

# Diccionarios

Son junto a las listas las colecciones más utilizadas y se basan en una estructura mapeada donde cada elemento de la colección se encuentra identificado con una clave única, por lo que no puede haber dos claves iguales. En otros lenguajes se conocen como arreglos asociativos.

Los diccionarios se definen igual que los conjuntos, utilizando llaves, pero también se pueden crear vacíos con ellas:

```
vacio = {}  
vacio
```

```
{}
```

Si consultamos el tipo de la variable que contiene un diccionario con la función `type()` encontraremos la palabra *dict*, esa es la clase que define los diccionarios:

```
type(vacio)
```

```
dict
```

## Definición

Para cada elemento se define la estructura **clave:valor**:

```
colores = {'amarillo':'yellow','azul':'blue'}  
colores
```

```
{'amarillo': 'yellow', 'azul': 'blue'}
```

Para consultar el valor de una clave utilizaremos la clave a modo de índice:

```
colores['amarillo']
```

```
'yellow'
```

## Mutabilidad

Los diccionarios son mutables, por lo que se les puede añadir elementos sobre la marcha a través de las claves:

```
colores['verde'] = 'green'
colores
```

```
{'amarillo': 'yellow', 'azul': 'blue', 'verde': 'green'}
```

Como los diccionarios son mutables también podemos sobrescribir un valor:

```
colores['amarillo'] = 'white'
colores
```

```
{'amarillo': 'white', 'azul': 'blue', 'verde': 'green'}
```

## Función del()

Sirve para borrar un elemento del diccionario:

```
del(colores['amarillo'])
colores
```

```
{'azul': 'blue', 'verde': 'green'}
```

Por cierto, las claves también pueden ser números, pero son un poco confusas:

```
numeros = {10:'diez',20:'veinte'}
numeros[10]
```

```
'diez'
```

Una utilidad de los diccionarios es que podemos trabajar directamente con sus registros como si fueran variables:

```
edades = {'Hector':27,'Juan':45,'Maria':34}
edades['Hector']+=1
edades
```

```
{'Hector': 28, 'Juan': 45, 'Maria': 34}
```

```
edades['Juan'] + edades['Maria']
```

## Lectura secuencial

Es posible utilizar iteraciones **for** para recorrer los elementos del diccionario:

```
edades = {'Hector':27, 'Juan':45, 'Maria':34}

for edad in edades:
    print(edad)
```

```
Maria
Hector
Juan
```

El problema es que se devuelven las claves en lugar de los valores. Para solucionarlo deberíamos indicar la clave del diccionario para cada elemento:

```
for clave in edades:
    print(edades[clave])
```

```
34
27
45
```

Si queremos mostrar tanto la clave como el valor podríamos hacerlo así:

```
for clave in edades:
    print(clave, edades[clave])
```

```
Maria 34
Hector 27
Juan 45
```

El **método .items()** nos facilita la lectura en clave y valor de los elementos. Devuelve ambos valores en cada iteración automáticamente y nos permite almacenarlos:

```
for clave, valor in edades.items():
    print(clave, valor)
```

```
Maria 34
Hector 27
Juan 45
```

# Listas de diccionarios

Podemos crear nuestras propias estructuras avanzadas mezclando ambas colecciones. Mientras los diccionarios se encargarían de manejar las propiedades individuales de los registros, las listas nos permitirían manejarlos todos en conjunto:

```
personajes = []

gandalf = {'Nombre': 'Gandalf', 'Clase': 'Mago', 'Raza': 'Humano'}
legolas = {'Nombre': 'Legolas', 'Clase': 'Arquero', 'Raza': 'Elfo'}
gimli = {'Nombre': 'Gimli', 'Clase': 'Guerrero', 'Raza': 'Enano'}

personajes.append(gandalf)
personajes.append(legolas)
personajes.append(gimli)


print(personajes)

[{'Clase': 'Mago', 'Nombre': 'Gandalf', 'Raza': 'Humano'},
 {'Clase': 'Arquero', 'Nombre': 'Legolas', 'Raza': 'Elfo'},
 {'Clase': 'Guerrero', 'Nombre': 'Gimli', 'Raza': 'Enano'}]
```

Como ahora tenemos una estructura común a través de diccionarios, podemos suponer que cada diccionario es un personaje y mostrar los registros mientras los recorremos dinámicamente con un *for*:

```
for pesonaje in personajes:
    print(pesonaje['Nombre'], pesonaje['Clase'], pesonaje['Raza'])

Gandalf Mago Humano
Legolas Arquero Elfo
Gimli Guerrero Enano
```

 **Curiosidades**

Un alumno me dijo que Gandalf no es humano, disculpadme por tal ofensa 🙏🙏🙏