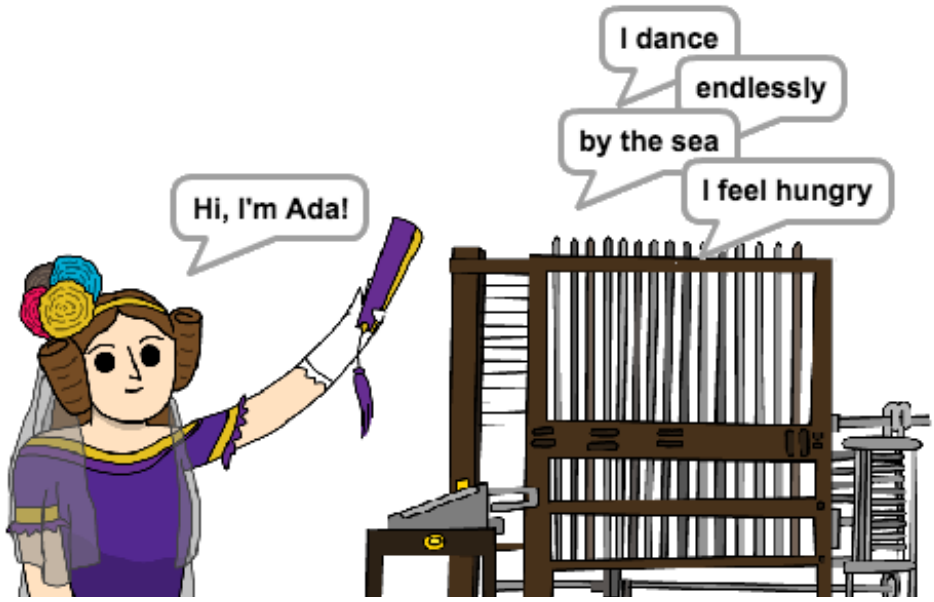
Ada's Poetry Generator

Introduction

You are going to learn how to program your own poetry generator!

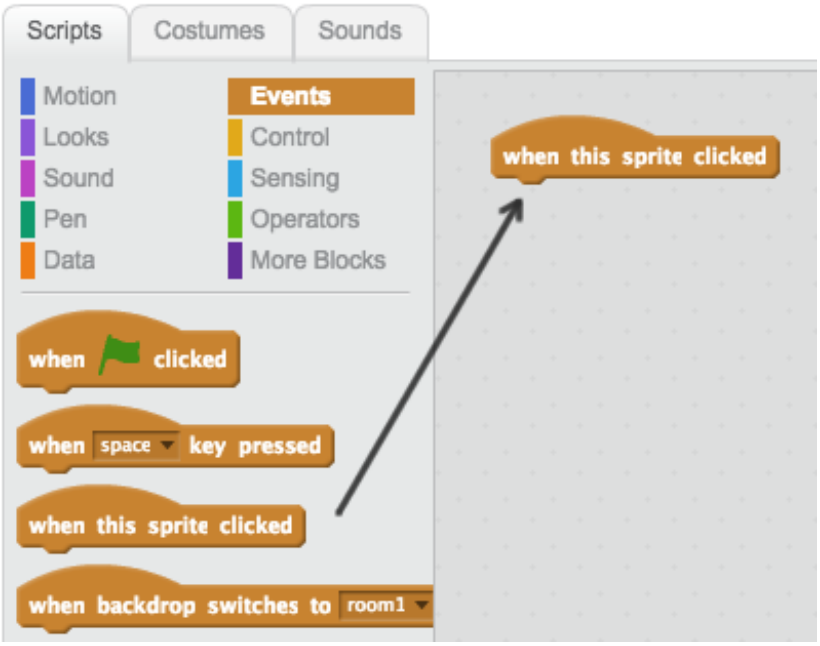


Step 1: Ada Lovelace

In 1842, Ada Lovelace wrote about using a machine called the 'Analytical Engine' to make calculations, and is seen as the world's first computer programmer! Ada was also the first to see that computers could be more than just big calculators.

