Intelligent Assistant with Keywords (ML4K)

Activity Guide

Version of October 27, 2022

Christina Gardner-McCune
University of Florida

David S. Touretzky
Carnegie Mellon University

This work was funded by a grant from NEOM Company.

NEOM

Intelligent Assistant with Keywords (ML4K)

Summary

Create a simulated intelligent assistant in Scratch that responds to voice input, modeled after Siri or Alexa. This is a simulation that outputs predetermined responses to selected keywords in the input; real assistants do much more.

Grade Bands: **3-5, 6-8, 9-12**

Learning Objectives

Construct a keyword-based intelligent assistant using the Machine Learning for Kids version of Scratch. Students will be able to:

- Use the Speech to Text extension to incorporate speech recognition into a program.
- Scan recognized utterances for keywords and construct responses using Scratch's string operations.
- Use the Text to Speech extension to produce spoken responses.

Materials/Resources Required

- Laptop or notebook with a microphone and speaker
- Chrome web browser
- Access to the Machine Learning for Kids (ML4K) web site (https://machinelearningforkids.co.uk). Note: you cannot use this code with Cognimates because its speech to text extension is different or with Scratch. There is a separate version of this activity for Cognimates.

Guidelines and Standards

AI4K12 Guidelines

Big Idea 4: Natural Interaction

4-A-iv: Natural language (Applications)

Other Relevant Educational Standards

Computer Science Teacher Association Standards

1A-AP-10 K-2	Develop programs with sequences and simple loops, to experess ideas or address a problem.
1A-AP-14 K-2	Debug (identify and dix) errors in an algorithm or program that includes sequences and simple loops.
1A-AP-09 K-2	Model the way programs store and manipulate data by using numbers or other symbols to represent information
1B-AP-09 3-5	Create programs that use varibales to store and modify data.
1B-AP-10 3-5	Create programs that include sequences, events, loops, and conditionals.
2-AP-11 6-8	Create clearly named variables that represent different data types and perform operations on their values.
2-AP-12 6-8	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-14 6-8	Create procedures with parameters to organize code and make it easier to reuse.
2-AP-16 6-8	Incorporate existing code, media, and libraries into original programs, and give attribution.
3A-AP-14 9-10	Use lists to simplify soultions, generalizing computational problems instead of repeatedly using simple variables.
3A-AP-15 9-10	Justify the selection of specific structures when tradeoffs involves implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
3A-AP-18 9-10	Create artifacts by using procedures within a program, combinations of data and procedures, or interrelated programs.
3A-AP-16 11-12	Demonstrate code reuse by creating programming solutions using libraries and APIs.

Other Relevant Activities

1. Intelligent Assistant with Keywords (Cognimates)
2. Chatbot with BERT
3. SpeechDemo

Vocabulary Terms

Flashcards for these vocabulary terms are located at the end of this guide.

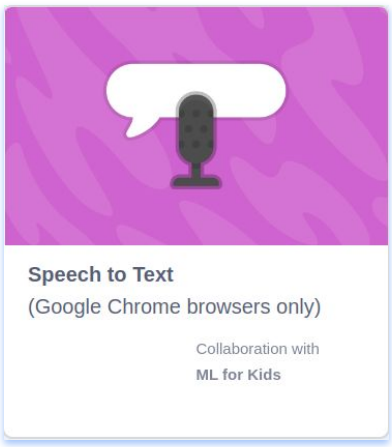
- Intelligent assistant
- Keyword
- Alexa
- Siri

Preparation

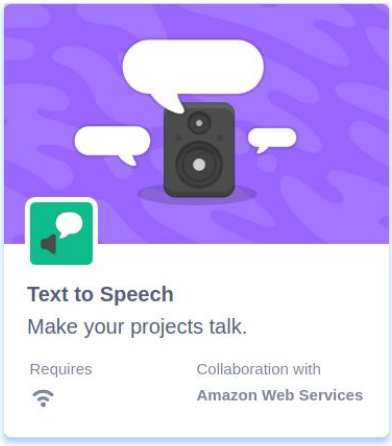
Setting up to Use Machine Learning for Kids

1. Here is a direct link to the Scratch3 page at https://scratch.machinelearningforkids.co.uk/ Note that you must use this version of Scratch, not the one at scratch.mit.edu: https://machinelearningforkids.co.uk/scratch3/. You can also get to this page by going to the home page and clicking on “Get started” and “Try it now”, then select “Pretrained” from the top nav bar and again “Try it now”.
2. In the bottom left corner of the Scratch window is a blue “Add Extension” button. Click on that.

