CHATTING WITH A HISTORICAL FIGURE

For Computational Thinking

TEACHER'S GUIDE

COMPUTE Project

2021

PART I

PROBLEM + SOLUTION
ENGLISH & PSEUDOCODE

- The Problem
- The Solution
- The Solution in Pseudocode

The Problem

We need to bring a historical figure to life. Not by acting, nor by summoning (did you really think of that?), 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?")

Cool. Lady Macbeth's cool. But now we want *you* to bring a *historical figure* 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: *Since this is just to make students start thinking computationally, all answers are valid as long as they show they are trying.*

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

The Solution in Plain English

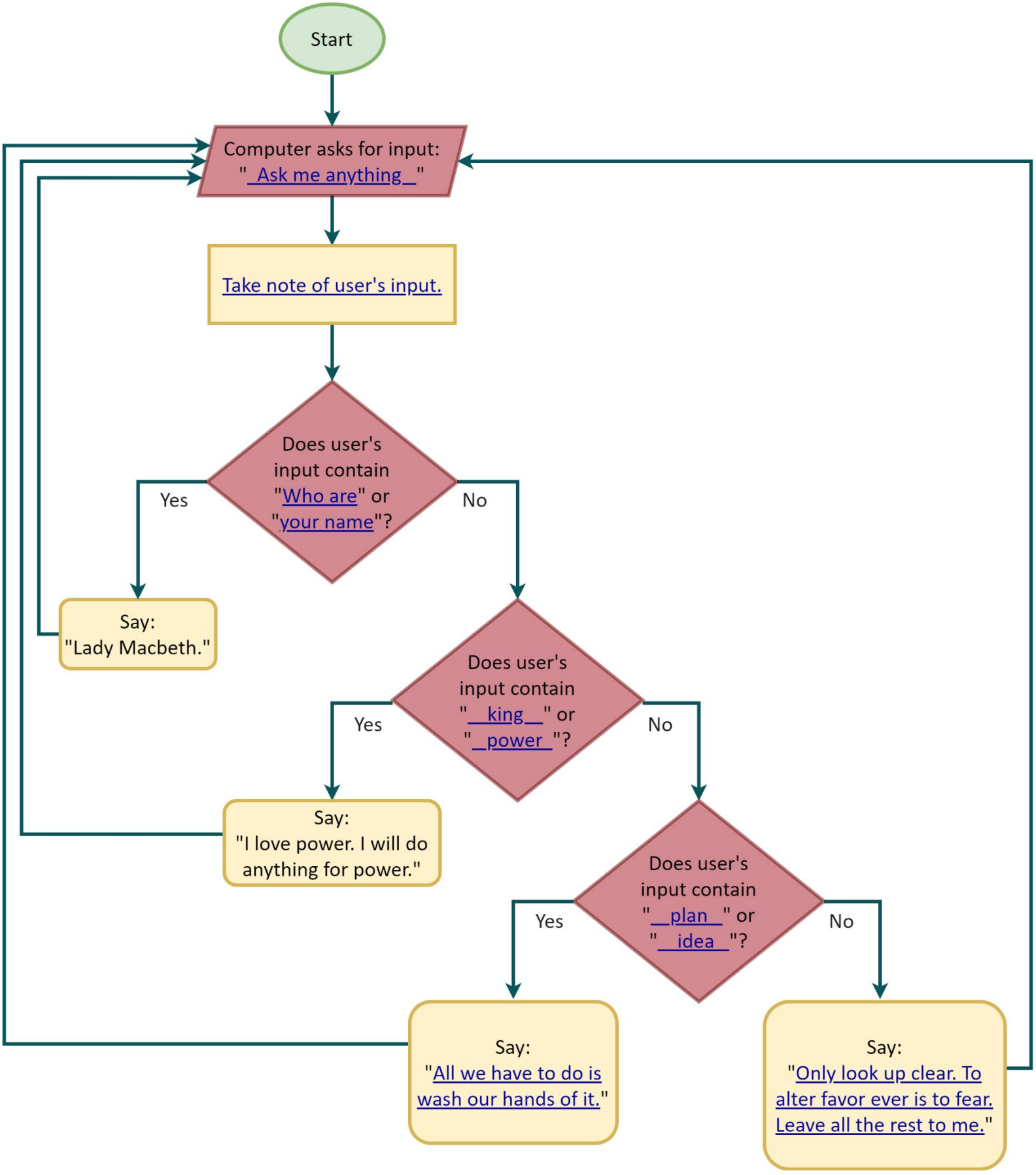
Fill in the Blanks

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 green 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 green 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: "___Ask me anything:___" and wait for the user to respond.	SET __userInput__ TO __answer__
2. After the user enters their answer, save this info in a variable called "userInput".	ASK ___Ask me anything:___ AND WAIT

3. If user's input contains "_ Who are _" or "your name", then say: "____ Lady Macbeth ____" for a few seconds.

If not, then:

IF userInput contains king

OR userInput contains power

THEN

SAY I love power. I will do anything for power. _____

_____ FOR 3 SECONDS

ELSE

4. If user's input contains "___ king ___" or "power", then say: "I love power. I will do anything for power." for a few seconds.

