

COL·LECCIONS

LLISTES, TUPLES, DICCIONARIS



M^a Belén Tortosa Pedrón



DICCIONARIS

- Un diccionari és una col·lecció no-ordenada de valors que són accedits a través d'una clau. És a dir, en lloc d'accedir a la informació mitjançant l'índex numèric, com és el cas de les llistes i tuples, és possible accedir als valors a través de les seves claus, que poden ser de diversos tipus.
- Les claus són úniques dins d'un diccionari, és a dir que no pot haver-hi un diccionari que tingui dues vegades la mateixa clau, si s'assigna un valor a una clau ja existent, es reemplaça el valor anterior.
- No hi ha una forma directa d'accedir a una clau a través del seu valor, i res impedeix que un mateix valor es trobi assignat a diferents claus
- La informació emmagatzemada en els diccionaris, no té un ordre particular. Ni per clau ni per valor, ni tampoc per l'ordre en què han estat agregats al diccionari.
- Qualsevol variable de tipus immutable, pot ser clau d'un diccionari: cadenes, enters, tuples (amb valors immutables en els seus membres), etc. No hi ha restriccions per als valors que el diccionari pot contenir, qualsevol tipus pot ser el valor: llistes, cadenes, tuples, altres diccionaris, objectes, etc.

En altres llenguatges, als diccionaris se'ls diu arrays associatius, matrius associatives, o també taules de hash.



DECLARACIÓ DE DICCIONARIS

- De la mateixa manera que amb llistes, és possible definir un diccionari directament amb els membres que contindrà, o bé inicialitzar el diccionari buit i després afegir els valors de a un o d'a molts.
- Per definir-juntament amb els membres que contindrà, es tanca el llistat de valors entre claus, les parelles de clau i valor se separen amb comes, i la clau i el valor se separen amb:

```
punt= {'x': 2, 'y': 1, 'z': 4}
```

- Per declarar buit i després ingressar els valors, es el declara com un parell de claus sense res al mig, i després s'assignen valors directament als índexs.

Diccionari buit

```
b=dict()  
c={}
```

```
materies = {}  
materies["Dilluns"] = ['M3', 'M2']  
materies["Dimarts"] = ['M5', 'M3']  
materies["Dimecres"] = ['M3', 'M11']  
materies["Dijous"] = []  
materies["Divendres"] = ['M3']  
print(materies)
```



ACCEDIR A UN ELEMENT

- Podeu utilitzar un cicle **for** en un diccionari per iterar per les seves claus, de la següent forma:

```
# Un simple diccionari  
d = {"foo" : "bar"}  
for key in d:  
    print d[key] #la nostra "bar"
```

- Els diccionaris estan desordenats, es a dir, que cada vegada que facis un cicle través d'un diccionari, vas a passar per cada clau, però no vas a tenir la seguretat de poder veure-les en un ordre particular.



DICCIONARIS: DEFINICIÓ D'UN DICCIONARI

16_exemple_diccionari.py x

```
1  dades_basiques = {
2      "nom": "Joannes Chrysostomus Wolfgang Amadeus ",
3      "cognom": "Mozart",
4      "NIF": "123456789-D",
5      "data_naixement": "27011756",
6      "lloc_naixement": "Salzburg, Viena",
7      "nacionalitat": "austriac",
8      "estat_civil": "casat"
9  }
10
11 print ('\nDetall del diccionari')
12 print ('=====')
13
14 print ('\nClaus del diccionari:', dades_basiques.keys())
15 print ('\nValors del diccionari:', dades_basiques.values())
16 print ('\nElements del diccionari:', dades_basiques.items())
17
18
19 # Exemple practic dels diccionaris
20 print ('\nInscripcio de Curs')
21 print ('=====')
22
23 print ('\nDades de participant')
24 print ('-----')
25
26 print ('NIF:', dades_basiques['NIF'])
27 print ('Nom: ' + dades_basiques['nom'] + dades_basiques['cognom'])
28
```

tall del diccionari

=====

claus del diccionari: dict_keys(['NIF', 'data_naixement', 'cognom', 'lloc_naixement', 'nom', 'estat_civil', 'nacionalitat'])

valors del diccionari: dict_values(['123456789-D', '27011756', 'Mozart', 'Salzburg, Viena', 'Joannes Chrysostomus Wolfgang Amadeus ', 'casat', 'austriac'])

