Introduction to Python Programming

1 - Output, variables and input.

repl.it

Resources created by Andy Colley (@MrAColley)

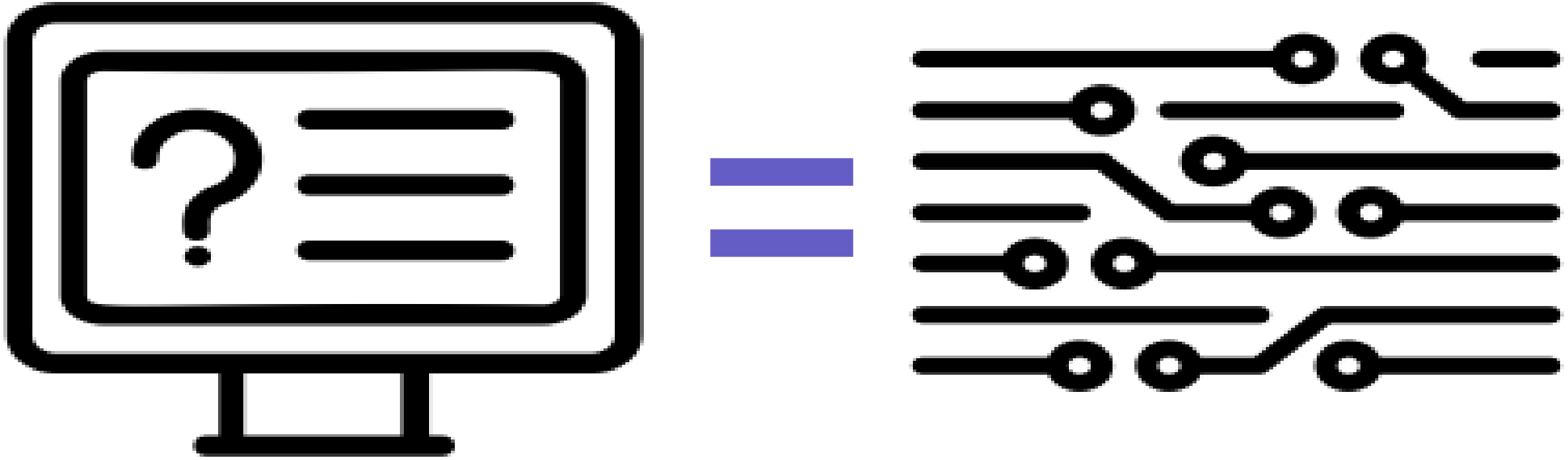
Learning Goals/Objectives

Be able to read, comprehend, trace, adapt and create Python code that:

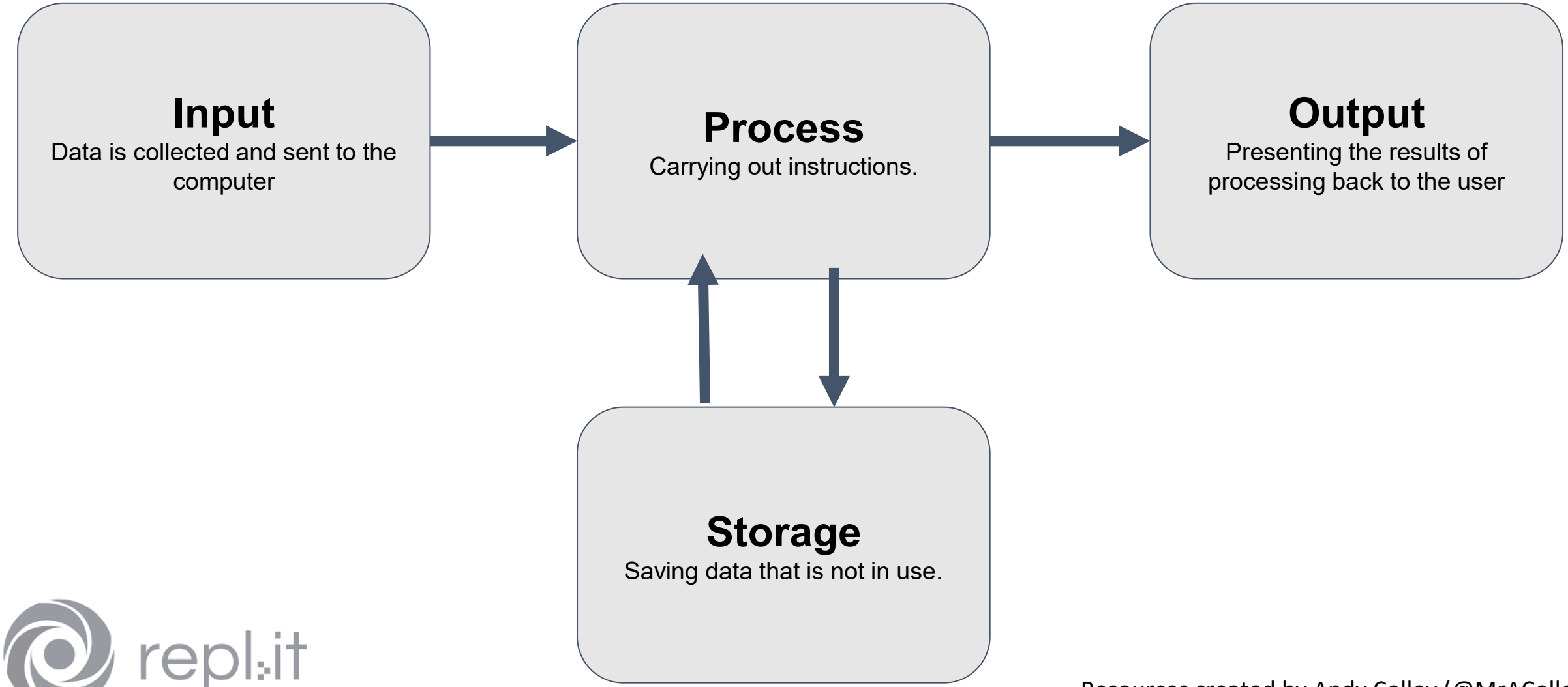
- Outputs text to the console
- Assigns data to variables
- Gets input
- Assigns input to variables.
- Refers to variables in other statements such as ‘print’

