Lesson 6: Fortune Teller

In this lesson, we'll create a fortune teller that can respond to a user's yes/ no questions.

In order to do this, we'll first need to learn how to use a list to store several values in a single variable.

In programming, a list is a group of values that can be accessed through a variable.

We'll use a list to store many responses our fortune teller can give to questions.

Like variables, lists can also be thought of as buckets. Unlike variables though, lists can contain multiple pieces of information.

PART 1: Creating a List

In order to create a list, we can use the following format:

```
list = [ value1, value2, value3, value4 ]
```

Lists use *square brackets* `[ ]` to surround the values, and they separate the values inside the square brackets with *commas* `,`
We want our program to respond to questions using different responses from a list.

Try typing the following code into a new program:

```
answers = ["Of course!","No way!","Probably!","Probably
not.","I'm too tired to answer. Ask again later."]
```

PART 2: Making a Loop

Next, we'll use a loop so that the user can ask more than one question at a time.

Try typing the following code in below your list definition:

```
"I'm too tired to answer. Ask again later."]

print("Hi! I'm Predicta, the computer fortune teller!")

while True:

    print("What would you like me to predict?")

    question = input()
```

In Python, we can tell a loop to repeat forever by using `while True:`

So far, if we ran our program, the fortune teller would introduce itself, and then keep asking what question the user had, but never respond.

When the user typed in a question, it would get stored in the variable called `question`, but nothing would be done with the question just yet.

PART 3: Accessing a Value in a List

Before our program can print a response, it has to access the response from the list. One way we can do this is by using the format: `value = list[index]`
Where `index` is a number representing which item to get from the list.

Like a lot of programming languages, lists in Python are *indexed at 0*.

This means that the 1st item in a list will be at position `0`, the 2nd at position `1`, etc.

When we type `response = answers[0]`, we are setting the variable `response` equal to the 1st string in the `answers` list, which is `"Of course!"`.

Try typing this code inside the while loop:

```
while True:
    print("What would you like me to predict?")
    question = input()
    response = answers[0]
    print(response)
```

Try running your program, and see what it does so far!

PART 4: Adding Random Responses

In order to let our fortune teller give multiple answers, we can use the random module.

To choose a random response from the list we made, we can use the `random` module's `choice()` function, which chooses a random value from an list.

Try modifying your program like this:

```
import random        # Import the random module
answers = [
.  .  .
while True:
    .  .  .
    question = input()
    response = random.choice(answers)    # Replace  answers[0]
    print(response)
```

⚠ Make sure to import the random module, or `choice()` won't work!

Here, we first imported the `random` module, and then we used `random.choice()`.

`random.choice()` is a pre-built Python function which will randomly select a single item from a list.

For example, `random.choice(answers)` will tell our program to give us a random value from the `answers` list, which we then store in the variable `response`.

This lets our program print a random message from the list we made earlier.

PART 5: Breaking Out of the Loop

The last thing we should add to our program is a way for the user to quit.

The while loop that we currently have in our program is set to run forever, but there is a way to stop repeating the loop and move on to the rest of the program.

In order to break out of a loop in Python, we can use the `break` command.

Try adding these lines to your program:

```
while True:

    .  .  .

    question = input()

    if(question == "quit"):

        break

    response = random.choice(answers)

    print(response)

print("Bye!")
```

Make sure the final `print()` statement is outside of the `while` loop!

Now, typing `quit` after the fortune teller asks for a prediction will cause the program to break out of the loop, and move on to the code below that block.

Right after the loop, our program prints a goodbye message, and then exits.

Once you've finished adding these lines, try running your program!

PART 6: BONUS: Adding More Responses

Once your program works, you can try adding your own predictions to the fortune teller!

There are some example of types of responses you could add below.

Try adding some more answers, like this:

```
answers = ["Of course!","No way!","Probably!","Probably
not.","I'm too tired to answer. Ask again later.",

"That's not very likely.","I think so.","Who knows?"]
```

⚠ Don't forget to put a comma **,** after every response except the last one!

When you're done, try testing your program and adding even more responses!

For another bonus, go to Lesson 6B to edit this program to make it seem like the computer is really engaged in a conversation with you.