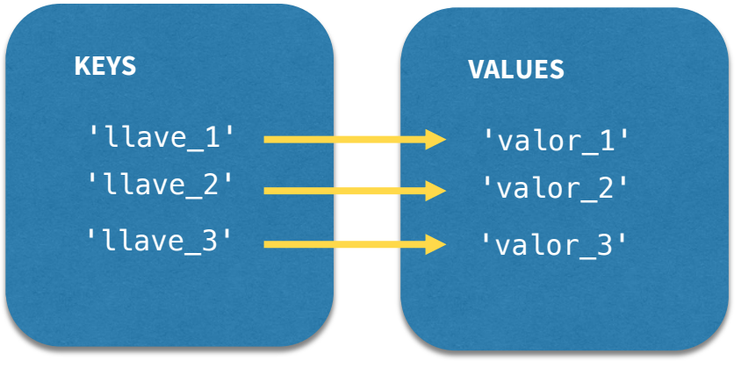
Diccionarios en Python . **EL_ACORDEÓN**

El diccionario (clásico)

Un diccionario es una colleción de llaves (keys) y valores (values). Cada llave se asocia, de modo único, con un valor. Como estructura contenedora de información, el diccionario está en la raíz de Python.



MÉTODOS

Acción	Comandos, ejemplo

Creación y asignación	`my_dict = {“enero”:154, “febrero”:240}`
	
	`my_dict = dict((‘enero’, 154), (‘febrero’, 240))`

Acceso a llaves	`my_dict.keys()`
Acceso a valores	`my_dict.values()`
Acceso a ambos, llaves y valores	`my_dict.items()`

Ciclo sobre un diccionario	`for key, value in my_dict.items():`
	`  print(`
	`  …`

Acceso a elementos concretos	`my_dict[[‘enero’, ‘febrero’]]`
Contiene una llave?	`‘enero’ in my_dict`
Ordenar por sus llaves	`sorted(my_dict)`

El defaultdict (collections)

El módulo `collections` contiene un diccionario llamado `defaultdict`. Para utilizarlo importaremos:

```
from collections import defaultdict
```

Uso para contar

USOS

El defaultdict tiene la ventaja de añadir llaves ausentes en un diccionario. Esto es particularmente útil para hacer diccionarios de conteos

PROBLEMA EJEMPLO

Tomemos una lista y contemos el número de ocurrencias de cada elemento en ella:

```
lista = [‘agua’, ‘fuego’, ‘tierra’, ‘agua’,
‘tierra’, ‘aire’, ‘aire’, ‘fuego’, ‘fuego’, ‘agua’]
```

SOLUCIÓN CON LA CLASE `dict`

```
num_elementos = {}
for elemento in lista:
    if elemento in num_elementos:
        num_elementos[elemento] += 1
    else:
        num_elementos[elemento] = 1
```

Necesitamos corroborar que sea una llave…

… y añadirla si no

SOLUCIÓN CON LA CLASE `defaultdict`

```
from collections import defaultdict
num_elementos = defaultdict(int)
for elemento in lista:
    num_elementos[elemento] += 1
```

La clase int será usada si la llave no está en el dict

El Counter (collections)

El módulo `collections` contiene una subclase del diccionario llamado `Counter`. Para utilizarlo importaremos:

```
from collections import Counter
```

Uso: contar y poco más

USOS

El Counter tiene la ventaja de contar de forma automática

PROBLEMA EJEMPLO

Tomemos una lista y contemos el número de ocurrencias de cada elemento en ella:

```
lista = [‘agua’, ‘fuego’, ‘tierra’, ‘agua’,
‘tierra’, ‘aire’, ‘aire’, ‘fuego’, ‘fuego’, ‘agua’]
```

SOLUCIÓN CON LA CLASE `Counter`

```
from collections import Counter
num_elementos = Counter(lista)
```

Automatizado

MÉTODOS

.elements()	Da un iterable sobre cada elemento el número de veces que aparece. Por ejemplo: `list(num_elementos.elements())`
.most_common(n)	Los elementos más comunes. Da los primeros `n`. El argumento es opcional. Por ejemplo: `num_elementos.most_common(2)`
Ordenar por sus llaves	`sorted(my_dict)`