CHATTING WITH A CHARACTER

For Computational Thinking

STUDENT GUIDE

COMPUTE
Project

2021

PART I

PROBLEM + SOLUTION
ENGLISH & PSEUDOCODE

- The Problem
- The Solution
- The Solution in Pseudocode

Tab 1

The Problem

We need to bring a character to life. Not by acting, nor by writing, but by coding!

We're looking for a program similar to this one. Check it out:

Lady Macbeth[1]

1. Open the following project: https://scratch.mit.edu/projects/554216995/
2. Click on the green flag to start.
3. Let's play! Ask Lady Macbeth the following questions:
   - Who are you?
   - Do you like power?
   - Is the king dead?
   - What is your plan?
   - Any random question (e.g. "What is the meaning of life?", "What color is your shoe?")

Fences

Cool. Lady Macbeth's cool. But now we want *you* to bring a character from *Fences* to life. Are you in? Don't worry—we will guide you through the solution.

Ready? Let's go!

The Solution

Let's start by analyzing how the Lady Macbeth[1] program works.

Think for a moment—how do you think the code work works? Write your best guess below. All answers are valid.

Hint: *What if instead of asking, "Who are you?" you asked, "What is your name?" or "Please tell me your name". Try it! Does the code work? And if you asked, "What is your first name?" or "How should I call you?" What happens? Try variations of the questions listed above (e.g. "Tell me your plan", "What idea do you have in mind?", "Any ideas in mind?") What happens? Are there certain words (or combinations of words) that trigger certain responses?*

Your best guess: _____

_____

_____

Did you guess the program logic correctly? Let's figure it out!

