

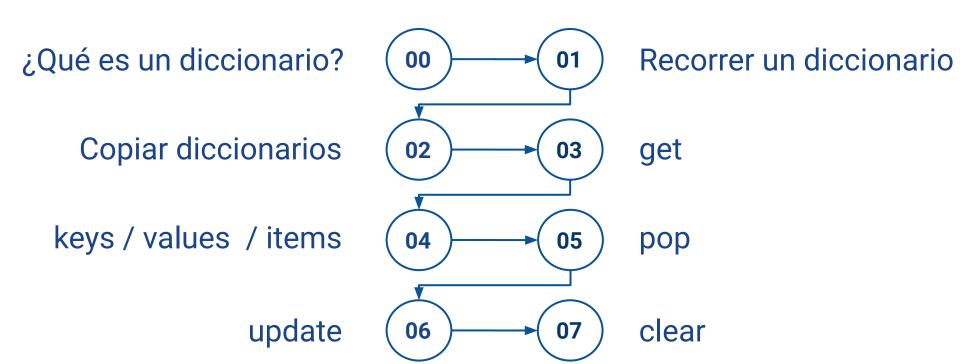


# Diccionarios Avanzado

Programación y Laboratorio I



#### **Diccionarios Avanzado**





#### ¿Qué es un Diccionario?

Un diccionario en Python funciona de manera similar al diccionario en el mundo real.

Las claves de un diccionario deben ser únicas y de un tipo de datos inmutable como cadenas, enteros y tuplas, pero los valores pueden repetirse y ser de cualquier tipo.



#### ¿Qué es un Diccionario?

Un diccionario es una colección de elementos, donde cada uno tiene una clave (key) y un valor (value).

```
diccionario = {'name' : 'Marty', 'edad' : 18}
print(diccionario['nombre']) # Marty
print(diccionario['edad']) # 18
```



#### Diccionario anidado

La definición de diccionario en python habilita la posibilidad de poder tener como valor de un diccionario cualquier objeto, inclusive otro diccionario.

```
diccionario = {'name' : 'Marty','addr': {'number': 9303,'street':'Lyon Drive'}}
print(diccionario['addr']['number'],diccionario['addr']['street'])
# 9303 Lyon Drive
```



#### Recorrer diccionarios

Al recorrer un diccionario utilizando un **for**, el valor devuelto son las claves del diccionario.

```
diccionario = {'name' : 'Marty', 'edad' : 18}
for clave in diccionario:
    print(clave, diccionario[clave])
# name Marty
# edad 18
```



### Copiar diccionarios

No es posible copiar un diccionario simplemente escribiendo dict\_2 = dict\_1, ya que en ese caso dict\_2 solo será una referencia a dict\_1.

#### Existen dos tipos de copias posibles:

- Superficial (shallow copy)
- Profunda (deep copy)



### **Copiar diccionarios**

Superficial (shallow copy), solamente se copian las referencias a los elementos contenidos en el objeto.

```
diccionario = {'name' : ['Marty','Mcfly'], 'edad' : 18}
diccionario_copia = diccionario.copy()
diccionario['name'][0] = "MARTY"
diccionario['edad'] = 19
print(diccionario_copia)
# {'name': ['MARTY', 'Mcfly'], 'edad': 18}
```



### **Copiar diccionarios**

Deep copy, si el objeto contiene subobjetos estos se copian recursivamente.

```
from copy import deepcopy
diccionario = {'name' : ['Marty','Mcfly'], 'edad' : 18}
diccionario_copia = deepcopy(diccionario)
diccionario['name'][0] = "MARTY"
diccionario['edad'] = 19
print(diccionario_copia)
# {'name': ['Marty', 'Mcfly'], 'edad': 18}
```



# get()

El método get() permite consultar el valor para una clave determinada. El segundo parámetro es el valor a devolver si no se encuentra la clave (es opcional).

```
diccionario = {'name' : 'Marty', 'edad' : 18}
print(diccionario.get('name','NO NAME'))# Marty
print(diccionario.get('nombre','NO NAME'))# NO NAME
```



# keys()

El método keys() devuelve una lista con todas las claves del diccionario.

```
diccionario = {'name' : 'Marty', 'edad' : 18}
print(list(diccionario.keys()))
# ['name', 'edad']
```



# values()

El método values() devuelve una lista con todas los valores del diccionario.

```
diccionario = {'name' : 'Marty', 'edad' : 18}
print(list(diccionario.values()))
# ['Marty', 18]
```



### items()

El método items() devuelve una lista con las claves y valores del diccionario. Si se convierte en lista se puede acceder utilizando el índice.

```
diccionario = {'name' : 'Marty', 'edad' : 18}
print(list(diccionario.items()))
# [('name', 'Marty'), ('edad', 18)]
```



# pop()

El método pop() busca y elimina la key que se pasa como parámetro y devuelve su valor asociado. Dará un error si se intenta eliminar una key que no existe.

```
diccionario = {'name' : 'Marty', 'edad' : 18}
diccionario.pop('edad')
print(diccionario)
# {'name': 'Marty'}
```



# pop()

También se puede pasar un segundo parámetro que es el valor a devolver si la key no se ha encontrado. En este caso si no se encuentra no habría error.

```
diccionario = {'name' : 'Marty', 'edad' : 18}
valor_eliminado = diccionario.pop('year','none')
print(valor_eliminado)
# none
```



### update()

El método update() se llama sobre un diccionario y tiene como entrada otro diccionario. Los value son actualizados y si alguna key del nuevo diccionario no esta, es añadida.

```
diccionario = {'name' : 'Marty', 'edad' : 18}
diccionario.update({'year' : '1973'})
print(diccionario)
# {'name': 'Marty', 'edad': 18, 'year': '1973'}
```



# clear()

El método clear() elimina todo el contenido del diccionario.

```
diccionario = {'name' : 'Marty', 'edad' : 18}
diccionario.clear()
print(diccionario)# {}
```