SPRINT3-T01-EstructuraDeDades

October 4, 2021

1 S03 T01: Estructura de dades

Descripció
Comencem a familiaritzar-nos amb les estructures de dades de Python

Nivell 1 - *Exercici 1*
Crea una llista que agrupi els mesos de l'any en trimestres (1T: Gener, Febrer i Març, 2T: Abril, Maig, Juny…), és a dir, una llista amb 4 llistes dins.

- ***Exercici 2*** Crea un codi que et permeti accedir a:
  El segon mes del primer trimestre Els mesos del primer trimestre Setembre i octubre
- ***Exercici 3***
  Crea una llista amb nombres desordenats i respon a les següents preguntes:

  ```
  - Quants números hi ha?
  - Quantes vegades apareix el número 3
  - Quantes vegades apareixen els nombres 3 i 4?
  - Quin és el número més gran?
  - Quins són els 3 números més petits?
  - Quin és el rang d'aquesta llista?
  ```

- ***Exercici 4***
  Crea un diccionari de la següent forma i respon a les preguntes:
  compra = { "Pomes" : {"Qty": 5, "€": 0.42}, "Peres" : {"Qty": 3, "€": 0.66} }

  ```
  - Afegeix alguna fruita més
  - Quant han costat les peres en total?
  - Quantes fruites hem comprat en total?
  - Quina és la fruita més cara?
  ```

1.1 1 Level 1

1.1 ***Exercice 1***

I will creat a list to group the month of the year in 4 quarters: First I will creat 4 lists, one for each quarter with the months of each one of them Second a list with that includes the 4 quarters

[3]:
```
q1 =["January", "February", "March"]
q2 =["April", "May", "June"]
q3 =["July", "August", "September"]
q4 =["October", "November", "December"]
```

```
yearQuarters = [q1, q2, q3, q4]
print(yearQuarters)
```

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

We see when I printed it that it worked! iuju!

1.1.1 1.2 *Exercice 2*

I want to check several things of my list. To acces to different items of the list of the list, Python allow to access throug its item index. To do so, I just put in a row the different items index, fom out to in, that I want to acces

- Print the second month of the 1st quarter:

[11]:
```
print(yearQuarters[0][1])
#this give me acces to the first item [0] of the list yearsquarter and the␣
↪second itmeof that memebre [1]
```

```
February
```

We see that I have printed the second month of the first quarte!! iuju!

- Acces the months of the first quarter:
  To do so, I will use two methods. First, just print the imte with index 0 of the year quarters list. Second, I will do a for loop that access to each quarter, so items of the yearQuarter list, and then prints the firts month of each list

[20]:
```
print(yearQuarters[0])

for i in yearQuarters[0]: #hereI get into yearQuarters item 0 (the first one)
    print(i) #here I print all its items
```

```
['January', 'February', 'March']
January
February
March
```

We can see I was able to print the months of each first quarter

I also wanted to do:

- Acces the first months of the quarter:
  To do so I will use a for loop that access to each quarter, so items of the yearQuarter list, and then prints the firts month of each list

[17]:
```
for q in yearQuarters: #here I say, for each item of the yearQuarter list
    print(q[0])#here I print the 1st item (index 0) of each item of the␣
↪yearQuarter list
```

```
January
April
July
October
```

I was also able to print the first month of each quarter! hurray!

- Acces months September and October
  To do so I will try to check if September and october are in the lists and print it.

[146]:
```
print(yearQuarters)

for q in yearQuarters:
    if "September" in q:
        print("September" in q)
        print("hurray, is here!")
        print(q)
    else:
        print("")

for q in yearQuarters:
    if "October" in q:
        print("hurray, is here!")
        print(q)
    else:
        print("")
```

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


True
hurray, is here!
['July', 'August', 'September']




hurray, is here!
['October', 'November', 'December']
```

While I was able to check separetley for September and October, I 'm still not able to check both at the same time. When I try to do the same with: if "September" or "October" in q: the ode give me a strange thing printing all the quarters saying is there…

1.1.2 1.3 *Exercice 3*

Creat A list ofunordered numbers an answer the following questions:
- 1.3.1. How many numbers are there?

[61]:
```
randomNumber = [2,4,1,5,10,7,24,12,34,78,61,12,10,3,5,4,3,24,12,34,78,61,12]
print(len(randomNumber)) #number of list elements, its lenght
```

