CHECKPOINT 4:

¿Cuál es la diferencia entre una lista y una tupla en Python?

The main difference between a list and a tuple in python is that, while lists are mutable, tuples are immutable.

In order to choose what is best for you in terms of using a list or a tuple, you should consider if the data structure you are going to work with should be able to be changed or if has to remain unchanged.

The syntax for creating lists is to use brackets [], while tuples use parentheses ().

Here's an example of a list and a tuple:

list = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday']

tuple = ('January', 'February', 'March', 'April')

¿Cuál es el orden de las operaciones?

In Python, when you need to calculate an operation, in order to remember the order in which Python performs the operators, is useful to remember the word PEMDAS, because the letters in PEMDAS are related with the order of the operators as it's shown below:

Please -> Parentheses ()

Excuse -> Exponents **

My -> Multiplication *

Dear -> Division /

Aunt -> Addition +

Sally -> Subtraction -

The order, starting in P and ending in S, is the same one that follows Python when calculates an operation.

Here's an example of an operation, showing the order used in Python to resolve it:

(4 + 6) ** 2 - 12 + 5 * 4

10 ** 2 - 12 + 5 * 4

100 - 12 + 5 * 4

100 - 12 + 20

108

¿Qué es un diccionario Python?

Dictionaries in Python are used to store data values associated to a key (key: value).

The syntax for creating dictionaries is to use curly brackets {}.

Dictionaries, as lists, are mutable so we can add, remove or change items after the dictionary has been created.

The values associated to each key of an item can be any kind of data type. They can be strings, integers, floats, booleans, lists, tuples or even they could be another dictionary.

Here's a dictionary example:

```
dictionary = {
   "name": "Luke",
   "age": 25,
   "height": 1.87,
   "studies": ["business", "law"]
}
```

In order to know the number of elements in a dictionary you can perform the len() function. Here's how it works for the defined dictionary:

```
print(len(dictionary)) # Result: 4
```

¿Cuál es la diferencia entre el método ordenado y la función de ordenación?

While sort() method only can be applied for lists, sorted function admits any iterable.

The difference between both sorting techniques is that, while sort() method sorts a created list, sorted function, when used with lists, creates a new sorted list from an already created list in a new variable.

In both cases you can sort in ascendant or descendant order.

Here's how both sorting techniques works with the following list:

```
list = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday']


list.sort()   #Ascendant order

print(list) # Result: ['Friday', 'Monday', 'Thursday', 'Tuesday', 'Wednesday']


new_list = sorted(list, reverse=True)   #Descendant order

print(new_list) # Result: ['Wednesday', 'Tuesday', 'Thursday', 'Monday', 'Friday']
```

¿Qué es un operador de reasignación?

As commented in the first question, tuples are immutable, same as strings. That means that, once they are created you can't modify them, at least directly.

If you really want to modify the content of a tuple, what you can do is create a new tuple with the same name as the tuple where you want to make a change and make the changes to that new element.

This way you are creating a new tuple by just overwriting the variable name.

Here's an example of how you can do it:

```
tuple = ('January', 'February', 'March', 'April')

tuple += ("May",)

print(tuple) # Result: ('January', 'February', 'March', 'April', 'May')
```