Scripting Mod 02 Tutorial – Python Lists

Save this script as Mod02Tutorial on your VM desktop. Remember to put your name and 'Mod 02 Tutorial' at the top of your script.

We are going to have the user enter 10 items into a blank list, including words and numbers. We will then do several things with that list.

Open a blank script, add your name and homework name and save it.

We need to initialize the list then we will use append to create the list. We will use while and a counter to get exactly 10 items from the user. We also need to initialize the counter. Do that before looking at my solution below.

```
#Jim Burkman
#Mod 02 Tutorial

counter = 0
my_list = []

while counter < 10:
    list_item = input('Please enter a word or an number:  ')
    my_list.append(list_item)
    counter += 1
```

Was this correct? We can make a quick loop to print out the elements. Use a For loop and iterate through the list, printing each element:

```
for i in my_list:
    print(i)
```

Run your script. If this worked then delete the for statement and the print statement. That was just a bit of code to check our work as we go along. This is a good idea.

I don't want to have to enter 10 things every time that I run this script while building it. So I'm going to comment out my code so far and replace it with a static list to use in development:

```
#! python3

##counter = 0
##my_list = []
##
##while counter < 10:
##    list_item = input('Please enter a word or an number:  ')
##    my_list.append(list_item)
##    counter += 1

my_list = ['seven', '7', '120', 'monkey', 'cat', 'the', 'heck', '3.1', '-1384', 'banana']
```

Task 1: Check the length of the list to ensure there are 10 entries. Print out the statement like "This list has 10 items. True" True is the boolean returned from checking the length.

We know how to do this, but might need to look at the slides to get the length of a list. Make a print statement and concatenate the Boolean result after converting it to a string. You can easily do this in just the print statement. Try, then look below:

```
#Task 1 - Check the length of the list
print('This list has 10 items.  ' + str(len(my_list) == 10))
```

Task 2: Print the list

We can print the elements in a list by using a For loop like we did for the quick check that we erased. Or we can simply print the list. Run your script and you'll see that the list has brackets around it, with each element separated by a comma:

```
#Task 2 - Print the list
print(my_list)
```

Task 3: Swap the first item with the last item in the list then print the list.

Logically, one way to do this is to take the first thing in the list and copy it into a variable. Then we will put the value of the last thing in the list in the first place. Then we will copy the value in the variable into the last place in the list. Then we will print it out. Give this a try then look:

```
first_thing = my_list[0]
my_list[0] = my_list[-1]
my_list[-1] = first_thing
print(my_list)
```

Task 4: Print the first 3 items in the list and the last three items in the list.

Read the part about slicing again. Separate your two slices in the print statement with a comma:

```
#Task 4: Print the first 3 items in the list and the last three i
print(my_list[0:3], my_list[-3:])
```

Task 5: Loop through and print all the items in the list.

Easy peasy. You've done this before:

```
#Task 5:  Loop through and print all the items in the list.
for i in my_list:
    print(i)
```

Task 6: Use an IF statement to check to see if the word "cat" is in the list and let the user know.

Ok, go check the slide on how to see if an element is in a list. Use an IF statement and print 'There is a cat in my list' if that's true, otherwise print 'There is no cat in my list'

```
#Task 6:  Use an IF statement to check to see it
if 'cat' in my_list:
    print('There is a cat in my list')
else:
    print('There is no cat in my list')
```

Task 7:

Get the name of a Marvel character from the user and pass that to a function that randomly inserts the name into the list (import random). Remember to put your function at the top of your script, after your personal comments and variable declarations. The order of things is homework comments, imports, variable declarations, functions, code body.

We can do this. Import random. Now define a function and give it a name. You'll pass it one value. In the function pick a random position between 0 and 9. Our list has 10 items indexed 0-9. If we wanted to make this work for a list of any size we would replace 9 with the code to get the length of our list (minus one). Use the insert function of your list to insert the new item. That function takes two elements: position, item being inserted. Try this then look at my function:

```
def rando_insert(thing_being_inserted):
    position = random.randint(0,9)
    my_list.insert(position, thing_being_inserted)
```

In the body of my script I'm going to get the user input (with no checks) then call my new function and pass that user input:

```
#Task 7:
another_item = input('Please insert the name of
rando_insert(another_item)
```

Task 8: Get the index for the Marvel character and print it out so that it looks something like 'Thor is at index 7', but Thor and 7 are the results of code.

