



Colecciones Python: Diccionarios

Contenidos

- [Introducción](#)
- [Acceso, modificación, añadir, eliminar](#)
- [Iterar](#)

Introducción

[al índice](#)

En este tipo de colecciones no solo podemos agrupar datos, sino que tenemos la posibilidad de establecer una relación entre los mismos, con la forma **clave: valor**.

La sintaxis es mediante llaves `{ }`

```
nombre_diccionario = {clave1: valor1, clave2: valor2, clave3:  
valor3}
```

Si veíamos que las listas son elementos mutables y ordenadas, y las tuplas inmutables y ordenadas, los diccionarios son:

- **Mutables:** podemos modificar sus elementos
- **Desordenado:** la potencia de acceso de los diccionarios es su formato clave-valor, no su orden. Aun así, el diccionario conserva el orden de los elementos según los vamos insertando.

Vamos a declarar un diccionario en el que definiremos elementos de un coche

```
In [4]: coche_dict = {
    "color": "Rojo",
    "marca": "Tesla",
    "modelo": "Cybertruck",
    "peso": 3500,
    "cambio automatico": True
}
coche_dict
```

```
Out[4]: {'color': 'Rojo',
'marca': 'Tesla',
'modelo': 'Cybertruck',
'peso': 3500,
'cambio automatico': True}
```

Se puede almacenar todo tipo de datos: numero, cadena, booleano, listas, tuplas, diccionarios...

Cuando declaremos el diccionario, lo podemos hacer en una sola linea `coche = {"color": "Gris", "marca": "...}"`, aunque se recomienda por sencillez a la hora de leer el código, realizar esta acción en varias líneas. Si escribimos el primer elemento y presionamos enter, Jupyter introduce automáticamente la tabulación.

Acceso, modificación, añadir, eliminar

[al índice](#)

```
In [14]: # Acceso
print(coche_dict["color"])
print(coche_dict["peso"])

# print(coche_dict["puertas"])

coche_dict["color"] = ["rojo", "gris"]
print(coche_dict)

# Añadir elemento:

coche_dict["puertas"] = 6
print(coche_dict["puertas"])

# Eliminar elemento:

del coche_dict["puertas"]
print(coche_dict)

# Conocer las claves de diccionario:
print(coche_dict.keys())
```

```
# Conocer Los valores del diccionario:
print(coche_dict.values())
```

```
['rojo', 'gris']
3500
{'color': ['rojo', 'gris'], 'marca': 'Tesla', 'modelo': 'Cybertruck', 'p
eso': 3500, 'cambio automatico': True}
6
{'color': ['rojo', 'gris'], 'marca': 'Tesla', 'modelo': 'Cybertruck', 'p
eso': 3500, 'cambio automatico': True}
dict_keys(['color', 'marca', 'modelo', 'peso', 'cambio automatico'])
dict_values(['rojo', 'gris', 'Tesla', 'Cybertruck', 3500, True])
```

Para más detalle de los métodos de los diccionario puedes [consultar la documentación](#)

```
In [17]: # Podemos empezar tambien mediante un diccionario vacío
tienda = {}
tienda["direccion"] = "Montesa 12"
print(tienda)
```

```
{'direccion': 'Montesa 12'}
```

ERRORES en diccionarios

Si hay una clave que no existe en el diccionario, saltará un error.

En este caso, el error es bastante intuitivo: `KeyError`, no encuentra la clave

```
In [18]: coche_dict["puertas"]
```

```
-----
---
KeyError                                Traceback (most recent call la
st)
Cell In[18], line 1
----> 1 coche_dict["puertas"]

KeyError: 'puertas'
```

¿Cómo solventar esto? Siempre tienes la opción de usar el bloque `try/except`, pero en este caso, los diccionarios tienen una solución más elegante: `mi_diccionario.get(clave, valor si no existe)`

```
In [20]: numero_de_puertas = coche_dict.get("puertas",4)
print(numero_de_puertas)
marca = coche_dict.get("marca","desconocida")
print(marca)
```

4

Tesla

Iterar

al índice

Que el diccionario no esté ordenado, no quiere decir que no podamos iterar sus elementos, ya que también es un **iterable**, como lo son las listas o las tuplas.

```
In [22]: for i in coche_dict:
          print(i)
          print(coche_dict[i])
```

```
color
['rojo', 'gris']
marca
Tesla
modelo
Cybertruck
peso
3500
cambio automatico
True
```

```
In [23]: for i in coche_dict:
          print(i, ":", coche_dict[i])
```

```
color : ['rojo', 'gris']
marca : Tesla
modelo : Cybertruck
peso : 3500
cambio automatico : True
```

```
In [24]: coche_dict.items()
```

```
Out[24]: dict_items([('color', ['rojo', 'gris']), ('marca', 'Tesla'), ('modelo', 'Cybertruck'), ('peso', 3500), ('cambio automatico', True)])
```

```
In [25]: for clave, valor in coche_dict.items():
          print(clave, ":", valor)
```

```
color : ['rojo', 'gris']
marca : Tesla
modelo : Cybertruck
peso : 3500
cambio automatico : True
```

```
In [ ]:
```