

# Diccionarios

Un diccionario es un conjunto de parejas **clave- valor** (key-value). Es decir, se accede a cada elemento a partir de su clave. Se definen de la siguiente manera:

```
1 estudiante = {  
2     "nombre": "Iñaki Perurena",  
3     "edad": 30,  
4     "nota_media": 7.25,  
5     "repetidor" : False  
6 }
```

Las **claves** tienen que ser **únicas** y estar formadas por un **string o un número**. Para acceder al valor de una clave existen dos maneras distintas:

```
1 # Acceder al valor de una clave  
2 edad = estudiante["edad"] # devuelve el valor de 'edad'  
3 nota_media = estudiante.get("nota_media") # devuelve el valor de 'nota_media'  
4  
5 # Insertar o actualizar un valor:  
6 estudiante["edad"] = 25 # actualiza el valor de 'edad'  
7 estudiante["suspensos"] = 3 # inserta una nueva pareja clave - valor  
8  
9 # insertar una pareja clave - valor o actualizar si ya existe:  
10 estudiante.update({'aprobados': '8'})
```

Algunos de los métodos más utilizados son los siguientes:

Método	Acción
diccionario.keys()	Devuelve todas las claves del diccionario
diccionario.values()	Devuelve todos los valores del diccionario
diccionario.pop(clave[, <default>])	Elimina la clave del diccionario y devuelve su valor asociado. Si no la encuentra y se indica un valor por defecto, devuelve el valor por defecto indicado.
diccionario.clear()	Vacía el diccionario
clave in diccionario	Devuelve True si el diccionario contiene la clave o False en caso contrario.
valor in diccionario.values()	Devuelve True si el diccionario contiene el valor o False en caso contrario.

## Recorrer un diccionario

La forma más habitual de recorrer un diccionario es mediante la sentencia `for`. Al recorrer un diccionario, por defecto se iterará sobre sus claves:

```
1 diccionario = {'a':1, 'b':2, 'c':3}
2 for key in diccionario:
3     print(key)
4
5 # Resultado: a b c
```

Es decir, el código anterior será equivalente al siguiente:

```
1 diccionario = {'a':1, 'b':2, 'c':3}
2 for key in diccionario.keys():
3     print key
4
5 # Resultado: a b c
```

Por lo tanto, para iterar accediendo a los valores, realizaremos lo siguiente:

```
1 diccionario = {'a':1, 'b':2, 'c':3}
2 for key in diccionario:
3     print(diccionario[key])
4
5 # Resultado: 1 2 3
```

Otra manera alternativa sería empleando la función `items()`, la cual devuelve el diccionario como tuplas de tipo `(key,value)`:

```
1 diccionario = {'a':1, 'b':2, 'c':3}
2 for key, value in diccionario.items():
3     print("El valor de %s is %d" % (key, value))
4
5 # Resultado:
6 # El valor de a is 1
7 # El valor de b is 2
8 # El valor de c is 3
```

## Borrar un elemento

Para borrar un elemento de un diccionario se utiliza la instrucción `del`.

```
1 edades = {  
2     "Ane" : 22,  
3     "Jokin" : 27,  
4     "Aitor" : 15  
5 }  
6 del edades["Aitor"]
```

Otra alternativa también utilizada y mencionada anteriormente es la función `pop()`, el cual devuelve el valor del elemento eliminado:

```
1 edades = {  
2     "Ane" : 22,  
3     "Jokin" : 27,  
4     "Aitor" : 15  
5 }  
6 edades.pop("Aitor")
```

Un diccionario nunca debería contener dos claves iguales. No obstante, en caso de contener una clave repetida, tanto `del` como `pop()` eliminarán todas las claves coincidentes.

## Coding time!

### Ejercicio 1

Crea un programa que recorra una lista y cree un diccionario que contenga el número de veces que aparece cada número en la lista.

- Ejemplo: [12, 23, 5, 12, 92, 5, 12, 5, 29, 92, 64, 23]
- Resultado: {12: 3, 23: 2, 5: 3, 92: 2, 29: 1, 64: 1}

### Ejercicio 2

Recorre un diccionario y crea una lista solo con los valores que contiene, sin añadir valores duplicados.

- Ejemplo: {'Mikel': 3, 'Ane': 8, 'Amaia': 12, 'Unai': 5, 'Jon': 8, 'Ainhoa': 7, 'Maite': 5}
- Resultado: [3, 8, 12, 5, 7]

### Ejercicio 3

Crea una programa de Login que compruebe el usuario y contraseña en el diccionario a continuación:

```
1 usuarios = {
2     "iperurena": {
3         "nombre": "Iñaki",
4         "apellido": "Perurena",
5         "password": "123123"
6     },
7     "fmuguruza": {
8         "nombre": "Fermín",
9         "apellido": "Muguruza",
10        "password": "654321"
11    },
12    "aolaizola": {
13        "nombre": "Aimar",
14        "apellido": "Olaizola",
15        "password": "123456"
16    }
17 }
```

El usuario tendrá un máximo de 3 intentos, y al acceder correctamente se mostrará el nombre y apellido del usuario.

## Ejercicio 4

Crea un programa que permita introducir a un profesor las notas de sus estudiantes (máximo 10 estudiantes). Los datos se deberán almacenar en un diccionario como el siguiente:

```
1 estudiantes = {
2     "1": {
3         "nombre": "Lorea",
4         "nota": 8
5     },
6     "2": {
7         "nombre": "Markel",
8         "nota": "4.2"
9     },
10    "3": {
11        "nombre": "Julen",
12        "nota": 6.5
13    }
14 }
```

Una vez introducidos todos los datos, el programa mostrará una lista con los nombres de los estudiantes que han suspendido y otra con los que han aprobado. También calculará y mostrará la nota media de la clase

