**DICS:**

[*LINK DOC COMPLETA*](https://drive.google.com/drive/folders/1H6YDuEaS5OVGLyFMEYs0ofsz7mrbR14f)

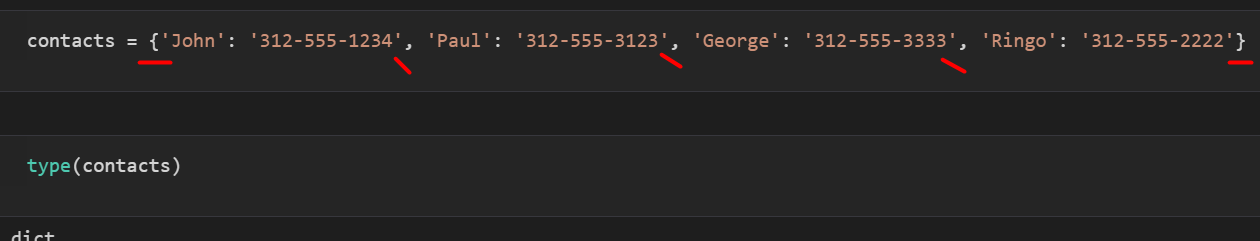
Un **único objeto de clave-valor** (key-value). CARACTERÍSTICAS:

* MUTABLE
* CHANGEABLE: podemos añadir, eliminar o sustituir keys
* ORDERED: desde la versión de python 3.7 los dics están ordenados
* UNDUPLICATED KEYS: no permite duplicados

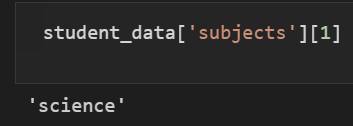
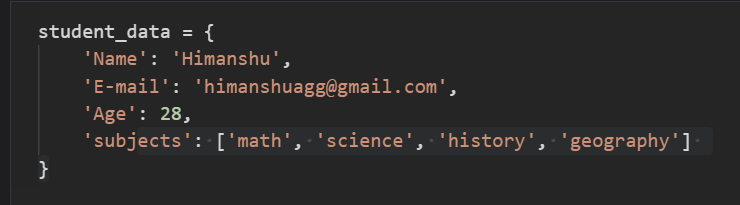
→ Se usan **“curly braces” {}** para definir a DICS

→ Dentro del DIC los **diferentes key-values** están **separados** por **comas “,”.**

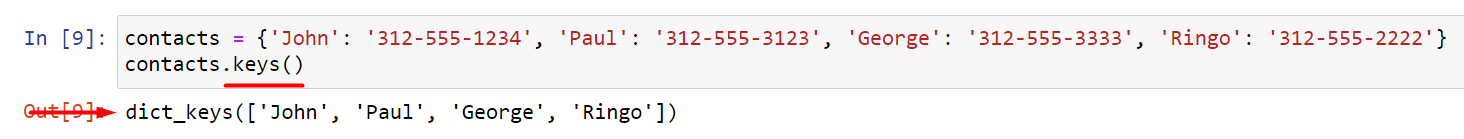
→ El **mismo key-value** está **separado** por **“:”** (colon).

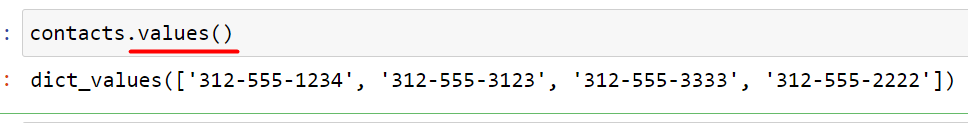
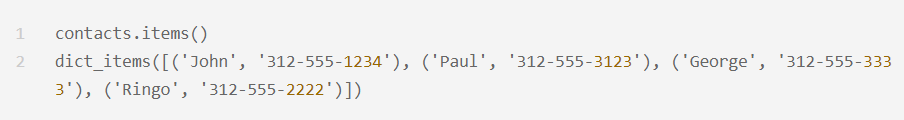
****

→ Un DIC puede **tener diferentes tipos de datos**, y puede **tener propios dics o lists** dentro de él (en este caso tiene un list dentro de él).

