

Lesson 4

if, elif, else



CS IN SCHOOLS

Learning objectives

By the end of this lesson, you should be able to:

- understand the syntax of `if`, `elif` and `else` statements that can handle three or more options
- produce a program that uses `if`, `elif` and `else` statements
- understand the `in` operator and how to apply it in code
- create input, process, output (IPO) diagrams



Today's Inputs and Outputs

- Today we will be using the following **inputs** and **outputs**:

	Input	Output	Processing	Communication
Your computer	Keyboard Mouse Touch screen Microphone	Monitor/Screen Speakers	CPU Graphics cards	Wifi Bluetooth Ethernet
Your Microbit	Buttons Thermometer Accelerometer Magnetometer Touch sensor Light sensor	25 x LED lights Speakers		Bluetooth Radio



if statements for multiple options



[Click here](#) for more info on `if` statements

- In the previous course, we learned about `if` statements and how to control what happens when the expression was either `True` or `False`
- Notice with an `if` statement, we can only have two answers - `True` or `False` - these are known as **Boolean** values
- For example, `if` you press the on button, turn on
 - True: press on button → turn on
 - False: do not press on button → do not turn on



if statements for multiple options



[Click here](#) for more info
on `if` statements

- When is it useful to be able to accept more than two responses?
 -
 -
 -
- What could you build if you could accept more than two responses?
 -
 -
 -
 -



if statements for multiple options



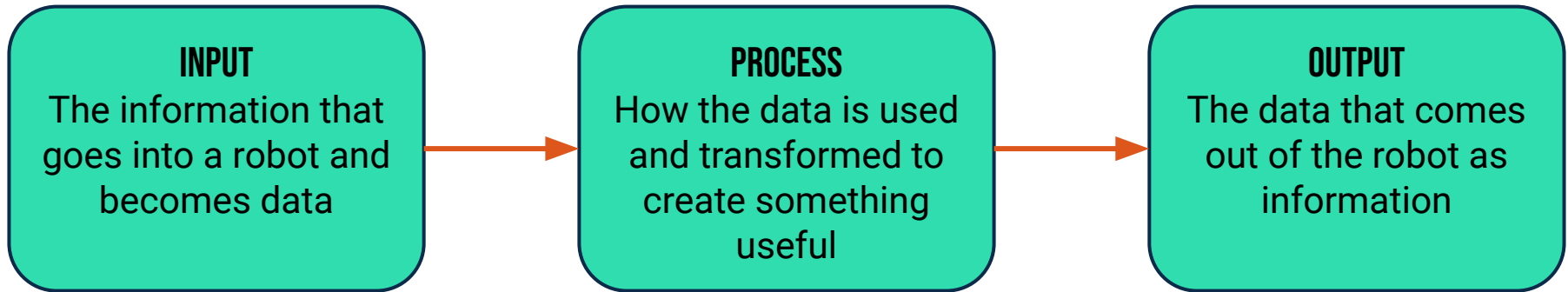
[Click here](#) for more info
on **if** statements

- When is it useful to be able to accept more than two responses?
 - Video games - direction, picking up objects, choosing what to do next
 - Changing the radio in your car - more than two radio stations!
 - Choosing a show on Netflix
- What could you build if you could accept more than two responses?
 - Games
 - Streaming sites
 - Remote controls
 - Online stores



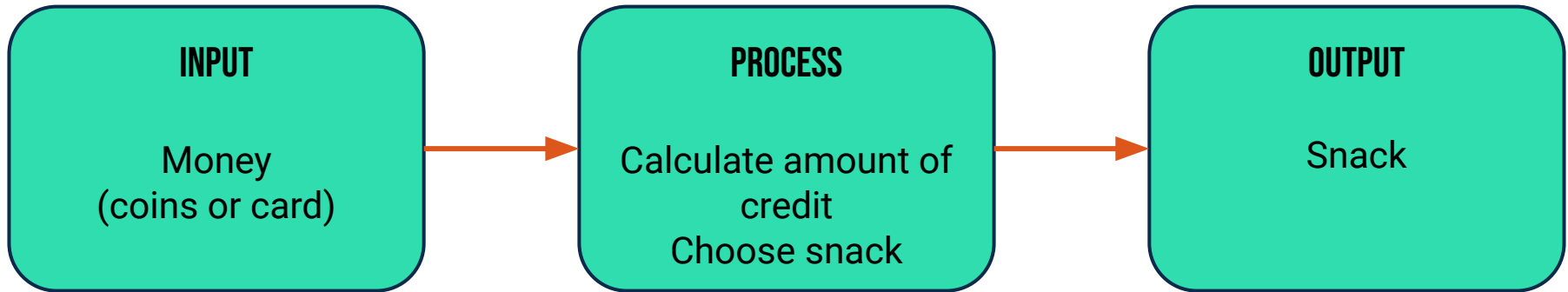
Input → Process → Output (IPO)

- **Input, Process, Output (IPO)** is a useful way to visualise how a robot takes in information, processes it as data, and then output the data as information
- We draw this as a small flowchart, seen below



Input → Process → Output (IPO)

- Let's consider the IPO for a vending machine:



Input → Process → Output (IPO)

- `if` statements let the micro:bit **process** the input, to produce a relevant output
- In the next demo, we will see the following **IPO** in action:



Demo: Voice Assistant



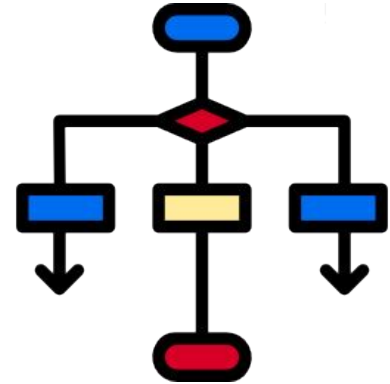
Voice Assistant

Voice Assistant Flowchart



Recap of Flowcharts

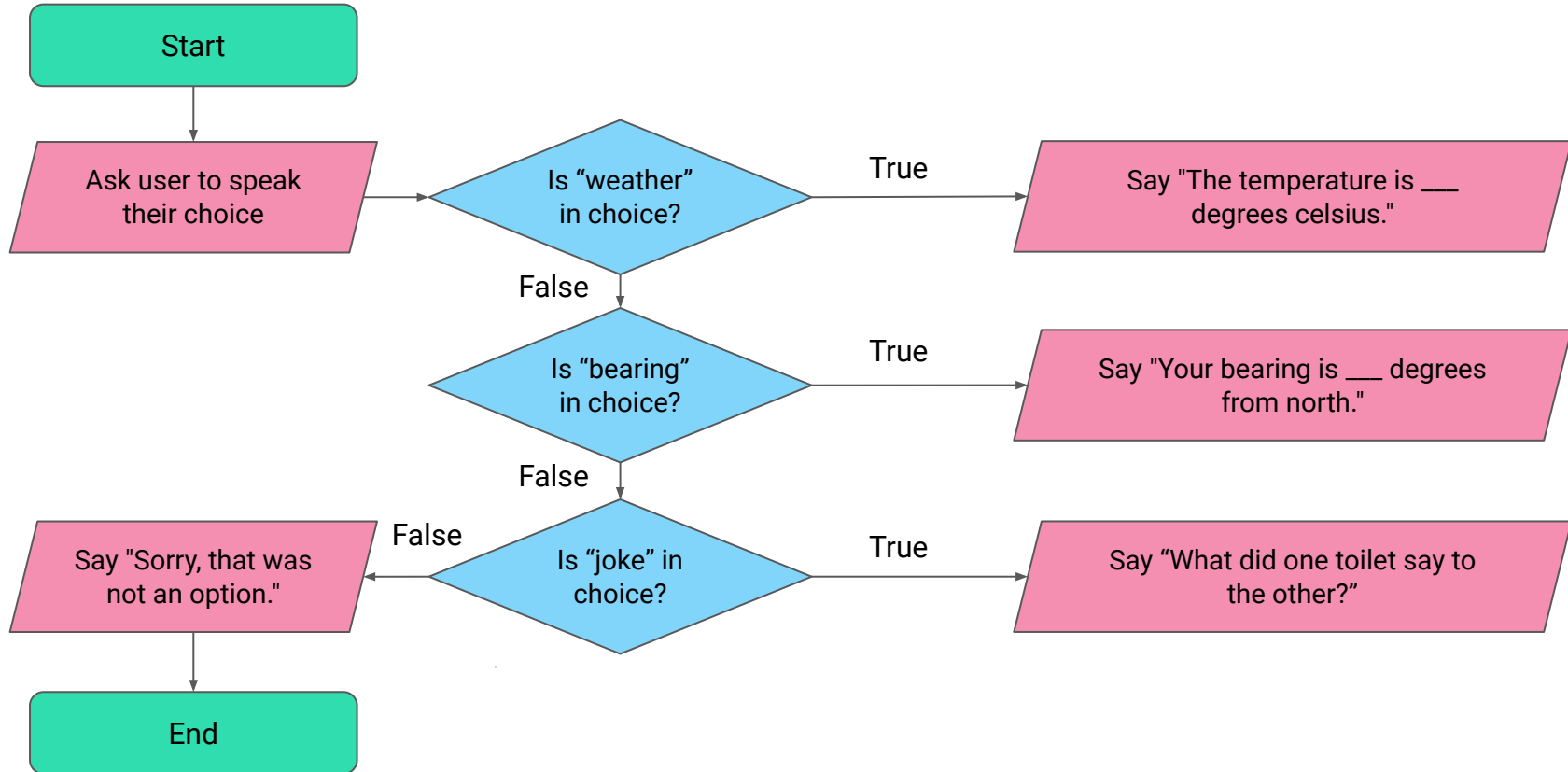
- A **flowchart** is a special type of diagram that is used by programmers to help them lay out their step-by-step design for their programs
- **Flowcharts** help us better understand the way computers make "decisions" and how they act differently based on different choices
- **Flowcharts** are written in English, and they often don't use the same keywords as a programming language