repl.it

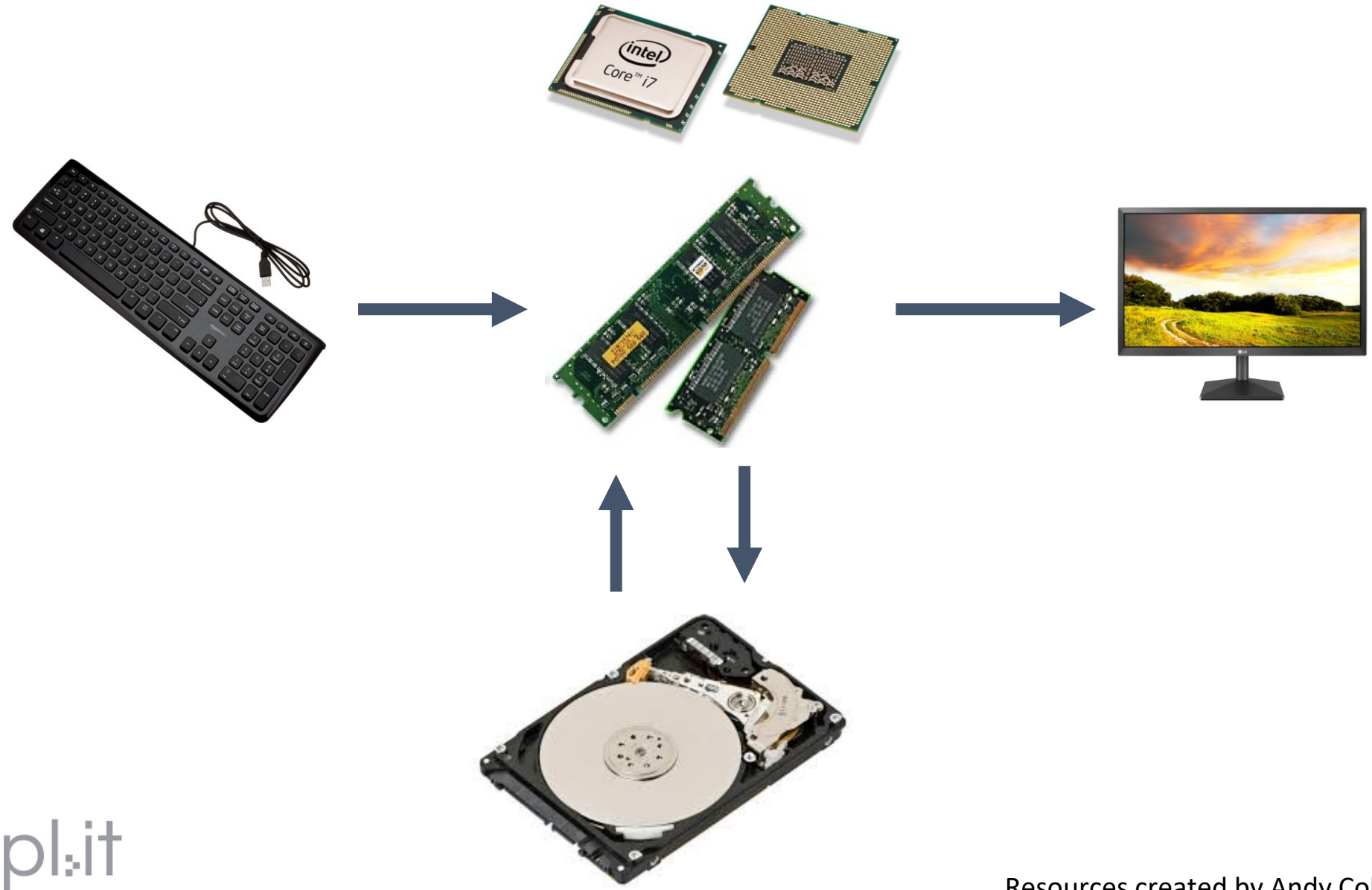
What is a Computer?



repl.it


Input
Data is collected and sent to the computer
Process
Carrying out instructions.
Output
Presenting the results of processing back to the user
Storage
Saving data that is not in use.

repl.it



repl.it

Resources created by Andy Colley (@MrAColley)

Input
Data is collected and sent to the computer

Process
Carrying out instructions.

Output
Presenting the results of processing back to the user

Storage
Saving data that is not in use.

repl.it

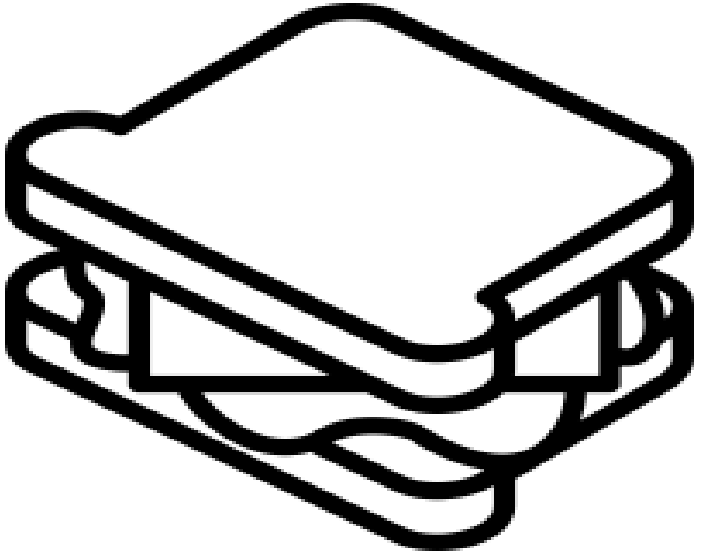
Resources created by Andy Colley (@MrAColley)

