

Estructuras de datos

AYUDANTÍA PYTHON CIENTÍFICO

Listas

Acepta repetidos, elementos mutables y accesibles por índice.

- `[]` o `list()`
- `lista.append(x)`: Agrega elemento x al final de la lista.
- `lista.insert(índice,x)`: Inserta el elemento en una posición o índice dado
- `lista.remove(x)`: Quita el primer elemento de valor x de la lista.
- `lista.pop([i])`: Elimina el elemento en el índice dado.
- `lista.clear()`: Elimina todos los elementos de la lista.
- `lista.count(x)`: Retorna el número de veces que x aparece en la lista
- `lista.sort()`: Ordena los elementos de la lista, puede tener argumento, de la forma `lista.sort(*, key=None, reverse=False)` para personalizar orden.
- `lista.reverse(x)`: Invierte los elementos de la lista.
- `lista.copy(x)`: Retorna una copia de la lista.

Tuplas

Acepta repetidos, elementos inmutables y accesibles por índice.

- `()` o `tuple()`
- Es inmutable. Solo es posible inicializarla en la creación, luego no se puede cambiar ni agregar índices.

Conjuntos

Colección no ordenada y sin elementos repetidos.

- `set()`
- Para usos de eliminación de datos duplicados y verificación de pertenencia
- `conjunto.add(x)`: Agregar elemento x
- `conjunto.discard(x)`: Remover elemento x

Diccionarios

Conjunto de pares *clave : valor* con el requerimiento de que las claves sean únicas. Claves inmutables.

- `{ }` o `dict()`
- Para usos de eliminación de datos duplicados y verificación de pertenencia
- `list.(diccionario)`: Retorna lista con todas las claves usadas en el diccionario
- Para comprobar si una clave está en el diccionario usar la palabra clave `"in"`
- `diccionario.items()`: Se puede obtener al mismo tiempo la clave y el valor correspondiente usando este método.

[Link hacia documentación oficial](#)