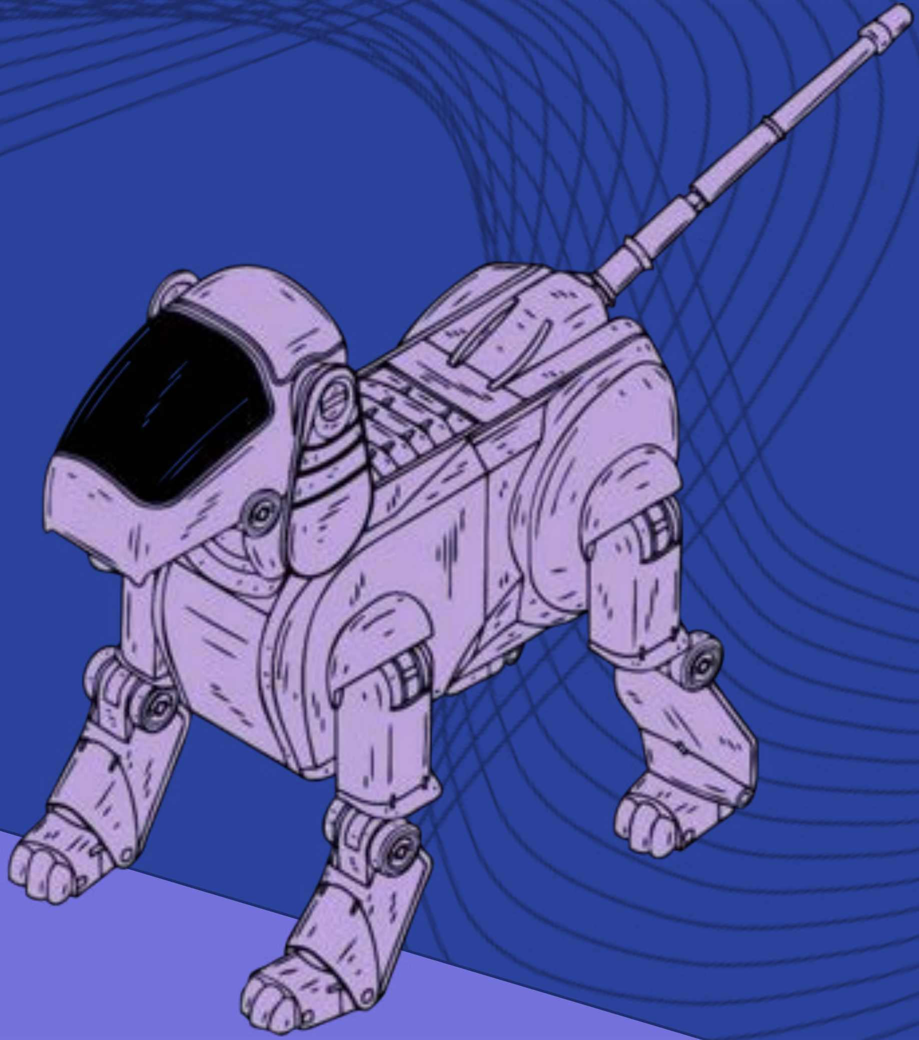
ChatBot

Level 3 – Python

At Home



cair
4 YOUTH

Introduction

What is Artificial Intelligence?

Less than a decade after breaking the Nazi encryption machine Enigma, Alan Turing, often renowned as the father of modern computer science, changed history a second time with a simple question:

‘Can machines think?’

Turing’s paper on ‘Computing Machinery and Intelligence’ established the fundamental goal and vision of AI