elements del diccionari: dict_items([('NIF', '123456789-D'), ('data_naixement', '27011756'), ('cognom', 'Mozart'), ('lloc_naixement', 'Salzburg, Viena'), ('nom', 'Joannes Chrysostomus Wolfgang Amadeus '), ('estat_civil', 'casat'), ('nacionalitat', 'austriac')])

inscripcio de Curs

=====

dades de participant

NIF: 123456789-D

Nom: Joannes Chrysostomus Wolfgang Amadeus Mozart



DICCIONARIS: MODIFICACIÓ D'UN DICCIONARI

- No pot haver claus duplicades en un diccionari. L'assignació d'un valor a una clau existent esborrarà el valor anterior:

```
16_exemple_diccionari_modificacio.py
1 # Exemple modificació d'un diccionari
2 d={"server":"mpilgrim", "database":"master"} #definició d'un diccionari
3 print(d)
4 print(d["server"])
5 d["server"]="Linux"
6 print(d)
7 print(d["server"])
8
9
10
```

```
C:\WINDOWS\system32\cmd.exe
{'database': 'master', 'server': 'mpilgrim'}
mpilgrim
{'database': 'master', 'server': 'Linux'}
Linux
```

- Es pot afegir nous parells clau-valor en qualsevol moment:

```
16_exemple_diccionari_modificacio_v2.py
1 # Exemple agregar un parell clau-valor al diccionari
2
3 d={"server":"mpilgrim", "database":"master"} #definició d'un diccionari
4 d['ui']="so"
5 print(d)
6
7
8
9
```

```
C:\WINDOWS\system32\cmd.exe
{'database': 'master', 'server': 'mpilgrim', 'ui': 'so'}
```

- Es diccionaris no tenen concepte algú d'ordre entre els seus elements. És incorrecte dir que els elements estan “desordenats”, simplement no hi ha ordre.

DICCIONARIS: BARREJA DE TIPUS DE DADES EN UN DICCIONARI



- Els diccionaris poden emmagatzemar qualsevol tipus de dades: cadenes, enters, decimals, objectes o inclús altres diccionaris. En un mateix diccionari, els elements no tenen perquè ser tots del mateix tipus, es poden barrejar.
- És poden barrejar i emparellar diferents tipus de claus en un diccionari.
- Les claus d'un diccionari poden ser cadenes, enters i alguns tipus més (més endavant).

16_exemple_diccionari_modificacio_v3.py

```
1  # Exemple barreja de tipus de dades en un diccionari
2
3  d={"server":"mpilgrim", "database":"master"} #definició d'un diccionari
4  d['ui']="so"
5  print(d)
6  d[42]='douglas'
7  d["retry"]=3
8
9  print(d)
10
```

C:\> Selecció C:\WINDOWS\system32\cmd.exe

```
{'ui': 'so', 'database': 'master', 'server': 'mpilgrim'}
{'ui': 'so', 'database': 'master', 'retry': 3, 42: 'douglas', 'server': 'mpilgrim'}
```

DICCIONARIS: ELIMINACIÓ D'ELEMENTS EN UN DICCIONARI



- La funció **del** permet esborrar elements individuals d'un diccionari per la seva clau.
- La funció **clear** esborra tots els elements d'un diccionari. **{ }** indica un diccionari sense elements

16_exemple_diccionari Eliminació.py

```
1 # Exemple barreja de tipus de dades en un diccionari
2
3 d={'ui':'so','server':"mpilgrim", "database":"master", 42:"douglas", "retry":3} #definició d'un diccionari
4
5 del d[42]
6 print(d)
7
8 d.clear()
9 print(d)
10
```

C:\WINDOWS\system32\cmd.exe

```
{'retry': 3, 'ui': 'so', 'server': 'mpilgrim', 'database': 'master'}
{}
```




■ copy()

 C:\WINDOWS\system32\cmd.exe

```
{'color': 'violeta', 'precio': 74.25, 'talla': 'L'}  
{'color': 'violeta', 'precio': 74.25, 'talla': 'L'}  
{}  
{'color': 'violeta', 'precio': 74.25, 'talla': 'L'}  
{'color': 'violeta', 'precio': 74.25, 'talla': 'L'}  
{}  
{}
```

```
I:\Curs2016-2017\M3\AWS1\UF1\codi>
```

CREAR UN NOU DICCIONARI AMB LES CLAUS DE SEQÜÈNCIA



■ *fromkeys(seqüència[,valor per defecte])*

```
sequencia = ["color", "talla", "marca"]  
dic1 = dict.fromkeys(sequencia)  
print(dic1)  
  
dic2 = dict.fromkeys(sequencia, 'valor x defecte')  
print(dic2)
```

