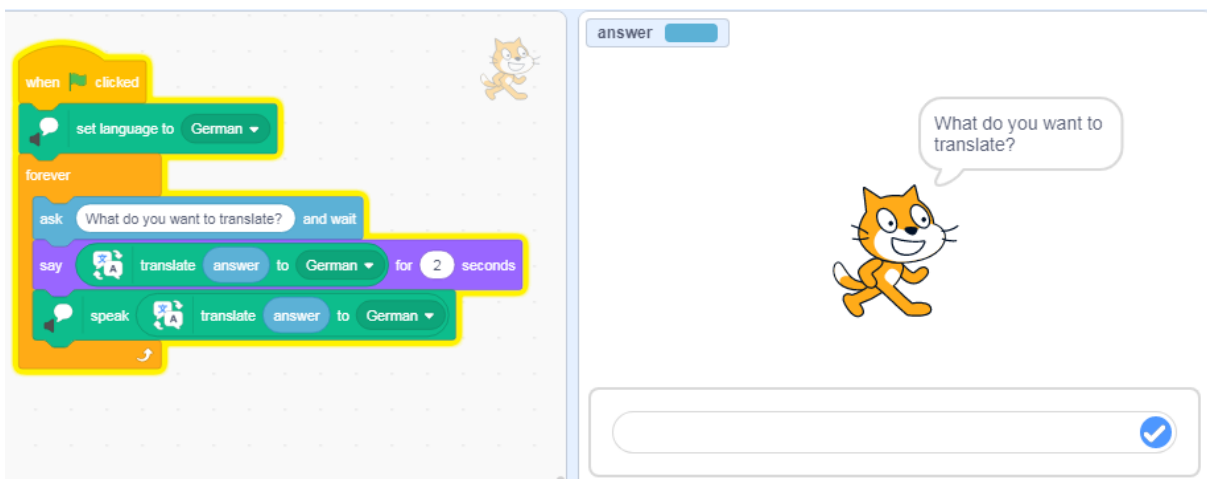


Image: Scratch program showing Sprite 1 coding blocks to translate input text and speak translated answer

*Some ideas for remix*

1. Change the language.
2. Change sprite and background.
3. Explore and build your own program.
4. Create your own code block to enable the user to change language to another language.

Sample code: [Text to speech translator](#)

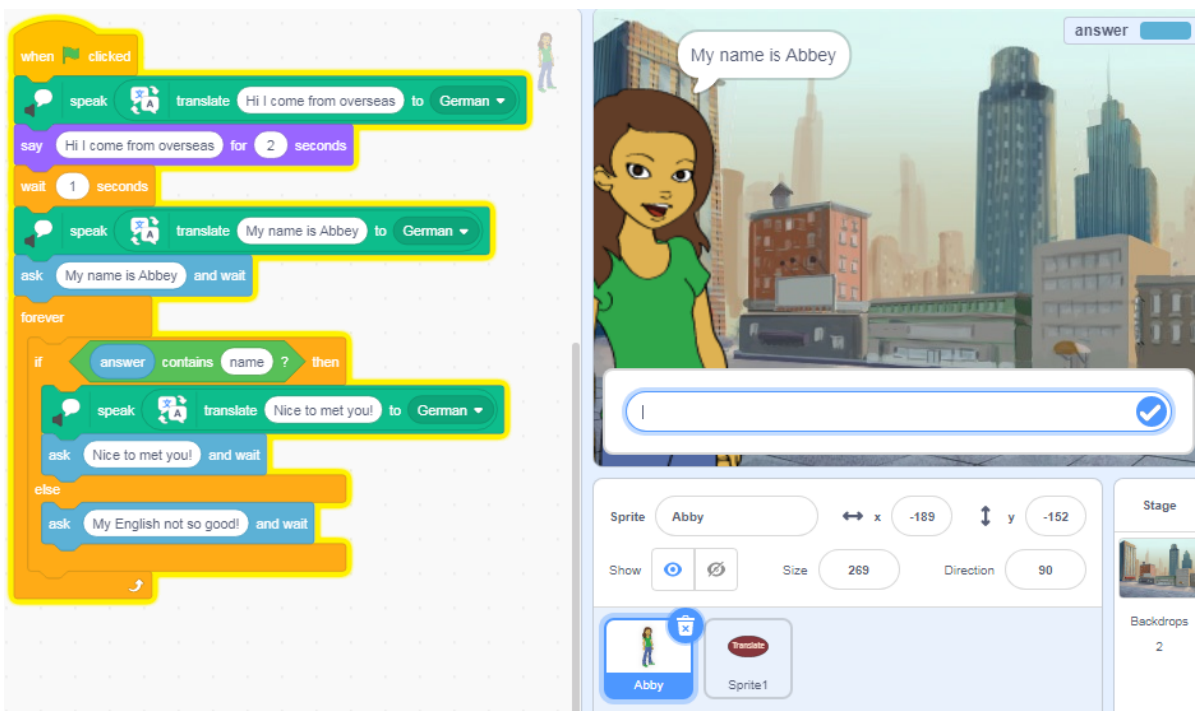
### Chatbot with language translation (Skill level: medium)

Use Scratch 3.0 and the additional blocks of 'Text to speech' and 'Translate'.