The Solution in Plain English

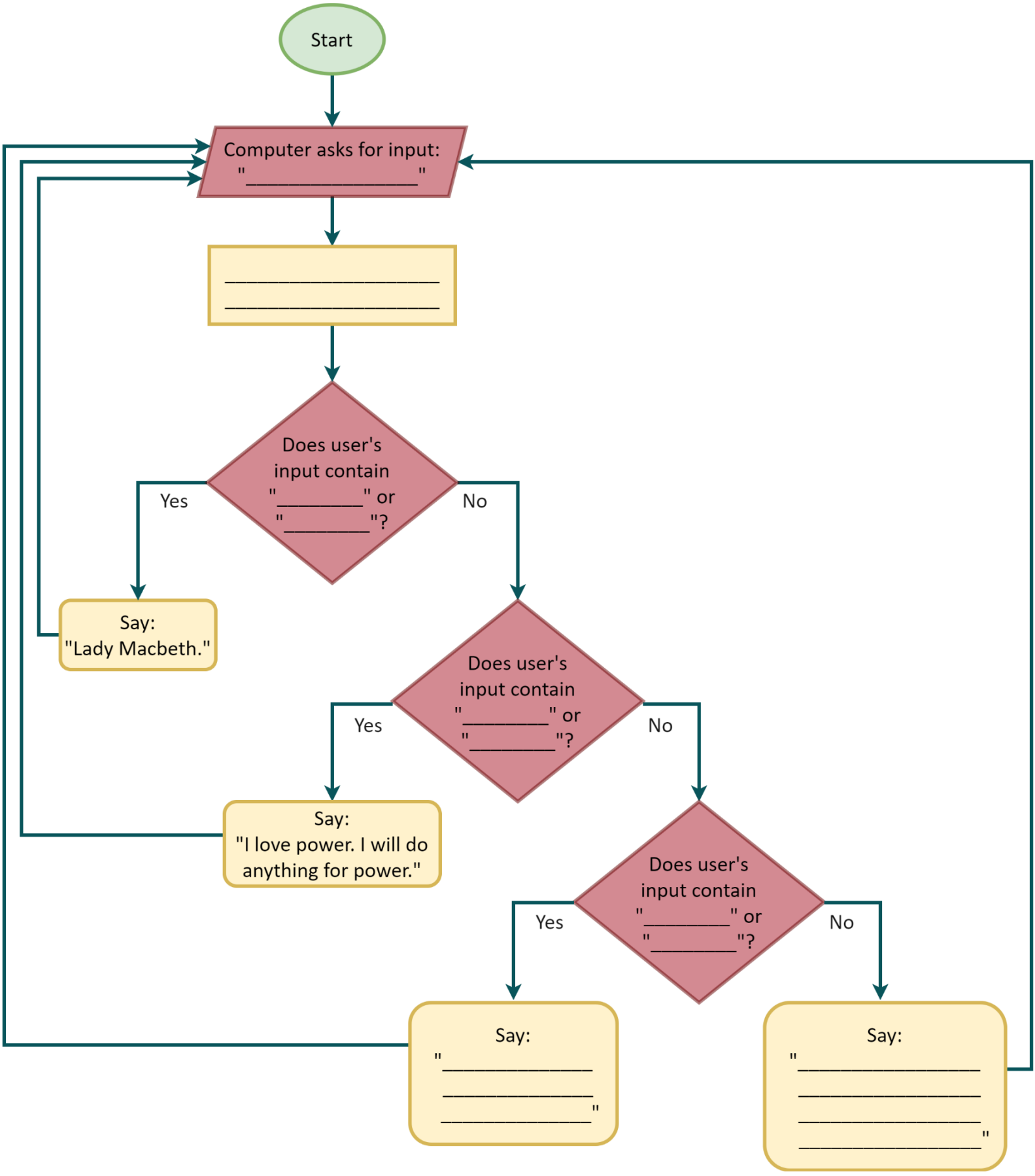
Fill in the Blanks

Choose from the following options to fill in the blanks:		
plan	Who are	Only look up clear. To alter favor ever is to fear. Leave all the rest to me.
your name	Take note of user's input.	power
All we have to do is wash our hands of it.	king	Ask me anything:
idea	ideas	name

Note: Some of these options will not be used.

Flowchart of Lady Macbeth's Code[1]

Fill in the blanks of the flowchart using the options from the blue table above.



Pseudocode

Now let's start translating our solution into a language that is *similar* to the languages that computers understand. On the limbo between plain English and programming languages, there is pseudocode. For this exercise, we will use a variation of the conventional pseudocode.

Variables

Computers cannot "remember" the user's input unless you *store* this information somewhere. To achieve this in programming, we use variables.

You can think of variables as containers. Like a regular container, the content you put inside a variable can *vary* (thus its name!) depending on what you want it to store.

And like a tag on a regular container, it is wise to give variables a distinct name that alludes to what the variable is storing (e.g. "User's Input"). Unlike a tag on a regular container, spaces between words and apostrophes are normally not allowed, and you also want to keep the name short. Thus, "User's Input" may become something like "User_Input", "userInput", etc.

Matching Pairs Game + Fill in the Blanks (Next Pages)

Match each action of the solution in plain English (left mustard column) with its pseudocode translation (right mustard column). Use the options on the blue chart to fill in all the blanks.

Tip: Start by filling in the blanks of the left mustard column. Note that the mustard columns continue on the next page. If possible, keep the colors of the words (e.g. "userInput" is orange).

Choose from the following options to fill in the blanks:		
Lady Macbeth	All we have to do is wash our hands of it.	userInput
king	userInput contains king	Ask me anything:
Only look up clear. To alter favor ever is to fear. Leave all the rest to me.	I love power. I will do anything for power.	Who are
userInput contains Who are	userInput contains plan	answer

Note: <u>All</u> options will be used, and some may be used more than once.

Solution in Plain English (For Programming)	Pseudocode
1. Asks for input by saying: "_____" and wait for the user to respond.	SET _____ TO _____
2. After the user enters their answer, save this info in a variable called "userInput".	ASK _____ AND WAIT

3. If user's input contains "_____" or "your name", then say: "_____" for a few seconds.

If not, then:

IF _____

OR userInput contains power

THEN

SAY _____

_____

_____ FOR 3 SECONDS

ELSE

4. If user's input contains "_____" or "power", then say: "_____ _____" for a few seconds.

If not, then:

IF _____

OR userInput contains idea

THEN

SAY _____

_____

_____ FOR 3 SECONDS

ELSE

5. If user's input contains "plan" or "idea", then say: "_____ _____" for a few seconds

If not, then:

IF _____

OR userInput contains your name

THEN

SAY _____ FOR 3 SECONDS

ELSE

6. say: "_____ _____ _____" for a few seconds

SAY _____

_____

_____ FOR 3 SECONDS

Reorder the Pseudocode

Now that you have matched each step with its pseudocode translation, write the whole pseudocode below in the right order. Pay close attention to the nested format of the lines below. We have helped you with some words to give you a sense of the nested structure, but feel free to check the flowchart you filled out earlier if you need a visual guide.

If possible, keep the color-coding of the pseudocode lines above. (E.g. "ASK" is turquoise; "SET" is orange)

_____

_____

IF _____ OR _____

    SAY _____

ELSE

    IF _____ OR _____

        SAY _____

    ELSE

        IF _____ OR _____

            SAY _____

        ELSE

            SAY _____

PART II

FENCES

- Characterization
- Pseudocode for Fences

From Macbeth to Fences

Let's begin adapting the Lady Macbeth's[1] program for *Fences.*

Steps

1. Determine which character from *Fences* you want to program:

_____

2. Now take a look at the novel. **What are some likely questions someone would ask this character? How would the character respond?** List 5 questions and answers below. For each question, **identify from one to two key words or combination of words** that the program should look for to give a certain answer. (e.g. Like the words "king" and "power" that triggered the "I will do anything for power" response from Lady Macbeth.)

For the answers, **you can search for some quirky quotes** by your chosen character in the novel (as we did for Lady Macbeth) or come up with your own answers. Just make sure to keep the essence of the character—you want users to think that they really are chatting with them!

Additionally, **choose one "wildcard" response**: A response your character will give in case the program does not find any of your key words in the user's question. (e.g. Just like the "Only look up clear . . . " response from Lady Macbeth.)

Have fun!

List 5 questions + answers + 1 wildcard response:

- Question 1: _____

  Key Words: _____ and _____

  Character's Answer: _____

  _____

- Question 2: _____

  Key Words: _____ and _____

  Character's Answer: _____

  _____

- Question 3: _____

  Key Words: _____ and _____

  Character's Answer: _____

  _____

- Question 4: _____

  Key Words: _____ and _____

  Character's Answer: _____

  _____

- Question 5: _____

  Key Words: _____ and _____

  Character's Answer: _____

  _____

- Wildcard Response: _____

  _____

3. Choose *three* questions and answers from the five you wrote above. (You will use the remaining two later).

4. Copy the pseudocode you wrote earlier but replacing the questions and answers from *Macbeth* with the three you chose for *Fences* plus your "wildcard response":

_____

_____

IF _____ OR _____

    SAY _____

ELSE

    IF _____ OR _____

        SAY _____

    ELSE

        IF _____ OR _____

            SAY _____

        ELSE

            SAY _____

PART III

SCRATCH

- Getting started
- Coding on Scratch

Tab 3

Getting Started

1. Open Scratch (https://scratch.mit.edu/) and create a free account.

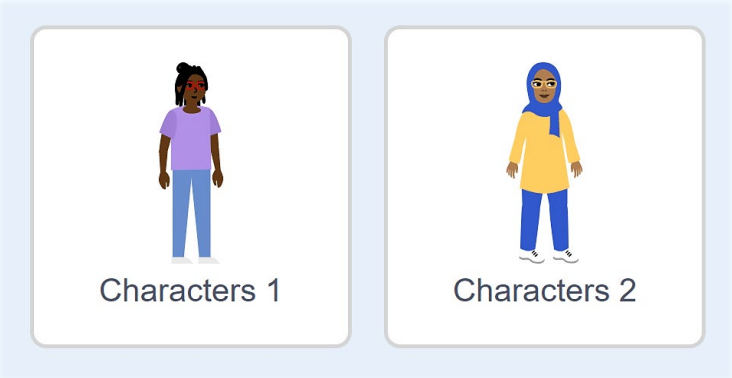
2. Follow this link: https://scratch.mit.edu/projects/563729274/editor/ to open the *Fences* template. **Do not edit this template!**

