

# Funzioni e procedure

- Suddividono il programma in moduli
- Migliorano leggibilità e manutenzione del codice
- Consentono riuso del codice
- Fondamentali in programmazione strutturata

# Funzioni

- Blocco di codice che restituisce un valore
- Chiamabile più volte nel programma
- Esempio Python:

```
def somma(a, b):  
    return a + b
```

- Uso: `risultato = somma(3, 5)`

# Procedure

- Blocco di codice che esegue operazioni ma non restituisce valore
- Chiamata per svolgere compiti specifici
- Esempio Python:

```
def stampa_messaggio():
    print("Ciao!")
```

- Chiamata: `stampa_messaggio()`

# Parametri

- Variabili passate a funzione o procedura
- Consentono personalizzazione delle operazioni
- Possono essere obbligatori o opzionali
- Permettono flessibilità e riuso

# Valore di ritorno

- Funzioni possono restituire valori
- Utili per calcoli, trasformazioni e decisioni
- Può essere assegnato a variabili
- Es.: risultato = somma(3,5)

# Vantaggi

- Codice modulare e organizzato
- Facilita test e debugging
- Riduce ripetizione di codice
- Base della programmazione avanzata

# Procedure ricorsive

- Funzioni che chiamano se stesse
- Utili per problemi suddivisibili in sottoproblemi
- Esempio classico: calcolo fattoriale
- Potenti ma da usare con attenzione

# Importanza in progetti reali

- Strutture modulabili e riutilizzabili
- Facilità di collaborazione tra programmati
- Essenziali in software complessi
- Fondamentali per qualità e manutenibilità