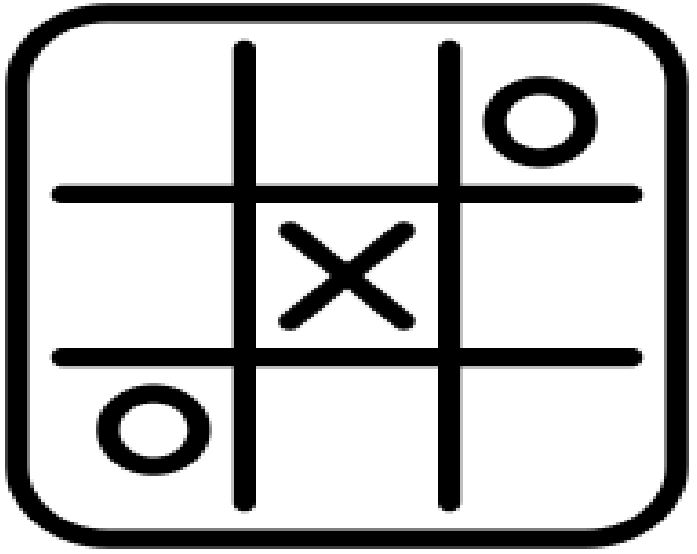
Algorithms

"An algorithm is a sequence of instructions for a task"

Algorithm ≠ Computer Program

Algorithms In The Real World





repl.it

A Computer Program

Algorithms that are converted into code become computer programs.

Algorithm

OUTPUT "Hello, World"

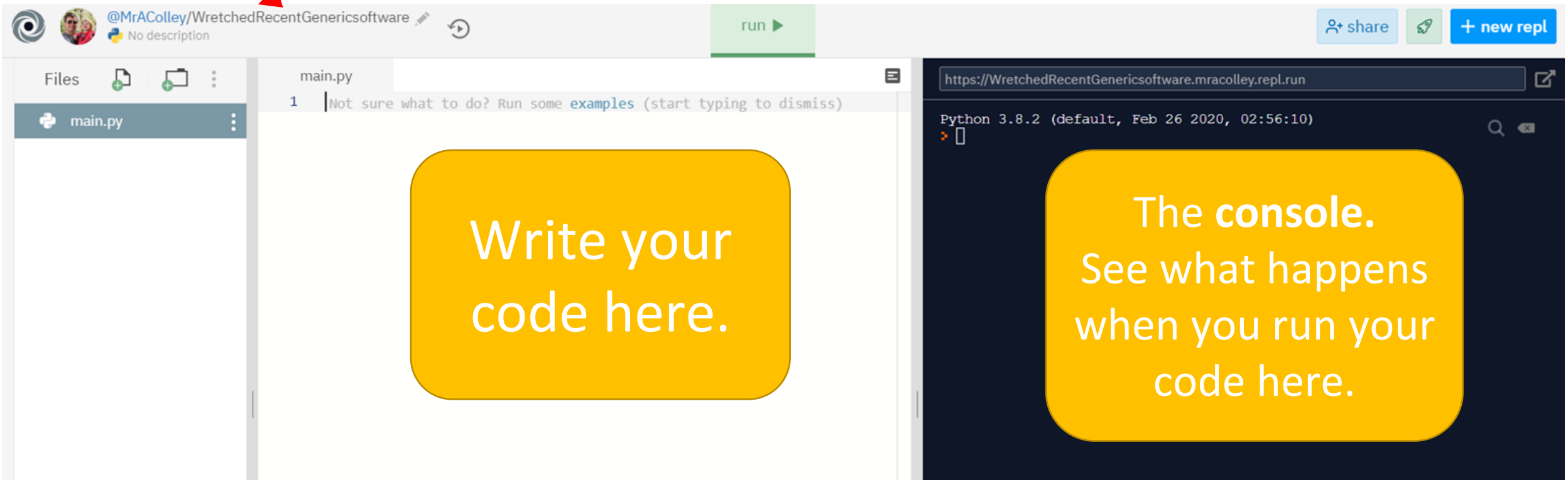
Scratch

Python

```
print("Hello, World!")
```

repl.it

Choose ‘Fork’ here



Write your code here.