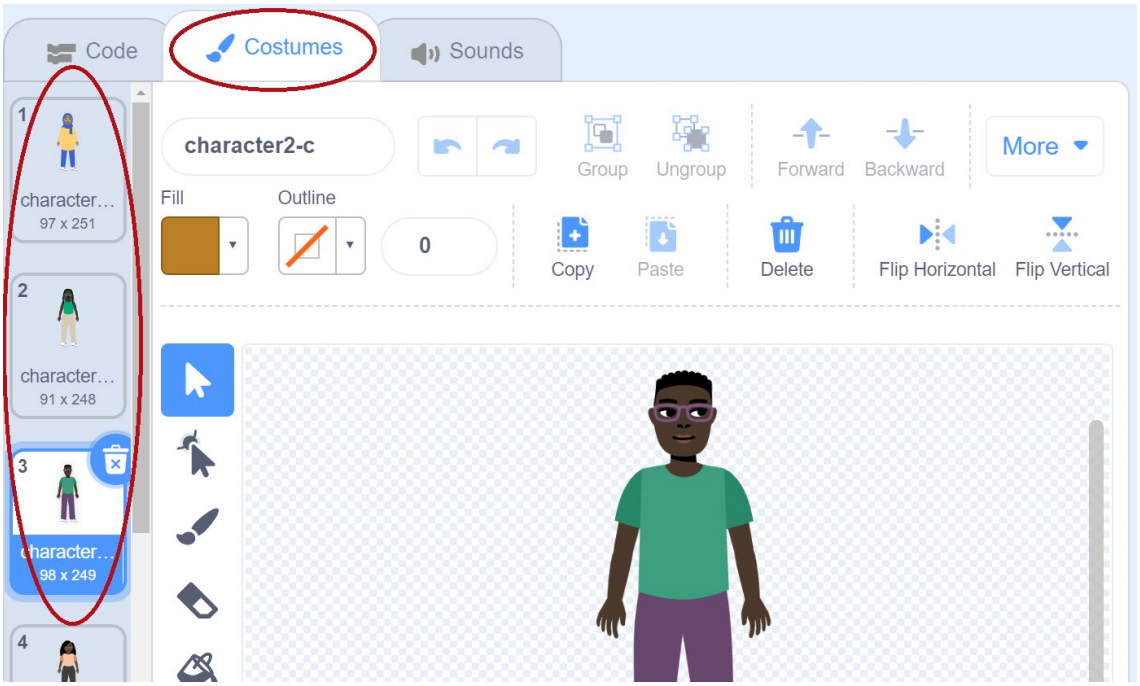
3. **Create a copy** of this template by clicking on *File > Remix*. Work on this copy ("remix") from now on.

4. You can use the backdrop provided on the template, or you can choose another one by clicking on "Backdrops" (bottom right).

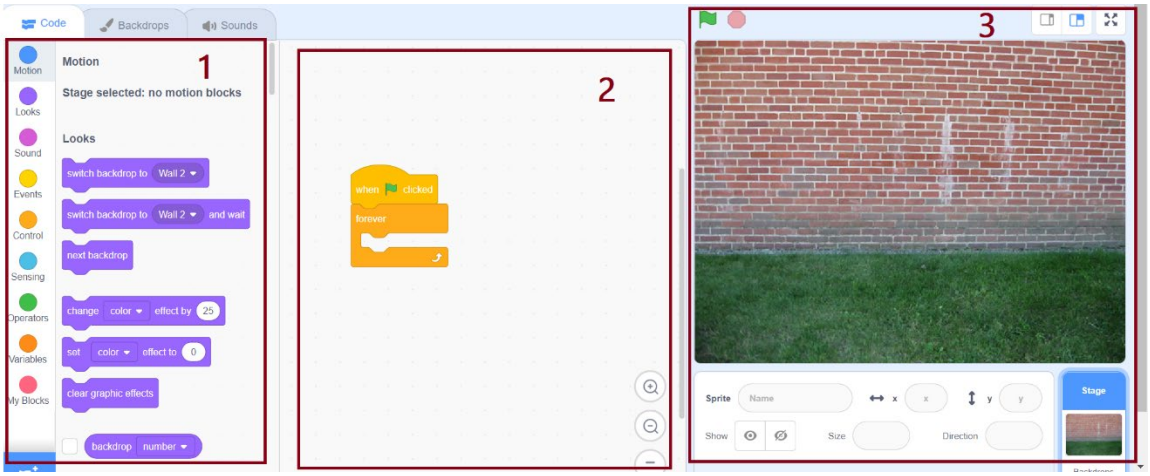
5. Click on "Sprites" (next to "Backdrops") and choose a character that most resembles the character you chose from *Fences*. Alternatively, you can upload your own image by clicking on "Upload Sprite".