Task

Python-Bot

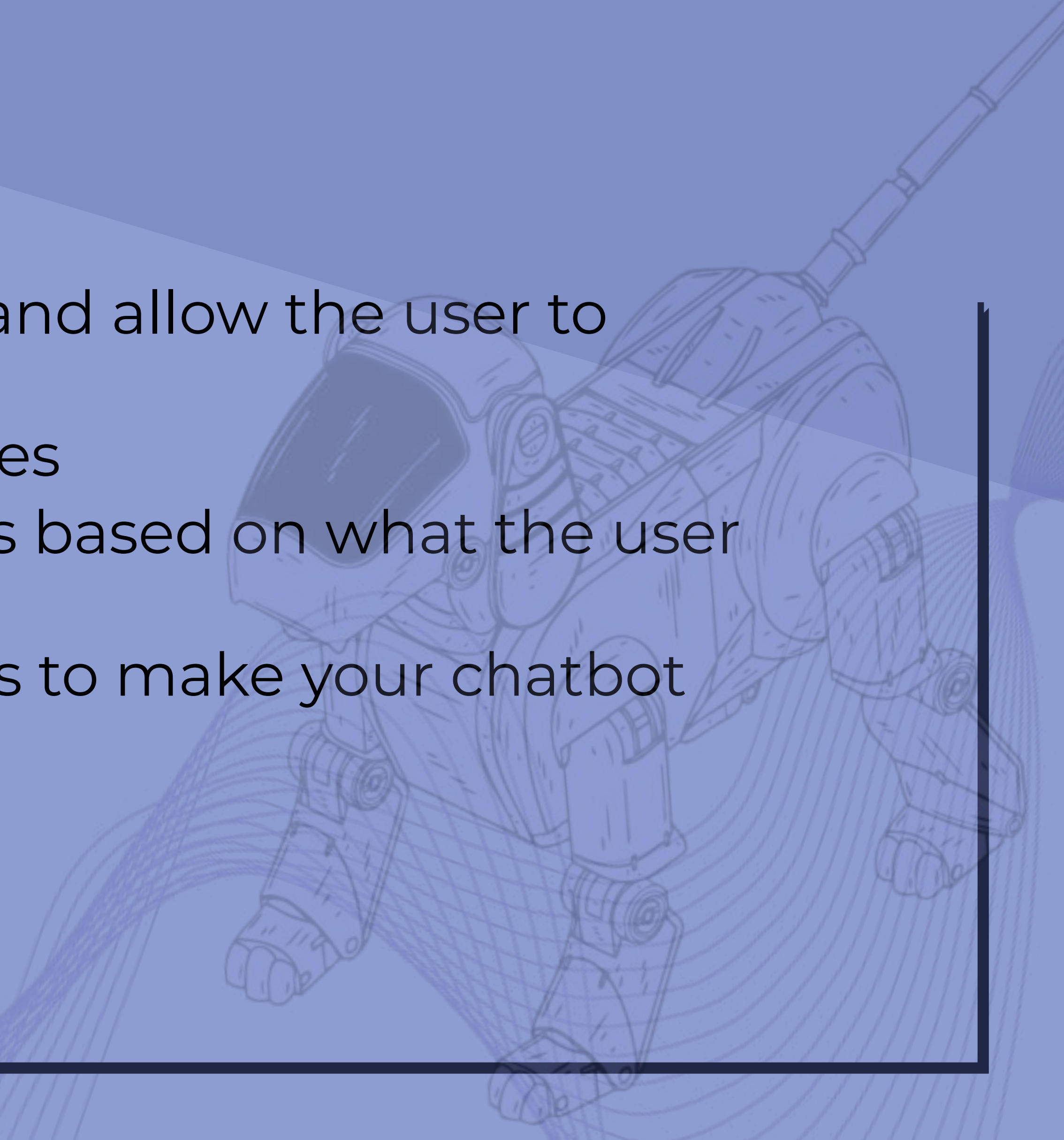
It is the goal of Artificial Intelligence to replicate or simulate human intelligence in machines.

- You have been tasked with designing an intelligent program that can have a conversation with the user
- These types of programs are often called **ChatBots**

Process

Python-Bot

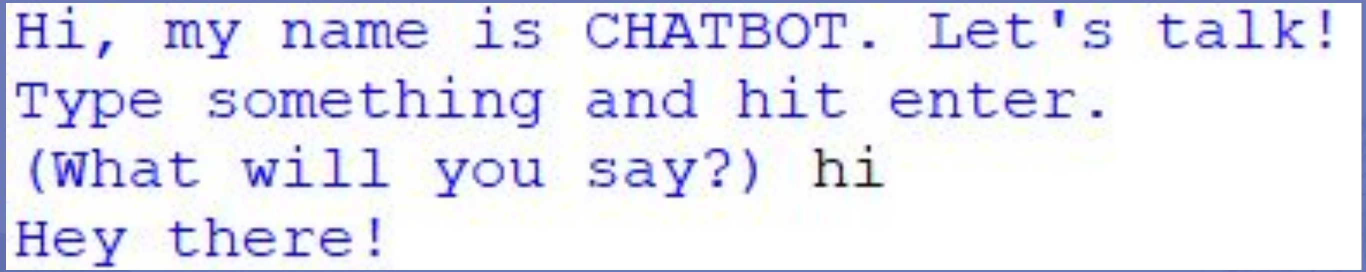
- Have your chatbot ask questions and allow the user to answer
- Store the user’s answers in variables
- Have your chatbot make decisions based on what the user enters into the program
- Expand and experiment with ways to make your chatbot more intelligent