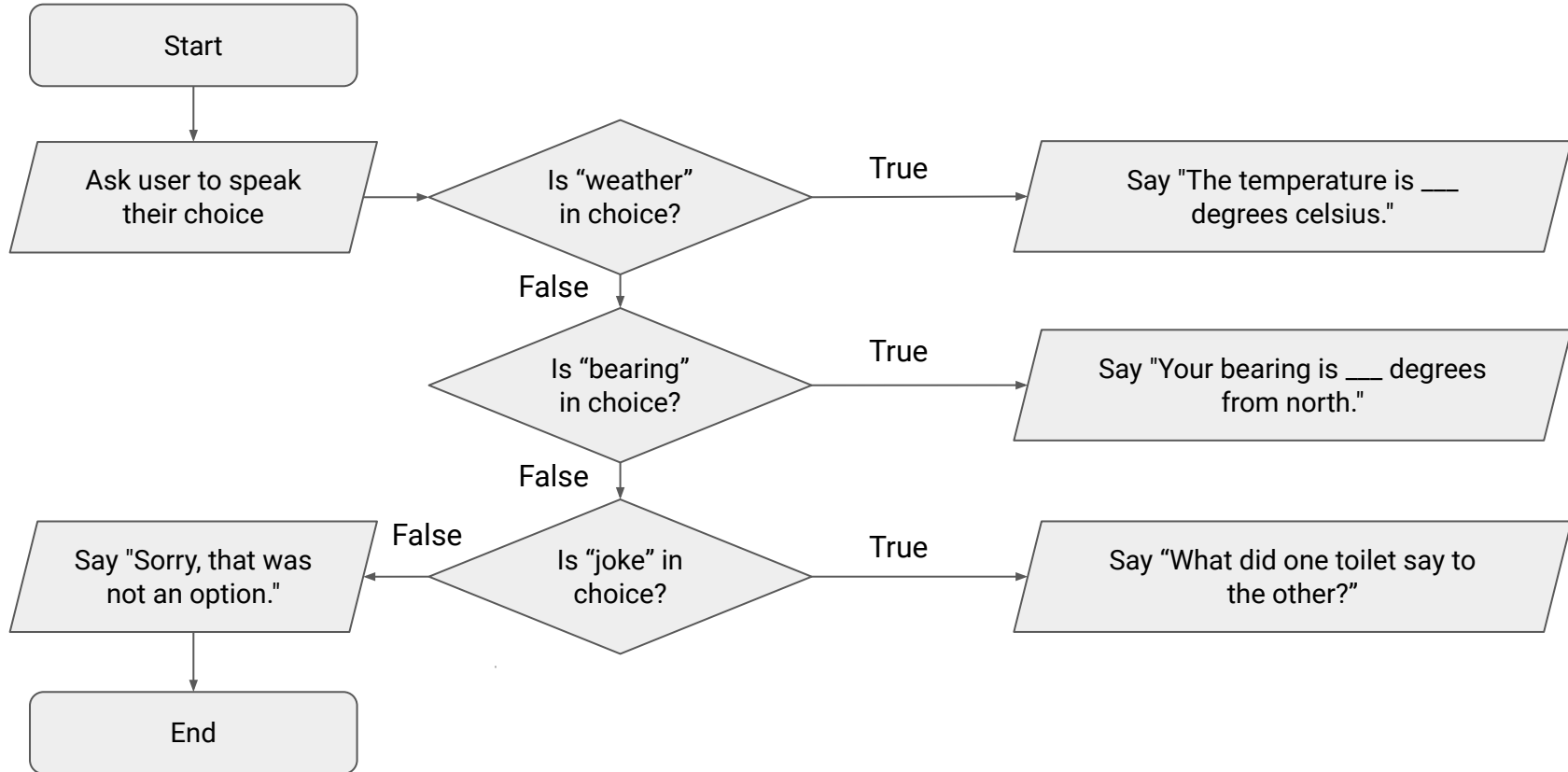
[Click here](#) for more info
on **flowcharts**