Tip: The options "Characters 1" and "Characters 2" hold a wider variety of characters than it seems. Just click on one of the options, and once you see a character appear on your backdrop, click on "Costumes" to choose from the character selection:





1. Let's start by familiarizing ourselves with Scratch:



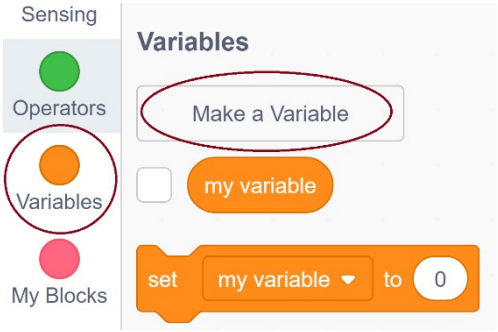
1.1 The first block (left) contains all the possible lines of code you can use on Scratch. The menu on the far left shows the different categories of code ("Motion", "Looks", "Sound", etc.)

1.2 The second block (middle) is “your workspace”. This is where you write your code.

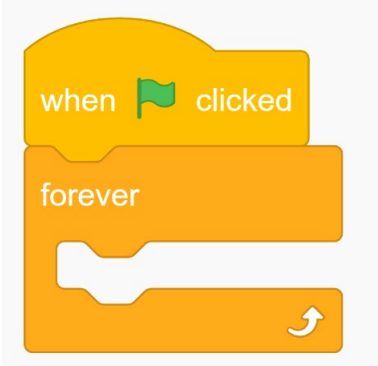
1.3 The third block (right) will show the end result of your code.

2. Let’s code!

2.1. Let’s start by creating our variable “userInput” by clicking on *Variables > Make a Variable.* Create the variable for all sprints.



2.2. We will help you with the first *two* lines of code, because they were not in your pseudocode. Look at all the possible lines of code listed on the left side on Scratch, and then choose these:



Hint: Have you noticed that the first line is yellow just like the “Events” category on Scratch?

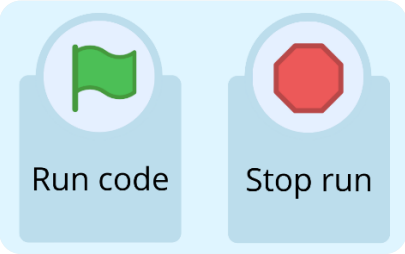
Can you guess what these first two lines do?

Your job is to fill everything inside the “forever” block with the pseudocode you wrote earlier. To do this, look at all the possible lines of code listed on the left side on Scratch, and then choose those that most resemble the lines of your pseudocode. Work line by line. Part by part.

