

## Glosario: Estructuras de Datos en Python

---

### 1. Estructura de Datos

Modelo lógico que permite almacenar, organizar y manipular datos de manera eficiente dentro de un programa. En Python, las estructuras de datos integradas ofrecen soluciones para múltiples tipos de problemas y patrones de uso.

---

### 2. Inmutabilidad

Propiedad de ciertos tipos de datos que impide que su contenido sea modificado después de su creación. En Python, los tipos `str` y `tuple` son inmutables, lo que garantiza su integridad a lo largo del programa y permite su uso como claves en diccionarios o elementos en conjuntos.

---

### 3. Cadenas de Texto (`str`)

Tipo de dato inmutable que representa secuencias de caracteres Unicode. Se utilizan ampliamente en procesamiento de datos, entrada de usuario, comunicación y visualización textual. Soportan operaciones de búsqueda, formato, reemplazo y segmentación.

---

### 4. Lista (`list`)

Estructura de datos ordenada y mutable que puede almacenar elementos heterogéneos. Permite operaciones como inserción, eliminación, recorrido, ordenamiento y búsqueda. Es ampliamente utilizada por su flexibilidad en el manejo de datos secuenciales.

---

## 5. Tupla (tuple)

Secuencia ordenada e inmutable de elementos. Ideal para representar estructuras fijas, como coordenadas, pares clave-valor, fechas o cualquier conjunto de datos que no deba ser alterado. Permiten desestructuración y retorno múltiple de valores en funciones.

---

## 6. Diccionario (dict)

Estructura de datos asociativa compuesta por pares clave-valor. Permite acceso rápido a los valores mediante claves únicas. Es mutable y extremadamente eficiente para búsquedas, indexación y almacenamiento de configuraciones o datos relacionales.

---

## 7. Conjunto (set)

Colección no ordenada de elementos únicos y mutables. Es útil para eliminar duplicados, realizar operaciones de teoría de conjuntos como unión, intersección o diferencia, y verificar pertenencia con alta eficiencia.

---

## 8. Hashing

Técnica utilizada para asignar de forma determinista una clave a una posición de almacenamiento. Es la base del acceso eficiente en estructuras como diccionarios y conjuntos en Python, garantizando tiempos de búsqueda y recuperación cercanos a  $O(1)$ .

---

## 9. Colección Heterogénea

Estructura de datos capaz de almacenar elementos de diferentes tipos dentro de una misma secuencia. Python permite esta flexibilidad en listas, tuplas y diccionarios, lo que las hace muy poderosas en programación dinámica.

---

## 10. Complejidad Computacional

Medida del rendimiento de un algoritmo o estructura en términos de tiempo y espacio, según el tamaño de la entrada. En estructuras de datos, afecta operaciones como acceso, inserción, eliminación y búsqueda.

---

## 11. Iterabilidad

Propiedad de una estructura que permite ser recorrida secuencialmente mediante bucles o funciones iteradoras (`for`, `map`, `enumerate`). Todas las estructuras de datos nativas de Python son iterables.

---

## 12. Acceso Posicional

Acceso a los elementos mediante su índice numérico dentro de una secuencia. Utilizado en listas, cadenas y tuplas. Permite operaciones rápidas pero exige conocer la posición exacta del dato.

---

## 13. Acceso por Clave

Mecanismo de acceso a un valor asociado a una clave única dentro de una estructura asociativa como un diccionario. Mejora la semántica del acceso y optimiza la eficiencia frente a estructuras secuenciales.

---

## 14. Mutabilidad Controlada

Capacidad de modificar una estructura sin afectar otras referencias externas, o limitando los cambios a ciertos elementos. Python permite combinar estructuras mutables e inmutables según el nivel de control requerido.

---

## 15. Desestructuración de Secuencias

Técnica que permite asignar múltiples valores desde una lista o tupla directamente a variables individuales. Facilita el retorno múltiple en funciones y la legibilidad del código.