Estructuras de datos avanzadas

Python

Nicky Alexander Flores bustamante

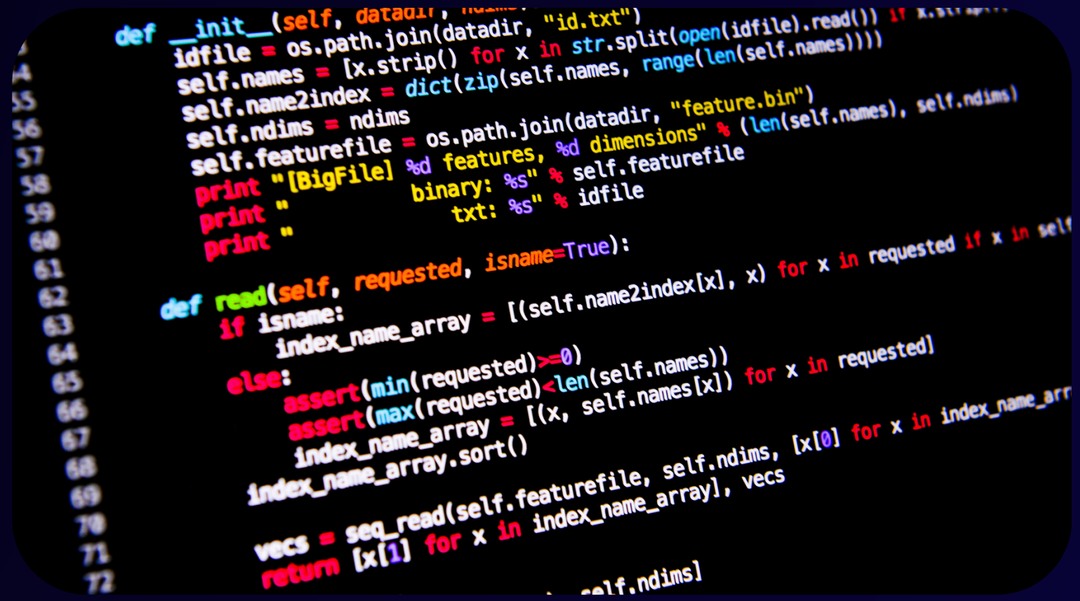
Juan Andres Menendez Villarraga

David Fernando Gomez Aristizabal

¿Que son?

Son aquellas que van más allá de las estructuras básicas (listas, tuplas, diccionarios y conjuntos) e incluyen constructores diseñados para resolver problemas mas complejos de almacenamiento y acceso eficiente a la información.

Estas estructuras de datos incluyen Counter, defaultdict,deque entre otras



Estas estructuras suelen estar disponibles a través de módulos específicos de la biblioteca estándar (como collections o heapq) o se implementan directamente en código para necesidades particulares.

Ejemplos de estructuras avanzadas

Counter

Counter es una subclase de diccionario que se utiliza para contar la frecuencia de elementos en un iterable (listas, cadenas, tuplas, etc.).

es decir, cuenta cuántas veces aparece cada cosa en una lista o una frase, lo cual nos resulta muy útil para saber los elementos más comunes, hacer estadísticas rápidas, etc.

Puedes usarlo no solo con listas, sino también con cadenas de texto, tuplas, etc y ademas cuenta con varios métodos útiles como .most_common(n) para saber cuáles elementos aparecen más veces.

Ejemplo

```python
from collections import Counter

lista = ['manzana', 'pera', 'manzana', 'naranja', 'pera', 'manzana']
contador = Counter(lista)
print(contador)
# Resultado: Counter({'manzana': 3, 'pera': 2, 'naranja': 1})

# Puedes acceder a los elementos más comunes:
print(contador.most_common(1))  # [('manzana', 3)]
```

Counter nos ayuda contando cuántas veces aparece cada elemento.

Ejemplos de estructuras avanzadas

defaultdict

En Python, defaultdict es una subclase de dict, que esta incluido en python de el modulo collections.

Es un tipo especial de diccionario en Python que crea automáticamente un valor para una clave que no existe cuando intentas usarla.

Especifica una función de fábrica que proporcionará el valor predeterminado para las nuevas claves. Esta función de fábrica podría ser int, lista, str o cualquier otro objeto llamable

Ejemplo

```python
from collections import defaultdict

palabras = ['gato', 'perro', 'raton']
contador = defaultdict(int)  # Valor por defecto: 0
for palabra in palabras:
    contador[palabra] += 1    # No hay que comprobar si existe la clave
print(contador)

# Así, no tienes que preocuparte si la clave no existe, porque defaultdict la crea para ti
automáticamente.
```

defaultldict evita errores al acceder a claves no existentes y facilitar la agregación de elementos.

Ejemplos de estructuras avanzadas

deque

Queue sigue estrictamente el modelo FIFO (el primero en entrar es el primero en salir), ideal para colas de tareas.
Deque es más flexible: permite tanto FIFO como LIFO, funcionando como cola o pila.

Queue bloquea automáticamente si está vacía o llena, útil para sincronización entre hilos.
Deque, en cambio, no bloquea y lanza errores si está vacía, lo que da más control y mejor manejo manual de errores.

Además, Queue no permite acceder por índices ni insertar al inicio, mientras que Deque sí lo permite.

Ejemplo

```python
dq = deque([10, 20, 30])

# Añade elementos en el final
dq.append(40) # [10, 20, 30, 40]

# Añade elementos al principio -> Esto no se puede en una cola normal
dq.appendleft(5) # [5, 10, 20, 30, 40]

# Le añade cada uno a la izquierda
dq.extendleft([0, 5]) # [5, 0, 5, 10, 20, 30, 40]

# remove method -> Queue no lo permite
dq.remove(20) # [5, 0, 5, 10, 30, 40]

# Elimina el último elemento
dq.pop() # [5, 0, 5, 10, 30]

# Elimina el primer elemento -> Queue no lo permite
dq.popleft() # [0, 5, 10, 30]

# Elimina toda la cola -> Queue no lo permite
dq.clear()  # []
```

deque se utiliza para implementar colas, pilas o realizar operaciones eficientes en ambos extremos de una lista.

Conclusiones

En conclusion:

- Counter: se utiliza para contar frecuencias de elementos.
- defaultdict: se utiliza para diccionarios con valores por defecto automáticos.
- deque: se utiliza para colas o pilas con operaciones eficientes en ambos extremos.

Working
Together in Code

Gracias por su atencion!