If not, then:

IF userInput contains plan

OR userInput contains idea

THEN

SAY All we have to do is wash our hands of it. _____

_____ FOR 3 SECONDS

ELSE

5. If user's input contains "plan" or "idea", then say: "All we have to do is wash our hands of it." for a few seconds

If not, then:

IF userInput contains What is

OR userInput contains your name

THEN

SAY _Lady Macbeth_ FOR 3 SECONDS

ELSE

6. say: "Only look up clear. To alter favor ever is to fear. Leave all the rest to me." for a few seconds

SAY Only look up clear. To alter favor ever is to fear. Leave all the rest to me. 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.)

TEACHERS

It is okay if students cannot keep the color-coding of the given words/phrases/symbols (e.g. they may not have color pencils at hand, or they may be filling out the worksheet digitally). However, when working on the Scratch part, make them refer to the previous exercise to check what color each word/phrase/symbol is supposed to be.

```
ASK ___ Ask me anything: ___ AND WAIT
SET  userInput  TO  answer
  IF userInput contains What is  OR  userInput contains your name
      SAY  Lady Macbeth  FOR 3 SECONDS
ELSE
      IF userInput contains king  OR  userInput contains power
          SAY I love power. I will do anything for power. FOR 3 SECONDS
      ELSE
          IF userInput contains plan  OR  userInput contains idea
              SAY All we have to do is wash our hands of it. FOR 3 SECONDS
```

```
ELSE
    SAY Only look up clear. To alter favor ever is to fear. Leave all the
        rest to me.  FOR 3 SECONDS
```

TEACHERS

Revise the students' pseudocode. It is important that they have it right before they proceed to the next part.

PART II

Tab 2

World War II

- Characterization
- Pseudocode for WWII

From Macbeth to WWII

Let's begin adapting the Lady Macbeth's[1] program to bring a historical figure to life instead. For this module, let's code one leader that led up to the start of World War II. Roosevelt, Churchill, Stalin, Mussolini, Hitler, De Gaulle—you choose!

Steps

1. Determine which important leader from WWII you want to program:

Answers will vary. (e.g., Churchill)

Hint: You don't have to limit your choices to the suggestions given above. You can code any leader from WWII you would like to!

2. Now take a look at your notes about this historical figure. (Feel free to do more research if you like!) **What questions would you like to ask this person? What are some likely questions that others would ask? How would this historical figure 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 historical figure (as we did for Lady Macbeth) or come up with your own answers. Just make

sure to keep the essence of this historical figure—you want users to think that they really are chatting with them!

Additionally, **choose one "wildcard" response**: A response your chosen historical figure 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:

(Answers will vary.)

- Question 1: (e.g., When did you win the Nobel Prize?)

  Key Words: (e.g., "Nobel") ~~and~~ _____

  Character's Answer: (e.g., 1953.)

- 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 your chosen historical figure. Don't forget to include your "wildcard response":

Teachers

Answers will vary, but they should be similar to the following structure. Remember that students may have *from one to two key words or combination of words* for each question. This means that the "OR" statement, and the statement that comes after it (the second key word/s), are optional.

```
ASK ___Ask me anything:___ AND WAIT
SET __userInput__ TO __answer__
  IF userInput contains (key word/s) OR userInput contains (key word/s)
      SAY (answer # 1) FOR 3 SECONDS
ELSE
      IF userInput contains (key word/s) OR userInput contains (key word/s)
          SAY (answer # 2) FOR 3 SECONDS
      ELSE
          IF userInput contains (key word/s) OR userInput contains (key word/s)
              SAY (answer # 3) FOR 3 SECONDS
          ELSE
              SAY wildcard response FOR 3 SECONDS
```

