

---

# Estructuras complejas: Diccionesarios

En Python, un diccionario es una colección de elementos, donde cada uno tiene una llave denominada *key* en inglés y un valor, en inglés, *value*.

Es un tipo de estructura de datos mutable que permite guardar un conjunto no ordenado de pares clave-valor.

También se los denomina “listas asociativas”, ya que nos permitirán crear un listado indicando nombres únicos y fáciles de recordar, y lograr asociar a valores.

---

# Crear un diccionario

# Diccionario vacío

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

# Diccionario con elementos clave valor.

```
otro_diccionario = {  
    "nombre": "Matías",  
    "usuario": "elmati22",  
}
```

---

## Acceso a los valores

Para acceder a los valores de un diccionario necesito conocer su clave o recorrer la estructura.

Así como en las listas podría acceder a un elemento a través de su índice, es decir, su posición en la lista, en este caso para un diccionario, podría acceder al valor asociado a una clave mencionando esa clave en la instrucción de acceso.

Por ejemplo:

```
print(otro_diccionario["nombre"])
```

```
print(otro_diccionario["usuario"])
```

# Mas Operaciones

---

Mientras que a las listas y tuplas se accede solo y únicamente por un número de índice, los diccionarios permiten utilizar una clave para declarar y acceder a un valor:

```
mi_diccionario = {'a': 1, 'b': 2, 'c': 3}
```

```
print(mi_diccionario['b'])    # Salida: 2
```

**Añadir un nuevo par clave-valor a un diccionario en Python:**

```
mi_diccionario['d'] = 4 # notación de corchetes
```

```
mi_diccionario.update({'d': 4}) # método update
```

**Un diccionario permite eliminar cualquier entrada:**

```
del(mi_diccionario['clave_2'])
```

**Al igual que las listas, el diccionario permite modificar los valores**

```
mi_diccionario['clave_1'] = 'Nuevo Valor'
```

---

## Recorrer un diccionario

El siguiente código crea un diccionario y lo recorre a través de un FOR:

```
mi_diccionario = {  
    "nombre": "Juan",  
    "edad": "23",  
    "usuario": "jn23",  
}  
  
# Recorriendo los elementos  
for clave in mi_diccionario:  
    print(clave, ": ", mi_diccionario[clave])
```

---

## Creando un Diccionario de listas: primera parte

El siguiente código crea un diccionario con valores de tipo listas, agregando datos con el bucle FOR:

```
# Creamos el diccionario con listas vacías en su interior
```

```
usuarios = {  
    "nombres": [ ],  
    "dni": [ ]  
}
```

```
# Definimos un tamaño para las listas del diccionario
```

```
# Lo puedes cambiar si quieres
```

```
tamano = 3
```

---

## Creando un Diccionario de listas: segunda parte

```
# Leemos los datos y los agregamos al diccionario
for i in range(tamano):
    print("Ingrese los datos de la persona", i + 1)
    nombre = input("Nombre: ")
    dni= input("Identificación: ")

    # La primera lista es para los nombres
    usuarios["nombres"].append(nombre)

    # La segunda lista es para las identificaciones
    usuarios["dni"].append(dni)

# Ahora devolveremos los valores en el diccionario
for i in range(tamano):
    print("Mostrando los datos de la persona", i + 1)
    print("Nombre:", usuarios["nombres"][i])
    print("DNI:", usuarios["dni"][i])
```