

✓ Clase 08 - Diccionarios en Python

✓ 1. Diccionarios: Clave-Valor, Acceso, Inserción y Eliminación

Definición

Un diccionario es una colección de elementos que, a diferencia de las listas o tuplas, no se almacenan en un orden secuencial, sino que cada elemento tiene una **clave** única y un **valor** asociado. Piensa en un diccionario real: buscas una palabra (la clave) para encontrar su definición (el valor). En Python, los diccionarios nos permiten almacenar información de forma lógica y acceder a ella de manera muy eficiente.

Propiedades y Características

- **Mutable:** Podemos modificar, añadir o eliminar elementos después de su creación.
- **Dinámicos:** Pueden crecer o decrecer en tamaño según lo necesitemos.
- **Claves únicas e inmutables:** No puede haber dos claves iguales en un mismo diccionario. Además, las claves deben ser de un tipo de dato inmutable (como strings, números o tuplas).
- **Ordenados (a partir de Python 3.7):** Desde esta versión, los diccionarios recuerdan el orden en que se insertaron los elementos. En versiones anteriores, eran desordenados.

Sintaxis

Se crean utilizando llaves `{}` y separando cada par clave-valor con dos puntos `:`.

```
# Creación de un diccionario
mi_diccionario = {
    "clave1": "valor1",
    "clave2": "valor2",
    "clave3": "valor3"
}

# Creación de un diccionario vacío
diccionario_vacio = {}
otro_vacio = dict()
```

✓ Ejemplos Resueltos

```
1 # 1. Creación y acceso a un diccionario
2 estudiante = {
3     "nombre": "Ana",
4     "edad": 22,
5     "curso": "Inteligencia Artificial",
6     "es_becado": True
7 }
8
9 print(f"Diccionario original: {estudiante}")
10
11 # Accedemos al valor asociado a la clave 'nombre'
12 nombre_estudiante = estudiante["nombre"]
13 print(f"El nombre del estudiante es: {nombre_estudiante}")
14
15 # ---
16
17 # 2. Inserción y Modificación de elementos
18
19 # Añadimos un nuevo par clave-valor para la universidad
20 estudiante["universidad"] = "Universidad Politécnica"
21 print(f"Diccionario con nueva clave: {estudiante}")
22
23 # Modificamos un valor existente. Ana ha cumplido años.
24 estudiante["edad"] = 23
25 print(f"Diccionario con edad actualizada: {estudiante}")
26
27 # ---
28
29 # 3. Eliminación de elementos
30
31 # Usamos la palabra clave 'del' para eliminar el par con la clave 'es_becado'
32 del estudiante["es_becado"]
33 print(f"Diccionario tras eliminar una clave: {estudiante}")
```

✓ Ejercicios Propuestos

1. **Crear un diccionario** llamado `mi_coche` que almacene la siguiente información:
`marca` (string), `modelo` (string), `año` (integer) y `color` (string).
2. **Acceder y mostrar** por pantalla el modelo del coche.
3. **Añadir una nueva clave** llamada `kilometraje` con un valor numérico.
4. **Actualizar el valor** de la clave `color` a un nuevo color que te guste.
5. **Eliminar la clave** `año` del diccionario.

2. Métodos de Diccionarios

Los diccionarios vienen con métodos integrados que nos facilitan mucho el trabajo. Veamos los más importantes.

✓ `.keys()`

Devuelve un objeto `dict_keys` que contiene todas las claves del diccionario. Es muy útil para iterar sobre ellas.

✓ Sintaxis

```
diccionario.keys()
```

```
1 configuracion = {
2     "idioma": "español",
3     "tema": "oscuro",
4     "notificaciones": True
5 }
6
7 claves = configuracion.keys()
8 print(f"Claves del diccionario: {claves}")
9
10 # Podemos convertirlo a lista si necesitamos
11 lista_claves = list(claves)
12 print(f"Claves como lista: {lista_claves}")
```

Ejercicio Propuesto

- Dado un diccionario `producto = {"nombre": "Laptop", "precio": 1200, "stock": 45}`, obtén y muestra por pantalla todas sus claves.

✓ `.values()`

Devuelve un objeto `dict_values` con todos los valores del diccionario.

✓ Sintaxis

```
diccionario.values()
```

```
1 configuracion = {
2     "idioma": "español",
3     "tema": "oscuro",
4     "notificaciones": True
5 }
6
7 valores = configuracion.values()
8 print(f"Valores del diccionario: {valores}")
9
10 # También podemos convertirlo a lista
11 lista_valores = list(valores)
12 print(f"Valores como lista: {lista_valores}")
```

Ejercicio Propuesto

- Para el diccionario `producto` del ejercicio anterior, obtén y muestra todos sus valores.

✓ `.items()`

Devuelve un objeto `dict_items` que contiene tuplas, donde cada tupla es un par `(clave, valor)`.