PART III

SCRATCH

- Getting started
- Coding on Scratch

Getting Started

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

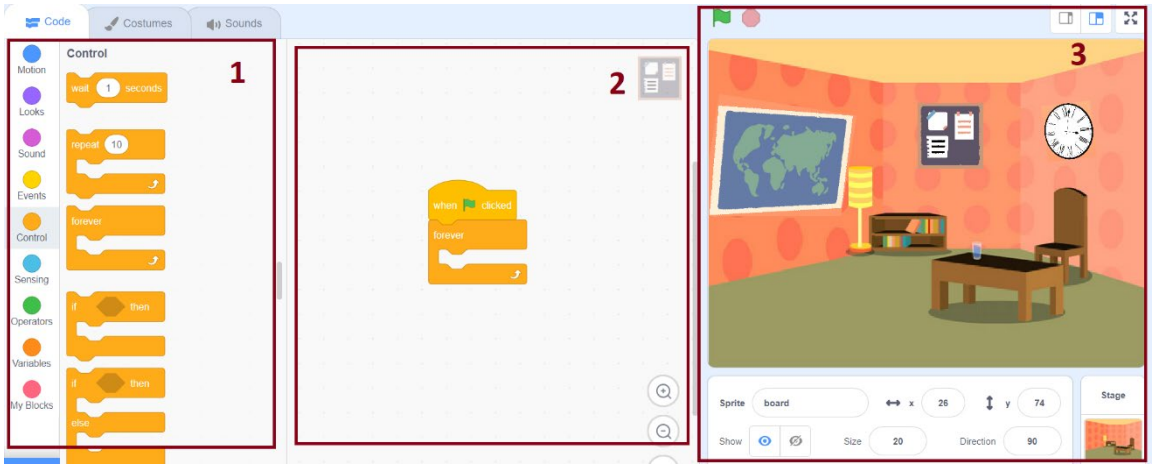
2. Follow this link: https://scratch.mit.edu/projects/618596837/editor/ to open the WWII template. **Do not edit this template!**

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

4. Find an image of your chosen historical figure and save it to your computer. *Hover over* the "Choose a Sprite" button (cat icon on the bottom right) and click on "Upload Sprite".

Coding on Scratch

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



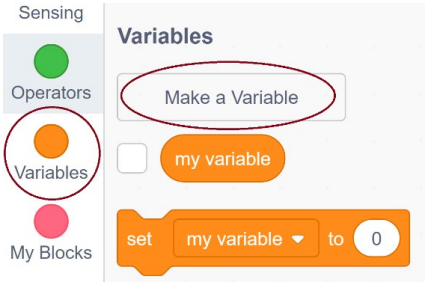
1.1 The first block (left) contains all the possible lines of code you can use on Scratch. The menu on the far left shows the different categories of code ("Motion", "Looks", "Sound", etc.)

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

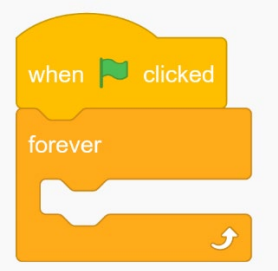
1.3 The third block (right) will show the end result of your code.

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!)

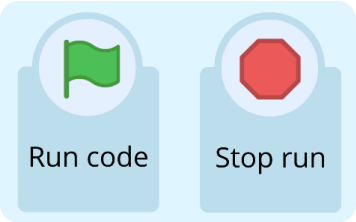
Teachers

The color-coding from the previous exercises is meant to guide students through the categories on Scratch.

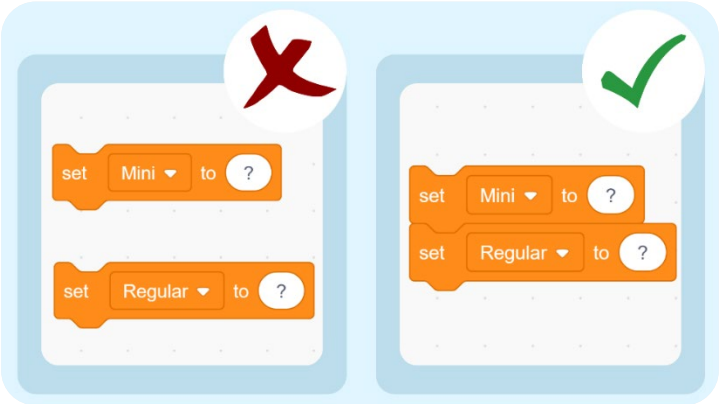
All variables ("userInput") were orange, just as the "Variables" category on Scratch. Operators were green, mirroring the green "Operators" category on Scratch.