1. Create a chatbot to chat with a visitor from overseas. In coding this program, text inputs (a string of text) are referred to as the 'answer'. The chatbot responds if the 'answer' contains a particular word, or else a generic response is given. The more words that are coded with a relevant response, the higher the chance a good match will be made with the questions input by the user. The program runs forever. [Note: in the program the user is actually asking the question but the program **deals with this as the 'answer'**, just a limitation of Scratch and its logic.]

Two sample programs have been provided to help scaffold the activity:

- [Foreign Chat](#)
  - [Foreign Chat basic 1 Question ONLY](#)
2. It may be useful for students to explore the fully functional program [Foreign Chat](#) first to input their question and see how the chatbot responds. Students could rate the chatbot on how many of their questions it was able to provide with a 'smart' response.
  3. A basic code [Foreign Chat basic 1 Question ONLY](#) is provided for students to remix. In remixing the project students could:
    - change the language
    - change the sprite and background
    - change the word contained in the answer and test
    - add new **if/else** blocks and duplicate the existing **if** code block. (To do this nest the new **if/else** block in the previous **else** option. Continue this process for each additional block until the last **else** has the generic answer for **no match found**.)



The image shows a Scratch 3.0 project for a chatbot. On the left, the code is as follows:

```

when clicked
  speak translate Hi I come from overseas to German
  say Hi I come from overseas for 2 seconds
  wait 1 seconds
  speak translate My name is Abbey to German
  ask My name is Abbey and wait
  forever
    if answer contains name ? then
      speak translate Nice to met you! to German
      ask Nice to met you! and wait
    else
      ask My English not so good! and wait
  
```

On the right, the chatbot's interface is visible. It features a character named 'Abby' with a speech bubble that says 'My name is Abbey'. Below the character is a text input field with a checkmark icon. The 'answer' variable is set to 'My name is Abbey'. The Stage area shows the chatbot's sprite and a 'Translate' block. The Stage Backdrops area shows a cityscape backdrop.

Image: Scratch program showing Sprite (Abby) coding blocks to respond to input text in selected language

Discuss:

- In what ways is the program smart?
- What are some limitations of the program?
- How well did the translation work?

### Automating a conversation (Skill level: med to hard)

1. Use Scratch 3.0 and the additional blocks of 'Text to speech' and 'Translate' to program a conversation between two people. Try to make the program as smart as possible.  
Note: this project does not include user input. Instead, it focuses on automating a conversation. For a focus on a program that requires user input consider chatbot or translate projects as shown above.

In this program skills required include:

- **Select sprites** (characters) and program with **say** blocks.
- **Randomise** responses using **Random** function (stored as a variable).
- **Create, store and use a variable**, make a list to store phrases.
- Use **broadcast message** to trigger responses.
- Use **loops** to continue the conversation.
- Use **'text to speech'** to read all conversation text.
- Use **translate** to translate one sprite's responses in the language being studied.