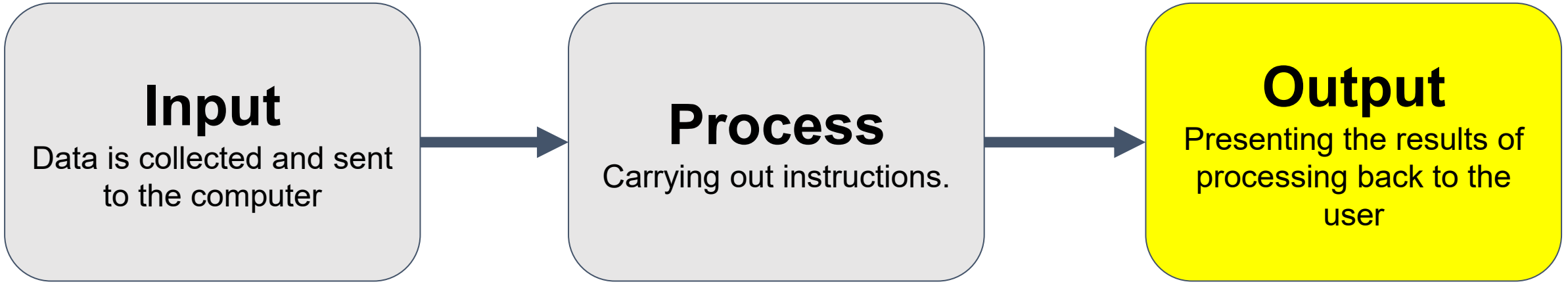
The **console.** See what happens when you run your code here.

repl.it

Resources created by Andy Colley (@MrAColley)

OUTPUT

Programming – Output




```
print("your text goes here")
```

repl.it

Programming – Output - Investigate

```
print("your text goes here")
```

Syntax – The Way The Code Is Written

print – no capital p.
Brackets – after print and at the end.
Text **must** go in speech marks.

Incorrect syntax = broken code that won't work = error!

repl.it

Programming – Output – Predict & Run

```
5   #Task 1 - Add a comment on line 7 to predict
    what the code on line 8 will do.
6
7
8   print("Hello World!")
```

Add comments to the code to predict exactly what it will output.

Run the code to see if you were correct.

repl.it

Programming – Output – Investigate

```
print("Hello World!")

# Task Investigate

# What would the output of the code print("I love Computing") be?

# What would happen if the code print("I love Comping") was run?

# What would happen if the code print("I love Computing" was run?
```

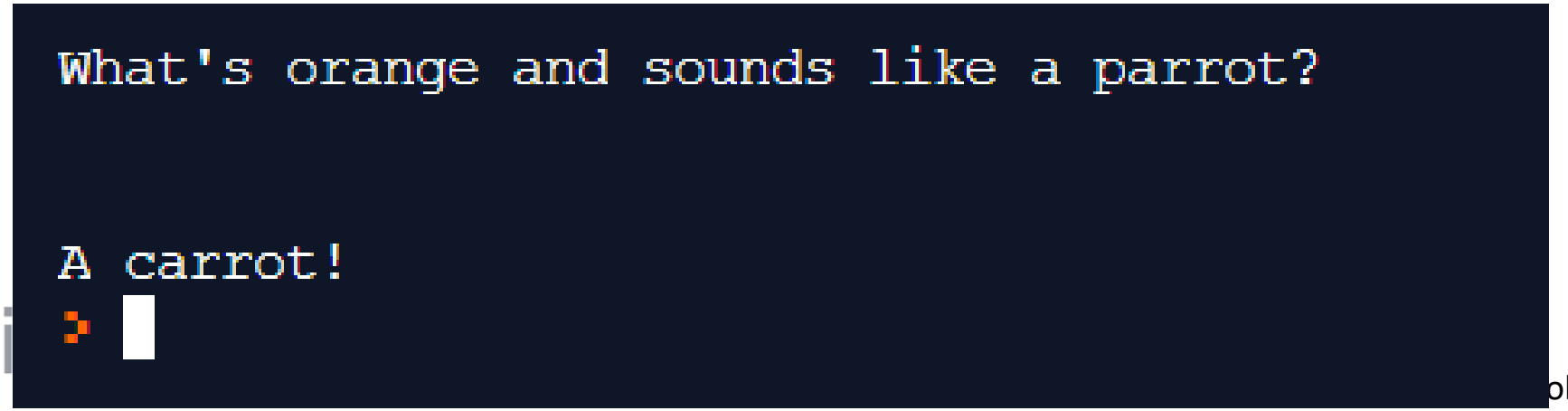
Add comments to the code to answer the questions.

https://repl.it/@MrAColley/11-Output

Programming – Output – Modify & Make

Modify – reuse the print statement to add your own single line message to the program.

