Lesson 3: Quiz Time

In this lesson, you will quiz the user on a topic of your choice. The questions will be multiple choice, so the user can type, A, B, C, or D to answer the questions. We will declare functions to print out the questions and to check if the answer is correct.

Part 1 – Defining Your First 2 Functions

Create a new file like before and save it as **YourName LastInitial Lesson 3 – Quiz Game.py** in your folder.

Let’s start the program by creating a variable called correctAnswers to store the number of questions the user got correct. We’ll also define a new function called printQuestion1.

Side note: the rules for function names are the same as variables, but function names are usually either written as an action or a question.

```python
correctAnswers = 0

def printQuestion1():
```

This function will not need any arguments so we don’t need any parameters inside the parentheses.

When you are defining a function, it is good to have it clear in your mind what exactly that function is going to do. Generally speaking, functions should have *one* clear task. Here, the purpose of this function is just to print the first question including the multiple-choice options. Don’t worry about the /t that you see in the code. You’ll see what that does in just a moment.

```python
def printQuestion1():
    print("C3PO was made by which Jedi?: ")
    print("\t A. Obi-Wan")
    print("\t B. Qui-Gon Jin")
    print("\t C. Anakin Skywalker") #correct answer
    print("\t D. R2D2")
```

As mentioned in the overview. This function declaration does absolutely nothing by itself. You are only telling the computer *how* to do something, not to actually *do* something. To use this function, you must call it.

```
3   def printQuestion1():
…       ...
9
10  printQuestion1() #Delete this when done testing
```

Go ahead and test your code. The question should print out to the terminal and the multiple-choice answers should be indented on the next 4 lines. Delete line 10 once you are sure that it works.

Remember our discussion of the escape character \ from before? The escape character \ changes the meaning of the character that follows it, so instead of adding a t to the string, \t actually places a tab or indent to the beginning of the string.

Now that you've seen how it works, we can make another function to print out our second question.

```
3   def printQuestion1():
…       ...
9
10  def printQuestion2():
11      print("Who trained Rey to be a Jedi?: ")
12      print("\t A. Leia Skywalker")
13      print("\t B. Yoda")
14      print("\t C. Kylo Ren")
15      print("\t D. Luke Skywalker") #correct answer
```

Part 2 – Setting up the Quiz

Now that we have our two functions ready, we can use them in our program. For now, our program will ask a question, get the user's answer, then repeat that again for the second question.

```
10  def printQuestion2():
...     ...
16
17  print("Welcome to Star Wars Trivia.")
18  print("Type A, B, C, or D to answer the questions")
19  printQuestion1()
20  choice = input()
21
22  printQuestion2()
23  choice = input()
24
```

Run your program and you should be able to see both questions and be able to answer them. What we haven't done is determine if the correct answer was given. While we could do that with just a conditional on line 21 and another on line 24, why not make it into a function?

Part 3 – Checking Answers

Before we make the function to check answers, let's think about what arguments this function would need. First, is the answer going to be same for every question? No! That means you need to know what question is being asked. Second, you need to know what the user typed in as a response in order to check if it's right.

Define a function called checkAnswer that has two parameters: questionNumber and answer.

```
10  def printQuestion2():
...     ...
16
17  def checkAnswer(questionNumber, response):
18
19  print("Welcome to Star Wars Trivia.")
20  print("Type A, B, C, or D to answer the questions")
... ...
```

Inside this function, we will check if the user typed correct response for the question. If they did, we will increase the correctAnswers variable. Don't worry about the meaning of line 18. We'll go over what global means in a future lesson.

```
def printQuestion2():
    ...

def checkAnswer(questionNumber, response):
    global correctAnswers
    if (questionNumber == 1 and response == "c"):
        correctAnswers += 1
    elif (questionNumber == 2 and response == "d"):
        correctAnswers += 1

print("Welcome to Star Wars Trivia.")
print("Type A, B, C, or D to answer the questions")
...
```

As you have probably seen for yourself, the computer is a *very* literal machine. Right now, if the user typed “C” for question 1, they would get it wrong because “C” does not equal “c”. Let’s add one simple line that solves this problem.

```
def printQuestion2():
    ...

def checkAnswer(questionNumber, response):
    global correctAnswers
    response = response.lower()
    if (questionNumber == 1 and response == "c"):
        correctAnswers += 1
    elif (questionNumber == 2 and response == "d"):
        correctAnswers += 1

print("Welcome to Star Wars Trivia.")
print("Type A, B, C, or D to answer the questions")
...
```

The function lower() can be used on any string to change all the uppercase letters to lowercase letters. If the letters were already lowercase, then there is no change. This makes it so that the user can type “c” or “C” and still get it right. In many cases, we can’t really account for *everything* the user may type, but it’s nice to give some leeway when you can.

Let’s call the function after we receive the input for each question. Then at the end, let’s print how many questions the user got right.

```
17  def checkAnswer(questionNumber, response):
    ...

25  print("Welcome to Star Wars Trivia.")
26  print("Type A, B, C, or D to answer the questions")
27  printQuestion1()
28  choice = input()
29  checkAnswer(1, choice)
30  printQuestion2()
31  choice = input()
32  checkAnswer(2, choice)
33  print("You got", correctAnswers, "out of 2 questions correct")
```

When we declared the function, we declared with two parameters: questionNumber and response. So, when we call it, we need two arguments to let the function know what question is being asked and what the user typed in.

Take a look at lines 27 – 29 and lines 30-32. Notice how readable the code is now because of our functions. These lines print out a question, get a response from the user, and then checks the answer. Readability is another benefit of creating functions.

Part 4 – Adding More Questions

Spend some time trying to add more questions to the quiz. For each question you want to ask, create a function to print the question, like we just did for the others. Then add another elif to the checkAnswers function to check the answer for your new question. If you are stuck, try asking the people around you.

You can also take this as an opportunity to change the theme of the quiz. You can make it about sports, art, language, a book series, video games, or whatever else you want.