



Chatbots with Personality

[HAL-9000 as Alexa](#)



Notable Code!

```
# Author: [REDACTED]  
# Date: Jan. 5, 2018  
  
# Displays a motivational quote on the screen  
print("Trust me. None of us have any idea what we're doing")
```

```
# Title: May the coding be with you  
# Author: [REDACTED]  
# Date: January 5th, 2018  
  
# A motivational quote to the screen you must print  
print("“Do. Or do not. There is no try.”- Yoda")
```

```
# Motivational Quote Generator  
# Author: [REDACTED]  
# Date: Jan 5th, 2018  
  
print ("sundays should come with a pause button")
```

```
# Author: [REDACTED]  
# Dates: Jan 5th, 2018 (ignore after the #, will not run it)  
  
# Step 1: Print to the screen a motivational quote  
print("Do not capital P.")
```

For more: <http://inspirobot.me/>



Question 1

Which symbol is used to designate **comments** in Python?

Question 2

What would you type in the comments the top of your Python file? (Hint: There should be at least 3 elements.)



Question 3

How would you display "Beep boop" on the output console?



What are some chatbots

today?

The study of natural language processing processes the text of natural languages like English and Japanese.

- ELIZA
- CleverBot
- MITSUKU
- Woebot





Unit 1 Chatbots

1/APPLICATIONS

In this unit, we'll learn about the computing science field of natural language processing, and its applications such as chat bots.

2/ALGORITHMS

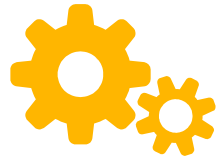
We'll learn about input, output, strings, conditionals, concatenation and other concepts to build a simple chatbot algorithm.

3/PROGRAMMING LANGUAGE

In Python 3, we'll be learning the syntax and keywords to implement our algorithms.

4/TESTING

We will learn how to tell if our program is any good or not!



Let's start!

First challenge: write a chatbot that can say hi and learn your name.



Design an Algorithm

Learn how to design solutions to problems by explaining ideas step-by-step



Write it in Python

Learn how to communicate your solution in a language that a machine understands



Test and Deploy

Make sure it works by testing your application



Change the World

Code is the new electricity. Change the world with what you've built.



Design an algorithm



saved



```
1 # Greetings Chatbot
2 # Author: Angelica Lim
3 # Date: Nov. 29, 2017
4
5 # Say hi, what's your name?
6 # Get the person's name
7 # Respond nice to meet you, <name>
8
```

Name of our
program

Algorithm steps

<http://interactivepython.org/runestone/static/thinkcspy/GeneralIntro/Comments.html>



Translate it to Python 3

```
1 # Greetings Chatbot
2 # Author: Angelica Lim
3 # Date: Nov. 29, 2017
4
5 # Say hi, what's your name?
6 print("Hi, what's your name?")
```

Output

Assigning a
value to a
variable

```
# Get the person's name
user = input()
```

Input

```
11 # Respond nice to meet you, <name>
12 print("Nice to meet you, " + user)
13
```

Concatenation

Using a variable

<http://interactivepython.org/runestone/static/thinkcspy/SimplePythonData/Variables.html>

<http://interactivepython.org/runestone/static/thinkcspy/Strings/OperationsonStrings.html>



Test your algorithm in

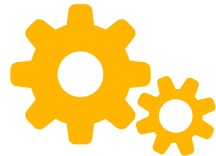
Python

```
saved [share icon] [run icon] [cloud icon] [play icon] [cube icon]
1 # Greetings Chatbot
2 # Author: Angelica Lim
3 # Date: Nov. 29, 2017
4
5 # Say hi, what's your name?
6 print("Hi, what's your name?")
7
8 # Get the person's name
9 user = input()
10
11 # Respond nice to meet you, <name>
12 print("Nice to meet you, " + user)
```

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
Hi, what's your name?
```

Try and
break it!

You can also ask others to
test it by sharing or
"deploying" your code.



About **variables**

There are some constraints to how you can name your variables, e.g.

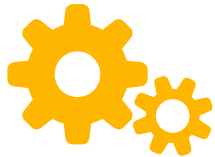


- Can contain letters, numbers, and underscores
- Should start with a letter



- They can't contain spaces or symbols
- They can't be one of the reserved keywords (see below for list)

<http://interactivepython.org/runestone/static/thinkcspy/SimplePythonData/VariableNamesandKeywords.html>



Add more **features**

After the greetings, make the chatbot **ask what your favourite book is**. The chatbot should then make a **comment** about your response.



Update your **algorithm**