C:\WINDOWS\system32\cmd.exe

```
{'color': None, 'talla': None, 'marca': None}  
{'color': 'valor x defecte', 'talla': 'valor x defecte', 'marca': 'valor x defecte'}
```

I:\Curs2016-2017\M3\AWS1\UF1\codi>_



CONCATENAR UN DICCIONARI

■ *update(diccionari)*

```
dic1 = {"color": "groc", "precio": 450}  
dic2 = {"talla": "XL", "marca": "Gucci"}  
dic1.update(dic2)  
print(dic1)
```

```
C:\WINDOWS\system32\cmd.exe
```

```
{'precio': 450, 'talla': 'XL', 'marca': 'Gucci', 'color': 'groc'}
```

```
I:\Curs2016-2017\M3\AWS1\UF1\codi>
```



ESTABLIR UNA CLAU I VALOR PER DEFECTE

■ ***setdefault("clave" [, None|valor_per_defecte])***

```
dic = {"color": "rosa", "marca": "Adidas"}  
clau= dic.setdefault("talla", "U")  
print(clau)  
print(dic)
```

```
dic2 = dic.copy()  
print(dic2)
```

```
clau = dic2.setdefault("estampat")  
print(clau)  
print(dic2)
```

```
clau = dic2.setdefault("marca", "Adidas")  
print(clau)  
print(dic2)
```

C:\WINDOWS\system32\cmd.exe

```
U  
{'color': 'rosa', 'marca': 'Adidas', 'talla': 'U'}  
{'color': 'rosa', 'marca': 'Adidas', 'talla': 'U'}  
None  
{'estampat': None, 'color': 'rosa', 'marca': 'Adidas', 'talla': 'U'}  
Adidas  
{'estampat': None, 'color': 'rosa', 'marca': 'Adidas', 'talla': 'U'}  
I:\Curs2016-2017\M3\AWS1\UF1\codi>
```



OBTENIR EL VALOR D'UNA CLAU

■ *get(clave[, "valor x defecte si la clau no existeix"])*

```
dic = {"color": "rosa", "marca": "Adidas"}  
print(dic.get("color"))
```

```
dic.get("stock")  
print(dic.get("stock", "sin stock"))
```

```
C:\WINDOWS\system32\cmd.exe
```

```
rosa
```

```
sin stock
```

```
I:\Curs2016-2017\M3\AWS1\UF1\codi>_
```



OBTENIR LES CLAUS D'UN DICCIONARI

■ `keys()`

```
dic = {"color": "rosa", "marca": "Adidas", 'talla': 'U'}
```

```
claus=dic.keys()  
print(claus)
```

```
C:\WINDOWS\system32\cmd.exe  
dict_keys(['marca', 'color', 'talla'])  
I:\Curs2016-2017\M3\AWS1\UF1\codi>_
```



OBTENIR ELS VALORS D'UN DICCIONARI

■ *values()*

```
dic = {"color": "rosa", "marca": "Adidas", 'talla': 'U'}
```

```
valors=dic.values()  
print(valors)
```

```
C:\> Seleccionar C:\WINDOWS\system32\cmd.exe
```

```
dict_values(['U', 'rosa', 'Adidas'])
```

```
I:\Curs2016-2017\M3\AWS1\UF1\codi>
```

OBTENIR LA QUANTITAT D'ELEMENTS D'UN DICCIONARI



■ *len()*

```
dic = {"color": "rosa", "marca": "Adidas", 'talla': 'U'}  
  
numElements=len(dic)  
  
print("Elements:",numElements)
```

```
C:\WINDOWS\system32\cmd.exe
```

```
Elements: 3
```

```
I:\Curs2016-2017\M3\AWS1\UF1\codi>
```




OBTENIR LLISTES A PARTIR DE DICCIONARIS

■ *items()*

- És possible, també, obtenir els valors com llistes a on el primer element és la clau i el segon el valor.

```
horari = {}  
horari["dilluns"] = ['M1', 'M3']  
horari["dimarts"] = ['M4']  
horari["dimecres"] = ['M3', 'M2']  
horari["dijous"] = []  
horari["divendres"] = ['M1']  
  
for dia, codigos in horari.items():  
    print (dia, ":", codigos)
```

```
C:\WINDOWS\system32\cmd.exe
```

```
dimecres : ['M3', 'M2']  
dimarts : ['M4']  
dilluns : ['M1', 'M3']  
dijous : []  
divendres : ['M1']
```