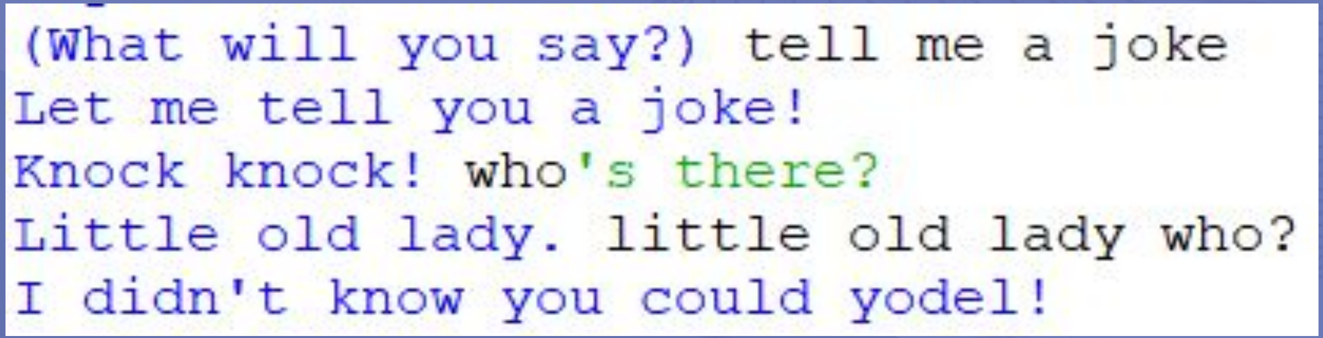
cair
4 YOUTH

Example

What our ChatBot will look like...



```
Hi, my name is CHATBOT. Let's talk!
Type something and hit enter.
(What will you say?) hi
Hey there!
```



```
(What will you say?) tell me a joke
Let me tell you a joke!
Knock knock! who's there?
Little old lady. little old lady who?
I didn't know you could yodel!
```