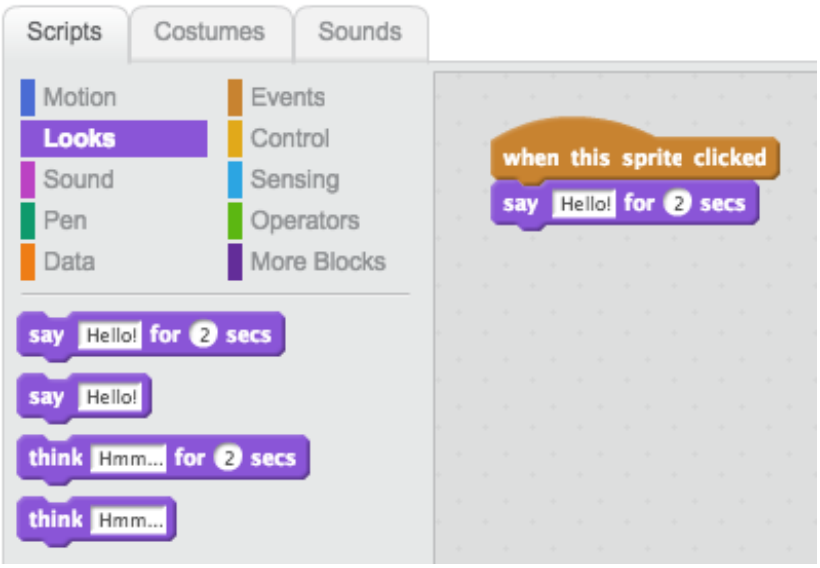
Activity Checklist

- [ ] Open the 'Poetry Generator' Scratch project online at **jumpto.cc/poetry-go** or download from **jumpto.cc/poetry-get** and then open if you are using the offline editor.
- [ ] Click on your 'Ada' sprite, and click the Events tab in the 'Scripts' coding section. Drag the when this sprite clicked block onto the coding area on the right.



Any code added underneath this block will run when Ada is clicked!

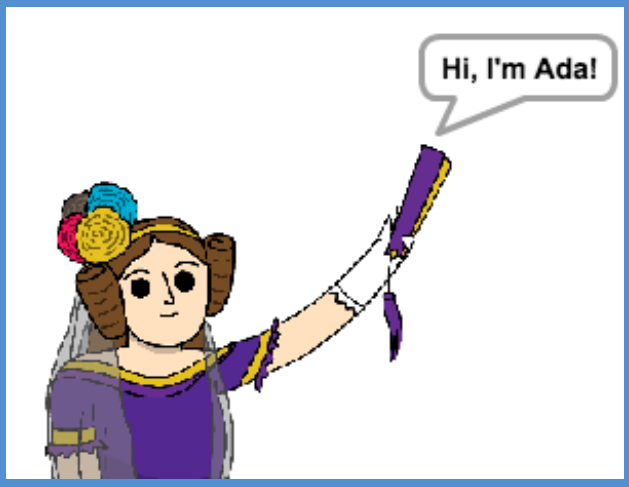
- [ ] Click the Looks tab, and drag the say Hello! for 2 secs block underneath the code you've already added.



- [ ] Click on Ada, and you should see her talk to you.

Challenge: Code Ada to introduce herself

Can you change your code, so that Ada says ‘Hi, I’m Ada!’ when you click on her?



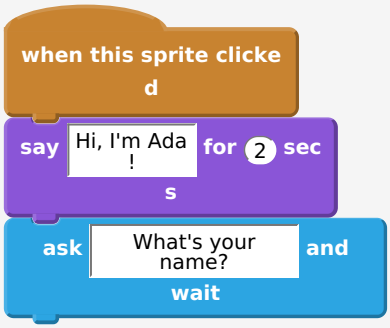
Save your project

Step 2: Telling Ada your name

Ada has introduced herself, but she doesn't know your name!

