

Resumen: Estructuras de Datos en Python

José Ángel Olmedo Guevara

¿Qué es una estructura de datos?

Forma de organizar y almacenar datos en memoria para optimizar operaciones como acceso, inserción, eliminación y recorrido. En Python contamos con estructuras integradas y otras en el módulo `collections`.

Contenedores básicos en Python

- **list**: Arreglo dinámico ordenado, admite duplicados.
- **tuple**: Secuencia ordenada e inmutable.
- **set**: Conjunto no ordenado sin duplicados.
- **dict**: Mapa clave-valor, ordenado desde Python 3.7.

Contenedores avanzados (collections)

- **deque**: Cola doblemente terminada, inserción/eliminación $O(1)$ en ambos extremos.
- **defaultdict**: Diccionario con valor por defecto al acceder claves inexistentes.
- **Counter**: Contador de elementos hashables.
- **namedtuple**: Tupla nombrada para acceder campos por nombre.

Ejemplos de uso

list

```
# Lista dinmica
nums = [1, 2, 3]
nums.append(4)      # [1,2,3,4]
print(nums[2])      # 3
```

tuple

```
# Tupla immutable
coords = (10, 20)
# coords[0] = 5 --> Error
```

set

```
# Conjunto nico
s = {1,2,2,3}
s.add(4)          # {1,2,3,4}
print(2 in s)     # True
```

dict

```
# Mapa clave-valor
edades = {'Ana':25, 'Luis':30}
print(edades['Ana']) # 25
edades['Ana'] = 26
```

deque y Counter

```
from collections import deque, Counter, defaultdict

dq = deque([1,2,3])
dq.appendleft(0)    # deque([0,1,2,3])
dq.pop()            # 3

cnt = Counter('abracadabra')
print(cnt['a'])      # 5

dd = defaultdict(int)
dd['x'] += 1         # 1
```

Comparativa rápida

Necesitas...	Usa
Secuencia mutable ordenada	list
Secuencia inmutable	tuple
Conjunto único de elementos	set
Acceso clave-valor	dict
Operaciones O(1) en extremos	collections.deque
Contar elementos	collections.Counter
Valores por defecto	collections.defaultdict
Tupla con campos nombrados	collections.namedtuple

Ventajas y Desventajas

list

Ventajas: - Dinámico, permite duplicados, acceso $O(1)$.

Desventajas: - Inserción/eliminación central $O(n)$.

tuple

Ventajas: - Inmutable, seguro para claves de dict.

Desventajas: - No cambia en tiempo de ejecución.

set

Ventajas: - Operaciones $O(1)$ promedio, sin duplicados.

Desventajas: - No ordenado, solo iteración.

dict

Ventajas: - Clave \rightarrow valor $O(1)$ promedio.

Desventajas: - Uso de memoria mayor que listas.

Notas adicionales

- Todas las estructuras integradas soportan comprensión (`[x for x in ...]`).
- `list`, `dict` y `set` son iterables y admiten funciones como `len()`, `in`, `min()`, `max()`.