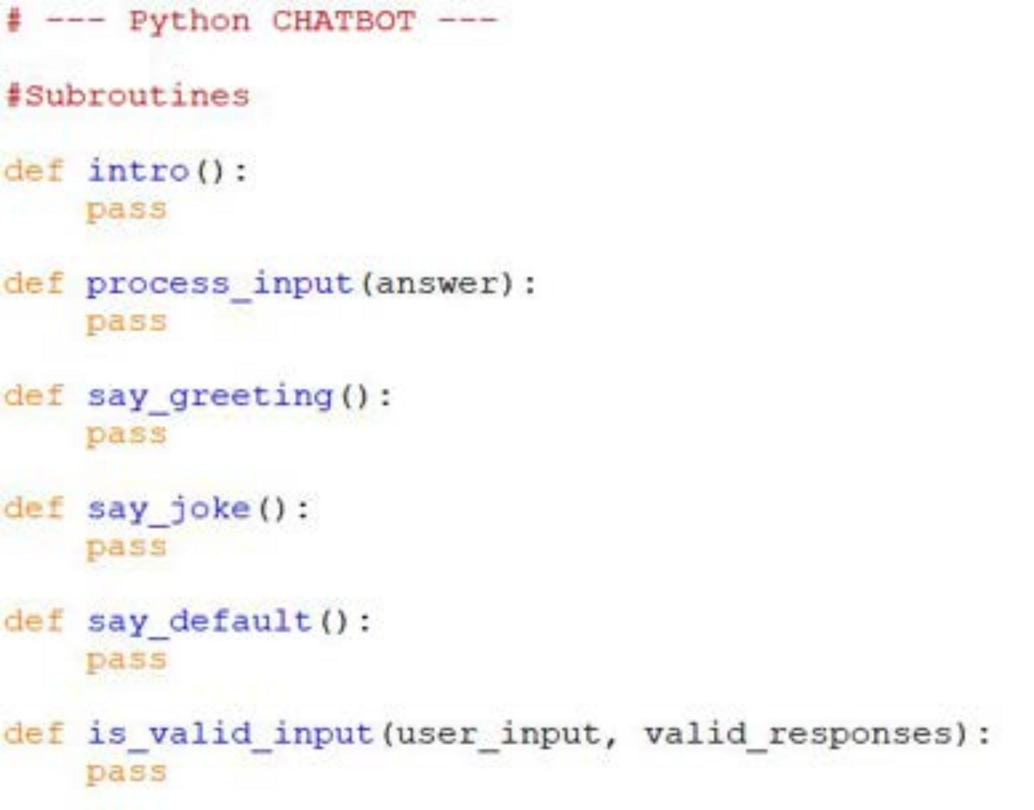
cair
4 YOUTH

Step 1

Setting up our program...

For this program, we are going to use subroutines – these are blocks of code that we can reuse in our program.

To start off, we're going to create the placeholders for our subroutines so that we have some structure to our program.



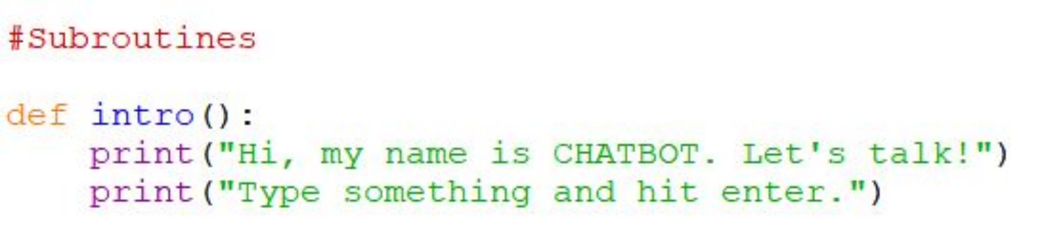
```
# --- Python CHATBOT ---

#Subroutines

def intro():
    pass

def process_input(answer):
    pass

def say_greeting():
    pass

def say_joke():
    pass

def say_default():
    pass

def is_valid_input(user_input, valid_responses):
    pass
```

cair
4 YOUTH

Step 2

Introducing our ChatBot

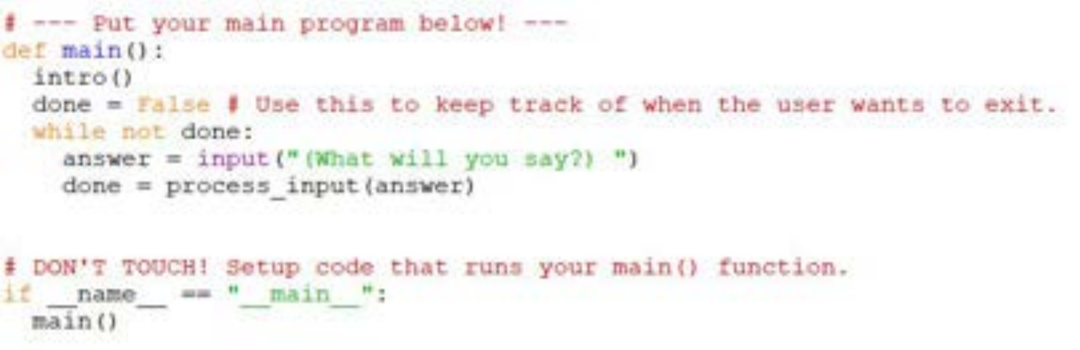
Let’s allow our ChatBot to introduce itself! Inside of the intro() subroutine, write the following code.



```
#Subroutines

def intro():
    print("Hi, my name is CHATBOT. Let's talk!")
    print("Type something and hit enter.")
```

However, at this point, when you run your code, nothing happens! This is because we haven’t ‘called’ our subroutine.

At the bottom of your program, add the following code. This will call our intro() subroutine and allow our user to ask ChatBot questions!