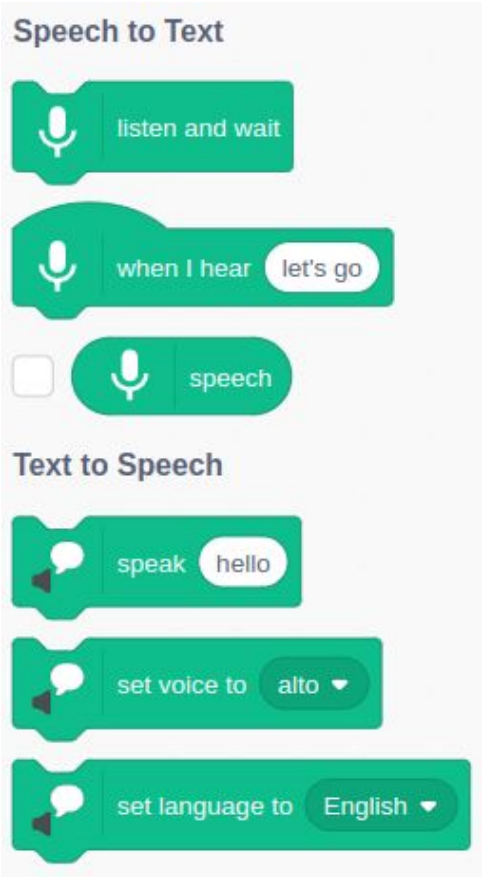
3. Scroll down until you see the “Speech to Text” extension, and select that:



4. Click on the “Add Extension” button again, scroll down until you see the “Text to Speech” extension, and select that:



5. This leaves the learner with two new sets of blocks in the palette:

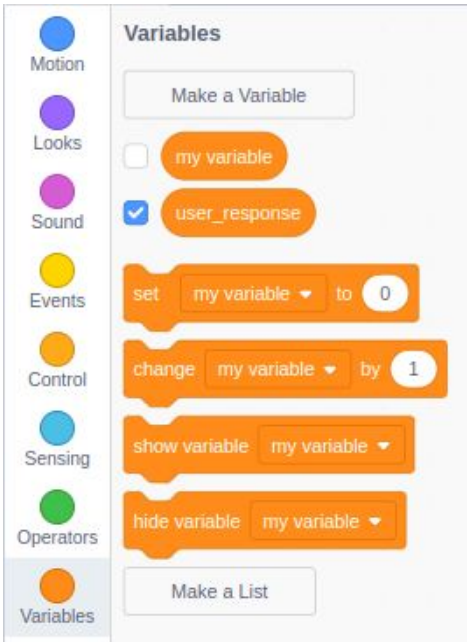


Writing the Keyword Response Functions

The intelligent assistant works by accepting input from the user, scanning it for keywords, and generating a canned response when a keyword matches. The assistant understands several types of requests; we will write a separate block to handle each type.

All block definitions appear at the end of this section; refer to those pages when you need to create a block in the steps below.

1. Go to “Variables” and create a variable called user_response. This is what you should see:



2. Go to “My Blocks” and click on the “Make Block” button. Create the block named “keyword-response” with inputs “keyword” and “response”.

3. Enter the definition of keyword-response using the code provided at the end of this section.

Try this!

👉 4. Let’s test the program before adding more response types. Enter all the code shown under “Main program”. This consists of three chunks: one to handle the “When green flag clicked” event, one to handle the “When this sprite clicked” event, and one to handle the “When I hear” event. In this last chunk, leave out the “two-step” and “riddle” blocks because we haven’t defined those yet.

5. Click on the green flag and say “Assistant, what time is it?” You should hear the response “There’s no time like the present.” Now say “Assistant, make me a sandwich.” You should hear the response “You can’t be serious.”

6. Next will provide a way for the assistant to tell us a joke. For this we introduce a more complex block that outputs a canned message, waits for the user to say something back, and then does a keyword-based response to that. It's called "two-step". Make a block with that name; its inputs are "keyword", "response", "keyword2", "response2", and "default".