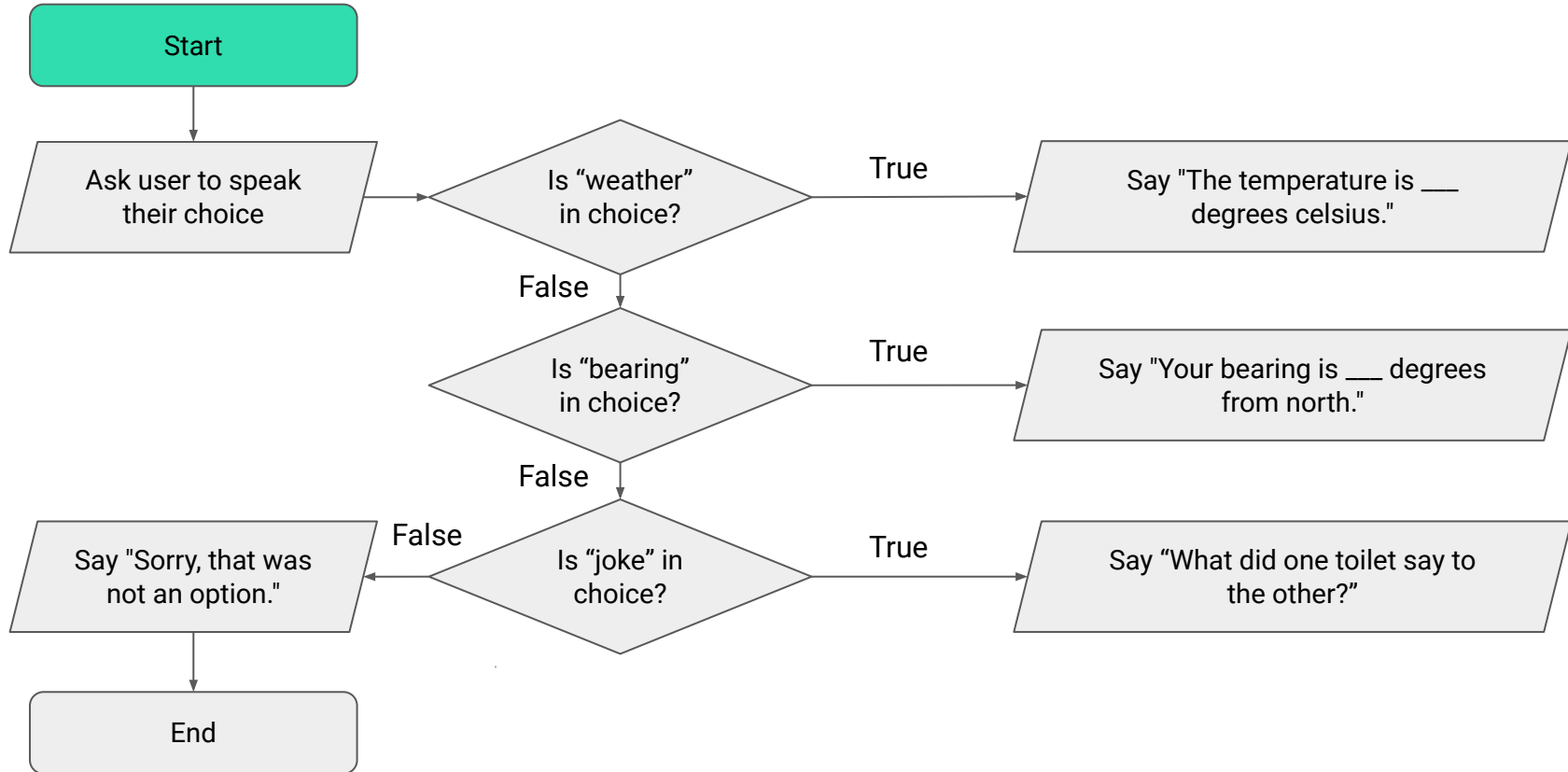
Voice Assistant Flowchart



Voice Assistant Flowchart

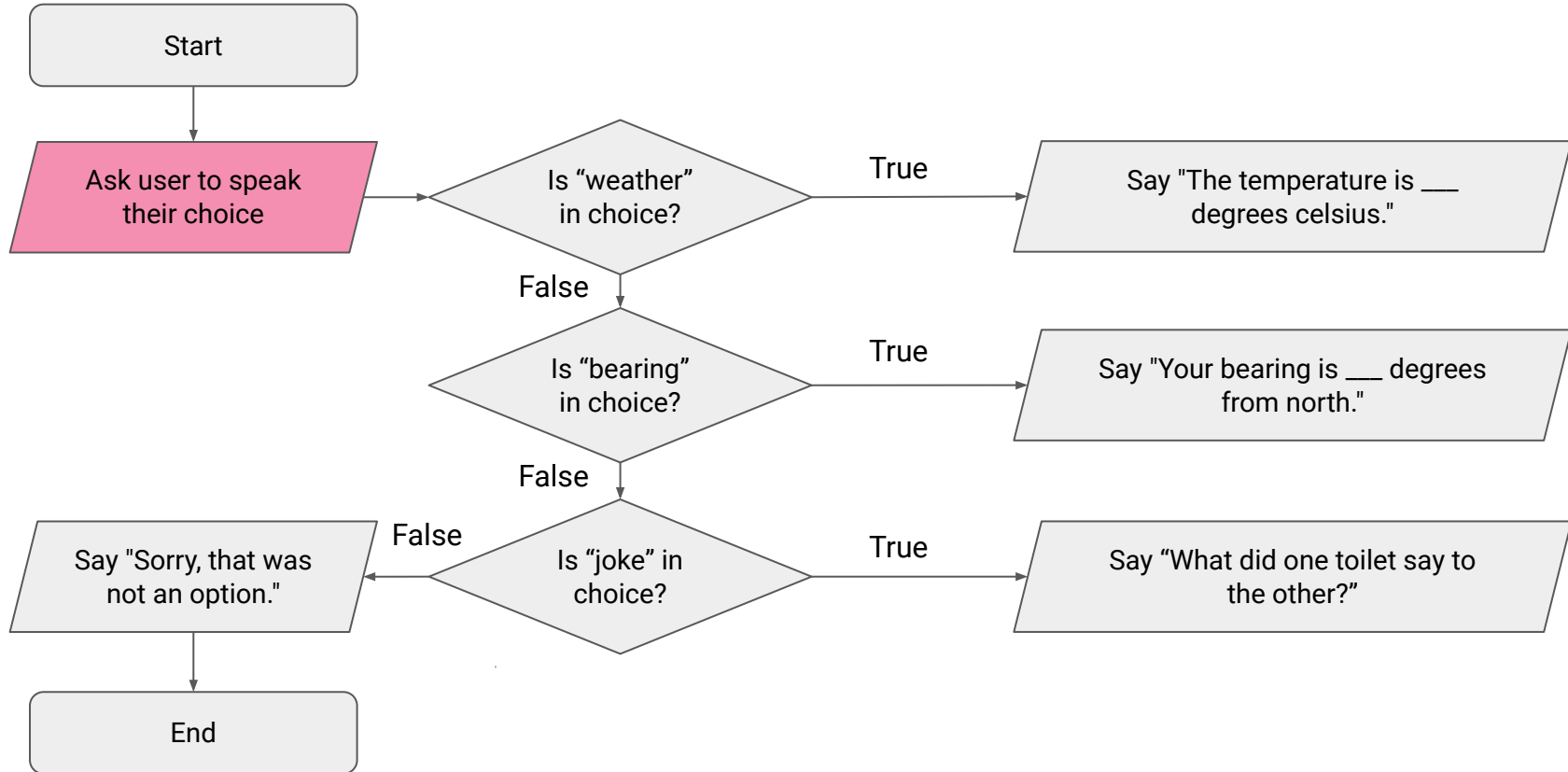


Voice Assistant Flowchart



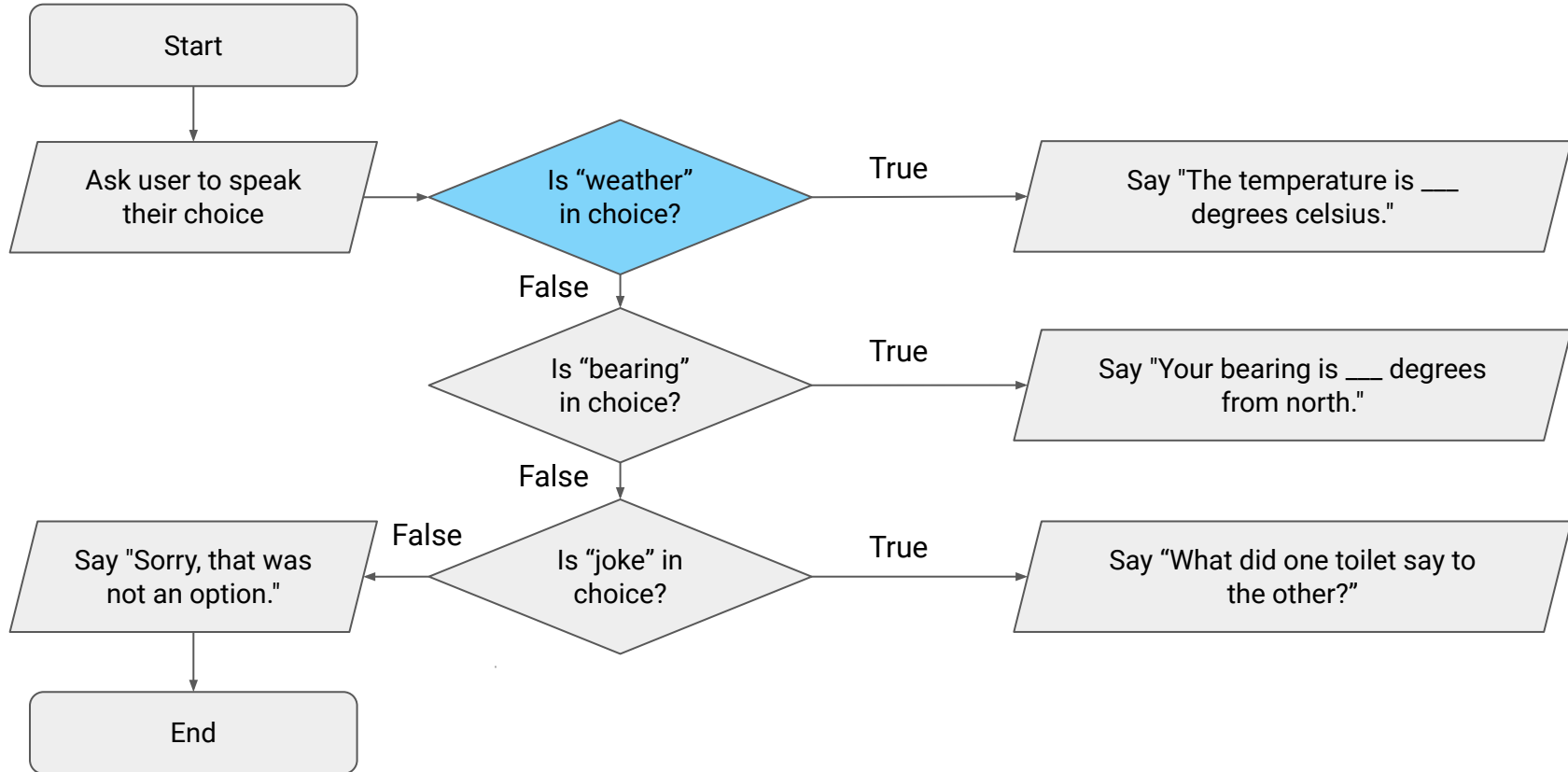
Voice Assistant Flowchart

Let's assume the user said:
"Tell me a joke."



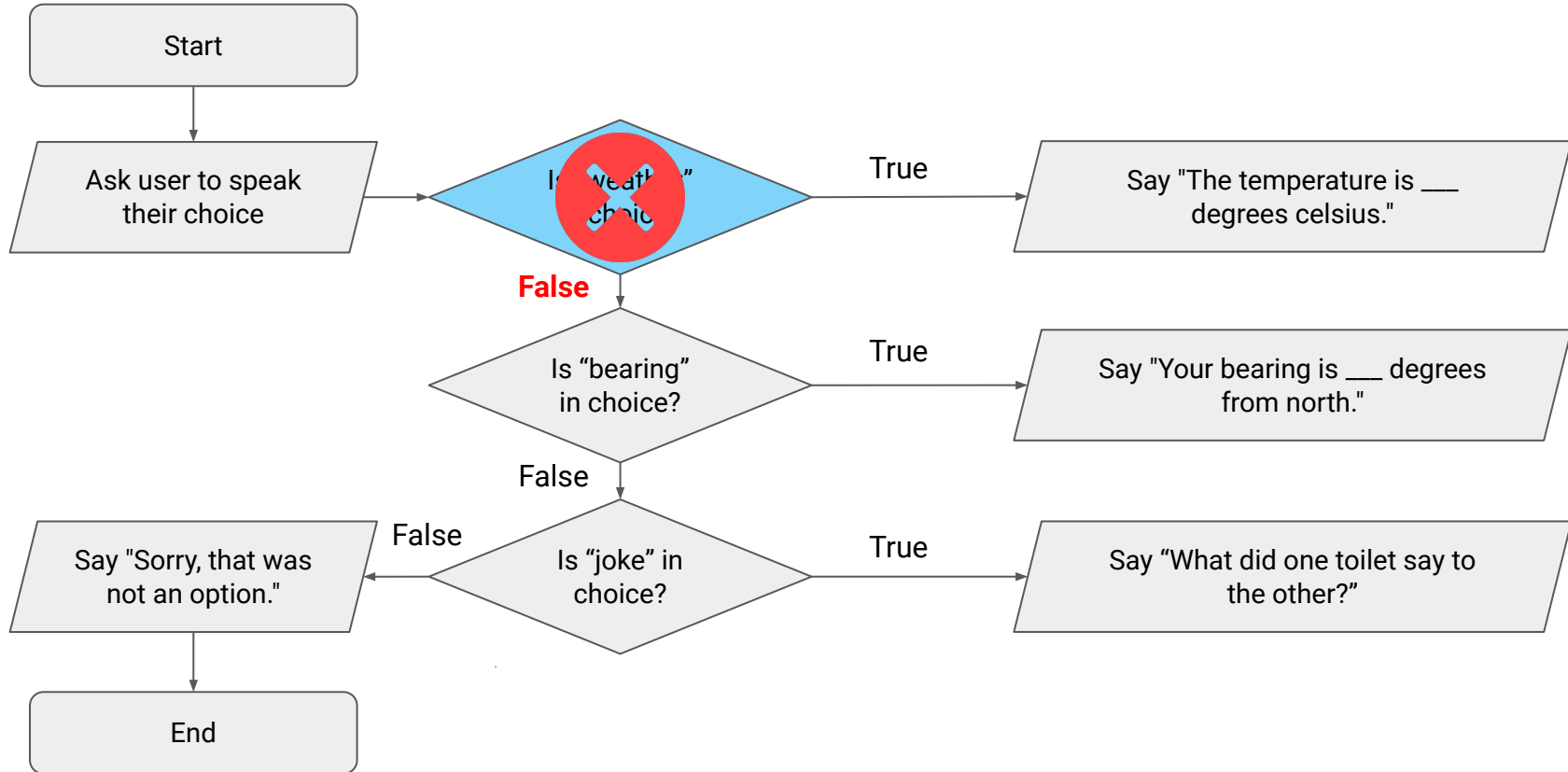
Voice Assistant Flowchart

Let's assume the user said:
"Tell me a joke."



