

## 18 – Sets

Un **set** in Python è una collezione **non ordinata di elementi unici**. I set sono molto utili per operazioni di **matematica insiemistica**, come unioni, intersezioni e differenze, e per eliminare rapidamente duplicati da una sequenza.

## Creazione di un set

Si può creare un set usando parentesi graffe `{}` o la funzione `set()`:

```
frutti = {"mela", "banana", "arancia"}  
print(frutti)  # l'ordine può variare  
  
vuoto = set()  # attenzione: {} crea un dizionario vuoto
```

## Caratteristiche principali

- Gli elementi sono **unici**: duplicati vengono automaticamente eliminati.
- Non ordinati: non puoi accedere agli elementi tramite indice.
- Mutabili: puoi aggiungere o rimuovere elementi.

## Aggiungere elementi

```
frutti.add("kiwi")  
frutti.update(["pera", "uva"]) # aggiunge più elementi  
print(frutti)
```

## Rimuovere elementi

```
frutti.remove("banana")    # solleva KeyError se non esiste
frutti.discard("mango")    # non solleva errore se l'elemento non c'è
ultimo = frutti.pop()      # rimuove un elemento casuale
frutti.clear()             # rimuove tutti gli elementi
```

## Operazioni tra set

I set supportano operazioni matematiche:

```
A = {1, 2, 3, 4}
```

```
B = {3, 4, 5, 6}
```

```
print(A | B) # unione -> {1,2,3,4,5,6}
```

```
print(A & B) # intersezione -> {3,4}
```

```
print(A - B) # differenza -> {1,2}
```

```
print(A ^ B) # differenza simmetrica -> {1,2,5,6}
```

## Verifica appartenenza

```
print(2 in A) # True  
print(5 not in A) # True
```

## Vantaggi

- Eliminano automaticamente i duplicati.
- Operazioni matematiche su insiemi molto efficienti.
- Ideali per problemi di appartenenza, confronto e deduplicazione.



Vuoi procedere con il **paragrafo 19 – Dictionaries**?