```
23
```

- 1.3.2. How many times 3 appears?

[77]:
```
counter = 0
for i in range(len(randomNumber)): #here we go throug the list
    if randomNumber[i] == 3: #here we check if the list item is equal to 3
        counter = counter + 1 #if it is equal to 3 we count 1

print(counter) #we print how many we have
```

```
2
```

Correct! we have 2 number 3 in the list!!

- 1.3.3. How many times 3 and 4 appears?

[80]:
```
counter3 = 0
counter4 = 0
for i in range(len(randomNumber)): #here we go throug the list
    if randomNumber[i] == 3: #here we check if the list item is equal to 3
        counter3 = counter3 + 1 #if it is equal to 3 we count 1
    elif randomNumber[i] == 4:
        counter4 += 1

print("we have " + str(counter3) + " number 3")
print("we have " + str(counter4) + " number 4")
#we print how many we have. I cast counter to string, since I just cant print␣
 ↪strings
```

```
we have 2 number 3
we have 2 number 4
```

Correct! we have 2 number 3 and 2 number 4 in the list!!

- 1.3.4. Which is the highest number of the list?

[83]:
```
highestNum = max(randomNumber) #htere is a built-in function to check which is␣
 ↪the highest number of the list
print("the highest number of the list is: " + str(highestNum))
```

```
the highest number of the list is: 78
```

We see that the highest number of the list is 78!!!

- 1.3.5. Which are the three smallest numbers of the list?

  First I want to sort the list, in that case I will do it from smalles to highest number. Then I will extract the 3 first items

[90]:
```
sortedList = randomNumber.copy() #here I copy the list to keep untoched the␣
 ↪first one
sortedList.sort() #I sort the copied list aascending allphanumerically
print(sortedList) #I print the sorted list so I can see it is sorted
print(sortedList[:3]) #I print the range of numbers from the beggining to 3 itme
```

```
[1, 2, 3, 3, 4, 4, 5, 5, 7, 10, 10, 12, 12, 12, 12, 24, 24, 34, 34, 61, 61, 78,
78]
[1, 2, 3]
```

We see that the smallest numbers of the list are, 1, 2 and 3!!!

- 1.3.6. Which is the list range?

  The **range** of a matrix/list can be defined as the **difference between the maximum and minimum among the elements of the matrix**. In NumPy, we have provided with an inbuilt function for this operation i.e. numpy.ptp(). It returns the range of the matrix by calculating maximum-minimum.

  I Will first do it manually and secondly using the built-in function of NumPy, importing it and so on

[93]:
```
highestNum = max(randomNumber)
smallestNum = min(randomNumber)
print("The highest number of the list is: " + str(highestNum) + "\nThe smallest␣
 ↪number of the list is: " + str(smallestNum))

listRange = highestNum - smallestNum
print("The range of the list is: " + str(listRange))
```

```
The highest number of the list is: 78
The smallest number of the list is: 1
The range of the list is: 77
```

Here I calculated the range of the list manually

now I will try with the built-in function Since a list is a matrix of 1 row with len(list) columns, we can calculate already the range of it

[96]:
```
import numpy as np #importing NumPy libary to use the built in function to␣
 ↪clculate the range of a matrix


print(randomNumber)
print("The range of the list is: ", np.ptp(randomNumber))


#Creating an array list from a list
a = np.array(randomNumber)
print ("Matrix A: ", a)
```

```
#Calculating the Range
matrixRange = np.ptp(a)

print("The range of the list is: ", matrixRange)
```

```
[1, 2, 3, 3, 4, 4, 5, 5, 7, 10, 10, 12, 12, 12, 12, 24, 24, 34, 34, 61, 61, 78,
78]
The range of the list is:  77
Matrix A:  [ 1  2  3  3  4  4  5  5  7 10 10 12 12 12 12 24 24 34 34 61 61 78
78]
The range of the list is:  77
```

In the web, I've see that they fists convert it to an array with the same numpy library. Not really sure why...maybe it allocates les memory?

1.1.3 1.4 *Exercice 4*

Creat A dicctionary like this:
compra = { "Pomes" : {"Qty": 5, "€": 0.42}, "Peres" : {"Qty": 3, "€": 0.66} }
- 1.4.1. Add some more fruits
I want to creat a nested dictionary Then I add some more itmes with the function update() or just assigning a new key and giveng the values

[100]:
```
shoppingList = {
  "Appels": {"Qty":5, "€": 0.42},
  "Pears": {"Qty": 3, "€": 0.66}}
print(shoppingList)