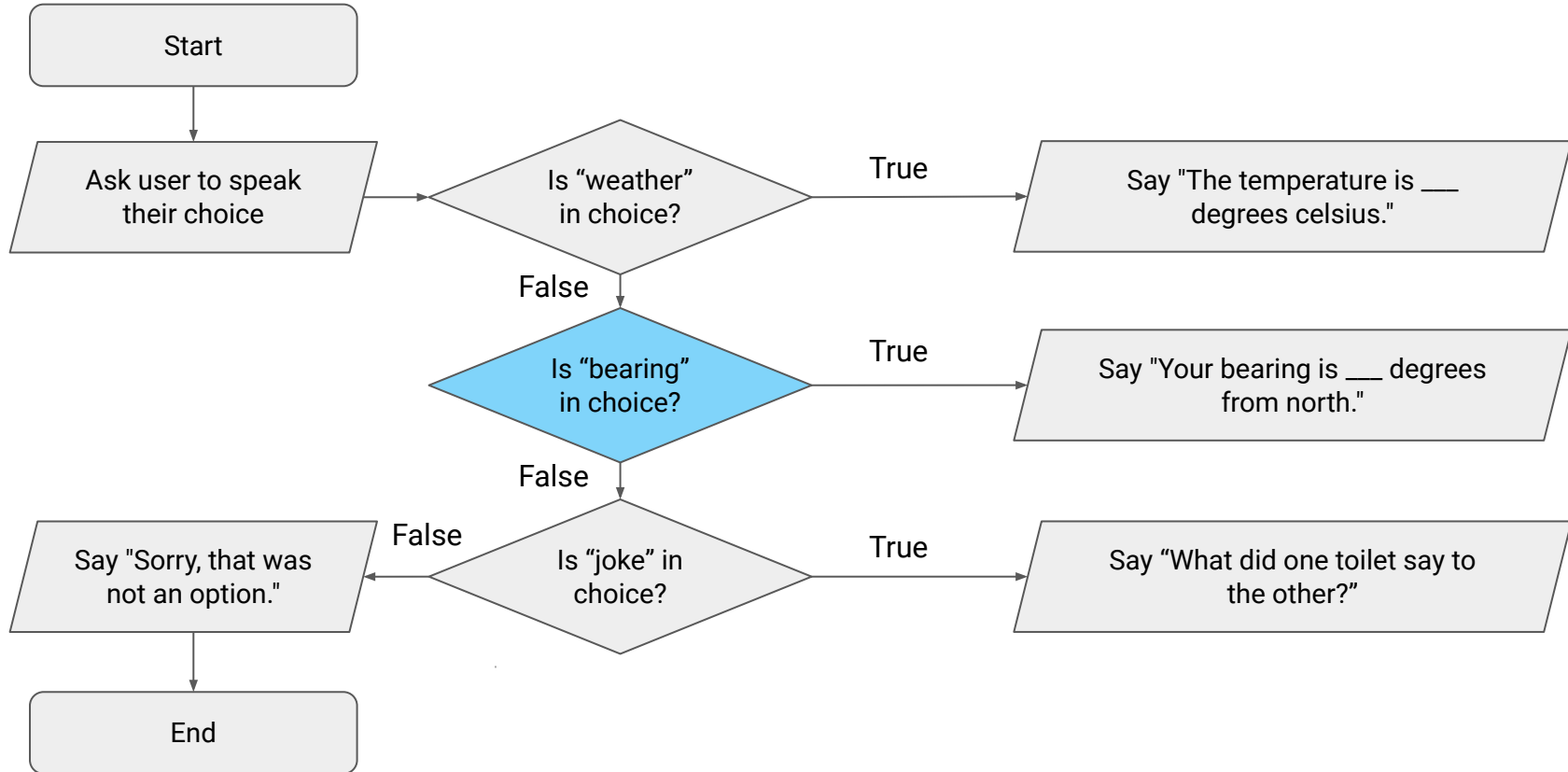
Voice Assistant Flowchart

Let's assume the user said:
"Tell me a joke."



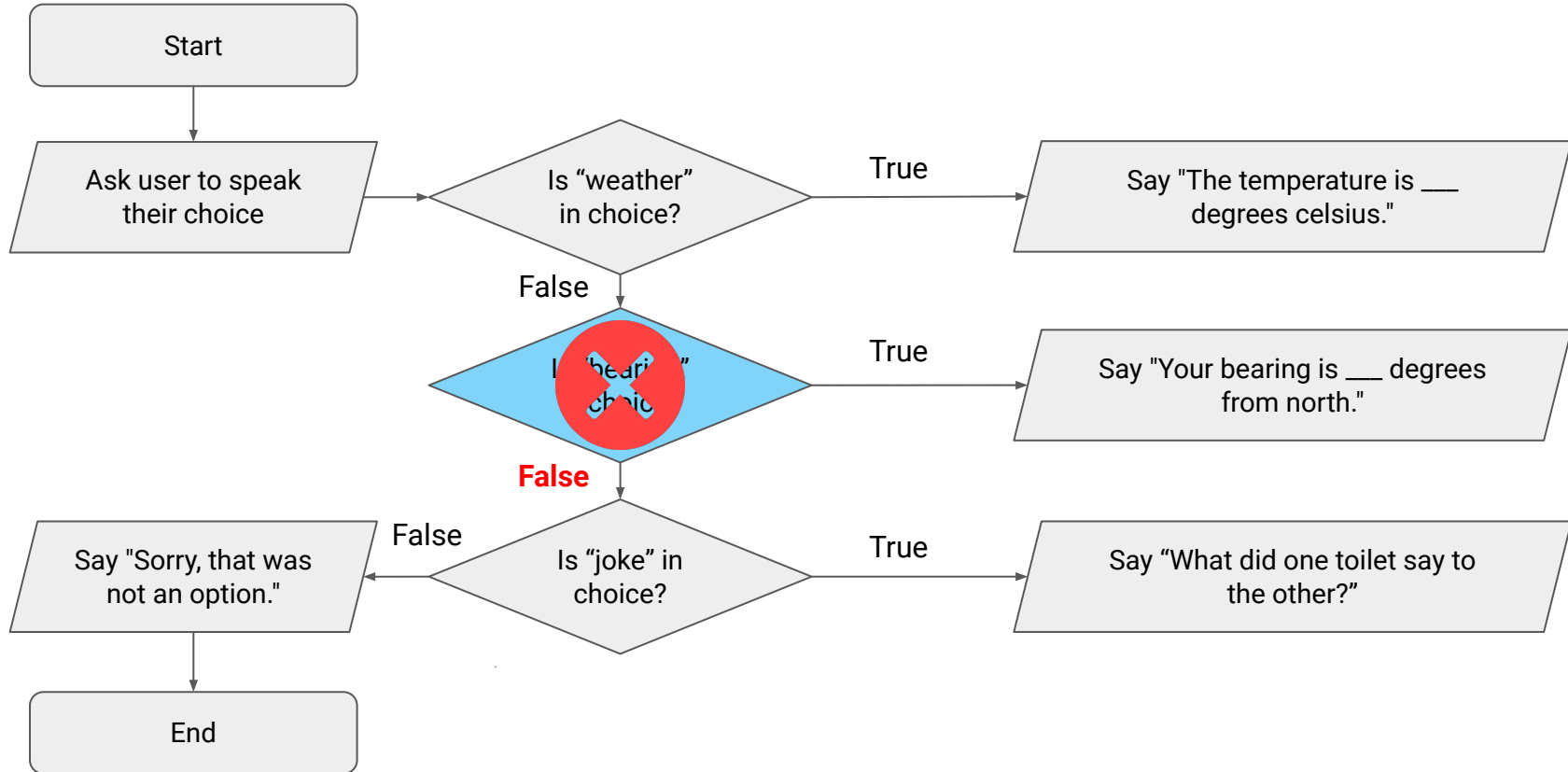
Voice Assistant Flowchart

Let's assume the user said:
"Tell me a joke."



