

## Estructuras de Datos (II) - dict (Diccionario)

Un **diccionario** en Python es una estructura de datos que almacena **pares de clave-valor**.

Se puede **crear un diccionario** vacío utilizando llaves `{}` o la función `dict()`. Por ejemplo:

```
mi_diccionario = {}  
mi_otro_diccionario = dict()
```

Ambas variables `mi_diccionario` y `mi_otro_diccionario` son diccionarios vacíos.

Para **acceder a los elementos de un diccionario** se utiliza la **clave entre corchetes**. Por ejemplo: `mi_diccionario['clave']`. Si intentas acceder a una clave que no existe en el diccionario, se generará una excepción `KeyError`.

```
mi_diccionario = {'clave1': 'valor1', 'clave2': 'valor2'}  
  
print(mi_diccionario['clave1']) # Imprime 'valor1'
```

También puedes verificar si una clave está presente (o no) en un diccionario utilizando el operador `in` y `not in`. Aquí tienes un ejemplo:

```
# Crear el diccionario  
mi_diccionario = {'nombre': 'Juan', 'edad': 30}  
  
# Verificar si una clave está presente en el diccionario  
print('nombre' in mi_diccionario) # True  
print('altura' in mi_diccionario) # False  
  
# Verificar si un elemento NO está presente en el diccionario  
print('altura' not in mi_diccionario) # True  
print('nombre' not in mi_diccionario) # False
```

Algunos métodos comunes de los diccionarios son:

- `.get(clave[, valor_por_defecto])`: Es otra forma de acceder al valor asociado a una clave. Si la clave no existe, devuelve el valor por defecto especificado o `None`.

```
mi_diccionario = {'clave1': 'valor1', 'clave2': 'valor2'}  
  
print(mi_diccionario.get('clave1')) # Imprime 'valor1'  
print(mi_diccionario.get('clave3', 0)) # Imprime 0  
print(mi_diccionario.get('clave3')) # Imprime None
```

- **.keys()**: Devuelve una vista de las claves del diccionario.

```
print(mi_diccionario.keys()) # Imprime dict_keys(['clave1', 'clave2'])
```

- **.values()**: Devuelve una vista de los valores del diccionario.

```
print(mi_diccionario.values()) # Imprime dict_values(['valor1', 'valor2'])
```

- **.items()**: Devuelve una vista de los pares clave-valor del diccionario.

```
print(mi_diccionario.items()) # Imprime dict_items([('clave1', 'valor1'), ('clave2', 'valor2')])
```

- **.clear()**: Elimina todos los elementos del diccionario.

```
mi_diccionario.clear()

print(mi_diccionario) # Imprime {}
```

- **.copy()**: Devuelve una copia superficial del diccionario.

```
nuevo_dicc = mi_diccionario.copy()

print(nuevo_dicc) # Imprime {'clave1': 'valor1', 'clave2': 'valor2'}
```

- **.fromkeys(iterable[, valor])**: Crea un nuevo diccionario con las claves especificadas en el iterable y el valor especificado para todas las claves.

```
nuevo_dicc = dict.fromkeys(['clave1', 'clave2'], 0)

print(nuevo_dicc) # Imprime {'clave1': 0, 'clave2': 0}
```

- **.pop(clave[, valor\_por\_defecto])**: Elimina y devuelve el elemento con la clave especificada. Si la clave no existe y se proporciona un valor por defecto, devuelve ese valor. Si la clave no existe y no se proporciona un valor por defecto, se genera una excepción `KeyError`.

```
print(mi_diccionario.pop('clave1')) # Imprime 'valor1'
print(mi_diccionario) # Imprime {'clave2': 'valor2'}
```

- **.popitem()**: Elimina y devuelve un par clave-valor arbitrario del diccionario. Si el diccionario está vacío, se genera una excepción `KeyError`.

```
mi_diccionario = {'clave1': 'valor1', 'clave2': 'valor2'}

print(mi_diccionario.popitem()) # Imprime ('clave2', 'valor2')
```

```
print(mi_diccionario) # Imprime {'clave1': 'valor1'}
```

- **.setdefault(clave[, valor\_por\_defecto]):** Devuelve el valor asociado a la clave si existe. Si la clave no existe, inserta la clave con el valor por defecto especificado y devuelve ese valor. Si no se proporciona un valor por defecto, el valor por defecto es **None**.

```
mi_diccionario = {'clave1': 'valor1', 'clave2': 'valor2'}

print(mi_diccionario.setdefault('clave3', 0)) # Imprime 0
print(mi_diccionario) # Imprime {'clave1': 'valor1',
'clave2': 'valor2', 'clave3': 0}
```

- **.update([otro\_diccionario]):** Actualiza el diccionario con los pares clave-valor del otro diccionario o de un iterable de pares clave-valor.

```
mi_diccionario = {'clave1': 'valor1', 'clave2': 'valor2'}
mi_diccionario.update({'clave3': 0})

print(mi_diccionario) # Imprime {'clave1': 'valor1',
'clave2': 'valor2', 'clave3': 0}
```