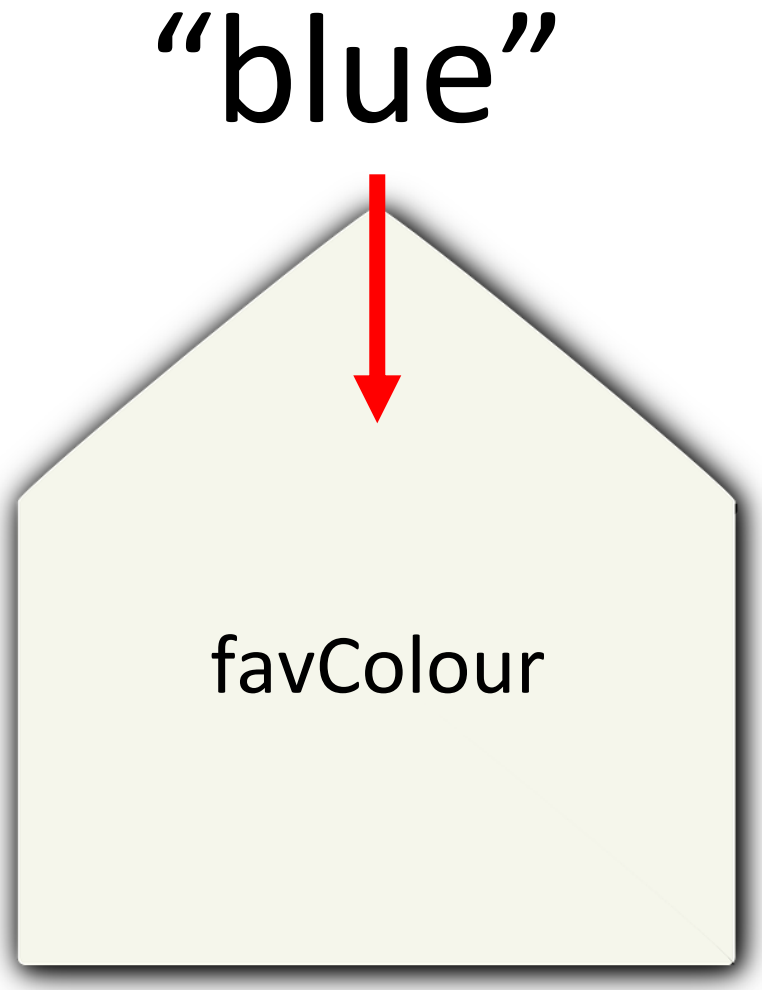
Make – use the print statement to output a joke that appears on multiple lines (keep it clean!). **Extra challenge** – can you figure out how to add blank lines like in my joke?



VARIABLES

Programming - Variables

- Placeholders for information.
- Can store one piece of information.
- Can be called whatever you like (keep it relevant)
- No spaces in the name.
- Name starts with a lower case letter.
- Can’t have 2 variables with the same name.
- Use the variable name whenever you want to
- output the information.




```
print(favColour)
print(“My favourite colour is “ + favColour)
```

repl.it

Programming – Variables – Predict & Run

```
#Task 1 - Add comments to the code below to explain what it does

name = "Axl"

print(name)
```

Add comments to the code to predict exactly what it will output.