Some useful tips:

- To run and stop the code:



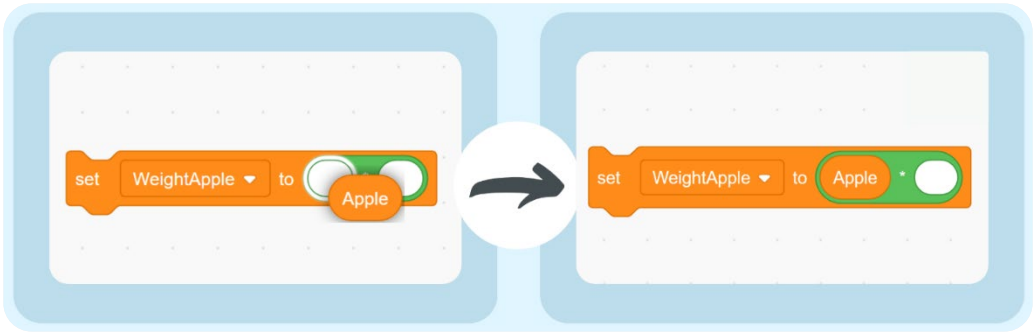
- To add new coding lines:



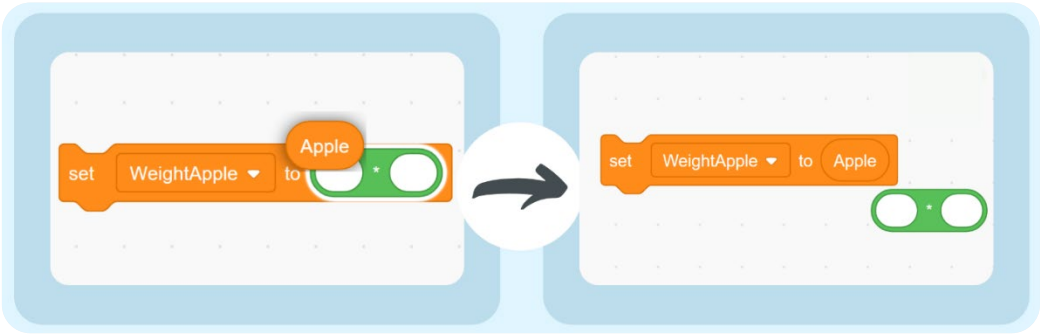
- To embed elements:

- 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?

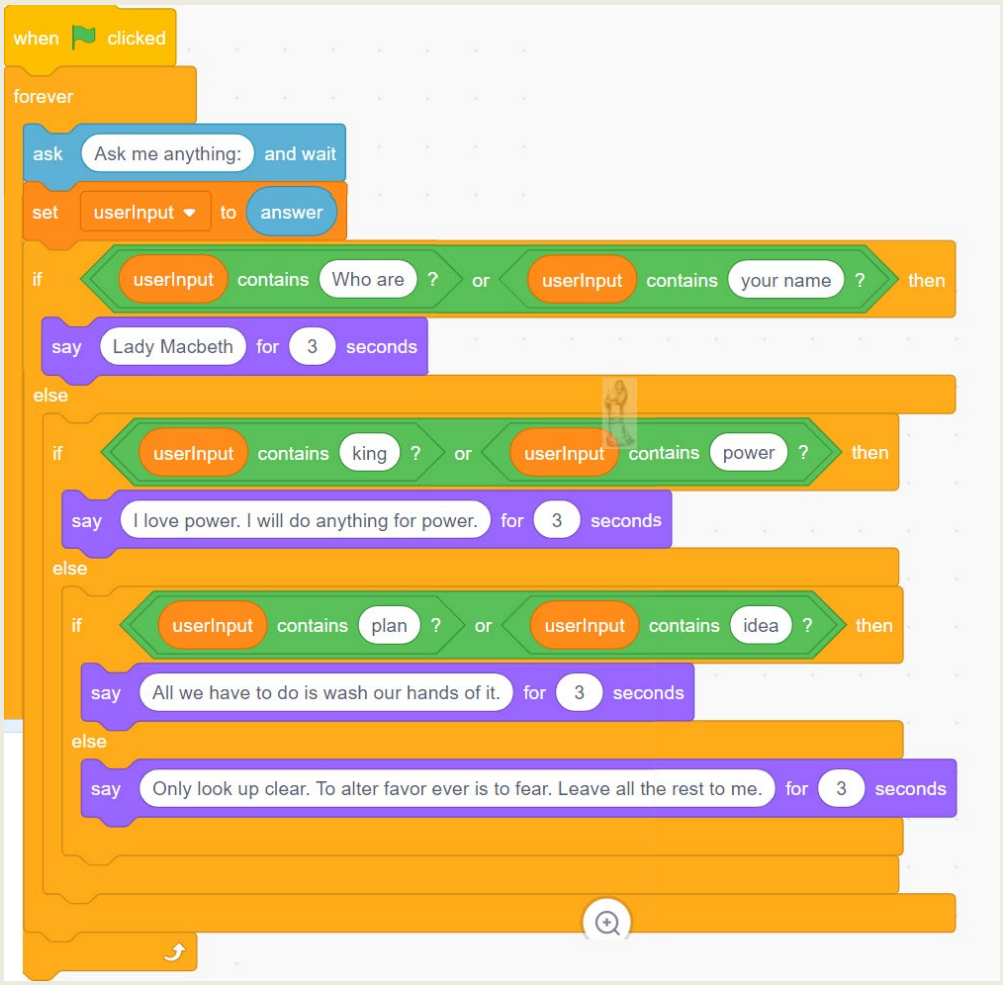
Hint: Remember the nested format of your pseudocode!

