Python Activity 18: Lists

Learning Objectives
Students will be able to:
Content:
- Define a **list**
- Identify **elements** of a list
- Explain the purpose of positive and negative indexes in a list**.**
- Explain how to access individual elements of a list
- Explain how following list functions: append(), insert(), remove(), count(), index()
- Explain how to replace an item

Process:
- Write code that prints a list
- Write code that edits a list – add, remove, and insert items

Alternate Python IDE: https://www.programiz.com/python-programming/online-compiler/

Role	Name	Email
Facilitator		
Spokesperson		
Quality Control		
Process Analyst		

Setup

1. The **spokesperson** should make a copy of this document in your own folder and then share the new document with professor Aydin (aydinn@kenyon.edu) and the graders: oppongkrampah1@kenyon.edu and coons1@kenyon.edu with edit access. Notify everyone.
2. The **facilitator** should create a new project in an online IDE of your choice. Paste link to the project in the box below after saving it. Make sure to save your code after editing it.
3. Have the spokesperson enter in your answers for each item below.
4. Use a different color to enter your answers

Link to the Project	

Critical Thinking Questions:

FYI: A ***sequence*** is an object that stores multiple data items in a contiguous manner. Two types of sequences are **strings** and **lists.** Each value stored in a list is called an **element.**

1. Examine the sample lists below.

Sample Lists in Python

```
digits = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
fruits = ["apple", "banana", "cantelope", "pear", "orange"]
studentData = ['Jones', 10234, 3.5, 'Brown', 23145, 2.8]
```

a. How many **elements** does the list named **digits** contain? _____

b. What type of data is stored in each list (String, numeric)?

- **digits** list: _____
- **fruits** list: _____
- **studentData** list: _____

c. How would you define a **list?** _____

_____

2. The second line of code in the following program prints the first **element** in the **digits** list.

```
fruits = ["apple", "banana", "cantaloupe", "pear"]
print(fruits[0])
```

a. What value in the list does **fruits[3]** represent? _____

b. In the interpreter, write a line of code that prints the last value.

_____

c. Edit your print statement in 'b' so that it prints **fruits[4]**. What is printed? Why?

_____

d. Edit your print statement so that it prints **fruits[-1]**. What is printed? _____

e. Change **-1** to **-2** in "d." What is printed? _____

FYI: The number used to locate an element in the list is called an **index**.

f. Explain how the positive and negative indexes locate specific elements.

_____

_____

g. What is printed with the following print statement: **print(fruits)**? How is the information displayed?

_____

_____

3. Enter and execute the following code:

```
gradebook = ["Abbot", 78, 89, "Barrava", 97, 86]
```

```
print(gradebook )
for x in gradebook:
  print(x, end = " ")
  print()
```

a. What is the output for the second line of code: `print(gradebook)`

_____

b. Examine the following code. It contains a FOR loop but does not use the **range()** function. In previous FOR loops the values resulting from the **range()** function were stored in **x** during each iteration of the loop.

```
for x in gradebook:
  print(x, end = " ")
print()
```

i. What is being stored in **x** for each iteration of the loop?

_____

ii. What is the output for this code?

_____

c. What are the similarities and differences between the output for "a." and "b."?

_____

_____

_____

d. Add each of the following print statements to the code above. What is the output for each statement? Explain the output.

```
print((gradebook[1] + gradebook [2]) /2)

print(gradebook [0] + gradebook [3] )

print(gradebook[2] + gradebook [3])
```

_____

_____

_____

_____

4. Enter and execute the following code:

```
flowers = ["rose", "peony", "tulip", "daffodil", "carnation", "daisy"]
print (flowers)

flowers.append( "gardenia")
print (flowers)
```

a. Explain what the following line of code does: `flowers.append('gardenia')`

_____

b. Write a line of code that would add the flower **lavender** to the list.

_____

5. Enter and execute the following code:

```
flowers = ["rose", "peony", "tulip", "daffodil", "carnation", "daisy"]
print (flowers)


flowers.insert(2,"lily")
print (flowers)
```

a. Explain what the following line of code does: `flowers.insert(2,'lily')`

_____

b. Write a line of code that would place the flower: **sunflower** at the beginning of the list.

_____

6. Enter and execute the following code:

```
flowers = ["rose", "peony", "tulip", "daffodil", "carnation", "daisy"]
print (flowers)

del flowers[2]
print (flowers)
```

a. Explain what the following line of code does: `del flowers[2]`

_____

b. Write a line of code that would delete the last flower in the list.

_____

7. Enter and execute the following code:

```
flowers = ["rose", "peony", "tulip", "daffodil", "carnation", "daisy"]
print (flowers)

flowers.remove("tulip")
print (flowers)
```

a. Explain what the following line of code does: `flowers.remove('tulip')`

_____

b. Write a line of code that would delete 'daffodil' from the list.

_____

c. Edit the code to determine what happens if the same flower appears in the list twice and use the `remove()` function to remove the word. Does it remove both instances of the word? _____

d. Write a line of code that attempts to remove the flower: **sweet pea.** What happens when the code is executed?

_____

8. Enter and execute the following code:

```
flowers = ["rose", "peony", "tulip", "daffodil", "carnation", "daisy"]
print (flowers)

flowers[1] = "freesia"
print (flowers)
```

a. Explain what the following line of code does: `flowers[1]= 'freesia'`

_____

b. Write a line of code that would replace 'daffodil' with 'gardenia'.

_____

c. Explain what the following line of code does: `flowers[-3]= 'lily'`

_____

d. Explain what happens when `flowers[8]= 'lily'` is added to the program:

_____

_____

9. Enter and execute the following code:

```
ballot = ["y", "y", "n", "y", "n", "y", "y", "n",
"y", "n", "y", "n", "n", "y", "n", "y", "y"]
numY = ballot.count("y")
numN = ballot.count("n")
print("Number of YES votes:", numY)
print("Number of NO votes: ", numN)
if numY > numN:
  print("The ballot was approved." )
else:
  print("The ballot was not approved.")
```

a. What is the output of this program?

_____

_____

_____

b. Explain the line of code: **`numY = ballot.count('y')`** What does the `count()` function do?

_____

Application Questions: Use the Python Interpreter to check your work

1. Create a program that prints a given list, prompts the user for a name and average, adds the new information to the list and prints the new list. You can use the **len() function** with the name of the list as an argument to determine the length of the list. It should produce output similar to the following:

```
LIST: ['Mary Smith', 132, 'Jean Jones', 156, 'Karen Karter', 167]
Name to add to the list: Ann Kert
Average: 189
UPDATED LIST: ['Mary Smith', 132, 'Jean Jones', 156, 'Karen Karter', 167, 'Ann Kert', 189]
There are now 8 items in the list.
```

_____

_____

_____

_____

_____

_____

_____

_____

_____

2. Revise the previous program so that it allows the user to enter the name of a person in the list whose average needs updating and also prompts the user for the new average. The program should then update the list and print the new list. Place the additional code below. The program should produce output similar to the following:

```
LIST: ['Mary Smith', 132, 'Jean Jones', 156, 'Karen Karter', 167]
Name to add to the list: Ann Kert
Average: 189
UPDATED LIST: ['Mary Smith', 132, 'Jean Jones', 156, 'Karen Karter', 167, 'Ann Kert', 189]
There are now 8 items in the list.
Whose average needs updating: Karen Karter
New Average: 170
REVISED LIST: ['Mary Smith', 132, 'Jean Jones', 156, 'Karen Karter', 170, 'Ann Kert', 189]
```

_____

_____

_____

_____

_____

_____