IT Academy - Data Science Itinerary

Sprint 3 - Data Structures

Assignment by: Kat Weissman

Python Learning Objectives:

- Lists
- Tuples
- Sets
- Dictionaries

Recommended learning resources:

- https://www.w3schools.com/python/python_lists.asp
- https://www.w3schools.com/python/python_tuples.asp
- https://www.w3schools.com/python/python_sets.asp
- https://www.w3schools.com/python/python_dictionaries.asp

Exercise 1

Create a list that groups the months of the year into quarters (Q1: January, February and March, Q2: April, May, June ...), that is, a list with 4 lists inside.

In [158…

```
quarters = [['January', 'February', 'March'],
            ['April', 'May', 'June'],
            ['July', 'August', 'September'],
            ['October', 'November', 'December']]
```

Exercise 2

Create a code that allows you to access:

- The second month of the first quarter

- The months of the first quarter
- September and October

In [159…

```
#Python indexing begins with 0, so the first element is [0] and the second element is [1]
q1m2 = quarters[0][1]
print ("The second month of the first quarter is:", q1m2)
```

```
The second month of the first quarter is: February
```

In [160…

```
q1 = quarters[0]
print ("The months of the first quarter are:")
for month in q1: print(month)
```

```
The months of the first quarter are:
January
February
March
```

In [161…

```
#The index method can be used to find the index of values in a list.
myMonths = ["September","October"]
indexMonths = []
for quarter in quarters:
    for month in myMonths:
        if month in quarter:
            indexMonths.append((quarters.index(quarter),quarter.index(month)))

for i in range (len(myMonths)):
    print (myMonths[i], "is located at the index", indexMonths[i])
```

```
September is located at the index (2, 2)
October is located at the index (3, 0)
```

In [162…

```
#The resulting indices can then be used to access the values at the location of the index.
for i in range(len(indexMonths)):
    print(quarters[indexMonths[i][0]][indexMonths[i][1]])
```

```
September
October
```

Exercise 3

Create a list of disordered numbers and answer the following questions:

- How many numbers are there?
- How many times does the number 3 appear
- How often do the numbers 3 and 4 appear?
- What is the largest number?
- What are the 3 smallest numbers?
- What is the range of this list?

In the code below, the random module is used to generate the list of numbers.

Reference:

- https://www.w3schools.com/python/module_random.asp
- https://www.w3schools.com/python/python_lists_methods.asp
- https://www.w3schools.com/python/python_ref_functions.asp

In [163…

```
#The random module can be used to generate random numbers
import random
#Optionally choose an integer to set the random seed in order to get the same results each time the code runs.
random.seed(5)

#Create a variable to randomly assign the length of the list in the range specified.
length = random.randint(20,50)

#Create an empty list and use a for loop to populate the list with random numbers in the range specified.
myList = []
for i in range(length): myList.append(random.randint(0,20))

print(myList)
```

```
[8, 11, 20, 16, 0, 14, 7, 20, 1, 5, 3, 11, 15, 7, 12, 17, 3, 18, 7, 0, 6, 13, 8, 5, 12, 5, 2, 4, 19, 19, 14, 4,
4, 0, 0, 6, 6, 5, 5]
```

In [164…

```
#Use the len function to return the length of the list.
print ("There are", len(myList), "numbers in the list.")

#Use the count method to return the number of times an item appears in the list.
print ("The number 3 appears", myList.count(3), "times in the list.")
print ("The numbers 3 and 4 appear", myList.count(3) + myList.count(4), "times in the list.")

#Use the max function to return the highest number in the list.
print ("The largest number in the list is", str(max(myList)) + ".")
```

```
#Assign the sorted list to a new variable in order to preserve the unsorted list.
mySortedList = sorted(myList)

#Use the sorted list to find the three smallest numbers.
print ("The three smallest numbers in the list are", str(mySortedList[0:3]) +".")

#Use the max and min functions to find the range of the list.
print ("The range of the list is", str(max(myList)-min(myList)) + ".")
```

```
There are 39 numbers in the list.
The number 3 appears 2 times in the list.
The numbers 3 and 4 appear 5 times in the list.
The largest number in the list is 20.
The three smallest numbers in the list are [0, 0, 0].
The range of the list is 20.
```

Exercise 4

Create a dictionary as follows and answer the questions:

purchase = {"Apples": {"Qty": 5, "€": 0.42}, "Pears": {"Qty": 3, "€": 0.66}}

- Add some more fruit
- How much did the pears cost in total?
- How many fruits did we buy in total?
- What is the most expensive fruit?

In [165…

```
purchase = {"Apples": {"Qty": 5, "€": 0.42}, "Pears": {"Qty": 3, "€": 0.66}}

#Add items to the dictionary using the update method. The object must be a dictionary.
purchase.update({"Oranges":{"Qty":2, "€": 0.45},
                 "Strawberries":{"Qty":1, "€": 2.90},
                 "Melons":{"Qty":2, "€": 2.70},
                 "Avocados":{"Qty":3, "€": 1.00},
                 "Pineapples":{"Qty":1, "€": 1.75}})

print(purchase)
```

```
{'Apples': {'Qty': 5, '€': 0.42}, 'Pears': {'Qty': 3, '€': 0.66}, 'Oranges': {'Qty': 2, '€': 0.45}, 'Strawberri
es': {'Qty': 1, '€': 2.9}, 'Melons': {'Qty': 2, '€': 2.7}, 'Avocados': {'Qty': 3, '€': 1.0}, 'Pineapples': {'Qt
y': 1, '€': 1.75}}
```

In [166…
```
#Access the values of items in the dictionary by using the key
costPears = purchase["Pears"]["Qty"]*purchase["Pears"]["€"]
print("The total cost of pears is €" + str(costPears) + ".")
```

```
The total cost of pears is €1.98.
```

In [167…
```
#initialize the counting variable
totalFruits = 0
#fruits represents the keys of the purchase dictionary.
for fruits in purchase:
    #increment the counter by value of "Qty"
    totalFruits += (purchase[fruits]["Qty"])

print ("The purchase includes", totalFruits, "fruits.")
```

```
The purchase includes 17 fruits.
```

In [168…
```
#The max function can return the maximum value, but doesn't return the name of the key
max(fruits["€"] for fruits in purchase.values())
```

Out[168…
```
2.9
```

In [169…
```
#initialize a tuple variable to track the most expensive fruit.
expensiveFruit =("",0)
#iterate through the dictionary to keep track of the most expensive fruit.
for fruits in purchase:
    if purchase[fruits]["€"] > expensiveFruit[1]:
        expensiveFruit = (fruits,purchase[fruits]["€"])

print("The most expensive fruit is {} for €{:.2f}.".format(expensiveFruit[0],expensiveFruit[1]))
```

```
The most expensive fruit is Strawberries for €2.90.
```