THE TURING TEST
CHATBOT CHALLENGE

"The Turing test is a test of a machine's ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human."

This document is designed to walk you through how to write your own code to create a ChatBot that aims to pass the Turing Test.

Stage 1: Research:

Here are some ChatBots that have tried (and failed) the Turing Test. Try them out and rate them in order of how human they seem:

1. http://www.jabberwacky.com/
2. http://www.cleverbot.com/
3. http://www.elbot.com/
4. http://www.mitsuku.com/
5. http://en.akinator.mobi/personnages/jeu (this one will try to guess what Celeb you're thinking of)
6. http://www.boibot.com/en/
7. http://www.eviebot.com/en/

 I take no responsibility for any insults you may receive from these chatbots!

Stage 2: Learning to code:

We're going to learn some simple Python code using create.withcode.uk which lets you write and run code in your web browser.

Some of your might have done loads of python, some of you might have done none. The best way to learn is by doing, making mistakes and correcting them.

💡 If you're stuck, search for code examples on www.stackoverflow.com, ask each other or ask your teacher.

Task	Description
1	Load my code from: https://create.withcode.uk/python/T
2	Change line 10 so that it says your name rather than mine `10  By P. Dring` 💡 You can also use http://patorjk.com/software/taag/#p=display&f=Crawford2&t=ChatBot to generate your own welcome message

3	Make your program ask how old you are:
	
	 ```25 # ask about their age```
	```26 age = int(input("How old are you?"))```
	```27▾ if age > 50:```
	```28 print("Oooh, that's old!")```
	```29▾ elif age > 100:```
	```30 print("I don't believe you!")```
	
	 💡 int(...) converts text into an integer (whole number)
	 💡 Notice that line 28 and 30 are indented (tabbed to the right). This means they will only run **if** age is greater than 50 or **if** age is greater than 100
	 💡 There is a **logical** error here: the code does exactly what we've told it to but doesn't do what we want it to. Can you spot the problem?

4	Extend your program to: ● Say "Snap! I'm … years old too!" if the user says they're the same age as you are ● Ask which primary school they go to if the user says they're under 11 ● Say something like: "That makes you 0.9 times as old as I am"

5	Make your program ask the user what their favourite food is:
| | ```python
food = input("What's your favourite food?")
``` |
| | Add this line to the very top of your code to let python choose random numbers and random items from a list: |
| | ```python
1  import random
``` |
| | Then, at the bottom of your code, make your program react randomly to whatever the user says: |
| | ```python
35  food = input("What's your favourite food?")
36  reactions = ["hate", "love", "like", "quite like"]
37  my_reaction = random.choice(reactions)
38  print("Really? I " + my_reaction + " " + food)
``` |
6	Extend your code to have 3 more random reactions
	💡 **reactions** is a **list** of **strings**. Each string is surrounded by quotation marks and separated by a comma. A list starts and ends with square brackets.
7	Make your program ask the user what sort of music they like
| | ```python
music = input("What sort of music do you like?")
``` |
| | Make your code respond with a different question for each genre of music: |
| | ```python
music = input("What sort of music do you like?")
favourites = {"rock": "Guns N' Roses",
              "pop": "Ellie Goulding",
              "RnB": "Kanye West"}

reactions = ["I'm not fussed about", "I absolutely love", "I can't stand"]
my_reaction = random.choice(reactions)
print(my_reaction + " " + music)
what_do_you_think = input("What do you think of " + favourites[music] + "?")
``` |
| 8 | Extend your code to have an option for **dance** and **jazz** |
| | 💡 **favourites** is a dictionary. It has **keys** (like "rock" and "pop") and **values** that you can look up based on the matching keys (like "Guns N' Roses" and "Ellie Goulding")
You'll need to add a new key and value for dance and jazz: |
| | ```python
favourites = {"rock": "Guns N' Roses",
              "pop": "Ellie Goulding",
              "RnB": "Kanye West",
              "dance": "Add a dance artist here"}
``` |

| 9 | If the user chooses a music genre that's not in your dictionary you get a runtime error:<br><br>What sort of music do you like? Funk<br>I absolutely love Funk<br>KeyError: Funk on line 49<br><br>You can avoid that by using error handling - see lines 50-53 below:<br><br>```python
41  music = input("What sort of music do you like?")
42  favourites = {"rock": "Guns N' Roses",
43                "pop": "Ellie Goulding",
44                "RnB": "Kanye West",
45                "dance": "Add a dance artist here"}
46
47  reactions = ["I'm not fussed about", "I absolutely love", "I can't stand"]
48  my_reaction = random.choice(reactions)
49  print(my_reaction + " " + music)
50  try:
51      what_do_you_think = input("What do you think of " + favourites[music] + "?")
52  except:
53      print("I don't know much about " + music + " to be honest")
``` |
| 10 | By now, you should be able to add some of your own questions and make your code react.

💡 Use print("...") to display text

💡 Use answer = input("...") to ask a question and save the answer

💡 Use if / elif / else to check what the user has said |
| 11 | Python is case sensitive, which means it's really fussy about capital letter. For example, "Pizza" is different to "pizza" in python.<br>Adapt your code so that it can cope with whatever the user types in, even if it's sometimes capitalised (LiKE THIs)<br><br>💡 Use the .lower() function to convert to lowercase or .upper() function to convert to uppercase text:<br><br>```python
music = input("What sort of music do you like?").lower()
``` |

Stage 3: Sabotage

It's hard to learn to code by watching someone else or listening to them explain how to do it.
It's better to learn by doing - have a go, make mistakes and then fix them.
It's even better to work in groups to find and fix each other's mistakes.

We're going to sabotage your own code by deliberately adding some syntax errors.

💡 A syntax error is a problem in your code that stops it from running.

Make 5 deliberate mistakes in your code then add comments at the bottom of your code saying where the mistakes are:

```
# TODO: fix the syntax error on lines, 34, 57, 22, 13 and 2
```

Share your code with each other and try to fix **and improve** each other's code