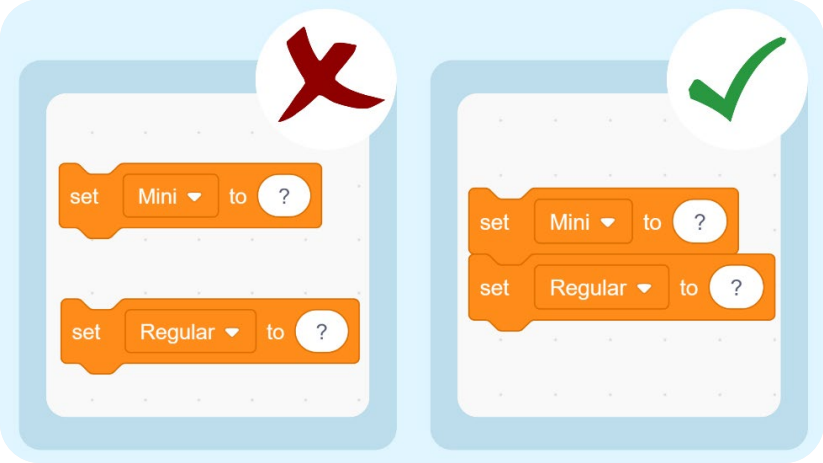
Hint: Remember that the pseudocode lines in our previous exercises were color-coded. (e.g. “ASK” was turquoise, just as the “Sensing” category on Scratch!)

Some useful tips:

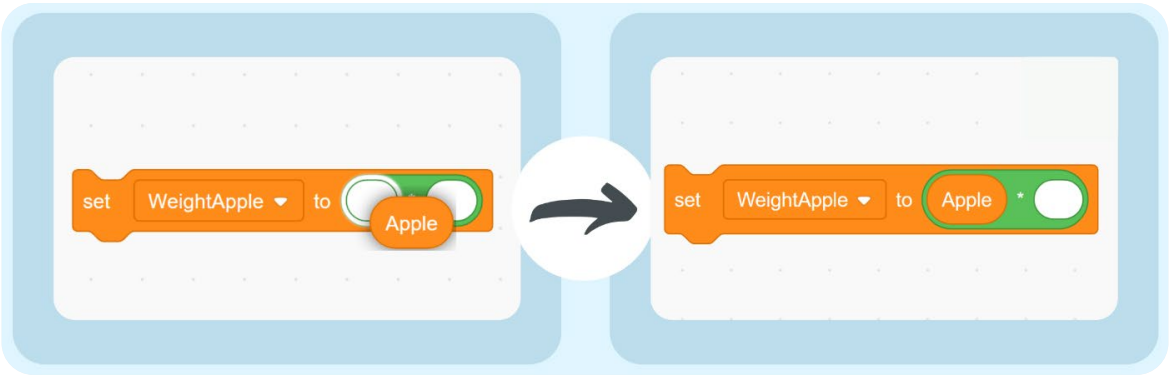
- To run and stop the code:



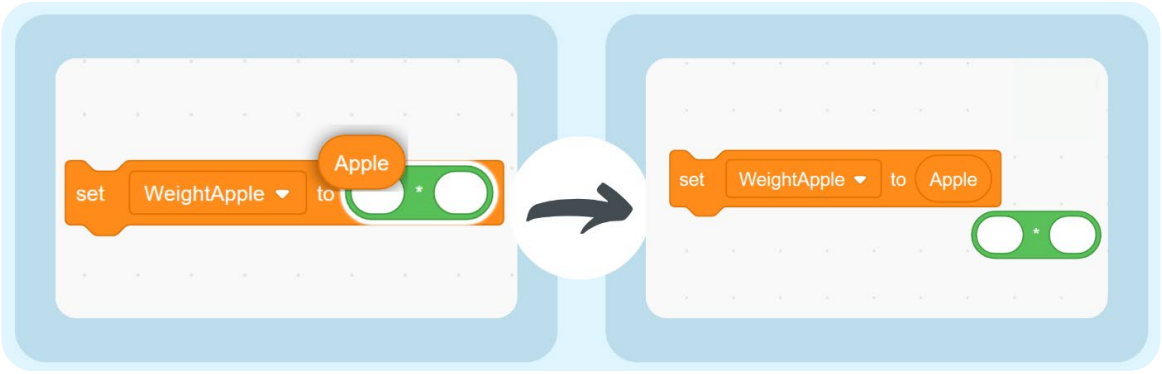
- To add new coding lines:



- To embed elements:
    - Drag an element towards the container block. Check the highlighted area before dropping it.
    - The highlighted area tells you where the new element will be placed:



The new element will be inserted into the first blank space.



