

## 13 – Stacks

Uno **stack** è una struttura dati di tipo **LIFO** (*Last In, First Out*), cioè l'ultimo elemento inserito è il primo ad essere rimosso. In Python, le **liste** possono essere facilmente usate come stack grazie ai metodi `append()` e `pop()`.

## Operazioni principali

Le operazioni fondamentali di uno stack sono:

1. **Push** – aggiungere un elemento in cima allo stack.
2. **Pop** – rimuovere l'elemento in cima.
3. **Peek/Top** – leggere l'elemento in cima senza rimuoverlo.
4. **IsEmpty** – verificare se lo stack è vuoto.

## Esempio semplice con lista

```
stack = []

# Push
stack.append(10)
stack.append(20)
stack.append(30)
print(stack) # [10, 20, 30]

# Pop
ultimo = stack.pop()
print(ultimo) # 30
print(stack) # [10, 20]

# Peek (ultimo elemento senza rimuoverlo)
cima = stack[-1]
print(cima) # 20
```

## Verificare se lo stack è vuoto

```
if not stack:  
    print("Lo stack è vuoto")  
else:  
    print("Lo stack contiene elementi")
```

## Usi comuni degli stack

- Gestione del **backtracking**, come nei percorsi o puzzle.
- Implementazione di **undo/redo** in editor di testo.
- Valutazione di **espressioni matematiche** (notazione polacca).
- Navigazione in strutture **ricorsive** come alberi.

## Vantaggi

- Operazioni molto veloci (  $O(1)$  per push/pop).
- Implementazione semplice con le liste di Python.
- Concetto chiaro e utile in molti algoritmi.

Vuoi procedere con il **paragrafo 14 – Queues**?