Exercises chapter 4: Lists

Saul SL

June 2023

1 Exercise 4-1 Pizzas

Think of at least three kinds of your favorite pizza. Store these pizza names in a list, and then use a for loop to print the name of each pizza.

- Modify your for loop to print a sentence using the name of the pizza instead of printing just the name of the pizza. For each pizza you should have one line of output containing a simple statement like I like pepperoni pizza.
- Add a line at the end of your program, outside the for loop, that states how much you like pizza. The output should consist of three or more lines about the kinds of pizza you like and then an additional sentence, such as I really love pizza!

```
pizzas = ['pepperoni', 'cheese', 'ny']
for pizza in pizzas:
    print(f"- {pizza}")

for pizza in pizzas:
    print(f"I like {pizza} pizza")

for pizza in pizzas:
    print(f"I like {pizza} pizza")
print('...\nI really like pizza')
```

2 Exercise 4-2 Animals

Think of at least three different animals that have a common characteristic. Store the names of these animals in a list, and then use a for loop to print out the name of each animal.

- Modify your program to print a statement about each animal, such as A dog would make a great pet.
- Add a line at the end of your program stating what these animals have in common. You could print a sentence such as Any of these animals would make a great pet!

```
animals = ['llama', 'armadillo', 'puma']
for animal in animals:
    print(f"- {animal.title()}")

for animal in animals:
    ianimal = f"{animal.title()}"
    if ianimal[0] == 'A':
        print(f"An {ianimal} has four legs")
    else:
        print(f"A {ianimal} has four legs")

print('\nAll the above animals are found in Bolivia')
```

3 Exercise 4-3 Counting to Twenty

Use a for loop to print the numbers from 1 to 20, inclusive.

```
for i in range(1, 21):
    print(i)
```

4 Exercise 4-4 One Million

Make a list of the numbers from one to one million, and then use a for loop to print the numbers. (If the output is taking too long, stop it by pressing ctrl-C or by closing the output window.)

```
million = list(range(1, 1_000_001))
for i in sorted(million, reverse=True):
    print(i)
```

5 Exercise 4-5 Summing a Million

Make a list of the numbers from one to one million, and then use min() and max() to make sure your list actually starts at one and ends at one million. Also, use the sum() function to see how quickly Python can add a million numbers.

```
million = list(range(1, 1_000_001))
print("There is a list of integers from 1 to 1000000")
print(f"The min value is: {min(million)}")
print(f"The max value is: {max(million)}")
print(f"The sum of all values is: {sum(million)}")
```

6 Exercise 4-6 Odd Numbers

Use the third argument of the range() function to make a list of the odd numbers from 1 to 20. Use a for loop to print each number.

```
odd_n = range(1, 20, 2)
for odd in odd_n:
    print(odd)
```

7 Exercise 4-7 Threes

Make a list of the multiples of 3 from 3 to 30. Use a for loop to print the numbers in your list.

```
imult = list(range(3, 33, 3))
for i in imult:
    print(i)
```

8 Exercise 4-8 Cubes

A number raised to the third power is called a cube. For example, the cube of 2 is written as 2**3 in Python. Make a list of the first 10 cubes (that is, the cube of each integer from 1 through 10), and use a for loop to print out the value of each cube.

```
cubes = []
for i in range(1, 11):
    cube = i**3
    cubes.append(cube)
    print(cube)
```

9 Exercise 4-9 Cube Comprehension

Use a list comprehension to generate a list of the first 10 cubes.

```
cubes_2 = [i**3 for i in range(1, 11)]
```

10 Exercise 4-10 Slices

Using one of the programs you wrote in this chapter, add several lines to the end of the program that do the following:

- Print the message The first three items in the list are:. Then use a slice to print the first three items from that program's list.
- Print the message Three items from the middle of the list are:. Use a slice to print three items from the middle of the list.
- Print the message The last three items in the list are:. Use a slice to print the last three items in the list.

```
print("The first 3 items of the list are:")
for i in cubes_2[:3]:
    print(i)


def sublist_even(ilist, n_subitems):
    n_items = len(ilist)
    # list even, sublist even
    if n_items % 2 == 0 and n_subitems % 2 == 0:
        mid_point = n_items / 2
        i_start = (mid_point + 1) - (n_subitems/2)

    # list odd, sublist even
    elif n_items%2 != 0 and n_subitems%2 == 0:
        mid_point = (n_items+1) / 2
        i_start = mid_point - (n_subitems/2)

    # list even, sublist odd
    elif n_items%2 == 0 and n_subitems%2 != 0:
        mid_point = (n_items) / 2
        i_start = mid_point - ((n_subitems-1)/2)

    # list odd, sublist odd
    elif n_items%2 != 0 and n_subitems%2 != 0:
        mid_point = (n_items + 1) / 2
        i_start = mid_point - ((n_subitems-1)/2)

    i_start = int(i_start - 1)
    i_end = i_start + n_subitems
    isublist = ilist[i_start:i_end]
    return isublist

```

```
print("The 3 items in the middle are:")
isublist = sublist_even(cubes_2, 8)
for i in isublist:
    print(i)

print(cubes_2)


print("The last 3 items are:")
for i in cubes_2[-3:]:
    print(i)
```

11 Exercise 4-11 My Pizzas, Your Pizzas

Start with your program from Exercise 4-1 (page 56). Make a copy of the list of pizzas, and call it friend_pizzas.Then, do the following:

- Add a new pizza to the original list.
- Add a different pizza to the list friend_pizzas.
- Prove that you have two separate lists. Print the message My favorite pizzas are:, and then use a for loop to print the first list. Print the message My friend's favorite pizzas are:, and then use a for loop to print the sec- ond list. Make sure each new pizza is stored in the appropriate list.

```
pizzas = ['pepperoni', 'cheese', 'ny']
friend_pizzas = pizzas[:]
pizzas.append('vegetarian')
friend_pizzas.insert(len(friend_pizzas), 'stone')

print("My pizzas are:")
for pizza in pizzas:
    print(f"- {pizza}")

print("My friend's pizzas are:")
for pizza in friend_pizzas:
    print(f"- {pizza}")
```

12 Exercise 4-12 More Loops

All versions of foods.py in this section have avoided using for loops when printing to save space. Choose a version of foods.py, and write two for loops to print each list of foods.

```
my_foods = ['pizza', 'falafel', 'carrot cake']
friend_foods = my_foods[:]

print("My favorite foods are:")
for food in my_foods:
    print(f"- {food}")

print("\nMy friend's favorite foods are:")
for food in friend_foods:
    print(f"- {food}")
```

13 Exercise 4-13 Buffet

A buffet-style restaurant offers only five basic foods. Think of five simple foods, and store them in a tuple.

- Use a for loop to print each food the restaurant offers.
- Try to modify one of the items, and make sure that Python rejects the change.
- The restaurant changes its menu, replacing two of the items with different foods. Add a line that rewrites the tuple, and then use a for loop to print each of the items on the revised menu.

```
foods = ('quinoa', 'chicken soup', 'pork', 'fish')
for food in foods:
    print(food)

# This will generate an error
foods[0] = 'beans'

# Change tuple
print("The food items has changed")
foods = ('beans', 'pizza', 'pork', 'fish')

for food in foods:
    print(food)
```