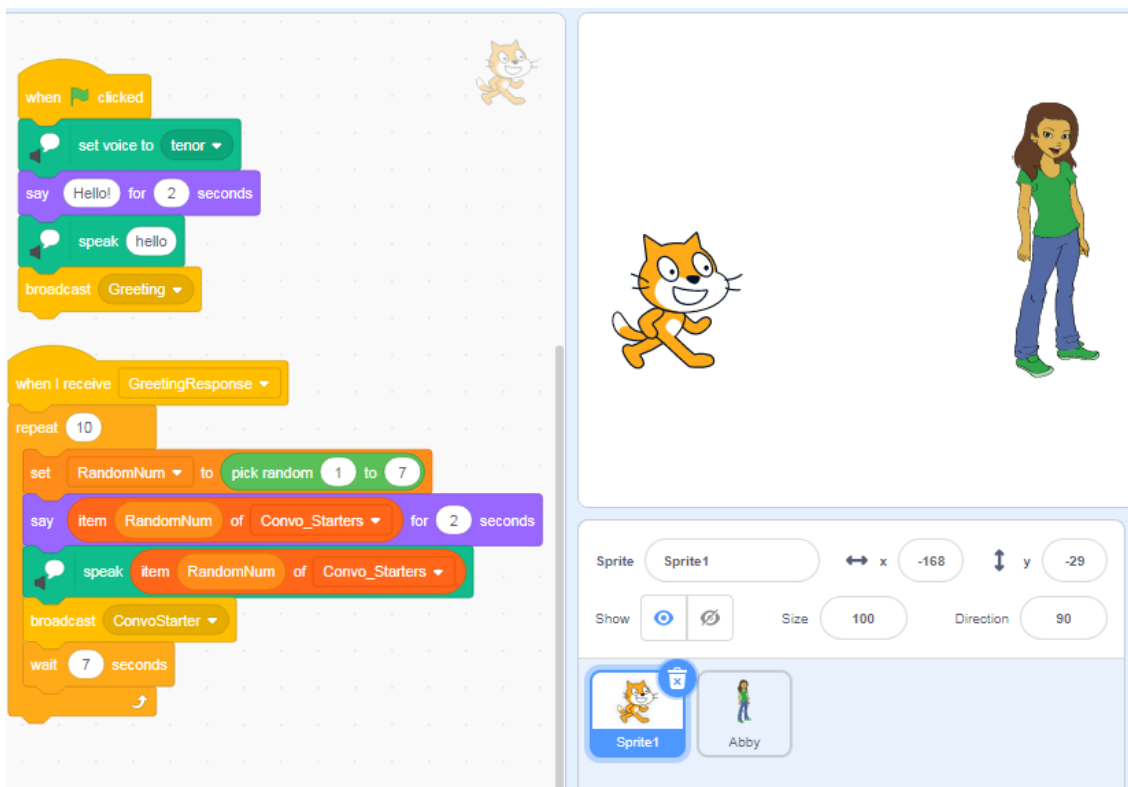


Image: Scratch program showing Sprite 1 coding blocks

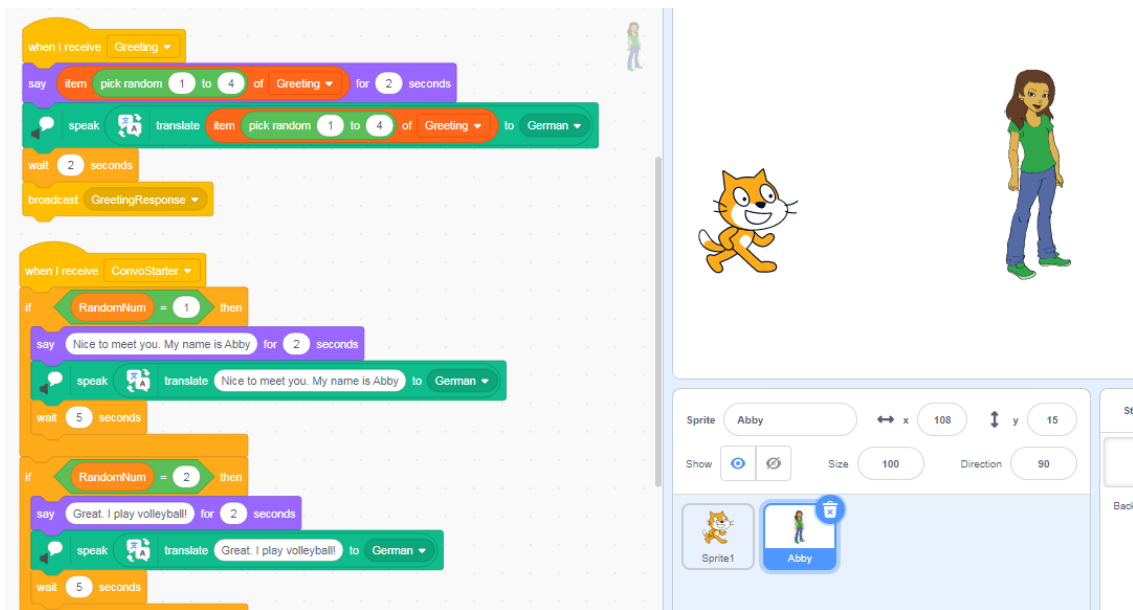


Image: Scratch program showing part of Sprite 2 (Abby) coding blocks

2. Use the sample code to explore text to speech and translate functions in Scratch. Play the program and then look inside to reveal how it was coded.

**Sample code:** Random convo: [Translate \(German\)](#)

*Some ideas for remix:*

- This example translates to German but you could remix in another language.
- Change greetings and convo starters and responses stored as lists.
- Work out a way not to repeat phrases used in the conversation.
- Create your own conversations using your own sprites.