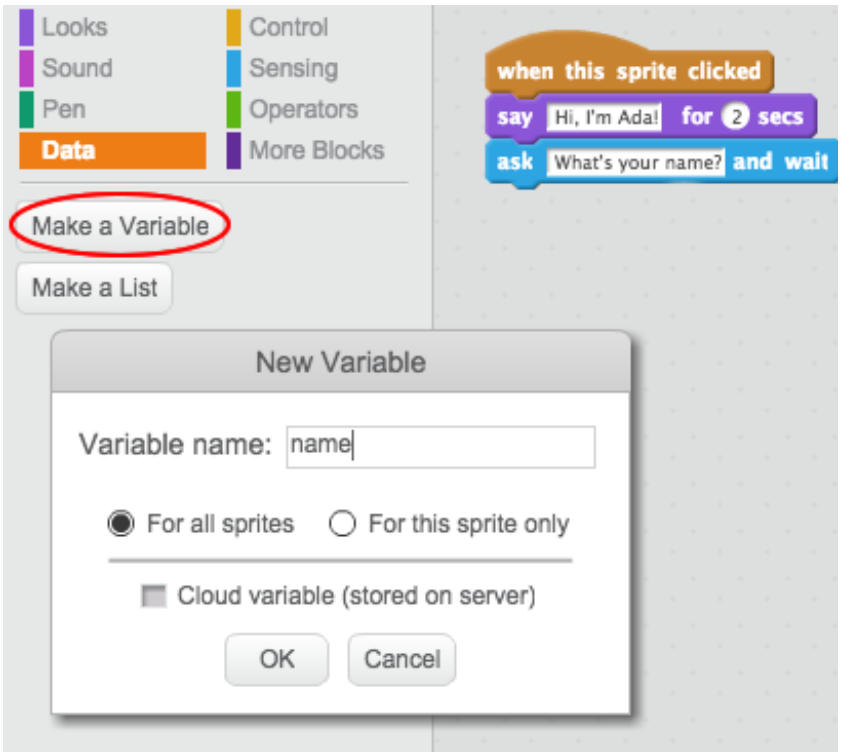
Activity Checklist

- [ ] Drag an ask block (from the sensing section) onto your code. Here's how your code should look:



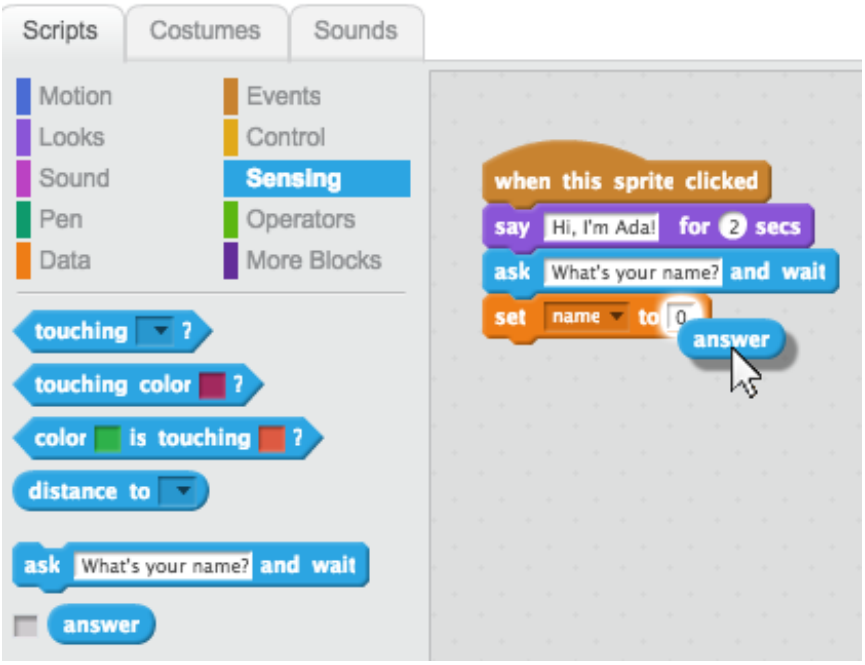
- [ ] Click on Ada to test your code. Ada should ask you your name, which you can type in!

- [ ] We can use a **variable** to store your name. Click Data, and then 'Make a Variable'. As this variable will be used to store your name, let's call the variable… name!



- [ ] To store your name, click the Data tab, and then drag the set name block onto the end of your code.

- [ ] Use the answer block to store the answer you type in.