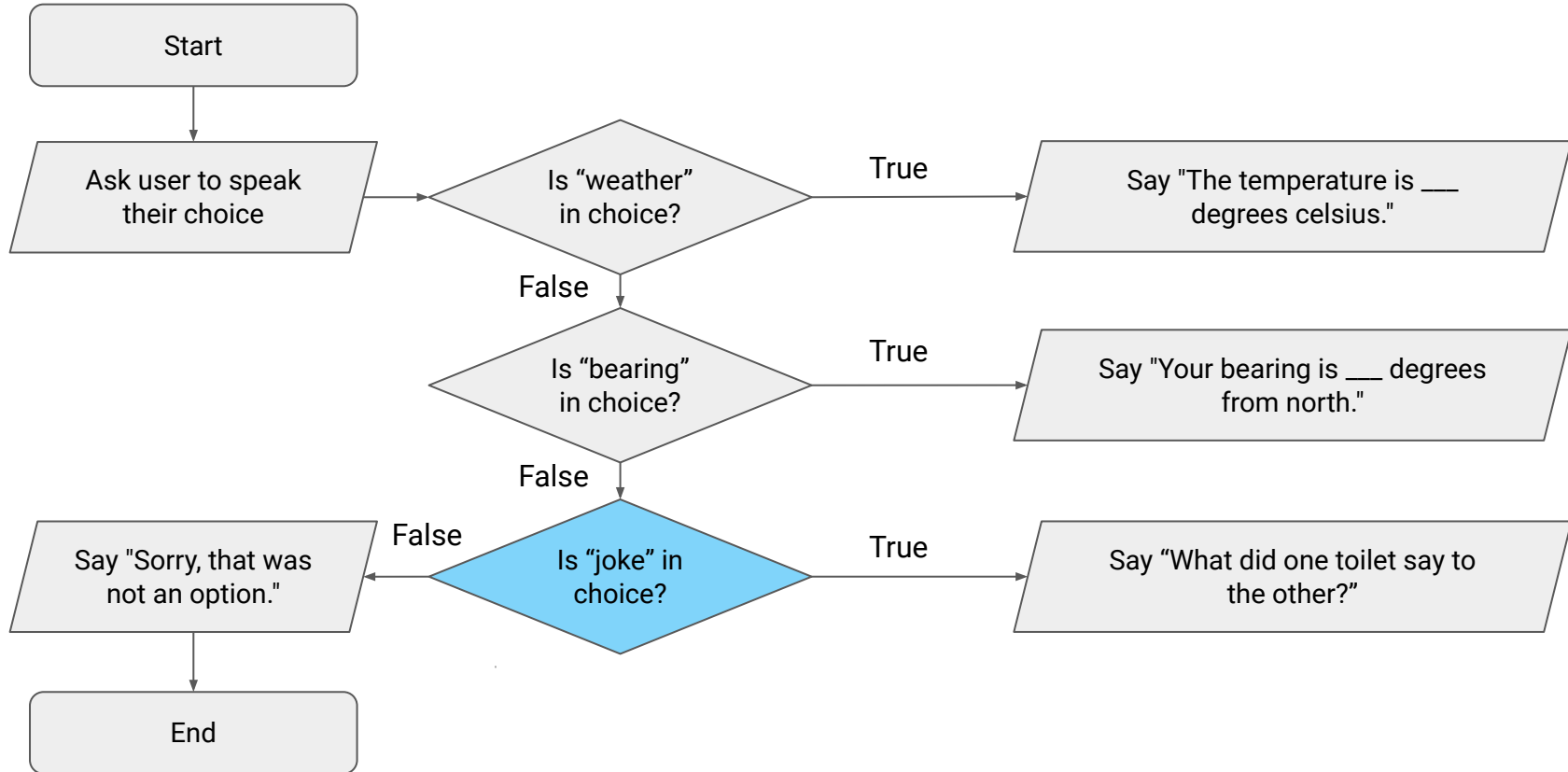
Voice Assistant Flowchart

Let's assume the user said:
"Tell me a joke."



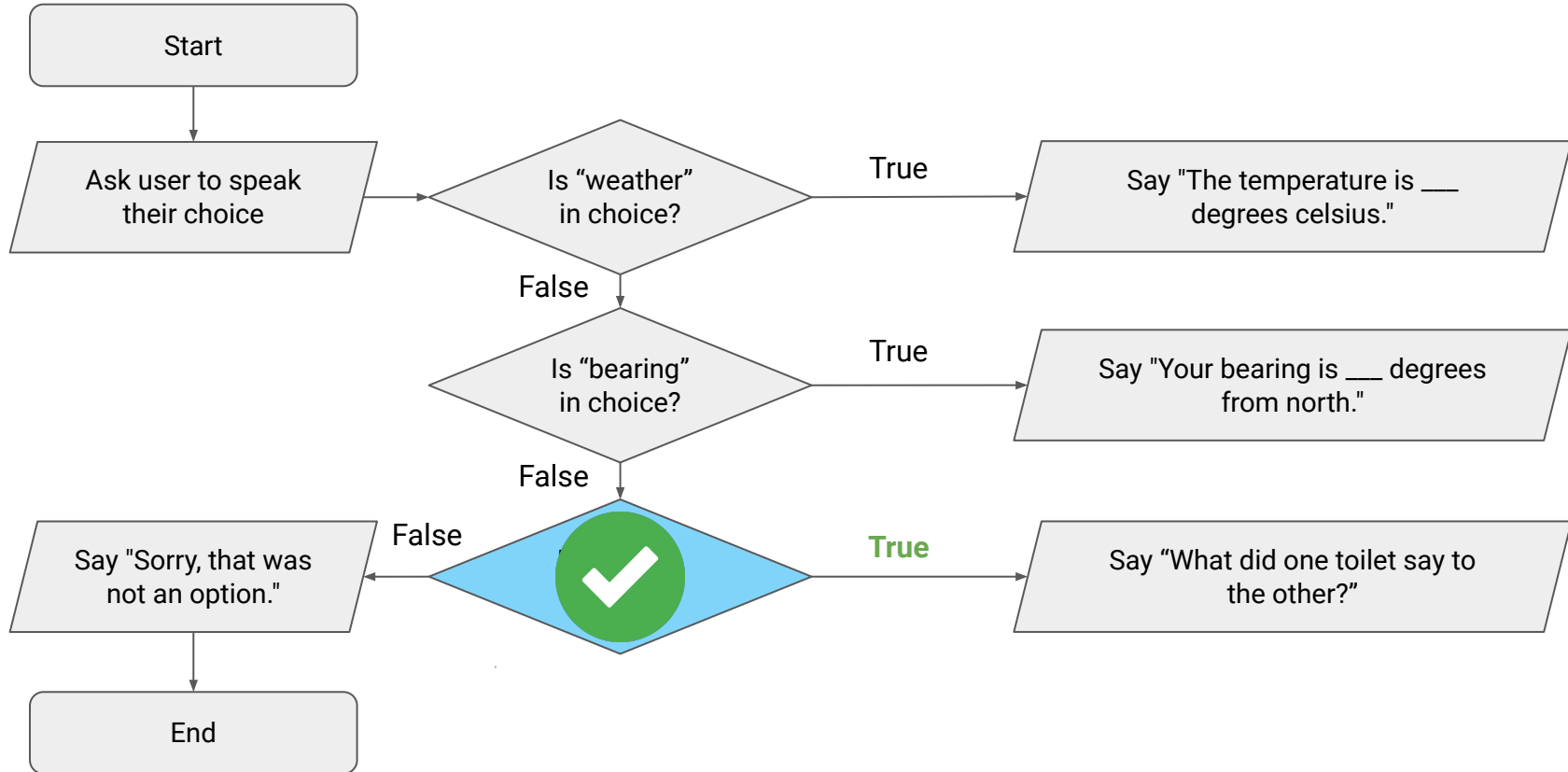
Voice Assistant Flowchart

Let's assume the user said:
"Tell me a joke."



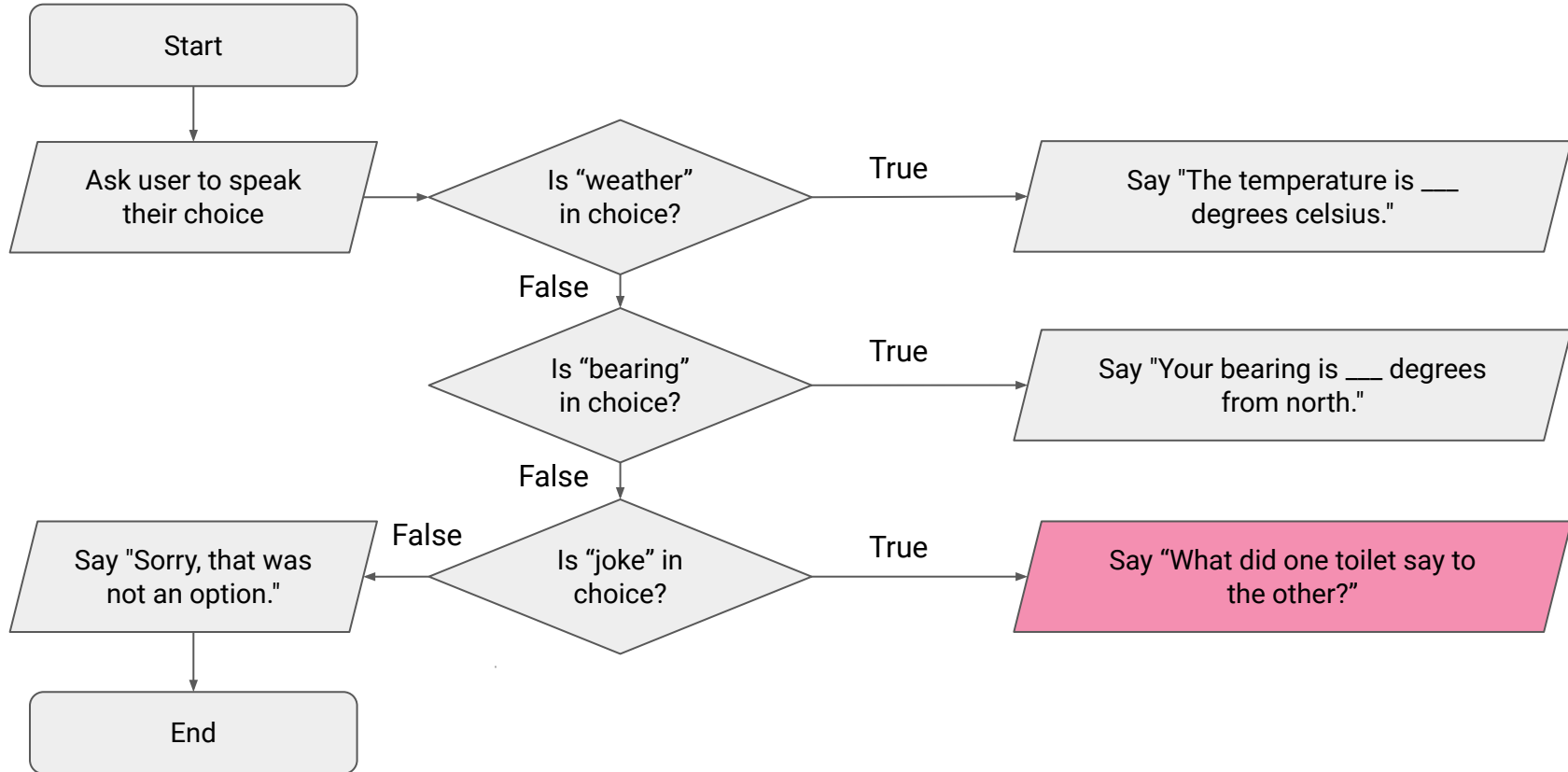
Voice Assistant Flowchart

Let's assume the user said:
"Tell me a joke."



