

# 09. Funzioni

Corso di Algoritmi e Linguaggi di Programmazione Python/C

# Outline

- Definizione di funzione
- Riutilizzo del codice
- Modularità
- Ambito di una variabile

# Definizione di funzione

- Costrutto che permette di raggruppare una o più istruzioni eseguite più volte all'interno del nostro programma
- Due parti principali: **firma** e **corpo**

```
// Questa è la firma!
tipo_ritorno nome_funzione(tipo_par_1 par_1, tipo_par_2 par_2)
{
    // Questo è il corpo
    istr_1;
    istr_2;
    tipo_ritorno valore_ritorno = istr_3;
    return valore_ritorno;
}
```

# Riutilizzo del codice

- Le funzioni servono a minimizzare il codice scritto
- Risultati immediati:
  - *Minor numero di errori*
  - *Coerenza*

```
def ipotenusa(c_1, c_2):  
    c_1_quad = c_1**2  
    c_2_quad = c_2**2  
    return (c_1_quad + c_2_quad)**1/2  
}  
  
if __name__ == "__main__":  
    a = 3  
    b = 4  
    i_1 = ipotenusa(a, b)  
    c = 6  
    d = 8  
    i_2 = ipotenusa(c, d)
```

# Modularità

- Ogni funzione deve essere intesa come *atomica*.
  - *In pratica, le elaborazioni compiute all'interno della funzione devono, per quanto possibile, essere indipendenti dal resto del programma.*
- Questo rende il programma *modulare*.
  - *Potremo usare la funzione ipotenusa in altri programmi, o magari cambiare esclusivamente questa se dovesse insorgere un errore.*

```
def ipotenusa(c_1, c_2):  
    return (c_1 + c_2)**1/2  
}  
  
if __name__ == "__main__":  
    a = 3  
    b = 4  
    i_1 = ipotenusa(a, b)  
    c = 6  
    d = 8  
    i_2 = ipotenusa(c, d)
```

# Ambito di una variabile

- Ad ogni variabile è associato un *ambito* di validità.
- Ciò significa che una data variabile può operare esclusivamente all'interno del suo ambito.
- Una variabile definita all'interno di una funzione o di un ciclo ha ambito *locale*, mentre una definita all'esterno di ogni funzione ha ambito solitamente *globale*.

```
numero_esami = 20;
miei_voti = lista_miei_voti;

float calcolo_voto_accesso_laurea(int[] voti_esami):
    somma_voti = 0;
    for i che va da 1 a numero_esami:
        somma_voti = somma_voti + voto_esame_i;
    endfor
    voto_medio = somma_voti / numero_esami;
    voto_accesso = voto_medio / 3 * 11;
    return voto_accesso;
```

# Ambito di una variabile – Analisi dell'esempio

- Le variabili **numero\_esami** e **miei\_voti** sono variabili *globali*.
  - Ciò implica che possono essere accedute all'interno della funzione `calcola_voto_accesso_laurea`.
- Le variabili **somma\_voti**, **voto\_medio** e **voto\_accesso** sono *variabili locali* relativamente all'ambito definito dalla funzione `calcola_voto_accesso_laurea`.
  - Non sono accessibili dall'esterno della funzione, ma rimangono accessibili nell'ambito individuato dal ciclo `for`.
- La variabile **voto\_esame\_i** è accessibile esclusivamente all'interno dell'ambito definito dal ciclo `for`.

```
numero_esami = 20;
miei_voti = lista_miei_voti;

float calcola_voto_accesso_laurea(int[] voti_esami):
    somma_voti = 0;
    for i che va da 1 a numero_esami:
        somma_voti = somma_voti + voto_esame_i;
    endfor
    voto_medio = somma_voti / numero_esami;
    voto_accesso = voto_medio / 3 * 11;
    return voto_accesso;
```

# Domande?

42