

12. Funzioni

Corso di Informatica

Corso di Laurea in Matematica (D.M. 270/04) - A.A. 2020/2021

Angelo Cardellicchio

angelo.cardellicchio@uniba.it

02/11/2020

Outline

- Definizione di funzione
- Riutilizzo del codice
- Modularità

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**
 - *Se dovesse cambiare il teorema di Pitagora, dovremo soltanto modificare la funzione **ipotenusa**!*

```
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)
```

Domande?

42