7. Enter the definition for the "two-step" block using the code provided at the end of this section. Note that whenever we want the computer to ask us something, we must first tell it to stop listening. This prevents it from hearing and responding to its own speech when it asks us a question. That's why two-step is more complicated than keyword-response.

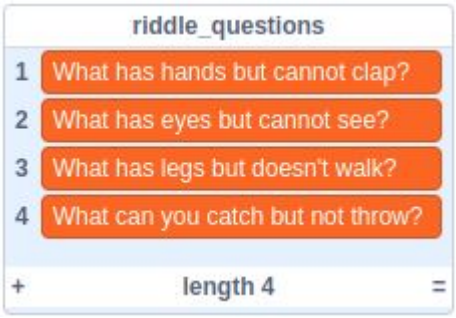
8. Add the call to two-step to the main program, after the calls to keyword-response.

Try this!

👉 9. Test the program by clicking on the green flag and saying "Assistant, please tell me a joke."

10. The last feature we'll add allows the assistant to ask us riddles. In order to use the "riddle" block you must create two list variables. Create a list variable called riddle_questions and set it to this list:
    - ★ What has hands but cannot clap?
    - ★ What has eyes but cannot see?
    - ★ What has legs but doesn't walk?
    - ★ What can you catch but not throw?

The result should look like this:



11. Create another list variable called riddle_answers and set it to this list:
    - ★ clock
    - ★ potato
    - ★ table
    - ★ cold

    The result should look like this:

12. Uncheck the boxes for riddle_questions and riddle_answers so they don't appear on the stage.

13. Create two new variables called "riddle_question" and "riddle_answer" and uncheck their boxes. Then create a new block named "riddle". Enter the definition for the riddle block using the code provided at the end of this section. Notice that riddle makes use of two-step.

14. Update the main program by inserting a "riddle" block after the "two-step" block. Now your program should exactly match the code shown in this activity guide.

Try this!

👉 15. Test your program by clicking on the green flag and saying "Assistant, ask me a riddle." Try giving the right answer. Then ask for another riddle and try giving the wrong answer.

Running the Complete Demo

Click on the green start flag to begin the demo. Begin each request with the wake word "Assistant". You can make requests such as:

- Assistant, what is the weather outside?
- Assistant, what time is it?
- Assistant, are you happy today?
- Assistant, can you tell me a joke?
- Assistant, do you know any good jokes?
- Assistant, please ask me a riddle.

To stop the demo, click on the sprite. This tells Scratch to stop listening to your microphone.

Limitations of the Demo

The demo looks for keywords and will match anything containing those characters, even if the meaning is different. For example, “Assistant, are most tents weatherproof?” will match “weather”, and “Assistant, do you go dancing sometimes?” will match “time”. We could prevent this erroneous matching by including a space on either side of the keyword. But then “<space>joke<space>” wouldn’t match “Assistant, do you know any jokes?”. And nothing we do will allow “happy” to match “Assistant, how is your happiness?”

Make It Your Own

Have students modify the code to create their own keywords and responses. For more advanced students, have them implement a “quiz” option modeled after the “riddle” block; it should pick a quiz question at random from a list the student creates. For example, it could be a trivia quiz about their school, or a quiz about their state (e.g., state motto, state bird, state flower, state capital).

Students could also modify the code to make the sprite change its appearance while it’s responding to a request, or while it’s waiting for you to supply an answer to a joke or riddle.