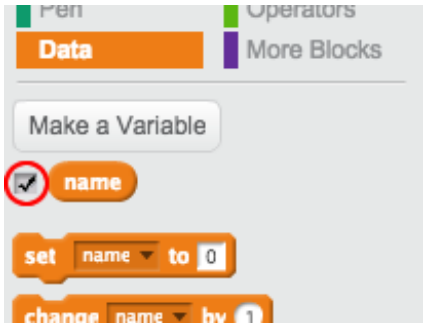


- [ ] Click on Ada to test your code, and enter your name when asked. You should see that your name has been stored in the name variable.

- [ ] You can now make use of your name in your code. Add this code:

To create this code, first drag a join block onto the say block, and then add your name block onto the join block.

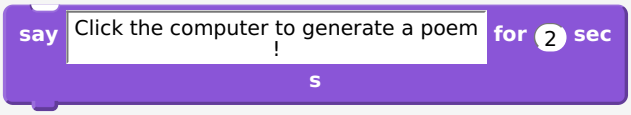
- [ ] To hide your name variable on the stage, click the tick next to the variable.



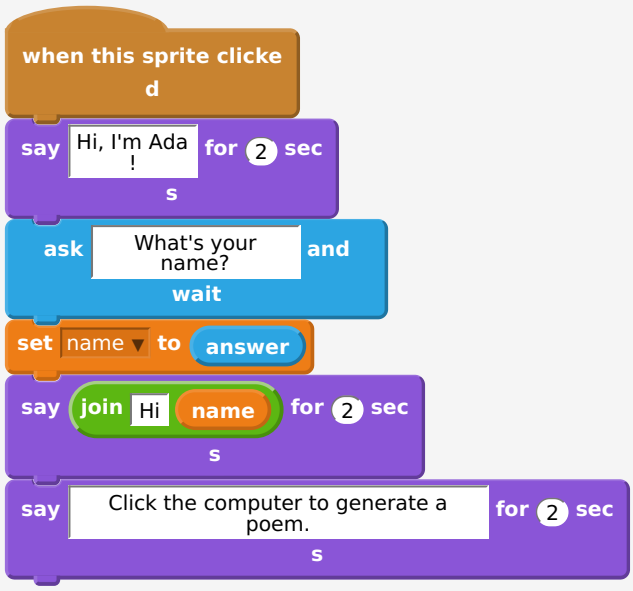
- [ ] Test your new code. Ada should say hello to you, using your name!

If there's no space between the word 'Hi' and your name, you'll need to add a space into the code yourself!

- [ ] Finally, add this code to explain what to do next:



- [ ] Test Ada's code one last time, to make sure that everything works. Here's How your code should look:



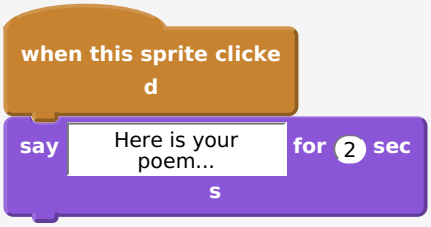
Save your project

Step 3: The Analytical Engine

Let's program Ada's computer (called the 'Analytical Engine') to generate poetry.

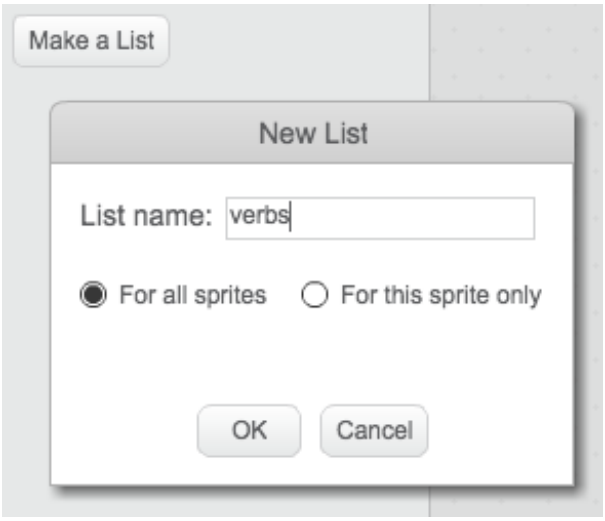
Activity Checklist

- [ ] Add this code to your 'Computer' sprite, so that it speaks when clicked:



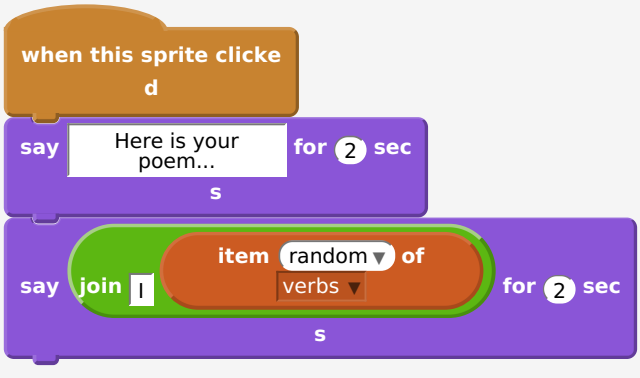
- [ ] To create a random poem, first you'll need a**list** of words to use. To create a new list, click the Data tab.

Let's use **verbs** (action words) in the first line of your poem. Create a new list called 'verbs'.



- [ ] Your new list will be empty. Click the + at the bottom of your empty list and add these verbs:

- [ ] The first line in your poem will be the word “I”, followed by a random verb. This is the code that you’ll need to add:



- [ ] Test your code a few times. Your computer should say a random word from your verb list each time.

Save your project

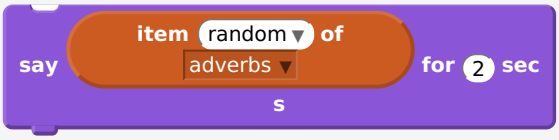
Step 4: More poetry

Your poem is quite short - let's add to it!

Activity Checklist

- [ ] Let's use adverbs in the next line of your poem. An**adverb** is a word that describes a verb. Create another list called adverbs, and add these 3 words:

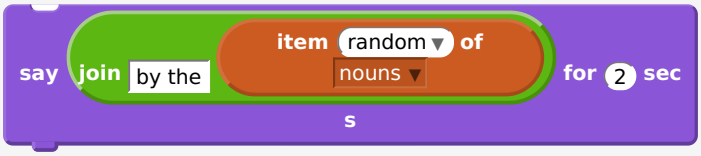
- [ ] Add this line to your computer's code, to say a random adverb on the next line of your poem:



- [ ] Test your code a few times. You should see a random poem each time.

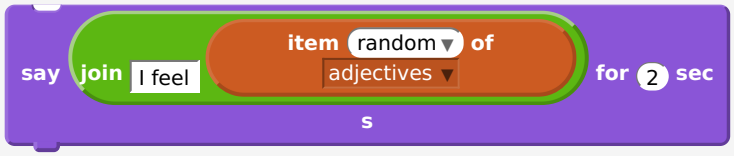
- [ ] Add a list of nouns to your project. A**noun** is a place or a thing.

- [ ] Add code to use the nouns in your poem.

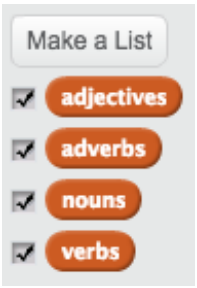


- [ ] Add a list of adjectives to your project. An**adjective** is a describing word.

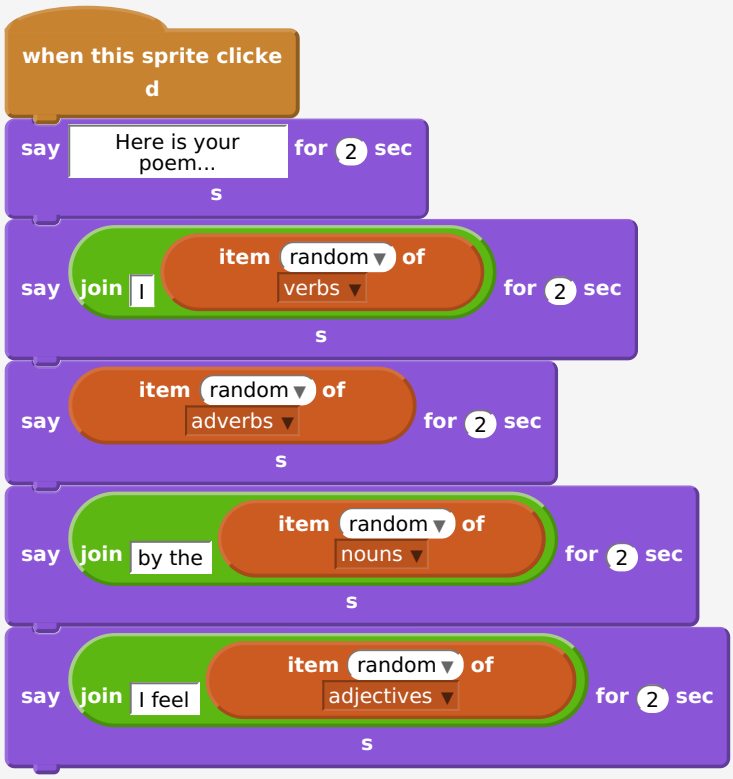
- [ ] Add code to use the adjectives in your poem:



- [ ] You can click the boxes next to your lists to hide them.



- [ ] Test out your new poem. Here's the code you should have:



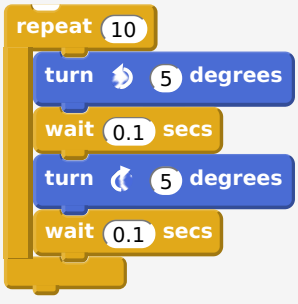
Save your project

Step 5: Animating the Analytical Engine

Let's animate your computer, so that it looks like it's generating poetry.

Activity Checklist

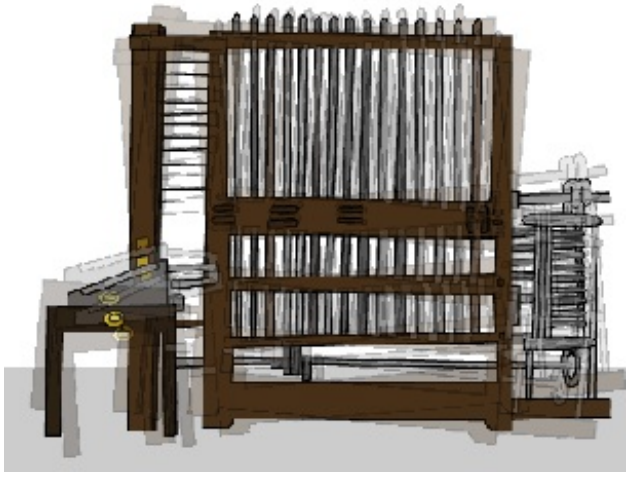
- [ ] Click on your computer sprite, and add this code after the first say block:



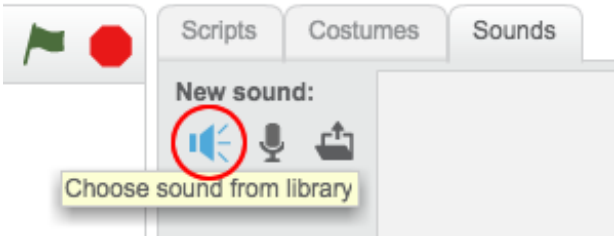
Here's how your code should look:

You'll find the repeat and wait blocks are in the Control section.

- [ ] Test your project. You should see the computer shake before producing a poem!

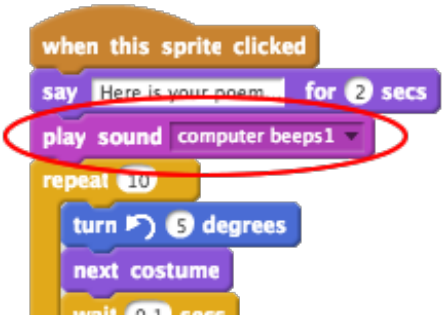


- [ ] Click the 'Sounds' tab, and click 'Choose sound from library'.



- [ ] Choose a ‘computer beeps’ sound and click OK.

- [ ] Add a play sound block, to play your sound just before your animation starts.



Save your project

Challenge: Personalise your poem

Can you use your name variable to personalise your poem?

Challenge: More words

Can you add more words to your lists, so that you can generate more poems?

Challenge: More poetry

Can you use your own lists to generate you own poetry?

Save your project