✓ Sintaxis

```
diccionario.items()
```

```
1 configuracion = {
2     "idioma": "español",
3     "tema": "oscuro",
4     "notificaciones": True
5 }
6
7 items = configuracion.items()
8 print(f"Items del diccionario: {items}")
9
10 # Es especialmente útil para recorrer el diccionario en un bucle
11 print("\nRecorriendo el diccionario:")
12 for clave, valor in configuracion.items():
13     print(f"- {clave}: {valor}")
```

Ejercicio Propuesto

- Para el diccionario `producto`, obtén sus items y luego usa un bucle `for` para imprimir cada clave y su valor en una línea separada, por ejemplo: `nombre:`

Laptop.

✓ `.get(clave, valor_por_defecto)`

Permite acceder a una clave de forma segura. Si la clave no existe, no lanza un error, sino que devuelve `None` o el valor por defecto que especifiquemos.

✓ Sintaxis

```
diccionario.get('clave_existente')  
diccionario.get('clave_inexistente', 'valor si no se encuentra')
```

```
1 estudiante = {  
2     "nombre": "Carlos",  
3     "edad": 25  
4 }  
5  
6 # Acceso a una clave que sí existe  
7 nombre = estudiante.get("nombre")  
8 print(f"Nombre obtenido con get: {nombre}")  
9  
10 # Intento de acceso a una clave que no existe (sin valor por defecto)  
11 apellido = estudiante.get("apellido")  
12 print(f"Apellido obtenido con get: {apellido}") # Devuelve None  
13  
14 # Intento de acceso con un valor por defecto  
15 curso = estudiante.get("curso", "No inscrito")  
16 print(f"Curso obtenido con get: {curso}")
```

Ejercicio Propuesto

- Usando el diccionario `mi_coche` de antes, intenta obtener el valor de la clave `cv` (caballos de vapor). Como no existirá, proporciona un valor por defecto de `'Dato no disponible'` y muéstralo por pantalla.

✓ `.update(otro_diccionario)`

Fusiona un diccionario con otro. Si hay claves coincidentes, los valores del diccionario que se pasa como argumento sobrescriben a los del original.

✓ Sintaxis

```
diccionario1.update(diccionario2)
```

```
1 perfil_usuario = {
2     "usuario": "dev_master",
3     "email": "user@example.com"
4 }
5
6 datos_adicionales = {
7     "pais": "España",
8     "email": "new_email@example.com" # Esta clave se actualizará
9 }
10
11 print(f"Perfil original: {perfil_usuario}")
12
13 # Actualizamos el perfil con los nuevos datos
14 perfil_usuario.update(datos_adicionales)
15
16 print(f"Perfil actualizado: {perfil_usuario}")
```

✓ Ejercicio Propuesto

- Tienes un diccionario `inventario_parcial = {"manzanas": 20, "naranjas": 15}`. Tienes otro diccionario con una nueva entrega: `nueva_entrega = {"naranjas": 30, "platanos": 25}`. Actualiza `inventario_parcial` con la `nueva_entrega` y muestra el resultado final.

✓ 3. Caso de Uso: Gestión de un Inventario Simple

Vas a gestionar el inventario de una pequeña tienda de fruta. Tu tarea es usar un diccionario para realizar un seguimiento de los productos y sus cantidades (en kg) a lo largo de un día de trabajo.

Sigue estos pasos:

1. Creación del Inventario:

- Crea un diccionario llamado `inventario`.
- Añade los siguientes productos con sus cantidades iniciales:
 - "manzanas": 50
 - "platanos": 75
 - "naranjas": 40
- Muestra el inventario inicial por pantalla

2. Elemento de lista

- Ha llegado un nuevo producto. Añade "peras" al inventario con una cantidad de 30 kg.
- Hemos recibido más manzanas. Actualiza la cantidad de "manzanas", añadiendo 25 kg a su valor actual.
- Muestra el inventario después de estas actualizaciones.

3. Registro de una Venta:

- Un cliente ha comprado 10 kg de "naranjas". Resta esta cantidad del stock.
- Muestra el inventario actualizado.

4. Producto Agotado:

- Los plátanos se han estropeado y hay que retirarlos. Elimina "platanos" del inventario.
- Muestra el inventario final.

5. Consulta de Stock:

- El encargado te pregunta si quedan uvas. Utiliza el método `.get()` para consultar el stock de "uvas", proporcionando 0 como valor por defecto si no se encuentran. Imprime el resultado de la consulta.

6. Informe Final:

- Al final del día, genera un pequeño informe. Imprime un mensaje que diga "Informe Final de Inventario:" y, a continuación, recorre los items del diccionario final para mostrar cada fruta y su cantidad disponible en una línea separada (ej: - Hay 55 kg de manzanas).

1 Empieza a programar o a crear código con IA.

