Nina Ngouabou

Loops assignments

10/12/2024

Lab cake 1: Looping through Athletes @channel**Objective:** Practice using loops to iterate through a list and display information.
Task: Write a Python program that uses a list of four U.S. women athletes who have competed in the 400 meters at the Olympics. Your program should do the following:

1. Create a list called `athletes` with the following names:

    - Allyson Felix
    - Sanya Richards-Ross
    - Shaunae Miller-Uibo
    - Phyllis Francis

1. Use a `for` loop to display each athlete's name along with the lap number they completed. The output should be in the following format:

```
Lap 1: Allyson Felix has completed their lap!
Lap 2: Sanya Richards-Ross has completed their lap!
Lap 3: Shaunae Miller-Uibo has completed their lap!
Lap 4: Phyllis Francis has completed their lap!
```

Requirements:

- Do not use the `enumerate()` function.
- Use a counter variable to keep track of the lap number.

Bonus Challenge:

- Modify your code to display a message at the end that says: "All athletes have completed their laps!"

Submission
Please submit your code in a file named `athlete_lap_assignment.py`and upload it to gihub. Make sure to test your code to ensure it produces the correct output.

```python
# write a Python program that uses a list of four U.S women athletes who have competed in the 400 meters at the Olympics with the following
# create a list called athletes

athletes =["Allison Felix", "Sanya Richards-Ross", "Shaunae Miller-Uibo", "Phyllis Francis"]

# Initialize the lap counter starting at 0. When I did 1 it did not work
lap = 0

# use a for loop to display each athlete's name along with lap number they have completed.

for athlete in athletes:
    lap += 1 # using the count method
    print (f"Lap {lap}: {athlete} has completed their lap!")

# print the end message
print ("\n All athletes have completed their laps!")
```

```
PROBLEMS 4  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS
TERMINAL

WIN10@DESKTOP-OL1MV38 MINGW64 ~/Documents/CloudFormation/Python (master)
$ python athlete_lap_assignment.py
Lap 1: Allison Felix has completed their lap!
Lap 2: Sanya Richards-Ross has completed their lap!
Lap 3: Shaunae Miller-Uibo has completed their lap!
Lap 4: Phyllis Francis has completed their lap!

 All athletes have completed their laps!
```

Lab cake 2: Reversing a List @channel
Objective: Practice reversing a list and transferring its elements into a new list using loops.
Task: Write a Python program that works with the list called laura_things containing the following items:

- "sewing machine"
- "scissor"
- "cutting mat"
- "television"

Your program should do the following:

1. Create a list called laura_things with the items listed above.
2. Reverse the order of the items in laura_things.
3. Transfer each item from the reversed list into a new list called reversed_things.
4. Print out the new list reversed_things to show that it contains the items in reverse order.

Requirements:

- You must reverse the list using slicing or a loop (do not use Python's built-in reverse methods like reverse()).
- The final output should look like this:
- ['television', 'cutting mat', 'scissor', 'sewing machine']

Bonus Challenge:

- After reversing the list and creating reversed_things, print a message that says: "The list has been successfully reversed!"

Submission
Please submit your code by [insert due date here] in a file named reverse_list_assignment.py. Make sure to test your code to ensure it produces the correct output (edited)

Using the for loop method to reverse the order of the list

```python
# 1- create a list called laura_things with the items listed above

laura_things = ["sewing machine", "scissor", "cutting mat", "television"]
print (f"The original order from the laura_things list is:\t {laura_things}\n")

for index in range(len(laura_things)):
    print (f"Item {index +1} in the list is {laura_things[index]} and is at index: {index}") # trying to replicate what we did in class
# 2- reverse the order of the list in laura_things using the loop method

# 3-4- Empty list to store reversed items and transfer each item from the list into a new list called reversed_things

reversed_things = []
# Loop through the original list in reverse order and append each item
for reverse_list in range(len(laura_things)):
    reversed_things.append(laura_things [::-1][reverse_list])

print(f" \nThe reversed order from the laura_things list is:\t {reversed_things}\n")

# using the slicing method
#reverse_things = (laura_things[::-1])
#print (f"The reverse order from the laura_things list is:\t {reverse_things}")

print (f" \nThe final output of laura_things list in reversed order is:\n {reversed_things}\n")

for index in range(len(laura_things)):
    print (f"Item {index +1} in the reversed list is {reversed_things[index]} and is at index: {index}") # trying to replicate what we did in class

# 5- Print a message that says 'The list has been successfully reversed!

print("\nThe list has been successfully reversed!")

```

```
WIN10@DESKTOP-OL1MV38 MINGW64 ~/Documents/CloudFormation/Python (master)
$ python reverse_list_assignment.py
The original order from the laura_things list is:       ['sewing machine', 'scissor', 'cutting mat', 'television']

Item 1 in the list is sewing machine and is at index: 0
Item 2 in the list is scissor and is at index: 1
Item 3 in the list is cutting mat and is at index: 2
Item 4 in the list is television and is at index: 3

The reversed order from the laura_things list is:       ['television', 'cutting mat', 'scissor', 'sewing machine']


The final output of laura_things list in reversed order is:
 ['television', 'cutting mat', 'scissor', 'sewing machine']

Item 1 in the reversed list is television and is at index: 0
Item 2 in the reversed list is cutting mat and is at index: 1
Item 3 in the reversed list is scissor and is at index: 2
Item 4 in the reversed list is sewing machine and is at index: 3

The list has been successfully reversed!
```

Using the slicing method to reverse the order of the list

The outcome is the same, but the code had to be changed on line 32. Instead of reversed_things used in loops, it is now reverse_things as it is the value I've assigned to the reverse to make the difference between the 2 programs.

```python

# using the slicing method
reverse_things = (laura_things[::-1])
print (f"The reverse order from the laura_things list is:\t {reverse_things}")

print (f" \nThe final output of laura_things list in reversed order is:\n {reversed_things}\n")

for index in range(len(laura_things)):
    print (f"Item {index +1} in the reversed list is {reverse_things[index]} and is at index: {index}") # trying to replicate what we did in class

# 5- Print a message that says 'The list has been successfully reversed!

print("\nThe list has been successfully reversed!")

```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

TERMINAL

```
WIN10@DESKTOP-OL1MV38 MINGW64 ~/Documents/CloudFormation/Python (master)
$ python reverse_list_assignment.py
The original order from the laura_things list is:       ['sewing machine', 'scissor', 'cutting mat', 'television']

Item 1 in the list is sewing machine and is at index: 0
Item 2 in the list is scissor and is at index: 1
Item 3 in the list is cutting mat and is at index: 2
Item 4 in the list is television and is at index: 3
The reverse order from the laura_things list is:        ['television', 'cutting mat', 'scissor', 'sewing machine']

The final output of laura_things list in reversed order is:
 []

Item 1 in the reversed list is television and is at index: 0
Item 2 in the reversed list is cutting mat and is at index: 1
Item 3 in the reversed list is scissor and is at index: 2
Item 4 in the reversed list is sewing machine and is at index: 3

The list has been successfully reversed!
```