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!

Teachers

This is what the finished Scratch code looks like for the Lady Macbeth program. For the WWII module, everything should look the same except for the key words, the number of key words, the answers from the character, and the number of questions (five instead of three):



You can find the fully functional Lady Macbeth Scratch code here:

https://scratch.mit.edu/projects/554216995/editor/

If clicking on the link above does not work, try copying it directly to the address bar. Once you open the link, click on the “Lady Macbeth” sprite (next to the “Key” sprite, below the picture of the stairs) to see the full code.

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 your chosen historical figure 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.

Teachers

Answers will vary, but they should be similar to the following structure (continues on the next page). Remember that students may have *from one to two key words or combination of words* for each question. This means that the "OR" statement, and the statement that comes after it (the second key word/s*),* are optional.

It is essential to keep the indented structure of the code.

```
userInput = input("Ask me anything:")
if "key word/s" in userInput or "key word/s" in userInput:
        print("answer #1")
else:
```

```
    if "key word/s" in userInput or "key word/s" in userInput:
        print("answer #2")
    else:
        if "key word/s" in userInput or "key word/s" in userInput:
            print("answer #3")
        else:
            print("wildcard response")
```

Teachers

You may also use the full Python code for the Lady Macbeth program as a guide:

```
userInput = input("Ask me anything:")
if "Who are" in userInput or "your name" in userInput:
    print("Lady Macbeth.")
else:
    if "king" in userInput or "power" in userInput:
        print("I love power. I will do anything for power.")
    else:
        if "plan" in userInput or "idea" in userInput:
            print("All we have to do is wash our hands of it.")
        else:
            print("Only look up clear. To alter favor ever is to fear. Leave all
                the rest to me.")
```

Run Your Code

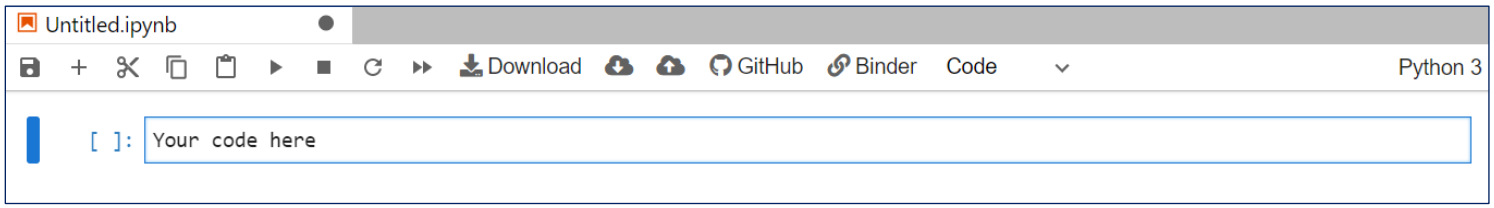
1. Open Jupyter Notebook: https://jupyter.org/try

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 historical figure to life by coding in Python.

Teachers

Here are some common typing errors in case you need to troubleshoot:

- **Not keeping the indented structure** of the code or **indenting at the wrong places.** (See the Python code given above to check where the indentation should go.)
- **Forgetting** a comma, parenthesis, or some other **punctuation mark**.
- **Order of characters.** (e.g. Misplacing a comma inside of the quotation marks when it should be outside, or vice versa.)
- **Mismatched quotation marks.** (e.g. Combining single with double quotation marks, or straight with curved quotation marks.)
- **Inconsistent capitalization of variables.** (e.g. Naming a variable "userInput" but then trying to use it as "UserInput".)

Note: Remind students that these are, not silly, but rather common mistakes even within the coding community. Even the most proficient coder can spend hours debugging his faulty program only to find that he had forgotten a comma!

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].