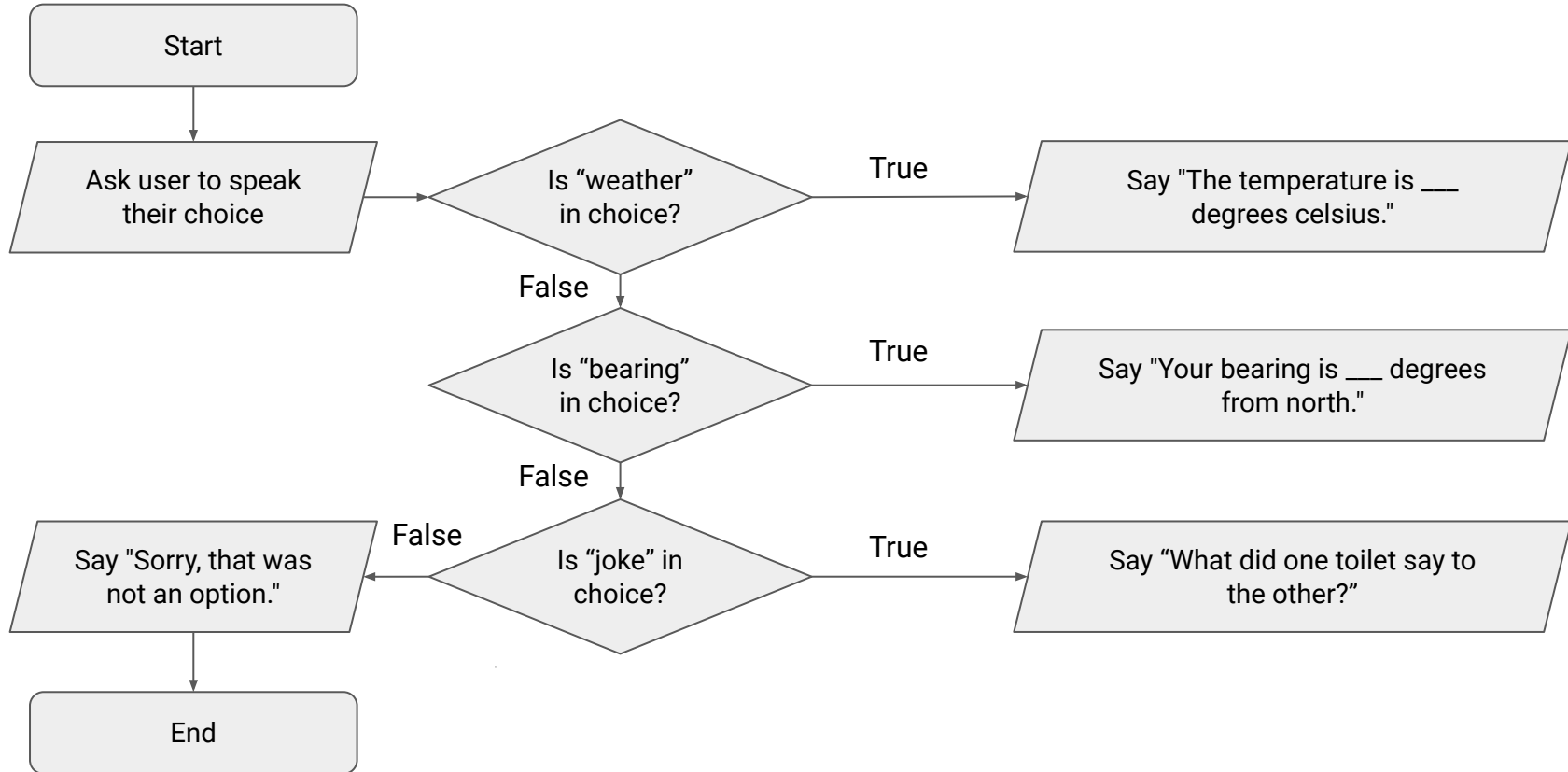
Voice Assistant Flowchart

Let's assume the user said:
"Tell me a joke."



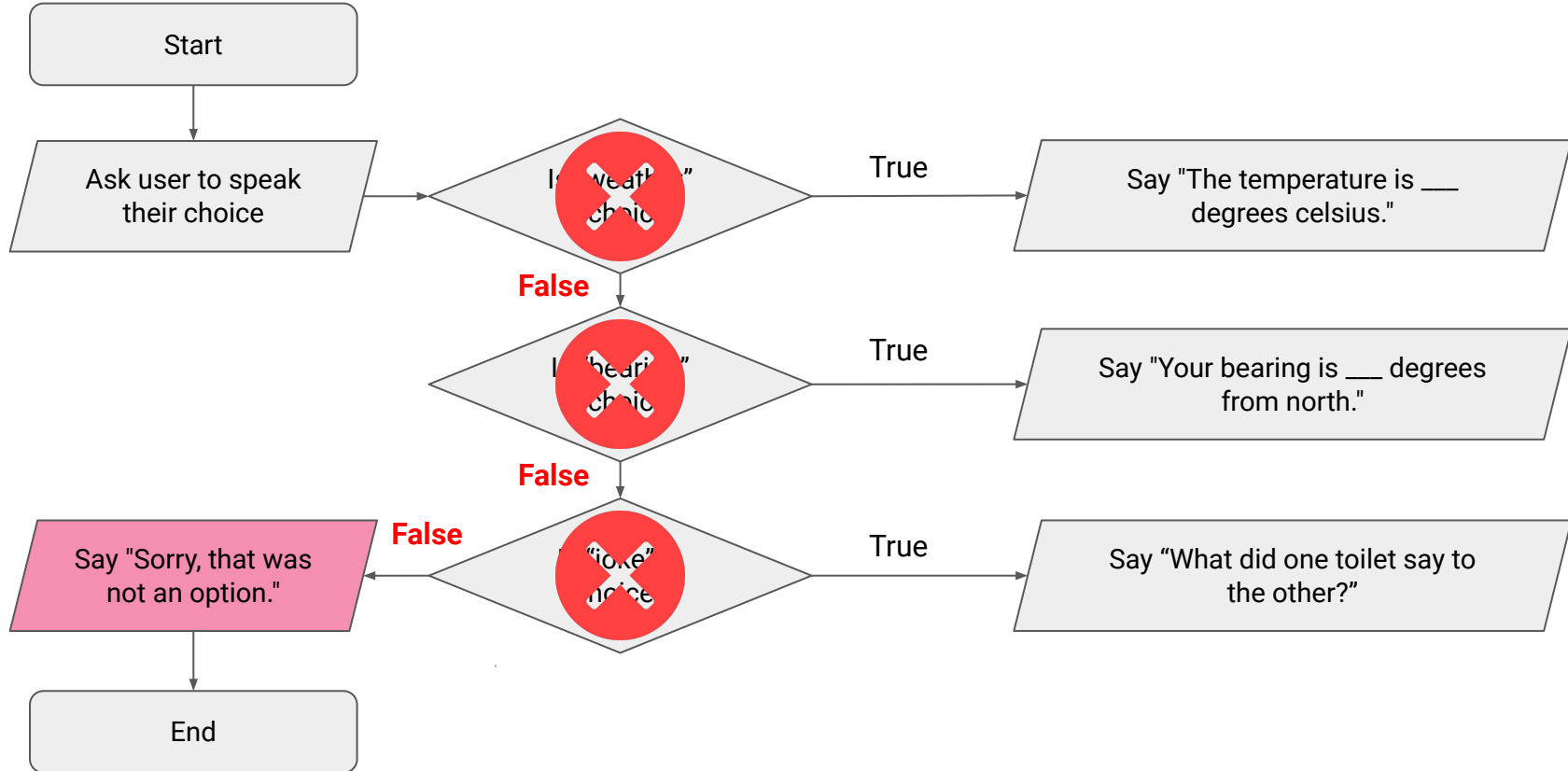
Voice Assistant Flowchart

What if the user said:
"How old are you?"

