

# TREBALLAR AMB DICCIONARIS

Els diccionaris són una col·lecció de dades que permeten accedir a la informació que contenen mitjançant una clau. Aquest tipus d'estructura de dades també es coneix per clau-valor. A continuació, es mostra un exemple per veure l'estructura i com es pot escriure un diccionari.

```
notes_dict = {'Joan': 4.85, 'Miquel': 7.80, 'Meritxell': 8.15, 'Laura': 4.75}
```

També podem crear diccionaris amb el mètode `dict`. Mostrem un exemple:

```
notes_dict = dict(Joan= 4.85, Miquel= 7.80, Meritxell= 8.15, Laura= 4.75)
```

Aquí tenim, doncs, el nom de la variable `notes_dict` on es guarda el diccionari. Les claus són 'Joan', 'Miquel', 'Meritxell' i 'Laura'. Els valors són 4.85, 7.80, 8.15 i 4.75. També veiem que, per obrir i tancar un diccionari, es fan servir els caràcters `{ }`. Les llistes, en canvi, recorda que fan servir els claudàtors `[]`.

Si volem accedir, per exemple, a la nota de la Meritxell i guardar-la en una variable, ho farem així:

```
nota_meritxell = notes_dict['Meritxell']
```

**Important:** els diccionaris accepten qualsevol tipus de dada com a valors. Per a les claus, normalment es fan servir *strings* o números enters.

També podem accedir al valor d'una clau determinada fent ús de la funció `get()`. Es faria així:

```
notes_dict.get('Laura')
```

Aquesta línia de codi ens permet accedir al valor 4.75.

Si volem afegir una nova parella clau-valor, només haurem d'escriure aquesta sentència:

```
notes_dict['Eduard'] = 6.80
```

Ara, el diccionari té una nova clau anomenada 'Eduard' amb un valor de 6.80.

Per saber el número de clau-valor que conté un diccionari, ho podem fer de la mateixa manera que amb les llistes: hem de cridar la funció *len()* i posar a dins dels parèntesis el nom de la variable que conté el diccionari:

```
print(len(notes_dict))
```

En aquest cas, ens apareixerà per pantalla el número de registres clau-valor perquè hem cridat la funció *len()* dins de la funció *print()*.

Si volem, per exemple, extreure la col·lecció de claus en forma de llista, ho podem fer amb dos passos:

```
claus = notes_dict.keys()  
claus_llista = list(claus)
```

La funció *keys()* està associada als diccionaris. Aquesta funció genera una col·lecció que conté les claus del diccionari, la qual es pot convertir en una llista fent ús de la funció *list()*. Aquí estem fent un canvi de tipus d'objecte de la mateixa manera que ho hem fet abans entre dades del tipus *int* i *float*.

També es pot generar una llista amb la col·lecció de valors. Ho farem de la mateixa manera que amb les claus, però en lloc de fer-ho cridant la funció `keys()`, cridarem la funció `values()`:

```
valors = notes_dict.values()
valors_llista = list(valors)
```

Això ens pot servir, per exemple, per saber si una clau es troba en un diccionari o no. Per saber si hi és o no, ho farem així:

```
print('Kim' in claus_llista)
```

Aquest codi ens imprimirà *True* (si està present) o *False* (si no hi és).

I, si per alguna raó ens interessa generar una llista amb les parelles clau-valor, ho podem fer cridant la funció `items()`. El codi s'escriuria així:

```
parelles = notes_dict.items()
parelles_llista = list(parelles)
```

Per tal d'eliminar una parella clau-valor del diccionari, ho podem fer a través d'aquesta sentència:

```
del notes_dict['Joan']
```

Ara, la parella clau-valor composta per '*Joan*': 4.85 ha quedat eliminada.

Un diccionari pot tenir una clau que el seu valor sigui un altre diccionari. Recorda que els valors poden ser qualsevol tipus d'objecte i els diccionaris també són un tipus d'objecte. Podem escriure aquest exemple per veure-ho més clarament:

```
nested_dict = {'a': 1, 'b': 2, 'c': {'c1': 3, 'c2': 4}, 'd': 5}
```

Una llista també pot ser un valor d'una clau d'un diccionari. Aquest seria un exemple:

```
nested_dict = {'a': 1, 'b': 2, 'c': [3,4,5,6], 'd': 7}
```

Si volem accedir i imprimir el valor a la posició 2 de la llista, que és el valor de la clau '*c*', es podria fer així:

```
print(nested_dict['c'][2])
```

Primer, accedim a la llista mitjançant la clau '*c*' i, després, accedim a la posició de la llista que ens interressi escrivint l'índex que correspongui.