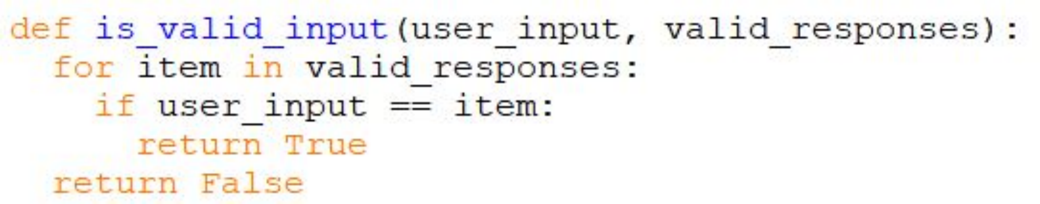


```
# --- Put your main program below! ---
def main():
  intro()
  done = False # Use this to keep track of when the user wants to exit.
  while not done:
    answer = input("(What will you say?) ")
    done = process_input(answer)


# DON'T TOUCH! Setup code that runs your main() function.
if __name__ == "__main__":
  main()
```

Step 3

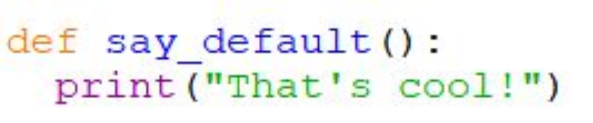
Default and validation

First, let’s make a subroutine that checks whether the user’s input is valid. This will come in handy later.



```
def is_valid_input(user_input, valid_responses):
  for item in valid_responses:
    if user_input == item:
      return True
  return False
```

Also, in case our ChatBot doesn’t know the answer to something, let's add some code that will run as his default response.

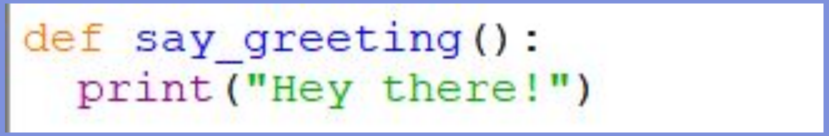


```
def say_default():
  print("That's cool!")
```

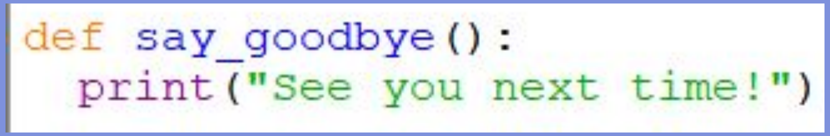
Step 4

Responses

We need to program some responses for our ChatBot. This is where you can get creative! Here are some examples of what your subroutines might look like.



```
def say_greeting():
  print("Hey there!")
```



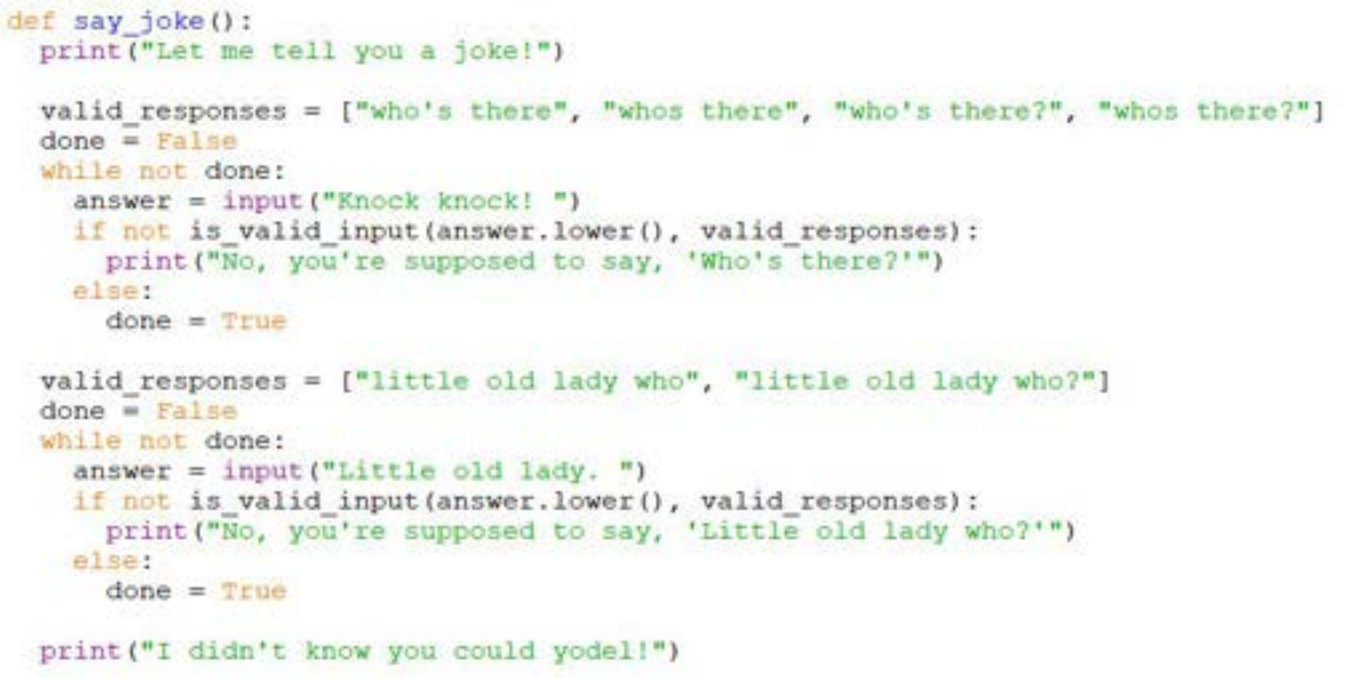
```
def say_goodbye():
  print("See you next time!")
```

