

# LENGUAJES DE PROGRAMACIÓN

**PYTHON**





## Diccionarios y conjuntos.

En el mundo de la programación, la capacidad de organizar y manipular datos es esencial. Python, como lenguaje de programación versátil, proporciona estructuras de datos eficientes y poderosas para este propósito. En esta lección, exploraremos dos estructuras fundamentales de Python: los diccionarios y los conjuntos. Estas estructuras nos permiten almacenar, acceder y manipular datos de manera eficiente. A lo largo de esta lección, veremos ejemplos y demostraciones que ilustran cómo utilizar diccionarios y conjuntos en Python.





## Diccionarios

Un diccionario en Python es una colección desordenada de pares clave-valor, donde cada clave está asociada a un valor específico. A diferencia de las listas y las tuplas, los diccionarios no están ordenados, lo que significa que los elementos no tienen una posición definida. Para acceder a los valores de un diccionario, se utilizan las claves correspondientes. Veamos un ejemplo:

```
# Crear un diccionario
mi_diccionario = {"nombre": "Juan", "edad": 25, "ciudad":
"Madrid"}

# Acceder a los valores del diccionario
nombre = mi_diccionario["nombre"]
edad = mi_diccionario["edad"]

# Modificar un valor del diccionario
mi_diccionario["ciudad"] = "Barcelona"

# Añadir un nuevo par clave-valor al diccionario
mi_diccionario["profesion"] = "Ingeniero"

# Eliminar un par clave-valor del diccionario
del mi_diccionario["edad"]

# Verificar si una clave existe en el diccionario
existe_ciudad = "ciudad" in mi_diccionario

# Número de pares clave-valor en el diccionario
num_pares = len(mi_diccionario)
```



En el ejemplo anterior, creamos un diccionario llamado `mi_diccionario` que contiene información sobre una persona. Utilizamos las claves "nombre", "edad" y "ciudad" para acceder a los valores correspondientes. Además, podemos modificar los valores asignando un nuevo valor a través de la clave correspondiente.

También podemos añadir nuevos pares clave-valor al diccionario utilizando la sintaxis `diccionario[clave] = valor`. Para eliminar un par clave-valor, utilizamos la declaración `del`. La expresión `clave in diccionario` nos permite verificar si una clave existe en el diccionario, y la función `len()` nos proporciona el número de pares clave-valor en el diccionario.

## Conjuntos

Un conjunto en Python es una colección desordenada de elementos únicos. A diferencia de las listas y las tuplas, los conjuntos no permiten elementos duplicados y no tienen un orden definido. Los conjuntos son útiles cuando necesitamos almacenar elementos sin importar su orden o cuando necesitamos realizar operaciones como intersecciones, uniones y diferencias entre conjuntos. Veamos un ejemplo:



```
# Crear un conjunto
mi_conjunto = {1, 2, 3, 4, 5}

# Añadir elementos al conjunto
mi_conjunto.add(6)

# Eliminar un elemento del conjunto
mi_conjunto.remove(3)

# Verificar si un elemento está en el conjunto
esta_presente = 4 in mi_conjunto

# Número de elementos en el conjunto
num_elementos = len(mi_conjunto)
```

En el ejemplo anterior, creamos un conjunto llamado `mi_conjunto` que contiene los números del 1 al 5. Utilizamos el método `add()` para añadir un elemento al conjunto y el método `remove()` para eliminar un elemento específico.

La expresión `elemento in conjunto` nos permite verificar si un elemento está presente en el conjunto, y la función `len()` nos proporciona el número de elementos en el conjunto.





## Comparación entre diccionarios y conjuntos

Los diccionarios y los conjuntos son estructuras de datos útiles en diferentes situaciones. A continuación, se presentan algunas consideraciones para tener en cuenta:

Los diccionarios son ideales cuando necesitamos almacenar información en pares clave-valor y acceder a ella a través de claves.

Los conjuntos son útiles cuando necesitamos almacenar elementos únicos y realizar operaciones matemáticas de conjuntos como uniones, intersecciones y diferencias.

Los diccionarios y los conjuntos no mantienen un orden específico, por lo que no se puede asumir el orden de los elementos en ninguna de las estructuras.

Los diccionarios y los conjuntos son estructuras de datos importantes en Python que nos permiten organizar y manipular información de manera eficiente. Los diccionarios nos permiten asociar valores a claves y acceder a ellos rápidamente, mientras que los conjuntos nos permiten almacenar elementos únicos y realizar operaciones de conjuntos. La elección entre diccionarios y conjuntos dependerá de las necesidades específicas de nuestro programa.