Run the code to see if you were correct.

repl.it

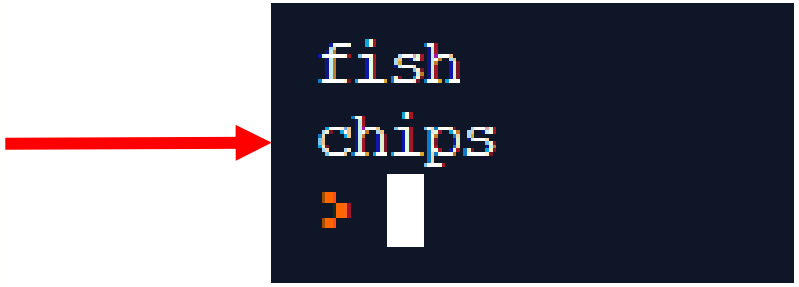
Programming – Variables – Modify

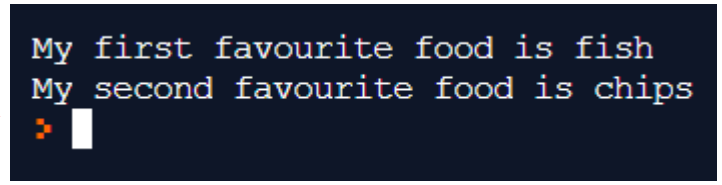
```
#Task 2 - Concatenation

#You can combine variables with strings in an output

name1 = "Axl"
name2 = "Slash"
#Add 2 more variables to store 'Duff' and 'Izzy'


#This line uses concatenation to join the variables together with the
string 'and' to make a sentence.

#Complete the line to output all of the variables
print(name1 + " and " + name2 + " and ")
```

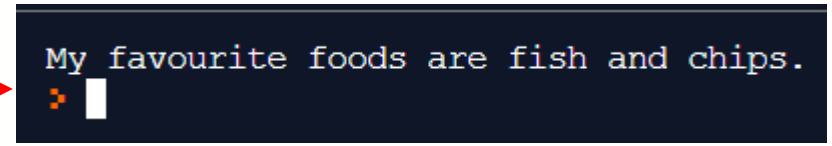
Resources created by Andy Colley (@MrAColley)

Programming – Variables – Make

```
#Task 3

########################## SIMPLE ####################
# Assign your name and your favourite food to 2 separate variables.
# Output the contents of the variables on 2 separate lines

############## MEDIUM ##############################

# Output two sentences (not just the contents of the variables). The
first with your name, the second with your favourite food.


############## COMPLEX ##############################
# Output both pieces of information as part of the same sentence.
# Make sure that you have spaces and punctuation in the correct
places.
```





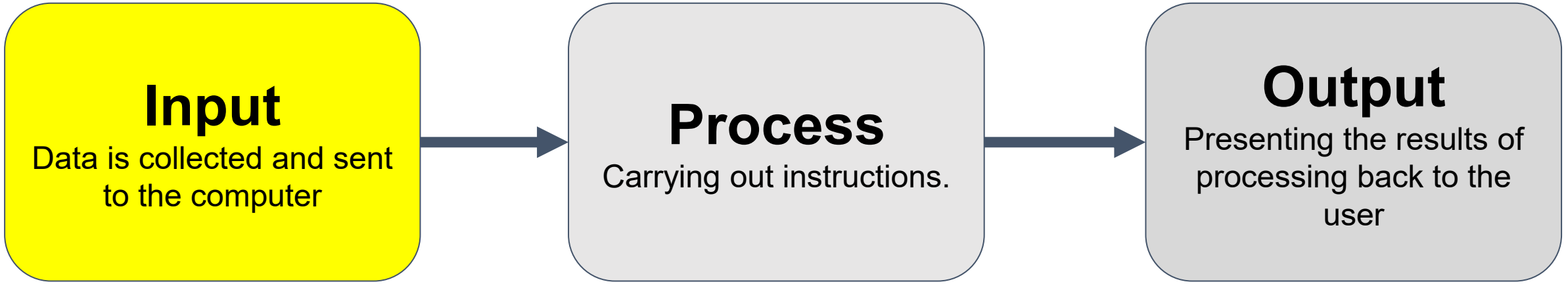


https://repl.it/@MrAColley/13-Variable-Assignment-Challenge

repl.it

INPUT

Programming – Input




```
print("What's your name?")
name = input()
print("Hello " + name)
```

repl.it

Programming – Input – Predict & Run

```
#Task 1 - Add comments to the code below to
predict what will happen when it is run.

#Task 2 - add comments to the code to explain
what the variables are called and where they
are used. Make sure to show where the
variable.

print("Hi! What's your name?")

name = "Dave"

print("Hi " + name + "! How are you today?")
```

Add comments to the code to predict exactly what it will output.