cair
4 YOUTH

Step 5

Responses

Your responses can vary in complexity. Here is the say_joke() subroutine. It has a list of what the user might enter and responds to the user based on what they input.



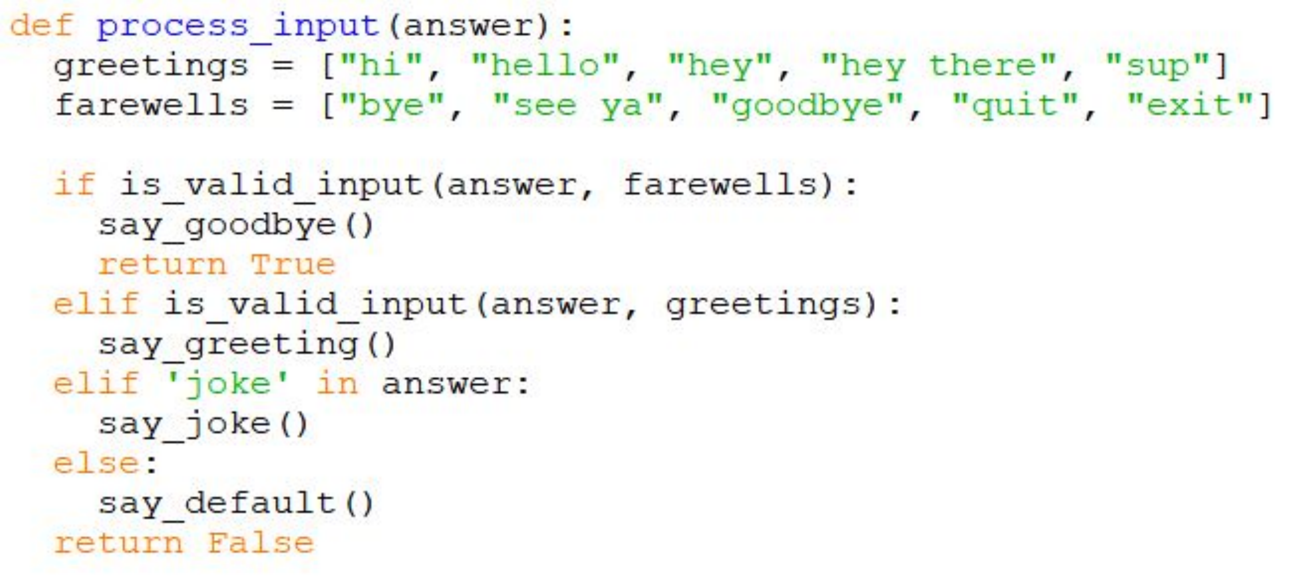
```
def say_joke():
  print("Let me tell you a joke!")

  valid_responses = ["who's there", "whos there", "who's there?", "whos there?"]
  done = False
  while not done:
    answer = input("Knock knock! ")
    if not is_valid_input(answer.lower(), valid_responses):
      print("No, you're supposed to say, 'Who's there?'")
    else:
      done = True

  valid_responses = ["little old lady who", "little old lady who?"]
  done = False
  while not done:
    answer = input("Little old lady. ")
    if not is_valid_input(answer.lower(), valid_responses):
      print("No, you're supposed to say, 'Little old lady who?'")
    else:
      done = True

  print("I didn't know you could yodel!")
```

Can you customise this subroutine to be one of your own jokes?

