

Diccionarios

Semestre 02, 2025

Definición

Un diccionario es una estructura de datos que permite almacenar valores y referenciar los mismos utilizando llaves o claves.

También son conocidos como matrices asociativas.

Los diccionarios si son mutables al igual que las listas por lo que se pueden modificar los valores de las llaves. Pero se tiene la restricción que las llaves Si son inmutables. Esto quiere decir que una vez definida la llave esta ya no puede cambiar.

Cada llave está asociada a un valor y mediante esta se puede hacer referencia al valor en sí justo como se haría con un índice.

Un diccionario usa la notación de llaves para definirse `{}`.

```
1
2  diccionario = {
3
4  "llave": "valor"
5
6  }
7
8
9
10 diccionario = {
11
12 "codigo": "cc2005",
13
14 "nombre": "Algoritmos",
15
16 "creditos": 6
17
18 }
19
20
```

```
21
22  diccionario = {
23
24  "carnet": "123456",
25
26  "notas": [100, 90, 80, 100]
27
28  }
29
```

El uso de llaves facilita el desarrollo ya que es una forma más “natural” de referenciar valores.

Al igual que las listas es posible imprimir in diccionario y poder ver el contenido del mismo:

```
1
2  diccionario = {
3
4  "codigo": "cc2005",
5
6  "nombre": "Algoritmos",
7
8  "creditos": 6
9
10 }
11
12
13
14 print(diccionario)
15
16
17
18 print(diccionario["codigo"])
19
```

Modificación de valores

Para modificar el valor de un atributo del diccionario se puede usar la llave para hacer la asignación:

```
1
```

```
2  diccionario = {
3
4  "codigo": "cc2005",
5
6  "nombre": "Algoritmos",
7
8  "creditos": 6
9
10 }
11
12
13
14 diccionario["creditos"] = 4
15
16
17
18 print(diccionario)
19
```

Adición de elementos

Para crear un nuevo elemento basta con crear una nueva llave:

```
1
2  diccionario = {
3
4  "codigo": "cc2005",
5
6  "nombre": "Algoritmos",
7
8  "creditos": 6
9
10 }
11
12
13
14 diccionario["catedratico"] = "Erick Marroquín"
15
16
17
18 print(diccionario)
19
```

Eliminar elementos

Para eliminar un elemento se puede hacer uso de la palabra reservada `del`:

```
1
2  diccionario = {
3
4  "codigo": "cc2005",
5
6  "nombre": "Algoritmos",
7
8  "creditos": 6
9
10 }
11
12
13
14 del diccionario["creditos"]
15
16
17
18 print(diccionario)
19
```

Errores comunes

Si se trata de acceder a una llave que no existe para tratar de leer el valor Python da error:

```
1
2  diccionario = {
3
4  "codigo": "cc2005",
5
6  "nombre": "Algoritmos",
7
8  "creditos": 6
9
10 }
11
12
```

```
13
14 print(diccionario["seccion"])
15
```

Funciones disponibles

<https://docs.python.org/3/tutorial/datastructures.html#more-on-lists>

Crear un diccionario

Un diccionario almacena pares **llave: valor**.

```
1
2 diccionario = {
3
4     "codigo": "cc2005",
5
6     "nombre": "Algoritmos",
7
8     "creditos": 6
9
10 }
11
```

Obtener el valor de una llave con `.get()`

Devuelve el valor asociado a una llave.

```
1
2 print(diccionario.get("codigo"))
3
4 # cc2005
5
```

`.get()` con valor por defecto

Permite especificar un valor alternativo si la llave no existe.

```
1
2 print(diccionario.get("seccion", -1))
3
4 # -1
5
```

`.keys()`

Devuelve todas las **llaves** del diccionario como un objeto iterable.

```
1
2 llaves = diccionario.keys()
3
4 print(llaves)
5
6 # dict_keys(['codigo', 'nombre', 'creditos'])
7
```

Convertir las llaves a lista

Puedes transformar el objeto iterable en una lista.

```
1
2 llaves = list(llaves)
3
4 print(llaves)
5
6 # ['codigo', 'nombre', 'creditos']
7
```

`.values()`

Devuelve todos los **valores** del diccionario como un objeto iterable.

```
1
2 valores = diccionario.values()
```

```
3
4 print(valores)
5
6 # dict_values(['cc2005', 'Algoritmos', 6])
7
```

Convertir los valores a lista

Igual que con las llaves, se pueden convertir a una lista.

```
1
2 valores = list(valores)
3
4 print(valores)
5
6 # ['cc2005', 'Algoritmos', 6]
7
```

.update()

Combina dos diccionarios, agregando o actualizando llaves y valores.

```
1
2 diccionario2 = {
3
4     "seccion": 600,
5
6     "catedratico": "Erick Marroquín"
7
8 }
9
10
11
12 diccionario.update(diccionario2)
13
14
15
16 print(diccionario)
17
```

Mostrar el diccionario completo

Permite visualizar todo el contenido del diccionario.

```
1
2  print(diccionario)
3
4  # {'codigo': 'cc2005', 'nombre': 'Algoritmos', 'creditos': 6}
5
```

Eliminar una llave específica

Permite borrar un elemento del diccionario mediante su llave.

```
1
2  del diccionario["creditos"]
3
4  print(diccionario)
5
6  # {'codigo': 'cc2005', 'nombre': 'Algoritmos'}
7
```

Verificar si una llave existe

Usa el operador `in` para comprobar si una llave está presente.

```
1
2  print("nombre" in diccionario)
3
4  # True
5
```