#Two whys to add itmes in the dictionary
shoppingList["Bananas"] = {"Qty": 4, "€": 0.34}
shoppingList.update({"Strwaberries":{"Qty":1,"€":3.3}})

print(shoppingList)
```

```
{'Appels': {'Qty': 5, '€': 0.42}, 'Pears': {'Qty': 3, '€': 0.66}}
{'Appels': {'Qty': 5, '€': 0.42}, 'Pears': {'Qty': 3, '€': 0.66}, 'Bananas':
{'Qty': 4, '€': 0.34}, 'Strwaberries': {'Qty': 1, '€': 3.3}}
```

We can see how I created the nested dictionaries, with the fruits as key and its quantity and price as a value.
Then I adde two more fruits in 2 different ways!!

- 1.4.2. How much have cost the pears in total?
  I will get the values of the key pears put them in variables and them add them

[111]:
```
pearsQtyPrice = shoppingList["Pears"]
pearsQty = shoppingList.get("Pears").get("Qty")
pearsPrice = shoppingList.get("Pears").get("€")
print(pearsQtyPrice)
print(pearsQty)
```

```
print(pearsPrice)

totalCoastpears = pearsQty * pearsPrice
print("The cost of the " + str(pearsQty) + " is " + str(totalCoastpears))
```

```
{'Qty': 3, '€': 0.66}
3
0.66
The cost of the 3 is 1.98
```

I've extracted the quantity and price of each pear and calculated the total price! :)

- 1.4.3. How many fruts have we bought in total?

[119]:
```
counter = 0
for x in shoppingList:
    print(x)
    #print(shoppingList[x])
    fruitsNum = shoppingList.get(x).get("Qty")
    counter = counter + fruitsNum
print(counter)
print(shoppingList)
```

```
Appels
Pears
Bananas
Strwaberries
13
{'Appels': {'Qty': 5, '€': 0.42}, 'Pears': {'Qty': 3, '€': 0.66}, 'Bananas':
{'Qty': 4, '€': 0.34}, 'Strwaberries': {'Qty': 1, '€': 3.3}}
```

I've combined searching inside each item of the dictionary key with a foor loop and getting the values of the nested quantity dictionaries. Then adding them to a counter of items :)

- 1.4.4. Which is the most expensive fruit?

  1- I want to creat a list with all the prices

  2- Then I get the maximum value and its index

  3- I get the keys and put them in a list, because if not I cannot acces with the index!

  4- finally I use this index to get the key with the same index

[148]:
```
listOfPrices = [] #I creat an empty list for all prices
for x in shoppingList: #loo throug the list
    print(x)
    prices = shoppingList.get(x).get("€") #I put in the variable each price I␣
 ↪get from each fruit
    listOfPrices.append(prices)#and put this price in the list

print(listOfPrices)#just to check if the list is correct I print it
```

```
maxPriceIndex = listOfPrices.index(max(listOfPrices))
print(maxPriceIndex)
listOfKeys = list(shoppingList.keys()) #I need a list of keys to be able to get␣
 ↪by index
print(listOfKeys[maxPriceIndex])#I get the item of the list I have the index of␣
 ↪maximum price
```

```
Appels
Pears
Bananas
Strwaberries
[0.42, 0.66, 0.34, 3.3]
3
Strwaberries
```

Another way to do so would be with defining a function I could define a function that gets a dictionary creats a list of values anfd a list of keys and then gets the maximum key for a value.

a way of doing it I found it in stackoverflow

definde a method to give you the key with the maxiumum value

def keywithmaxval(d):
a) create a list of the dict's keys and values;
b) return the key with the max value
v=list(d.values())
k=list(d.keys())
return k[v.index(max(v))]

```
[ ]:
```