

### Qué es collections

collections es un módulo en Python que contiene una variedad de clases que ofrecen diferentes tipos de estructuras de datos útiles y eficientes

### Qué es deque

deque es un tipo de datos en Python que representa una lista doblemente ligada. Esto significa que los elementos pueden ser agregados o eliminados tanto desde el principio como desde el final de la lista en tiempo constante

### Métodos para agregar elementos

```
from collections import deque
d = deque()
# Agregar elementos a la derecha (final) del deque
d.append(1)
d.append(2)
d.append(3)
print(d) # deque([1, 2, 3])
# Agregar elementos a la izquierda (inicio) del deque
d.appendleft(0)
d.appendleft(-1)
d.appendleft(-2)
print(d) # deque([-2, -1, 0, 1, 2, 3])
# Agregar varios elementos al final de la lista
d = deque([1, 2, 3])
d.extend([4, 5, 6])
print(d) # deque([1, 2, 3, 4, 5, 6])
# Agregar varios elementos al inicio de la lista
d = deque([1, 2, 3])
d.extendleft([4, 5, 6])
print(d) # deque([6, 5, 4, 3, 2, 1])
```

### Más métodos...

```
#reverse invierte los elementos en el deque.
d = deque([1, 2, 3, 4, 5])
d.reverse()
print(d) # deque([5, 4, 3, 2, 1])
#maxlen: Establece un tamaño máximo para el deque.
Si se supera, los elementos más antiguos se eliminan automáticamente.
d = deque([1, 2, 3, 4, 5], maxlen=3)
print(d) # deque([3, 4, 5])
d.append(6)
print(d) # deque([4, 5, 6])
```

### ¿Cómo se usa?

```
from collections import deque
d = deque()
```

### Métodos para eliminar elementos

```
from collections import deque
d = deque([1, 2, 3, 4, 5])
# Eliminar un elemento desde el final del deque
popped_element = d.pop()
print(popped_element) # 5
print(d) # deque([1, 2, 3, 4])
# Eliminar un elemento desde el principio del deque
popped_element = d.popleft()
print(popped_element) # 1
print(d) # deque([2, 3, 4])
# Para eliminar todos los elementos del deque
d = deque([1, 2, 3])
d.clear()
print(d) # deque([])
# Eliminar la primera ocurrencia de un elemento especificado
d = deque([1, 2, 3, 4, 5, 4, 3, 2, 1])
d.remove(3)
print(d) # deque([1, 2, 4, 5, 4, 2, 1])
# Tratar de eliminar un elemento que no está en el deque
try:
    d.remove(6)
except ValueError as ve:
    print(ve) # 'deque.remove(x): x not in deque'
```

### Otro más...

```
#rotate() para rotar los elementos en la lista a la izquierda o a la derecha.
d = deque([1, 2, 3, 4, 5])
d.rotate(2)
print(d) # deque([4, 5, 1, 2, 3])
d.rotate(-1)
print(d) # deque([5, 1, 2, 3, 4])
```