```
1  # Greetings Chatbot
2  # Author: Angelica Lim
3  # Date: Nov. 29, 2017
4
5  # Say hi, what's your name?
6  print("Hi, what's your name?")
7
8  # Get the person's name
9  user = input()
10
11 # Respond nice to meet you, <name>
12 print("Nice to meet you, " + user)
13
14 # Ask what your favourite book is
15 # Let the user respond
16 # Make a comment about it
17
```



Translate to Python 3

```
1 # Greetings Chatbot
2 # Author: Angelica Lim
3 # Date: Nov. 29, 2017
4
5 # Say hi, what's your name?
6 print("Hi, what's your name?")
7
8 # Get the person's name
9 user = input()
10
11 # Respond nice to meet you, <name>
12 print("Nice to meet you, " + user)
13
14 # Ask what your favourite book is
15 print("What is your favourite book?")
16
17 # Let the user respond
18 input()
19
20 # Make a comment about it
21 print("Oh, nice!")
```

Here we stored the user's name in a **variable** (called user) because we wanted to use it later.

But we don't need to store the answer in a variable if we're not going to use it.



Test **how it works**

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
Hi, what's your name?
  Jeanne
Nice to meet you, Jeanne
What is your favourite book?
  Harry Potter
Oh, nice!
> █
```

What happens if you run it twice, with different book names?



Let's **review** some concepts

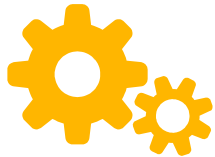
How do you print in Python?

What is the command or **function** to get input from the user?

How do you concatenate two words in Python?

How do you store an input from the user?

What symbol do we use to assign a value to a variable?



Add more features

First challenge was to write a chatbot that can learn your name.

Secondly, to ask what your favourite book is and make a comment about it.

>> The comment **should not be too obviously repetitive.**



Update your algorithm

```
1 # Greetings Chatbot
2 # Author: Angelica Lim
3 # Date: Nov. 29, 2017
4
5 # Say hi, what's your name?
6 print("Hi, what's your name?")
7
8 # Get the person's name
9 user = input()
10
11 # Respond nice to meet you, <name>
12 print("Nice to meet you, " + user)
13
14 # Ask what your favourite book is
15 print("What is your favourite book?")
16
17 # Let the user respond
18 input()
19
20 # Make a comment about it that is not too
  repetitive
21
22 # Have a list of possible comments
23 # Choose one randomly from the list
24 # Say that random comment
```



Translate to Python 3

```
13
14 # Ask what your favourite book is
15 print("What is your favourite book?")
16
17 # Let the user respond
18 input()
19
20 # Make a comment about it that is not too repetitive
21 # Make a list of possible comments
22 comments = ["Oh, nice!", "That's a good one.", "Hmm, strange taste.", "blah
    blah blah", "Whoa there.", "Hahahhaa!"]
23
24 # Choose one randomly from the list
25 import random
26 random_comment = random.choice(comments)
27
28 # Say that random comment
↓ print(random_comment)|
```

This is a Python **list**, indicated with brackets. All the options are separated by commas. Everything should be on one line (here it's wrapped due to space, but it's all on line 22.)

This is a **module** that we can **import** to have more functionality. We can access the module's functions with a **dot**.

Test **how it works**

Remember the feature we wanted to make:

The comment **should not be too obviously repetitive**.

```
4
5 # Say hi, what's your name?
6 #print("Hi, what's your name?")
7
8 # Get the person's name
9 #user = input()
10
11 # Respond nice to meet you, <name>
12 #print("Nice to meet you, " + user)
13
14 # Ask what your favourite book is
15 print("What is your favourite book?")
16
17 # Let the user respond
18 input()
19
20 # Make a comment about it that is not too repetitive
21 # Make a list of possible comments
22 comments = ["Oh, nice!", "That's a good one.", "Hmm, strange taste.", "blah
    blah blah", "Whoa there.", "Hahahhaa!"]
23
24 # Choose one randomly from the list
25 import random
26 random_comment = random.choice(comments)
27
28 # Say that random comment
29 print(random_comment)
```

Tip! **Comment** out the other bits of your program to **focus on testing** the part you're working on.

What happens when you run it multiple times? Is it still repetitive?

```
input
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
What is your favourite book?
Green Eggs and Ham
Hahahhaa!
```



Let's **review** some concepts

How do we make a list in Python?

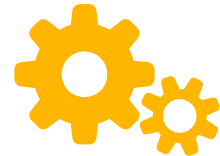
What package do we need to import to randomly choose something from a list?

How can we test smaller pieces of our Python code?

What is concatenation?

What does a dot after a module name do?

What is the only kind of symbol we can have in a variable name?



For next class

Be **creative**!



Make your chatbot ask at least 3 questions. Include some randomness in replies.

Remember to design your algorithm in English first, translate to Python, then test it works!