

Diccionarios

Siempre escribe tu código como si la persona que lo fuera a mantener fuera un peligroso psicópata que sabe quien eres y donde vives



Introducción a los Diccionarios

- Los diccionarios son estructuras de datos en Python que almacenan información en pares clave-valor.
- Son dinámicos, indexados y pueden ser anidados.

```
primerDiccionario = { 'color': 'blanco' }  
print(primerDiccionario['color']) # Imprime blanco
```

Creación de Diccionarios

Uso de llaves {}

```
usuario = {  
    "Nombre": "Sara",  
    "Edad": 27,  
    "Documento": 1003882  
}
```

Usando dict()

```
usuario = dict(Nombre='Sara', Edad=27, Documento=1003882)
```

Acceder y Modificar Elementos

Acceso a valores:

```
print(usuario['Nombre']) # Sara  
print(usuario.get('Nombre')) # Sara
```

Modificar valores:

```
usuario['Nombre'] = "Laura"
```

Agregar nuevos elementos:

```
usuario['Direccion'] = "Calle 123"
```

Iterar un Diccionario

Iterar claves:

```
for clave in usuario:  
    print(clave)
```

Iterar valores:

```
for valor in usuario.values():  
    print(valor)
```

Iterar claves y valores:

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

Métodos Más Comunes

Método	Descripción	Ejemplo
<code>clear()</code>	Elimina todos los elementos del diccionario.	<code>d.clear()</code>
<code>copy()</code>	Devuelve una copia superficial del diccionario.	<code>d2 = d.copy()</code>
<code>get(key[, default])</code>	Obtiene el valor asociado a una clave. Si no existe, devuelve <code>default</code> (o <code>None</code>).	<code>valor = d.get("clave", "valor_por_defecto")</code>
<code>items()</code>	Devuelve una vista de pares (clave, valor) del diccionario.	<code>for k, v in d.items():</code>
<code>keys()</code>	Devuelve una vista de las claves del diccionario.	<code>for k in d.keys():</code>
<code>values()</code>	Devuelve una vista de los valores del diccionario.	<code>for v in d.values():</code>
<code>pop(key[, default])</code>	Elimina y devuelve el valor asociado a <code>key</code> . Si no existe, devuelve <code>default</code> (o lanza <code>KeyError</code>).	<code>d.pop("clave")</code>

Métodos Más Comunes

`popitem()`

Elimina y devuelve un par (clave, valor) arbitrario.
Si el diccionario está vacío, lanza `KeyError`.

```
par = d.popitem()
```

`update([other]`

Actualiza el diccionario con pares clave-valor de `other` (puede ser otro diccionario o iterable).

```
d.update({"clave":  
"valor"})
```

`setdefault(key
[, default])`

Devuelve el valor de `key` si existe; si no, inserta `key` con `default` (o `None`) y lo devuelve.

```
d.setdefault("clave",  
"valor_por_defecto")
```

`fromkeys(seq[,
value])`

Crea un nuevo diccionario con claves de `seq` y valores iguales a `value` (o `None`).

```
dict.fromkeys(["a",  
"b"], 0)
```


Ejercicio 1

- Pedir un producto y cantidad.
- Calcular el precio total.
- Verificar disponibilidad.