The new element will replace the whole green block.

Once you finish your code on Scratch, test it! Assume the role of “user” and ask the program some questions. Is the program answering as you intended to?

Hint: Remember the nested format of your pseudocode!

Revise your code if necessary; test it again. Revise it until you get it right. *When your code works with three questions and answers, add the remaining two.*

Happy coding!

PART IV

PYTHON

- Writing in Python
- Jupyter Notebook

Python

Look at the following Python code. Can you guess what it does?

```
userInput = input("Ask me anything:")
if "Who are" in userInput or "your name" in userInput:
    print("Lady Macbeth.")
else:
    print("Was hope drunk wherein you dressed yourself?")
```

Did you guess correctly? Check it out:

`userInput = input("Ask me anything:")`	Asks for input by saying: "Ask me anything:" and stores the user's answer in a variable called "userInput".

`if "Who are" in userInput or "your name" in userInput:`
`  print("Lady Macbeth.")`
`else:`
`  print("Was hope drunk wherein you dressed yourself?")`
	If the program finds the words "Who are" in "userInput" or "your name" in "userInput":
	then it prints on screen: "Lady Macbeth."
	If not:
	then it prints on screen: "Was hope drunk wherein you dressed yourself?"

Fill in the Blanks

Take the following coding structure for Python and fill in the blanks with *three* of the five questions and answers you wrote for *Fences* plus your "wildcard response". Use the previous Python code as a guide. You may also refer to your Scratch code and the flowchart you filled out earlier if you need a visual guide of the code structure.

Make sure to include the required punctuation (use the previous Python code as a guide). Every character matters in coding!

Tip: It is not necessary to keep the color-coding this time.

```
userInput = _____
if _____
    print _____
else:
    if _____
        print _____
    else:
        if _____
            print _____
        else:
            print _____
```

Run Your Code

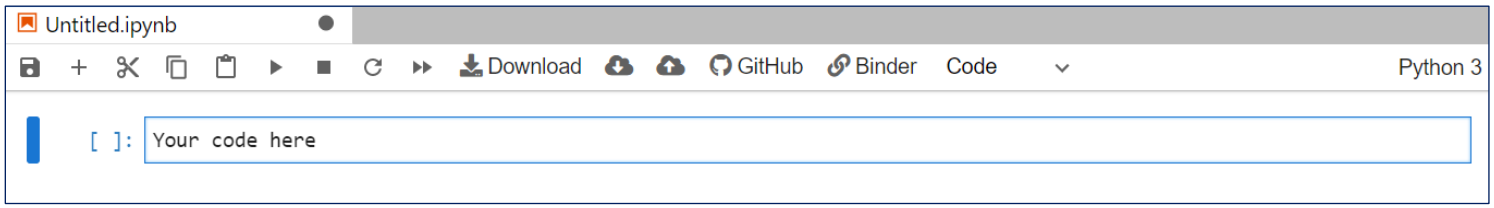
1. Open Jupyter Notebook: https://jupyter.org/try

2. Open JupyterLab and wait a few seconds for it to load.

3. Click on "Python 3" under "Notebook".

4. Type the Python code you wrote above in the coding block as shown below. Make sure to include all the required punctuation and *to maintain the indented structure of the code.* (Each indentation matters in Python!) Press the "Tab" key to indent a line.

Note: The user will only be able to ask one *question this time. If you want to ask another question, you will have to copy and paste the code into a new coding block.*



5. Click on the "Play" button to run your code and test it.

6. Is the program running correctly? Revise it if necessary. Once the program runs correctly with three questions and answers, add the remaining two.

You may add even more questions and answers if you like!

7. Test your code until you get it right.

8. Congratulations! You just brought a character to life by coding in Python.

References

[1] *Module Adapted from* W. McKnight and Exploring Computational Thinking team at Google, "ECT Pencil Code Program: Lady Macbeth Chat Bot," *ISTE*, 09-Jun-2015. [Online]. Available: https://learn.iste.org/d2l/lor/viewer/view.d2l?ou=6606&loIdentId=194. [Accessed: 16-Dec-2021].