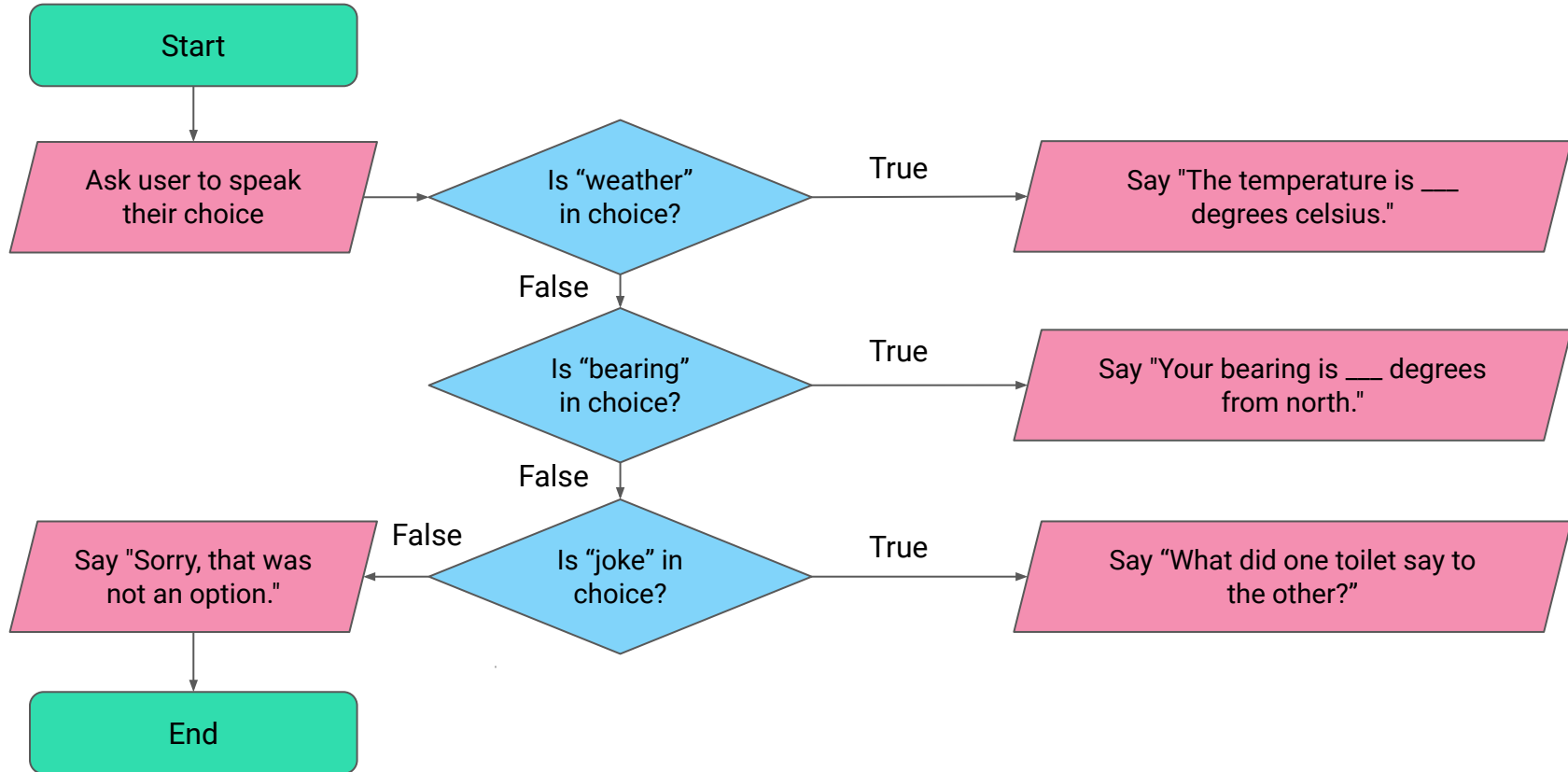


Voice Assistant Flowchart

What if the user said:
"How old are you?"



Voice Assistant Flowchart



Voice Assistant Code Walkthrough



Voice Assistant: Part 1

```
from speech import *
from microbit import *
from csinsc import *

# Connect to the micro:bit
assistant = Microbit()

label .again

print("Press any button...")
assistant.waitForButtonPress()

# Get the instruction
print("Speak!")
choice = listen(5)
print("I heard you say: " + choice)
```

Voice Assistant: Part 1

Code we are familiar with:

- What do these lines of code do?

```
from speech import *  
from microbit import *  
from csinsc import *
```

```
# Connect to the micro:bit  
assistant = Microbit()
```

```
label .again
```

```
print("Press any button...")  
assistant.waitForButtonPress()
```

```
# Get the instruction  
print("Speak!")  
choice = listen(5)  
print("I heard you say: " + choice)
```

Voice Assistant: Part 1

Code we are familiar with:

- What do these lines of code do?
 1. Let's us use `say()` and `listen()`

```
from speech import *
```

```
from microbit import *
```

```
from csinsc import *
```

```
# Connect to the micro:bit
```

```
assistant = Microbit()
```

```
label .again
```

```
print("Press any button...")
```

```
assistant.waitForButtonPress()
```

```
# Get the instruction
```

```
print("Speak!")
```

```
choice = listen(5)
```

```
print("I heard you say: " + choice)
```

Voice Assistant: Part 1

Code we are familiar with:

- What do these lines of code do?
 1. Let's us use `say()` and `listen()`
 2. Let's us connect to and use the **micro:bit**

```
from speech import *
```

```
from microbit import *
```

```
from csinsc import *
```

```
# Connect to the micro:bit
```

```
assistant = Microbit()
```

```
label .again
```

```
print("Press any button...")
```

```
assistant.waitForButtonPress()
```

```
# Get the instruction
```

```
print("Speak!")
```

```
choice = listen(5)
```

```
print("I heard you say: " + choice)
```

Voice Assistant: Part 1

Code we are familiar with:

- What do these lines of code do?
 1. Let's us use `say()` and `listen()`
 2. Let's us connect to and use the **micro:bit**
 3. Let's us use `label`, `goto` and colours

```
from speech import *  
from microbit import *  
from csinsc import *
```

```
# Connect to the micro:bit
```

```
assistant = Microbit()
```

```
label .again
```

```
print("Press any button...")
```

```
assistant.waitForButtonPress()
```

```
# Get the instruction
```

```
print("Speak!")
```

```
choice = listen(5)
```

```
print("I heard you say: " + choice)
```


Voice Assistant: Part 1

Code we are familiar with:

- What is this?

```
from speech import *
from microbit import *
from csinsc import *

# Connect to the micro:bit
assistant = Microbit()

label .again

print("Press any button...")
assistant.waitForButtonPress()

# Get the instruction
print("Speak!")
choice = listen(5)
print("I heard you say: " + choice)
```

Voice Assistant: Part 1

Code we are familiar with:

- What is this?
 - A **comment**! Only humans read this, Python ignores it

```
from speech import *  
from microbit import *  
from csinsc import *
```

```
# Connect to the micro:bit
```

```
assistant = Microbit()
```

```
label .again
```

```
print("Press any button...")
```

```
assistant.waitForButtonPress()
```

```
# Get the instruction
```

```
print("Speak!")
```

```
choice = listen(5)
```

```
print("I heard you say: " + choice)
```

Voice Assistant: Part 1

Code we are familiar with:

- What does this do?

```
from speech import *  
from microbit import *  
from csinsc import *
```

```
# Connect to the micro:bit
```

```
assistant = Microbit()
```

```
label .again
```

```
print("Press any button...")
```

```
assistant.waitForButtonPress()
```

```
# Get the instruction
```

```
print("Speak!")
```

```
choice = listen(5)
```

```
print("I heard you say: " + choice)
```

Voice Assistant: Part 1

Code we are familiar with:

- What does this do?
 - **Creates the connection** between the micro:bit and the computer

```
from speech import *  
from microbit import *  
from csinsc import *
```

```
# Connect to the micro:bit
```

```
assistant = Microbit()
```

```
label .again
```

```
print("Press any button...")
```

```
assistant.waitForButtonPress()
```

```
# Get the instruction
```

```
print("Speak!")
```

```
choice = listen(5)
```

```
print("I heard you say: " + choice)
```

Voice Assistant: Part 1

Code we are familiar with:

- This **label** pairs with the **goto** later on to create an **infinite loop** - a loop that will never end
- This means we can make multiple requests to our assistant without having to re-run the program

```
from speech import *  
from microbit import *  
from csinsc import *
```

```
# Connect to the micro:bit  
assistant = Microbit()
```

```
label .again
```

```
print("Press any button...")  
assistant.waitForButtonPress()
```

```
# Get the instruction  
print("Speak!")  
choice = listen(5)  
print("I heard you say: " + choice)
```

Voice Assistant: Part 1

Code we are familiar with:

- Here we **print** the instruction to the user
- We want them to press any button, then speak, so in the demo we tell them to do this

```
from speech import *
from microbit import *
from csinsc import *

# Connect to the micro:bit
assistant = Microbit()

label .again

print("Press any button...")
assistant.waitForButtonPress()

# Get the instruction
print("Speak!")
choice = listen(5)
print("I heard you say: " + choice)
```

Voice Assistant: Part 1

Code we are familiar with:

- What does this line of code do?

```
from speech import *
from microbit import *
from csinsc import *

# Connect to the micro:bit
assistant = Microbit()

label .again

print("Press any button...")
assistant.waitForButtonPress()

# Get the instruction
print("Speak!")
choice = listen(5)
print("I heard you say: " + choice)
```

Voice Assistant: Part 1

Code we are familiar with:

- What does this line of code do?
 - It will make the program pause until the user **presses a button**

```
from speech import *
from microbit import *
from csinsc import *

# Connect to the micro:bit
assistant = Microbit()

label .again

print("Press any button...")
assistant.waitForButtonPress()

# Get the instruction
print("Speak!")
choice = listen(5)
print("I heard you say: " + choice)
```


Voice Assistant: Part 1

Code we are familiar with:

- Another comment...
- Another print statement...

```
from speech import *
from microbit import *
from csinsc import *

# Connect to the micro:bit
assistant = Microbit()

label .again

print("Press any button...")
assistant.waitForButtonPress()

# Get the instruction
print("Speak!")
choice = listen(5)
print("I heard you say: " + choice)
```

Voice Assistant: Part 1

- Here we have the `listen()` function
- You may have used this before
- It **listens** to the user for the amount of seconds you specify, then stores what the user says inside the variable on the left
- In this case, the program will listen for **5 seconds**, and store what was said inside `choice`

```
from speech import *
from microbit import *
from csinsc import *

# Connect to the micro:bit
assistant = Microbit()

label .again

print("Press any button...")
assistant.waitForButtonPress()

# Get the instruction
print("Speak!")
choice = listen(5)
print("I heard you say: " + choice)
```

Voice Assistant: Part 1

- The **choice** variable now contains the words that were spoken by the user, as a **string**

```
from speech import *
from microbit import *
from csinsc import *

# Connect to the micro:bit
assistant = Microbit()

label .again

print("Press any button...")
assistant.waitForButtonPress()

# Get the instruction
print("Speak!")
choice = listen(5)
print("I heard you say: " + choice)
```

Voice Assistant: Part 1

- The `choice` variable now contains the words that were spoken by the user, as a `string`
- So this `print` statement will display "I heard you say:" and whatever the user said

```
from speech import *  
from microbit import *  
from csinsc import *
```

```
# Connect to the micro:bit  
assistant = Microbit()
```

```
label .again
```

```
print("Press any button...")  
assistant.waitForButtonPress()
```

```
# Get the instruction  
print("Speak!")  
choice = listen(5)
```

```
print("I heard you say: " + choice)
```

Voice Assistant: Part 2

- Part 1 of the code demonstrated how to get the **choice** from the user using the **listen()** function
- Now, we will discuss how to use this **choice** to make a decision

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```

Voice Assistant: Part 2

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```

Voice Assistant: Part 2

- Here is where our program can make more than one decision
- It will check the first statement, then the second, then the third, then the fourth
- If it makes a check and the statement is **True**, it will skip the rest and move onto the next part of the program
- Let's see this in action...