Step 6

Processing Responses

We need a subroutine that will manage all of our different responses. It uses IF statements to examine what the user has typed in, and calls one of our response subroutines accordingly.



```
def process_input(answer):
  greetings = ["hi", "hello", "hey", "hey there", "sup"]
  farewells = ["bye", "see ya", "goodbye", "quit", "exit"]

  if is_valid_input(answer, farewells):
    say_goodbye()
    return True
  elif is_valid_input(answer, greetings):
    say_greeting()
  elif 'joke' in answer:
    say_joke()
  else:
    say_default()
  return False
```

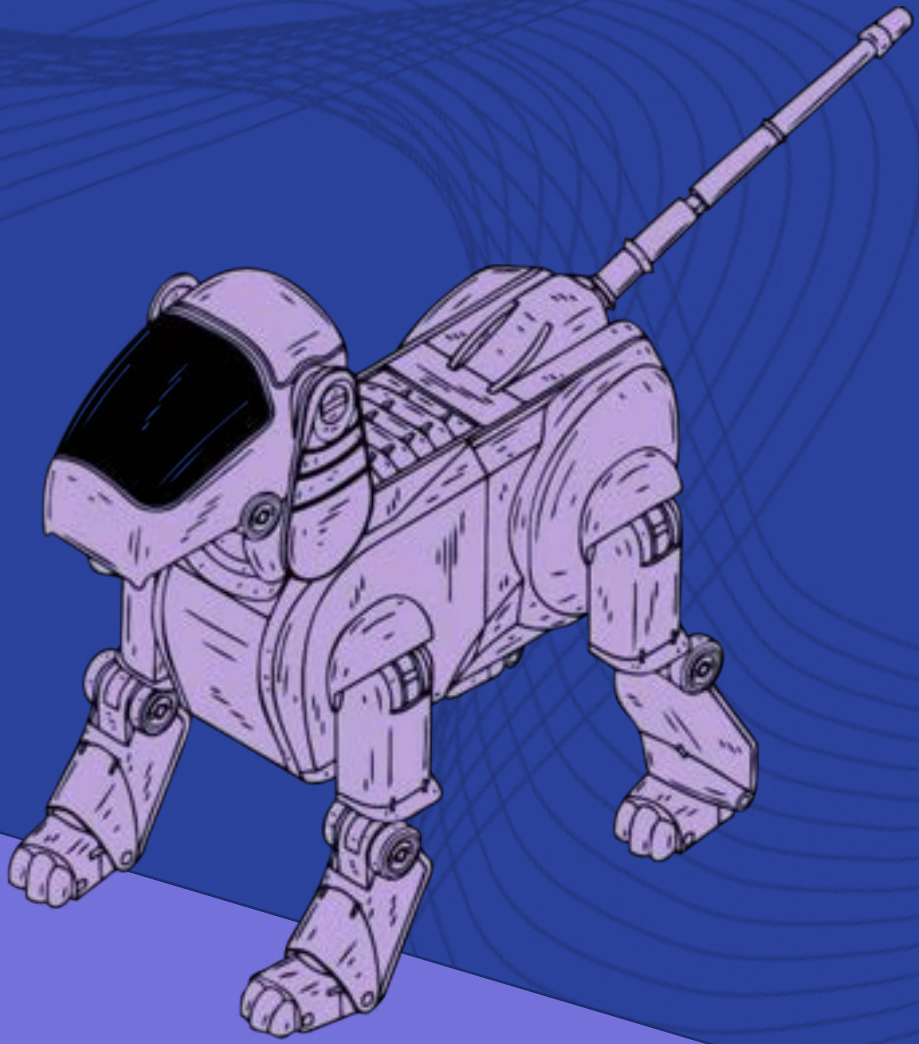
Conclusion

Learning outcomes

- ✔ Learn how to create subroutines that can be reused throughout a programme.
- ✔ Call subroutines.
- ✔ Understand how basic AI uses IF statements to respond to potential inputs.
- ✔ Customise your chatbot and make it your own!

Congratulations!

You have completed the project



cair
4 YOUTH