Els diccionaris són molt útils i encara ens queden moltes coses per descobrir de Python. Seguim!

# Descobreix tot el que Barcelona Activa pot fer per a tu



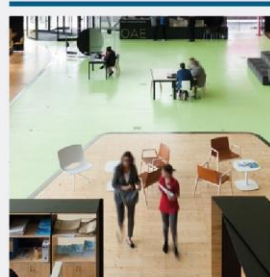
Acompanyament durant tot el procés de recerca de feina

[barcelonactiva.cat/treball](http://barcelonactiva.cat/treball)



Suport per posar en marxa la teva idea de negoci

[barcelonactiva.cat/emprenedoria](http://barcelonactiva.cat/emprenedoria)



Serveis a les empreses i iniciatives socioempresarials

[barcelonactiva.cat/empreses](http://barcelonactiva.cat/empreses)

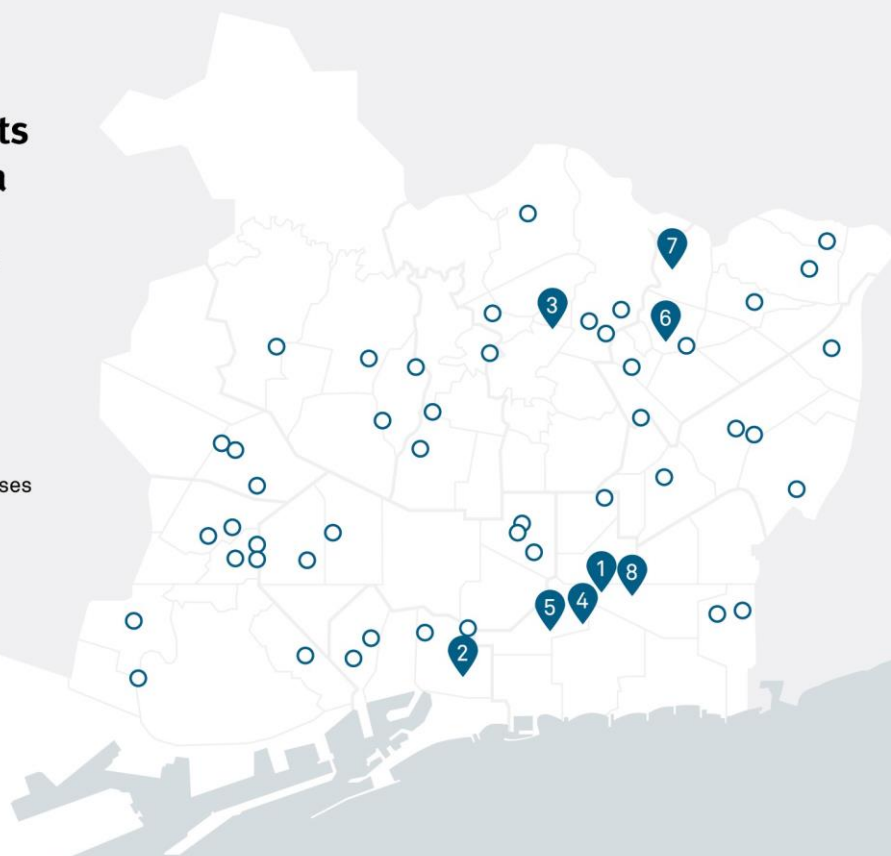


Formació tecnològica i gratuïta per a la ciutadania

[barcelonactiva.cat/cibernarium](http://barcelonactiva.cat/cibernarium)

## Xarxa d'equipaments de Barcelona Activa

- 1 Seu Central Barcelona Activa  
Porta 22  
Centre per a la Iniciativa  
Emprenedora Glòries  
Incubadora Glòries
- 2 Convent de Sant Agustí
- 3 Ca n'Andalet
- 4 Oficina d'Atenció a les Empreses  
Cibernàrium  
Incubadora MediaTIC
- 5 Incubadora Almogàvers
- 6 Parc Tecnològic
- 7 Nou Barris Activa
- 8 innoBA
- Punts d'atenció a la ciutat



© Barcelona Activa  
Darrera actualització 2022

Cofinançat per:



**UNIÓ EUROPEA**  
Fons Europeu de Desenvolupament Regional

**Segueix-nos a les xarxes socials:**



[barcelonactiva.cat/cibernarium](http://barcelonactiva.cat/cibernarium)



[barcelonactiva](#)



[barcelonactiva](#)



[company/barcelona-activa](#)