CARACTERÍSTIQUES D'UN DICCIONARI

- No pot haver-hi claus duplicades en un diccionari. L'assignació d'un valor a una clau existent esborrarà el valor anterior.
- Es poden afegir nous parells clau-valor en qualsevol moment.
- Aquesta sintaxi és idèntica a la de la modificació de valors existents.

```
dic={'Nom':'Pepito','Cognom':'Garcia','Edat':25,'Nom':'Anna'}  
  
print(dic)
```

```
C:\WINDOWS\system32\cmd.exe
```

```
{'Cognom': 'Garcia', 'Edat': 25, 'Nom': 'Anna'}
```

```
I:\Curs2016-2017\M3\AWS1\UF1\codi>
```



CARACTERÍSTIQUES D'UN DICCIONARI

- Els diccionaris no serveixen només per emmagatzemar cadenes. Els valors d'un diccionari poden ser de qualsevol tipus, incloses cadenes, enters, objectes o fins i tot altres diccionaris. I en un mateix diccionari, els valors no han de ser tots del mateix tipus: es poden barrejar i aparellar com sigui necessari.
- Les claus d'un diccionari estan més restringides, però poden ser cadenes, enters, i tuples.



BUSCAR ELEMENT EN UN DICCIONARI

- L'algoritme per buscar un element en un diccionari és molt diferent que el que s'utilitza per buscar en llistes.
- Per buscar a les llistes, s'utilitza un algoritme de comparació que triga cada vegada més a mesura que la llista es fa més llarga.
- En canvi, per buscar en diccionaris s'utilitza un algoritme anomenat **hash**, que es basa en realitzar un càlcul numèric sobre la clau de l'element, i té una propietat molt interessant: sense importar el número d'elements que tingui el diccionari, el temps de recerca és sempre aproximadament igual.
- Aquest algoritme de hash és també la raó per la qual les claus dels diccionaris han de ser immutables, ja que l'operació feta sobre les claus ha de donar sempre el mateix resultat, i si s'utilitzés una variable mutable això no seria possible.
- No és possible obtenir porcions d'un diccionari usant [:], ja que en no tenir un ordre determinat pels elements, no seria possible agafar-los en ordre.



DIFERENCIA ENTRE MINÚSCULES I MAJÚSCULES

- A l'assignar un valor a una clau existent, simplement es substitueix el valor antic amb el nou.

```
16_exemple_diccionari_minuscules.py
1  # Exemple diferencia entre majúscules minúscules en un diccionari
2
3  d={'clau1':"Nuevo valor"} #definició d'un diccionari
4  print(d)
5
6  d ['Clau1']="Altre valor"
7  print(d)
8
9
10
```

C:\WINDOWS\system32\cmd.exe

```
{'clau1': 'Nuevo valor'}
{'Clau1': 'Altre valor', 'clau1': 'Nuevo valor'}
```

- En aquest exemple, no s'assigna un valor a una clau existent en el diccionari, ja que com Python diferencia entre majúscules i minúscules, considera que **clau1** i **Clau1** són claus diferents, per tant afegeix la clau **Clau1** al diccionari.



RESUM MÈTODES D'UN DICCIONARI

- # Declaració d'un diccionari
 - ❖ **diccionario = dict()**
- # Retorna el numero d'elements que té el diccionari
 - ❖ **len(dict)**
- # Retorna el valor de l'element amb clau **key**. Sino en cas contrari, retorna **default**
 - ❖ **dict.get(key, default=None)**
- # Insereix un element en el diccionari clau:valor. Si la clau existeix, no la insereix
 - ❖ **dict.setdefault(key, default=None)**
- # Eliminem l'element del diccionari amb clau **key**
 - **dict.pop('key', None)**
- # Retorna la copia d'un diccionari
 - **dict2 = dict.copy()**
 - **dict.copy()**
- # Crea un nou diccionari posant com a claus les que hi ha en la llista i els valors per defecte si se'ls passa
 - **dict.fromkeys(list, defaultValue)**
- # Retorna true si existeix la clau. En cas contrari, retorna **false**
 - **dict.has_key(key)**
- # Retorna una llista de tuples formades pels parells clau:valor
 - **dict.items()**
- # Afegeix els elements d'un diccionari a altre
 - **dict.update(dict2)**