****

**extraer valores** (los devuelve en un tuple)



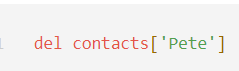
****

METHODS para DICS (al final hay un listado completo con links)

1. Llamar a todos los keys, values o ambos: (si iteramos nos daría individuales)

* nombre dic**.keys() :** nos **devuelve el conjunto de keys** que hay en el DIC
* nombre dic**.values() :** nos **devuelve el conjunto de values** del DIC
* nombre dic**.items() : nos devuelve ambos key y value** como un tuple
* **len(**nombre dic**) : nos devuelve el número de items**

1. Llamar a un key individual para obtener su valor:
   * nombre dic**[nombre del key]**
   * nombre dic**.get(nombre del key)**
2. CHANGE the Dic:

* ELIMINAR registro
  + **del** nombre dic**[key existente]** 
  + nombre dic**.pop(key existente)**
* ELIMINAR el último registro
  + nombre dic**.popitem()**
* ELIMINAR el dic ENTERO
  + nombre dic**.clear() →** lo deja vacío {}
  + **del** nombre dic  **→** lo elimina entero
* INSERTAR registro
  + nombre dic**.key[nuevo key] = nuevo valor**
* REEMPLAZAR registro
  + nombre dic**.key[key existente]** = **nuevo valor**
  + nombre dic**.update({key existente : nuevo valor})**

1. CHECKEAR SI EXISTE un registro específico

* **key existente** **in** nombre dic → devuelve True o False

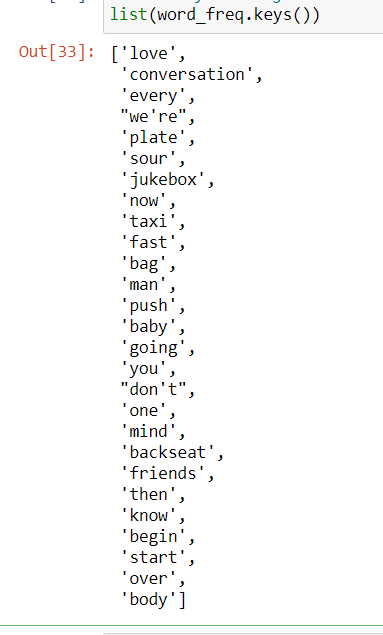
1. COPIAR un Dic en otro:

* **new\_dict =** nombre dic**.copy()**
* **new\_dict = dic(**nombre dic**)**

LISTAR UN CONJUNTO DE KEYS

→ TE DEVUELVE LOS VALORES como un TUPLE, **no te los lista** (para usarlos podemos **LISTAR sus KEY() o sus VALUE()** (NO .items() o **ITERARLOS**)

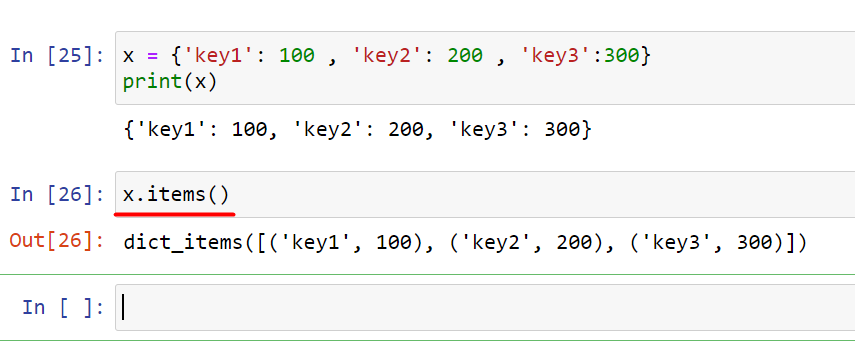
1. Llamamos a los keys y utilizamos **list(**nombre dic**.key())** ,
2. Llamamos a los values y utilizamos **list(**nombre dic**.values())**



[ejercicio alvaro](https://drive.google.com/file/d/1FqzcSVdfwqy7OXijqs49bSc81COWUStp/view?usp=sharing)