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```

Voice Assistant: Part 2

The user said: "Tell me a joke"

- In the first statement, Python is checking to see if the word "weather" is contained in the variable choice
- Remember, choice contains what the user said!

```
if "weather" in choice:
```

```
    message = "The temperature is..."
```

```
elif "bearing" in choice:
```

```
    message = "Your bearing is..."
```

```
elif "joke" in choice:
```

```
    message = "What did one toilet say..."
```

```
else:
```

```
    message = "Sorry, I did not understand"
```

```
# Say the message and return to the start
```

```
print("My response was: " + message + "\n")
```

```
say(message)
```

```
goto .again
```


Voice Assistant: Part 2

The user said: "Tell me a joke"

- In the first statement, Python is checking to see if the word "weather" is contained in the variable choice
- Remember, choice contains what the user said!

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```

Voice Assistant: Part 2

`choice: "Tell me a joke"`

- In the first statement, Python is checking to see if the word `"weather"` is contained in the variable `choice`
- Remember, `choice` contains what the user said!

```
if "weather" in choice:
```

```
    message = "The temperature is..."
```

```
elif "bearing" in choice:
```

```
    message = "Your bearing is..."
```

```
elif "joke" in choice:
```

```
    message = "What did one toilet say..."
```

```
else:
```

```
    message = "Sorry, I did not understand"
```

```
# Say the message and return to the start
```

```
print("My response was: " + message + "\n")
```

```
say(message)
```

```
goto .again
```

Voice Assistant: Part 2

choice: "Tell me a joke"

- Is the word "weather" contained in the choice variable?

```
if "weather" in choice:
```

```
    message = "The temperature is..."
```

```
elif "bearing" in choice:
```

```
    message = "Your bearing is..."
```

```
elif "joke" in choice:
```

```
    message = "What did one toilet say..."
```

```
else:
```

```
    message = "Sorry, I did not understand"
```

```
# Say the message and return to the start
```

```
print("My response was: " + message + "\n")
```

```
say(message)
```

```
goto .again
```

Voice Assistant: Part 2

choice: "Tell me a joke"

- Is the word "weather" contained in the choice variable?
- No - False!
- So we check the next option

```
if "weather" in choice:
```

```
    message = "The temperature is..."
```

```
elif "bearing" in choice:
```

```
    message = "Your bearing is..."
```

```
elif "joke" in choice:
```

```
    message = "What did one toilet say..."
```

```
else:
```

```
    message = "Sorry, I did not understand"
```

```
# Say the message and return to the start
```

```
print("My response was: " + message + "\n")
```

```
say(message)
```

```
goto .again
```

Voice Assistant: Part 2

choice: "Tell me a joke"

- Is the word "bearing" contained in the choice variable?

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```

Voice Assistant: Part 2

choice: "Tell me a joke"

- Is the word "bearing" contained in the choice variable?
- No - False!
- So we check the next option

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```

Voice Assistant: Part 2

choice: "Tell me a joke"

- Is the word "joke" contained in the choice variable?

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```

Voice Assistant: Part 2

choice: "Tell me a joke"

- Is the word "joke" contained in the choice variable?
- Yes - True!
- Now, just like an `if` statement, we execute the code indented and underneath

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```


Voice Assistant: Part 2

choice: "Tell me a joke"

- Is the word "joke" contained in the choice variable?
- Yes - True!
- Now, just like an `if` statement, we execute the code indented and underneath
- Here we are storing a string inside the `message` variable, to be said later

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```

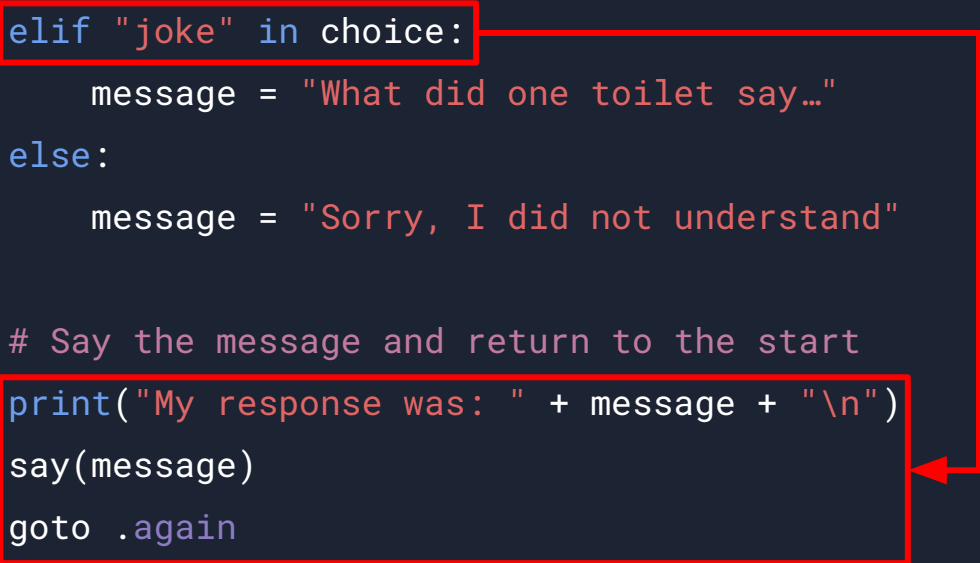
Voice Assistant: Part 2

choice: "Tell me a joke"

- Because this `elif` statement was `True`, we skip the remaining statements and jump to the next section of code
- This next section will:
 - `print()` the message
 - `say()` the message
 - And return to the `label` from before

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```



Voice Assistant: Part 2

- These statements are all bundled together
- Notice the first one says "if", the next two say "elif", and the final one says "else"?
- You can have as many elif statements as you want, but you can only have one else!
- You can even have no elif or else

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"
```

```
# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```

Voice Assistant: Part 2

- Suppose the user said "Please tell me the weather", what would the program do?

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```

Voice Assistant: Part 2

- Suppose the user said "Please tell me the weather", what would the program do?
- It would say the temperature that the **micro:bit** read off the thermometer!

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```

Voice Assistant: Part 2

- What if the user said "Please tell me the age of my uncle", what would the program do?

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```

Voice Assistant: Part 2

- What if the user said "Please tell me the age of my uncle", what would the program do?
- It would say "Sorry, I did not understand" because there is no `if` or `elif` statement to check for any of those words

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```

Voice Assistant: Part 2

- When all the `if` or `elif` statements are `False`, the `else` statement will run
- This makes it useful as an error checker to make sure the user says something that the program can handle!

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```


Voice Assistant: Part 2

What would be said by the program if the user said these things:

- "hello, tell me a joke"
- "weather, now"
- "what way am I facing?"
- "bearing"
- "how old are you?"

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```

Voice Assistant: Part 2

What would be said by the program if the user said these things:

- "hello, tell me a joke"
→ joke
- "weather, now"
→ weather
- "what way am I facing?"
→ sorry
- "bearing"
→ bearing
- "how old are you?"
→ sorry

```
if "weather" in choice:
    message = "The temperature is..."
elif "bearing" in choice:
    message = "Your bearing is..."
elif "joke" in choice:
    message = "What did one toilet say..."
else:
    message = "Sorry, I did not understand"

# Say the message and return to the start
print("My response was: " + message + "\n")
say(message)
goto .again
```



Exercise 1: Fixing Reggie



[Activity 04.01](#)



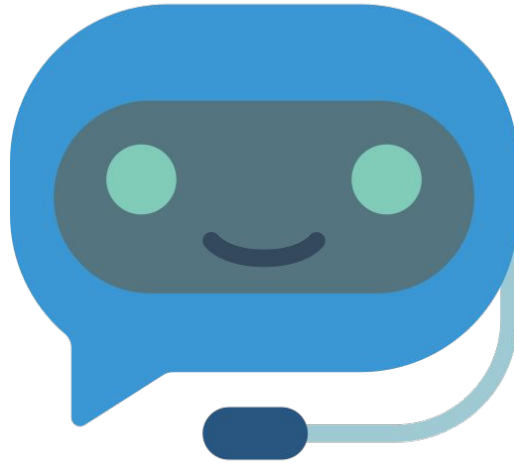
Exercise 2: Upgrading Reggie



[Activity 04.02](#)



Exercise 3: Your Own Assistant



[Activity 04.03](#)



Summary

- `if`, `elif` and `else` statements allow our programs to handle three or more options
- `else` ensures that if the option selected by the user is not available, the program still works
- `in` tells Python to check for something inside something else (e.g. is `weather` in `"I want the weather."`)
- **Input, Process, Output (IPO)** diagrams are useful to highlight the information that goes in, the processing that happens to it, and what comes out



License Information

These [CS in Schools](#) lessons plans, worksheets, and other materials were created by the CS in Schools team. They are licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).

Images all taken from [Flaticon](#).

