

ESTRUCTURAS COMPLEJAS

Estructuras Complejas

Además de los tipos de datos básicos o primitivos ya utilizados anteriormente (int, float, str, bool y None),

Python tiene cuatro tipos más complejos que permiten almacenar colecciones de datos. Estos tipos son:

- Tuplas
- Conjuntos
- Listas
- Diccionarios

Estos cuatro tipos pueden almacenar colecciones de datos de diferentes tipos. Se diferencian principalmente en su **sintaxis** y en la **manera en que los datos pueden ser manipulados**.

Tuplas

Una **tupla** es una colección de datos **inmutables**, lo que significa que no pueden ser modificados una vez que han sido creados.

Las tuplas pueden almacenar **elementos de diferentes tipos**, incluyendo otros objetos como listas, diccionarios o incluso otras tuplas.

```
mi_tupla = ('cadena de texto', 15, 2.8, 'otro dato', 25)
```

Puedes acceder a cada uno de los elementos de la tupla utilizando su **índice**, siendo **0** el índice del primer elemento:

```
print(mi_tupla[1]) # Salida: 15
```

Para crear una tupla con un solo elemento, debes agregar una coma después del valor, de lo contrario, Python lo interpretará como el tipo de dato del valor:

```
tupla_un_elemento = (5,) # Tupla con un solo elemento
```

Conjuntos

Un conjunto es una colección **no ordenada** de elementos **únicos**. A diferencia de las listas y las tuplas, los conjuntos no permiten elementos duplicados.

Para crear un conjunto, especificamos sus elementos entre llaves ({}):

```
mi_conjunto = {True, 3.14, None, 'hola', (1, 3)}
```

Importante: Los conjuntos **no pueden** contener elementos mutables como listas, diccionarios e incluso otros conjuntos. Esto es porque los elementos deben ser **inmutables** (hashables).

Ejemplo de uso de un conjunto:

```
s = {1, 2, 3, 4}
s.add(5)      # Agrega el elemento 5 al conjunto
s.discard(2)  # Elimina el elemento 2 del conjunto (sin error si no existe)
print(s)      # Salida: {1, 3, 4, 5}
```

*Aunque los conjuntos son **mutables** en Python (se pueden modificar después de ser creados), existe un tipo similar llamado **frozenset** (conjunto congelado) que es **inmutable**.*

Listas

Una **lista** es similar a una tupla, pero con la diferencia fundamental de que **las listas son mutables**, lo que significa que puedes modificar sus elementos una vez creadas:

```
mi_lista = ['cadena de texto', 15, 2.8, 'otro dato', 25]
```

Al igual que las tuplas, las listas se acceden utilizando su **índice** (el índice comienza en 0):

```
print(mi_lista[1]) # Salida: 15
```

A diferencia de las tuplas, **las listas son mutables**, lo que permite **modificar** sus elementos después de su creación:

```
mi_lista[2] = 3.8 # El tercer elemento ahora es 3.8
```

Además, las listas permiten **agregar nuevos elementos**:

```
mi_lista.append('Nuevo dato') # Se agrega un nuevo dato al final de la lista
```

Diccionarios

A diferencia de las listas y tuplas, que se acceden **únicamente por índice**, los **diccionarios** permiten acceder a los valores mediante una **clave**. La clave se utiliza tanto para **declarar** como para **acceder** a un valor asociado:

```
mi_diccionario = {'clave_1': valor_1, 'clave_2': valor_2, 'clave_7': valor_7}
```

Acceder a un valor en el diccionario se realiza usando su clave:

```
print(mi_diccionario['clave_2']) # Salida: valor_2
```

Un diccionario también permite **eliminar** una entrada utilizando la palabra clave **del**:

```
del(mi_diccionario['clave_2']) # Elimina la entrada con 'clave_2'
```

Al igual que las listas, los diccionarios permiten **modificar** los valores asociados a una clave:

```
mi_diccionario['clave_1'] = 'Nuevo Valor' # Modifica el valor de 'clave_1'
```

Métodos

En Python, las diferentes estructuras de datos (tuplas, listas, diccionarios y conjuntos) tienen métodos propios que permiten manipular sus elementos de diversas maneras. A continuación, se presentan algunos de los métodos conocidos para cada una de estas estructuras:

- Tuplas:
 - **count(x)**: Devuelve el número de veces que el valor **x** aparece en la tupla
 - **index(x)**: Devuelve el índice del primer elemento cuyo valor es **x**
 - Las tuplas no tienen métodos para agregar, eliminar ni modificar elementos. Inmutables.
- Listas:
 - **append(x)**: Agrega un elemento **x** al final de la lista.
 - **sort()**: Ordena la lista en su lugar
- Diccionarios:
 - **get(key)**: Devuelve el valor asociado a la clave **key**. Si la clave no existe, devuelve **None** o un valor predeterminado si se especifica.
 - **items()**: Devuelve una vista de todos los pares clave-valor.
- Conjuntos:
 - **add(x)**: Agrega un elemento **x** al conjunto
 - **clear()**: Elimina todos los elementos del conjunto.