Run the code to see if you were correct.

https://repl.it/@MrAColley/14-Input-With-Variables

repl.it

Programming – Input - Efficiency

```
print("What's your name?")
name = input()
```

```
name = input("What's your name?")
```

Programming – Input – Modify

```
# Task 3 - Adapt the code so that it assigns input into the 'name'
variable. CHALLENGE - put a prompt in the input command to ask the
user for their name.

# Task 4 - Combine lines 24 and 25  so that it the input has a prompt
in it.

#Task 5 - Adapt the output on line 26 so that it includes both the
name and the answer variables.

name = "Billy"
print("We want to know if you like progamming!")
print()
print("Do you like programming " + name + "?")
answer = input()
print("Great! You said " + answer + "!")
print("Let's learn some Python today")
```

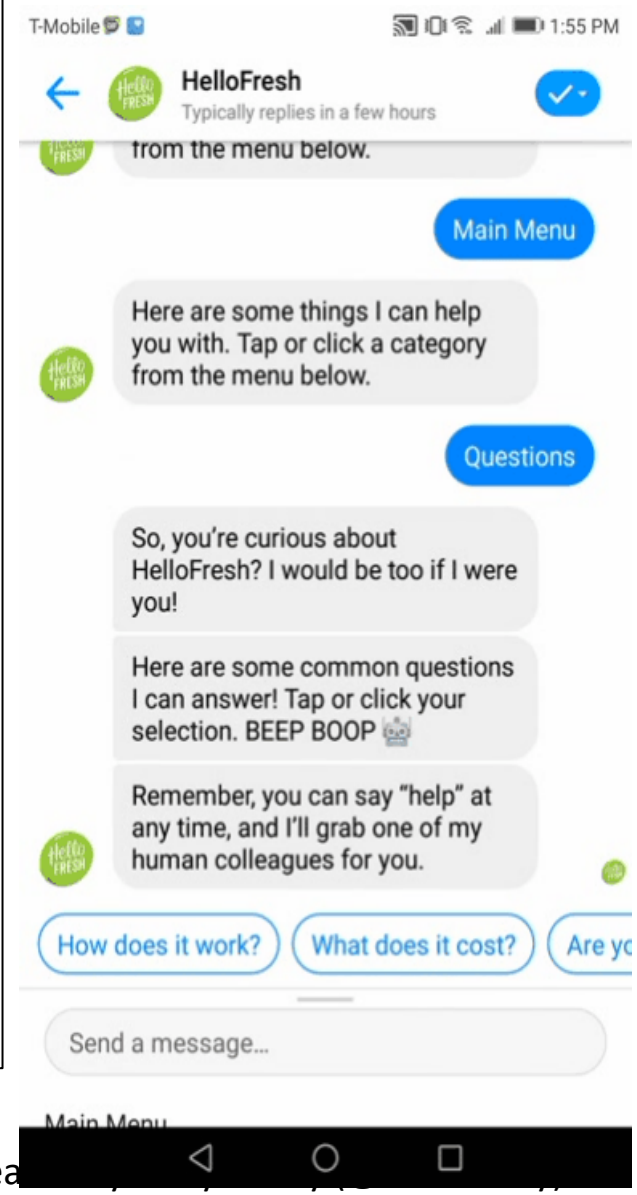
Programming – Variables – Make Homework

Chat-Bot Challenge

Lots of websites use chat bots to interact with their customers. These chat bots are often very sophisticated and use AI to learn and adapt to the user. Our chat bot is going to be a bit simpler.

The chat bot should work like this:

- Ask the user their name and store it in a variable.
- Greet the user by name.
- Ask the user three questions about themselves and store their responses in three **different** suitably named variables.
- Respond to each of the questions one by one, using the user's name in the response.
- Output a summary of all of the user's answers in a single sentence.



repl.it

Resources crea