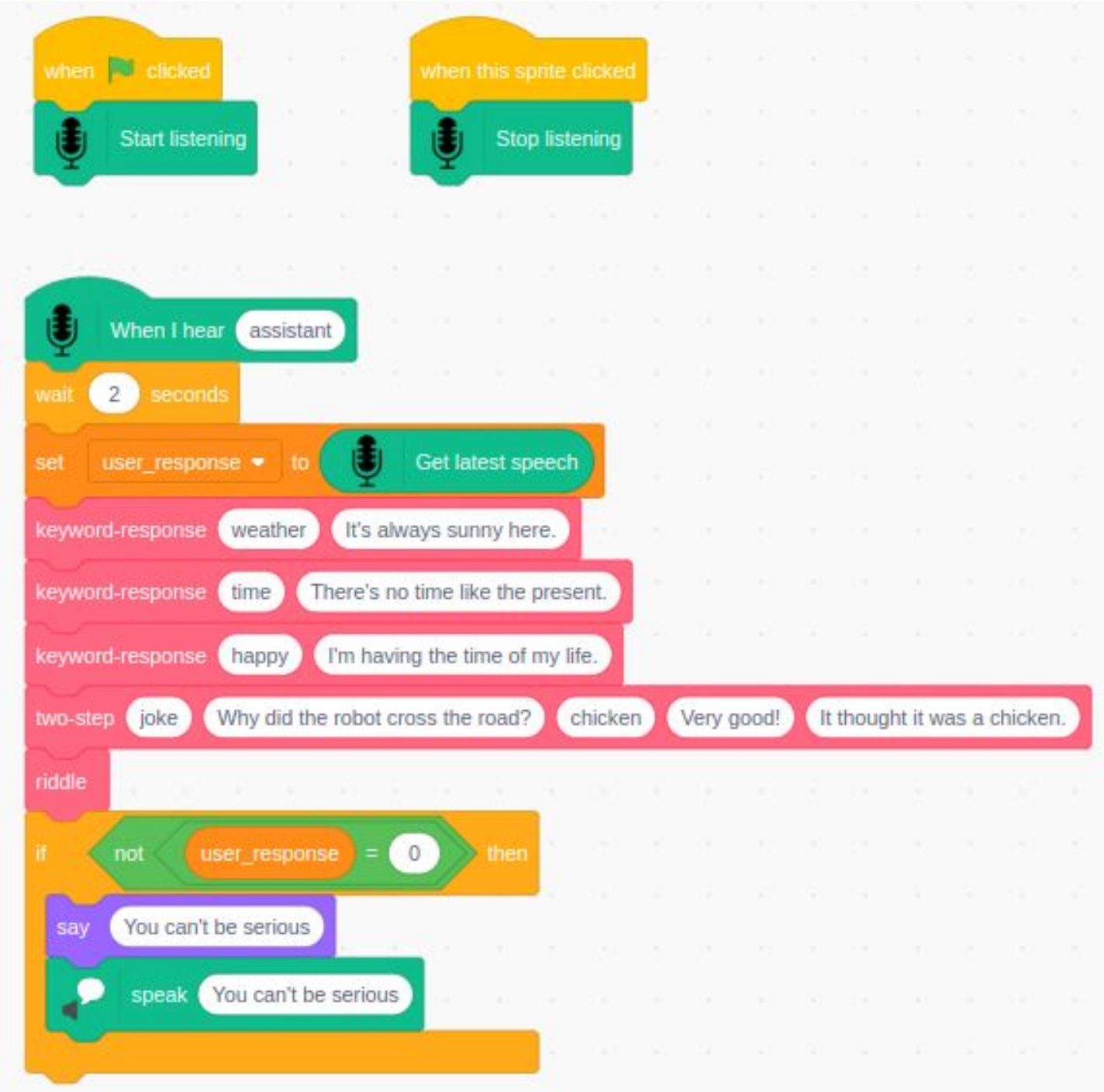
Assessments

To be determined.

