**Practice 7:**

Text, letter

Description automatically generatedDictionaries: instead of an index, we use keys

**Key : value**

A tuple is a list and you can put whatever you want in it i.e. any type. It is immutable. Tuples needs less ram memory than lists.

Separate example:

S = “Hello World”

L = list(s)

For elem in L:

Print(s)

**Dictionaries:**

* In Python, a dictionary is an object that contains a collection of data or items
* Element are made of two parts: a key and a value. The keys are unique and can be any kind of immutable objects. Values can be any kind of object, mutable or immutable.
* To create a dictionary it is necessary to write in curly brackets { } the elements divided by commas (,). Every element is made of a **key** followed by a colon **(:)** and a **value**
* Alternatively you can use the dict function, which creates a new dictionary with no elements
* Dictionaries are mutable objects

**Keys and values: how to create dictionaries**

• The order of items in a dictionary is unpredictable, for this reason we cannot use indexing as we can do in sequences

• To retrieve a value stored in the dictionary we use the corresponding key : **dict\_name[key]**

• In order to add key-value pairs: **dict\_name[key] = value**

myDict = {10: “K”, “A”:0, 2.5:(1,2,3), ”Greetings”:”hello”}

print(myDict)

print(“myDict[‘A’], (“myDict[“A”])

myDict[5] = 500 **: to assing a new value**

myDict[“A”] = 48 🡪 changing existing content. i.e. assigning a new value to A.

print(myDict)

**>>> {10: “K”, “A”:48, 2.5:(1,2,3), ”Greetings”:”hello”,5:500}**

Operations with dictionaries:

• **in** allows to check the presence of a key  
• **del** removes a key-value pair from a dictionary

• **len** returns the number of **key-value pairs** (i.e. the couples of key and values) e.g.len(myDict) -- > **elements are made by couples.**

Text

Description automatically generated

**Traversing dictionaries**

• The keys of a dictionary can be traversed using a for loop:

>>> address\_book = { 'pippo': '555-123456', 'pluto': '555-654321' } >>> for key in address\_book:

print(key)

* Output: pippo pluto: it shows us the **keys**

Elements:

🡪 to show us **elements**:

* For elem in myDict:

Print(myDict[elem])

🡪 special method to print elements (using methods in chart above):

* For elem in myDict.**values**():

Print(elem)

• We can also traverse key-value pairs using the for loop of the list dict.**items**()

>>> **for myKey, myValue in myDict.items():**

**Print(myKey,myValue)**

**Practice 7 worksheet**

1. Def top\_cities(myList, num\_cities = 5):

Import random

Top = {}

For key in range(1,number+1):

Rnd\_num = **random.randrange**(len(myList))

Town = myList[rnd\_num]

Top[key] = town

Return top

Y = Top\_cities(destinations,7)

Print(y)

**Isinstane():** function checks if the object (first argument) is an instance or subclass of classinfo class (second argument).