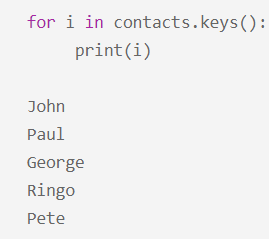
NO SE PUEDE LISTAR EL DIC ENTERO

Utilizamos **.items()**, y nos lo devuelve en un listado como un TUPLE

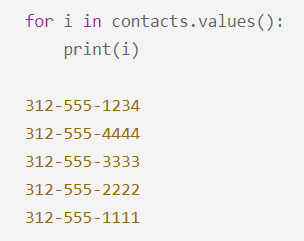


ITERAR

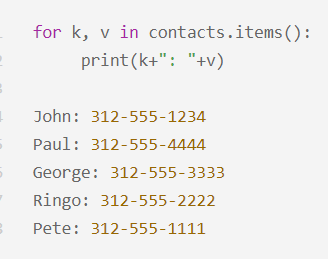
* keys para iterar a través de las claves,



* values()para iterar a través de los valores

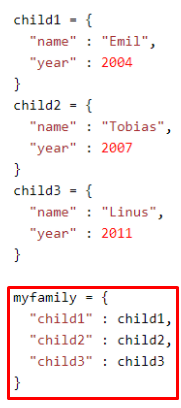
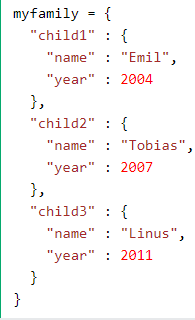


* items()para iterar a través de ambos simultáneamente



Nested Dictionaries

Un diccionario puede contener diccionarios, esto se llama diccionarios anidados.

 **→**  

ORDENAR UN DIC POR SUS KEYS

1º obtenemos sus keys y los convertimos en una Lista

2º ordenamos la lista → .sort()

3º creamos un nuevo dic vacío

4º hacemos un loop para iterar y pasar al dic vacío los nuevos keys ordenados



ORDENAR UN DIC POR SUS VALUES (seguir el orden que marco en rojo)

**1º Ejecutar import operator**

****

**2º usar la función “sorted” con el módulo “operator”** (que nos devolverá un listado de tuples ordenados)

word\_freq = { x:1, y:4, z:2}

word\_freq\_sorted = **sorted(**word\_freq.items(), key=**operator**.itemgetter(**1**)**)**

print(word\_freq\_sorted)

or

**2º usar la función “sorted” con la función “lambda”** (que nos devolverá un listado de tuples ordenados)

word\_freq\_sorted = **sorted(**word\_freq.items(), key=**lambda** x: x[1]**)**

*# equivalent version*

*# sorted\_d = sorted(d.items(), key=lambda (k,v): v)*

or

**2º usar la función “sorted” con list comprehension**

word\_freq\_sorted = **sorted(**(value, key), **for** (key,value) **in** word\_freq.items()**)**

**2º crear un dic vacío, iterar el tuple ordenado y añadir sus elementos al dic**

**word\_freq2 = {}**

**for e in word\_freq\_sorted:**

**word\_freq2[e[0]]= e[1]**

**print(word\_freq2)**

**LISTADO DE METHODS COMPLETOS:**

| **Method** | **Description** |
| --- | --- |
| [clear()](https://www.w3schools.com/python/ref_dictionary_clear.asp) | Removes all the elements from the dictionary |
| [copy()](https://www.w3schools.com/python/ref_dictionary_copy.asp) | Returns a copy of the dictionary |
| [fromkeys()](https://www.w3schools.com/python/ref_dictionary_fromkeys.asp) | Returns a dictionary with the specified keys and value |
| [get()](https://www.w3schools.com/python/ref_dictionary_get.asp) | Returns the value of the specified key |
| [items()](https://www.w3schools.com/python/ref_dictionary_items.asp) | Returns a list containing a tuple for each key value pair |
| [keys()](https://www.w3schools.com/python/ref_dictionary_keys.asp) | Returns a list containing the dictionary's keys |
| [pop()](https://www.w3schools.com/python/ref_dictionary_pop.asp) | Removes the element with the specified key |
| [popitem()](https://www.w3schools.com/python/ref_dictionary_popitem.asp) | Removes the last inserted key-value pair |
| [setdefault()](https://www.w3schools.com/python/ref_dictionary_setdefault.asp) | Returns the value of the specified key. If the key does not exist: insert the key, with the specified value |
| [update()](https://www.w3schools.com/python/ref_dictionary_update.asp) | Updates the dictionary with the specified key-value pairs |
| [values()](https://www.w3schools.com/python/ref_dictionary_values.asp) | Returns a list of all the values in the dictionary |