Easy. Go hit the slides to find that code to get the index for a value. Concatenate up a string. Try, then look:

```
#Task 8:
print('\nTask 8')
print(another_item + ' is at index ' + str(my_list.index(another_item)))
```

You could have also assigned the index location to a fresh variable then printed that in your concatenated string. That's ok.

Task 9:

Copy all the integers in the original list to a new list, then sort and print out that list.

Hint: use a for loop with try/except. Try to int(each list item). If it's not an int it will throw an error, otherwise put it in a new list of just ints.

Initialize a new list up top on your script. Like we did with our currently commented out my_list. Make a For loop to iterate through my_list. For each item try to cast it to an int. If that works then add it to your new list (NOTE: cast it as you add it to the list, otherwise you'll get a list of strings that look like the integers that you want, not actual integers). If it fails just continue so that processing returns to the For loop to get the next item.

Check the slides to figure out how to sort a list. Sort, then print your list (not the items in the list…just the list). Try all this then look:

```
#Task 9:  Copy all the integers in the original l

for jim in my_list:
    try:
        int(jim)
        ints_only.append(int(jim))
    except:
        continue

ints_only.sort()
print ('These are the integers from the list')
print (ints_only)
```

If you didn't cast the items to ints when you made your new list then your output won't be sorted like numbers. It will be sorted like strings. In that case this (1, 120, 2, 3, 400, 41) is correct for string, but we want numbers like (1, 2, 3, 41, 120, 400).

Task 10: Convert the original list to a tuple and print the tuple

Back to the slides. This is pretty self-explanatory though, so give it a try before looking below:

```
#Task 10:
my_tuple = tuple(my_list)
print(my_tuple)
```

Task 11: Try and change the first item in the tuple to "cat", but catch the error and print out "Tuples are immutable!"

This is also self-explanatory so give it a try then look below:

```
#Task 11:
try:
    my_tuple[0] = 'cat'
except:
    print('Tuples are immutable!')
```

Task 12: One last task. Copy this new list in this text box into your script but just under your Task 12 comment:

```
list_in_list = [[1,2,3],['a','b','c']]
```

Make a For loop and iterate through the list and print out each element. See how each element is an “inner” list? That makes sense. Our list is filled with lists. If we want to unpack the “inner” lists we would have another for loop inside the first one. Delete your print statement and replace it with another for loop (use a new variable name) and print the items in the inner loops. Try this then look:

```
#Task 12:
print('\nTask 12')
list_in_list = [[1,2,3],['a','b','c']]
for i in list_in_list:
    for j in i:
        print(j)
```

This is a very, very important concept that you need to feel comfortable with. In this example I is each "inner" list. Then j is each item in each "inner" list. I can nest lists inside of lists inside of lists….as much as I want. When it comes time to unpack things just think of one For loop for each nesting layer. Play with this idea on your own because we use nesting often.

Go back and make nice comments and good spacing. Remove your static list and uncomment your early code to get values from the user. Add print() and \n as needed to get your output to look like mine. Run your script. Run it again in a command window. Leave the script on your VM desktop for grading and back it up to the proper sftp server.

My output:

```
Please enter a word or an number:  cat
Please enter a word or an number:  dog
Please enter a word or an number:  1
Please enter a word or an number:  15
Please enter a word or an number:  6.7
Please enter a word or an number:  100
Please enter a word or an number:  elephant
Please enter a word or an number:  egg
Please enter a word or an number:  run
Please enter a word or an number:  24

Task 1
This list has 10 items.  True

Task 2
['cat', 'dog', '1', '15', '6.7', '100', 'elephant', 'egg', 'run', '24']

Task 3
['24', 'dog', '1', '15', '6.7', '100', 'elephant', 'egg', 'run', 'cat']

Task 4
['24', 'dog', '1'] ['egg', 'run', 'cat']

Task 5
24
dog
1
15
6.7
100
elephant
egg
run
cat
```

```
Task 6
There is a cat in my list

Task 7
Please insert the name of a Marvel character: Loki

Task 8
Loki is at index 9

Task 9
These are the integers from the list
[1, 15, 24, 100]

Task 10
('24', 'dog', '1', '15', '6.7', '100', 'elephant', 'egg', 'run', 'Loki', 'cat')

Task 11
Tuples are immutable!

Task 12
1
2
3
a
b
c
```