



# Clase 4: Diccionarios en Python

## 📌 ¿Qué es un diccionario en Python?

Un **diccionario** es una estructura de datos en Python que permite almacenar pares de clave-valor. La **clave** es el identificador único y el **valor** es el dato asociado a esa clave. Puedes verlo como un diccionario físico, donde la palabra es la clave y su significado es el valor.

### Ejemplo simple de un diccionario:

```
mi_diccionario = {  
    1: "Juan",  
    2: "María",  
    3: "Pedro"  
}
```

Aquí tenemos claves (1, 2, 3) y valores ("Juan", "María", "Pedro").

## 🔑 Propiedades Clave de los Diccionarios

1. **Clave única:** No pueden existir dos claves iguales en un diccionario.
  2. **Mutable:** Puedes modificar sus valores y agregar o eliminar pares de clave-valor.
  3. **Ordenado (Python 3.7+):** Los diccionarios mantienen el orden de inserción de los elementos.
- 

## Operaciones Comunes con Diccionarios

### 1. Crear un Diccionario

Los diccionarios se definen usando llaves `{ }` y los pares clave-valor se separan por dos puntos `:`.

```
persona = {  
    "nombre": "Caitlyn",  
    "apellido": "Brigham",  
    "altura": 1.60,  
    "edad": 29  
}  
print(persona)
```

### 2. Acceder a Elementos

Para acceder a un valor en particular, se usa la clave entre corchetes `[ ]`.

```
print(persona["nombre"]) # Caitlyn
```

Si intentas acceder a una clave que no existe, Python arroja un error `KeyError`.

### 3. Modificar un Valor

Puedes modificar el valor de una clave existente de esta forma:

```
persona["edad"] = 30  
print(persona["edad"]) # 30
```

## 4. Eliminar un Elemento

Utilizamos la función `del` para eliminar una clave y su valor asociado.

```
del persona["edad"]  
print(persona)
```

## 5. Métodos Útiles de Diccionarios

- `keys()` : Retorna todas las claves del diccionario.

```
claves = persona.keys()  
print(claves) # dict_keys(['nombre', 'apellido', 'altura'])
```

- `values()` : Retorna todos los valores del diccionario.

```
valores = persona.values()  
print(valores) # dict_values(['Caitlyn', 'Brigham', 1.60])
```

- `items()` : Retorna pares clave-valor en forma de tuplas.

```
pares = persona.items()  
print(pares) # dict_items([('nombre', 'Caitlyn'), ('apellido', 'Brigham'), ('altura', 1.60)])
```

## Ejemplos Prácticos: Casos de la Vida Real

### 1. Agenda de Contactos con Diccionarios

Supongamos que tenemos una agenda donde queremos almacenar la información de varias personas. Podemos hacerlo usando diccionarios anidados:

```
agenda_contactos = {  
    "Carla": {
```

```
        "nombre": "Carla",
        "apellido": "Gómez",
        "edad": 29,
        "altura": 1.65
    },
    "Diego": {
        "nombre": "Diego",
        "apellido": "Fernández",
        "edad": 32,
        "altura": 1.80
    }
}

# Acceder a los datos de Carla
print(agenda_contactos["Carla"])
```

Esto permite almacenar información detallada de cada contacto y acceder a ella fácilmente.

## Ejercicio Práctico: Gestión de Productos en una Tienda

Imagina que trabajas en una tienda y necesitas gestionar información sobre productos. Usaremos un diccionario para almacenar cada producto junto con su descripción y precio.

### Ejercicio: Crear y gestionar productos

1. Crea un diccionario para almacenar los productos.
2. Agrega al menos 3 productos con sus respectivos nombres, descripciones y precios.
3. Escribe una función que permita buscar un producto por su nombre.
4. Escribe una función que permita actualizar el precio de un producto.

### Solución paso a paso:

```

# 1. Crear el diccionario de productos
productos = {
    "Producto1": {"nombre": "Laptop", "descripcion": "Laptop
de 14 pulgadas", "precio": 750},
    "Producto2": {"nombre": "Smartphone", "descripcion": "Tel
éfono inteligente de 6.5 pulgadas", "precio": 300},
    "Producto3": {"nombre": "Auriculares", "descripcion": "Au
riculares inalámbricos", "precio": 50}
}

# 2. Función para buscar un producto por nombre
def buscar_producto(nombre_producto):
    for producto in productos.values():
        if producto["nombre"].lower() == nombre_producto.lower():
            return producto
    return "Producto no encontrado"

# 3. Función para actualizar el precio de un producto
def actualizar_precio(nombre_producto, nuevo_precio):
    for producto in productos.values():
        if producto["nombre"].lower() == nombre_producto.lower():
            producto["precio"] = nuevo_precio
            return f"El nuevo precio de {producto['nombre']}
es {producto['precio']}"
    return "Producto no encontrado"

# Ejemplo de uso:
print(buscar_producto("Laptop")) # {'nombre': 'Laptop', 'des
cripcion': 'Laptop de 14 pulgadas', 'precio': 750}

actualizar_precio("Smartphone", 280)
print(productos["Producto2"]) # {'nombre': 'Smartphone', 'de

```

```
scripcion': 'Teléfono inteligente de 6.5 pulgadas', 'precio':  
280}
```

## 💡 Aplicaciones de Diccionarios en la Vida Real

- **Sistemas de Inventarios:** Puedes almacenar productos con sus características y precios.
- **Gestión de Contactos:** Como en el ejemplo de la agenda de contactos, puedes guardar información detallada de personas.
- **Análisis de Datos:** En procesamiento de datos, los diccionarios son útiles para agrupar y manipular grandes volúmenes de información.

## Ejercicio Final: Encuestas y Análisis de Datos

Imagina que tienes que procesar las respuestas de una encuesta. Las respuestas de cada persona están almacenadas en un diccionario donde la clave es el nombre de la persona y el valor es otro diccionario con las preguntas y sus respuestas.

### Ejercicio:

1. Crea un diccionario con al menos 3 personas y sus respuestas.
2. Escribe una función que calcule el porcentaje de personas que respondieron "Sí" a una pregunta en particular.

### Solución:

```
# 1. Crear el diccionario de encuestas  
encuestas = {  
    "Juan": {"P1": "Sí", "P2": "No", "P3": "Sí"},  
    "María": {"P1": "No", "P2": "Sí", "P3": "Sí"},  
    "Pedro": {"P1": "Sí", "P2": "Sí", "P3": "No"}  
}  
  
# 2. Función para calcular el porcentaje de respuestas "Sí"  
def porcentaje_si(pregunta):
```

```
total_personas = len(encuestas)
si_respuestas = sum(1 for respuestas in encuestas.values
() if respuestas[pregunta] == "Sí")

return (si_respuestas / total_personas) * 100

# Ejemplo de uso:
print(f"Porcentaje de 'Sí' en P1: {porcentaje_si('P1')}%")
```

## Conclusión

Los diccionarios son una herramienta poderosa para almacenar datos estructurados. Te permiten manejar información compleja de una manera eficiente y clara. ¡Con esta guía y los ejercicios, ya puedes dominarlos para resolver problemas del mundo real! 💡