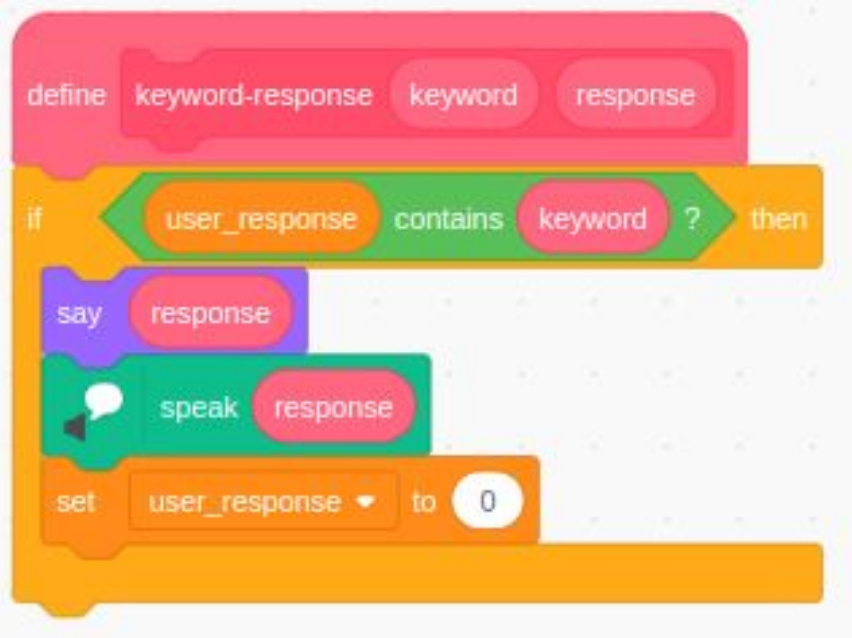
Other resources

Code for This Demo

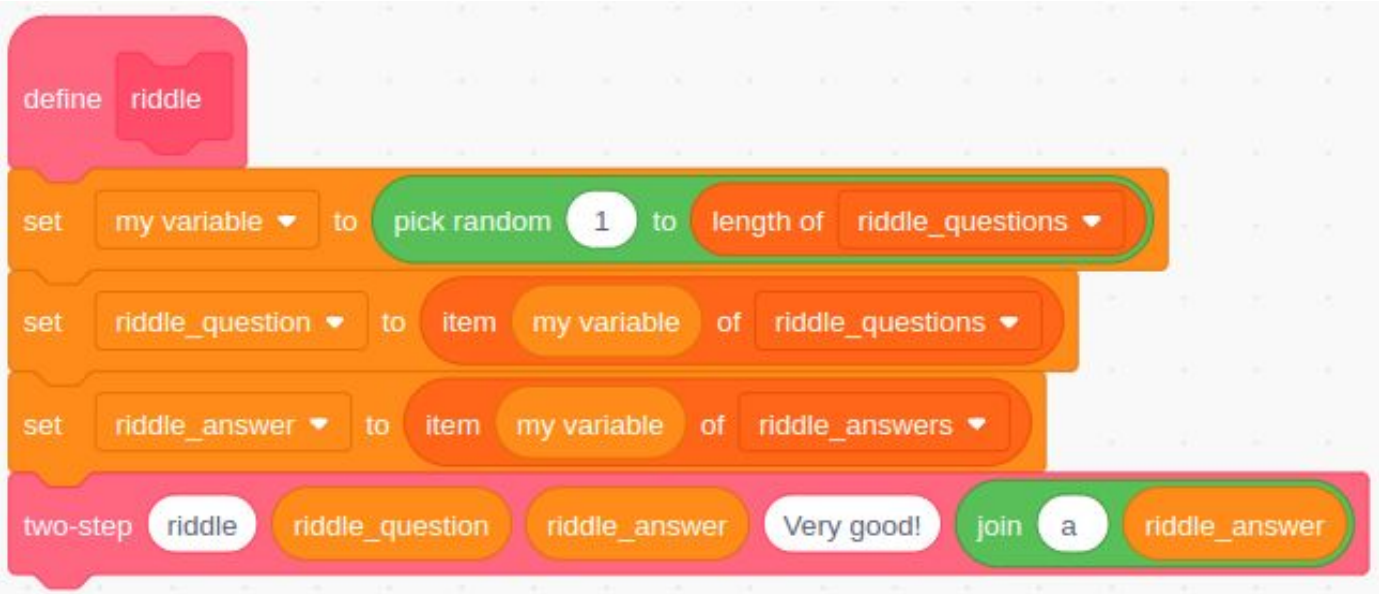
Main program



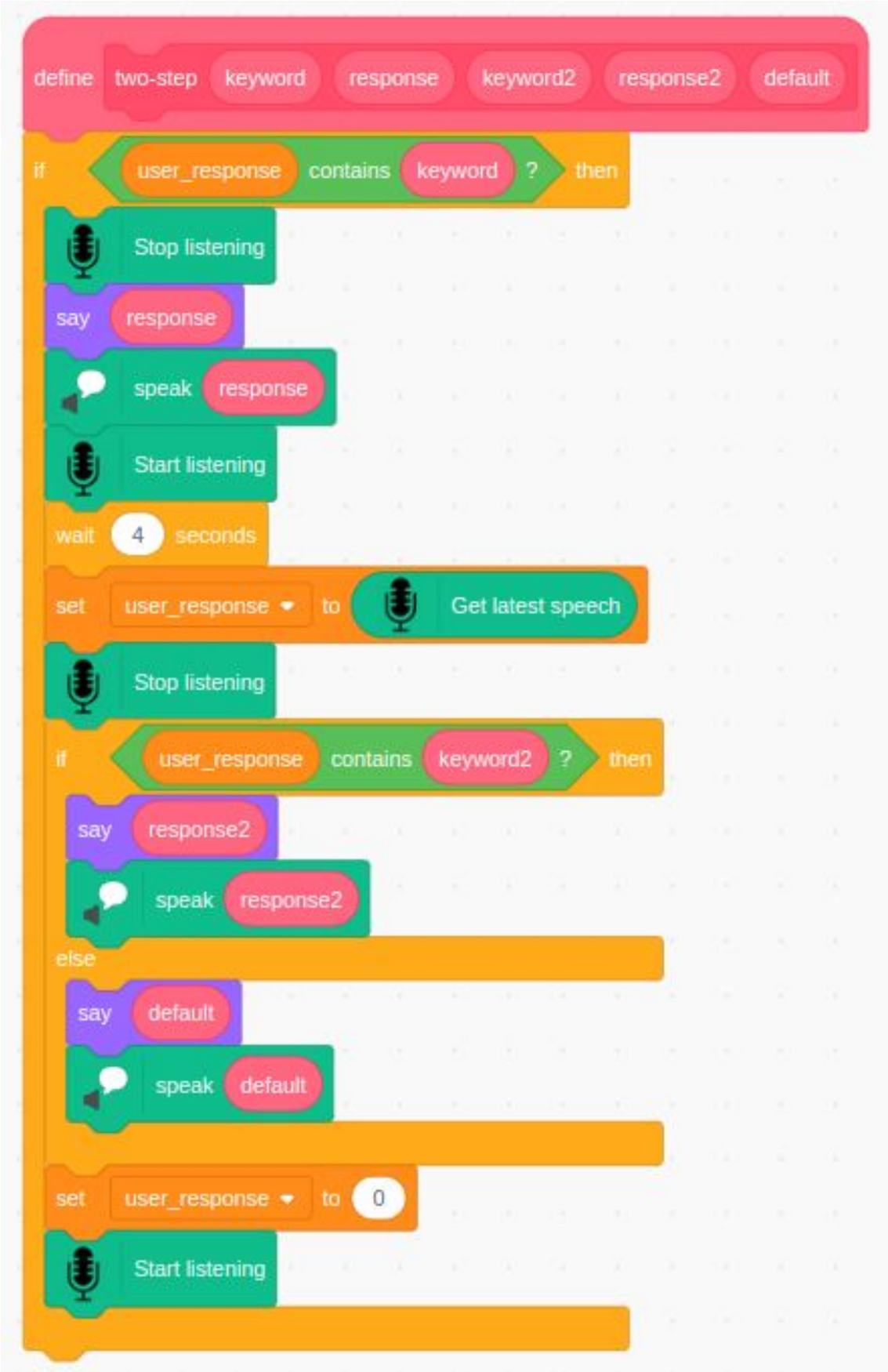
keyword-response



riddle



two-step


define two-step keyword response keyword2 response2 default
if user_response contains keyword ? then
Stop listening
say response
speak response
Start listening
wait 4 seconds
set user_response to Get latest speech
Stop listening
if user_response contains keyword2 ? then
say response2
speak response2
else
say default
speak default
set user_response to 0
Start listening

Vocabulary Flash Cards

ALEXA

Amazon's cloud-based voice service which uses a natural voice experience that allows people to interact with the technology they use every day.

INTELLIGENT ASSISTANT

Digital assistants that emulate human interaction to perform certain tasks, especially simple, repetitive ones.

KEYWORD

A word that a program specifically checks for that has a special meaning for it, typically a command such as "play" or a parameter like "time".

SIRI

Apple's personal assistant for iOS, macOS, tvOS and watchOS devices that uses voice recognition and is powered by artificial intelligence (AI)