**Assignment: End of Course Challenge!**

Can you make 3 incremental improvements to the Munch MVP code in Python? Let's go!...

**Assignment instructions**

**We've set 3 simple coding challenges.**

Each challenge is a simple improvement to the Munch app that we've already coded in Python.

If you open your Munch python file in the background, can you solve all three questions in this challenge?

There's an **easier**, a **medium** and a **hard** challenge - we'll let you decide which is which! Fear not though, if you get stuck, particularly on the harder question, the **solutions are available**!

We recommend you make a copy of your Munch python file, so that you can play around with the code, while you keep your tested and original version unchanged if you ever need it again. Of course, the source code for Munch is available via our website if you do happen to need it.

Best of luck and have fun...

**Questions for this assignment**

**Challenge One: Add numbers to your lists.**

**We coded two automatically generated lists in Python: The myMenu list and myShoppingList.**

**We also changed our code to use FOR loops in order to print each item in the list on a new line, without all of the special list related characters like commas and brackets (which appear when simply printing the whole list with a statement like print(myMenu) or similar).**

**So if we're printing each list item on a new line..**

**- Can you think of a way to print numbers before each item - so the numbers appear on the left, at the start of each line, like so:**

**1. Item 1**

**2. Item 2**

**3. Item 3**

**and so forth..**

**Here's a clue from the Python tutorial: https://docs.python.org/3/tutorial/datastructures.html#more-on-lists**

**Consider getting an element's position in the list, during the for loop cycle ;)**

**Have a go, and if you get stuck, check out the solution!**

**Challenge Two: Validate the y/n user input.**

**In step 3 of our Munch app in Python, we asked the user: "Would you like a shopping list for this menu?"**

**Let's add some validation, to make sure the user enters a valid input before proceeding.**

**- Can you think of a way to make it really clear to the user that we want them to enter a yes or no answer?**

**- Can you think of a way to validate the answer, so that if the user enters an input other than a lowercase 'y', or 'n', the app asks them to try again?**

**- Can you think of a way to allow them to proceed by also accepting an uppercase 'Y' or 'N', for example, if they have their caps lock on?**

**This is a tricky one! The clue is to use a loop, but exiting a loop hasn't been covered in detail yet, so don't worry if you need the solution. This is a good one to think about on your own before seeing the solution, so you really understand the problem, before you see how many coders solve it :)**

**Challenge 3: Make your code a little more modular.**

**In the Munch app MVP, we signed off using an else statement, which we wanted to run if the user chooses not to ask Munch to build a shopping list. We used the code:**

**else:**

**print()**

**print("You got it! Bye for now :)")**

**How about we make our end program code a little more modular, so that if we ever choose to give the user a different ending, we can work on that code separately?**

**- Can you move the two print statements into a new function?**

**- Can you call the new function if the else condition evaluates true?**

**You can name your third function anything that makes sense to you!**

**Good luck! :)**