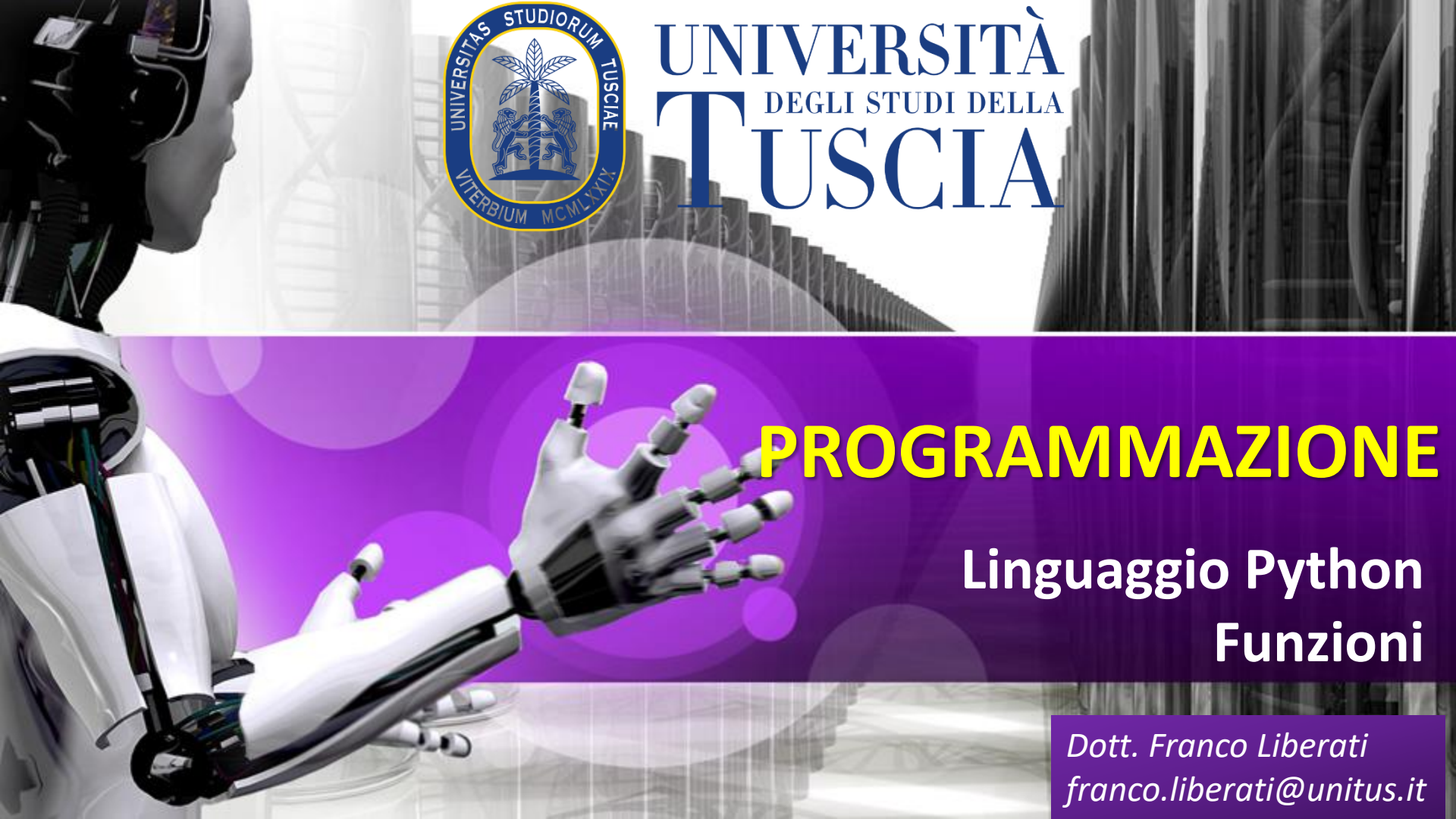


UNIVERSITÀ DEGLI STUDI DELLA TUSCIA

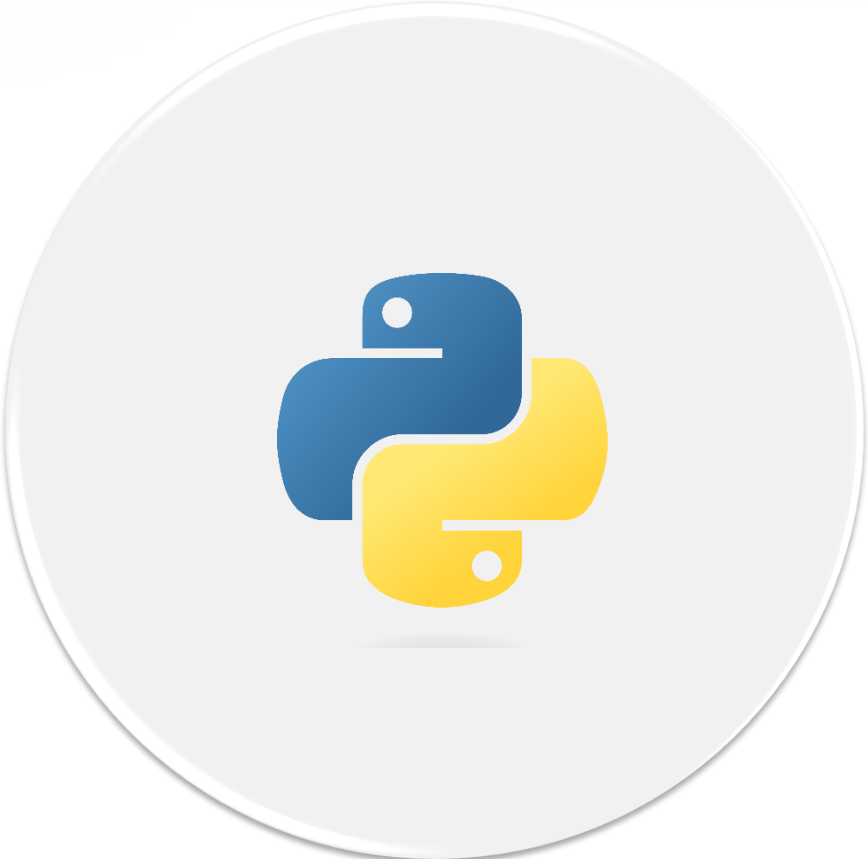
PROGRAMMAZIONE

Linguaggio Python
Funzioni

Dott. Franco Liberati
franco.liberati@unitus.it

LINGUAGGIO PYTHON

Argomenti del corso

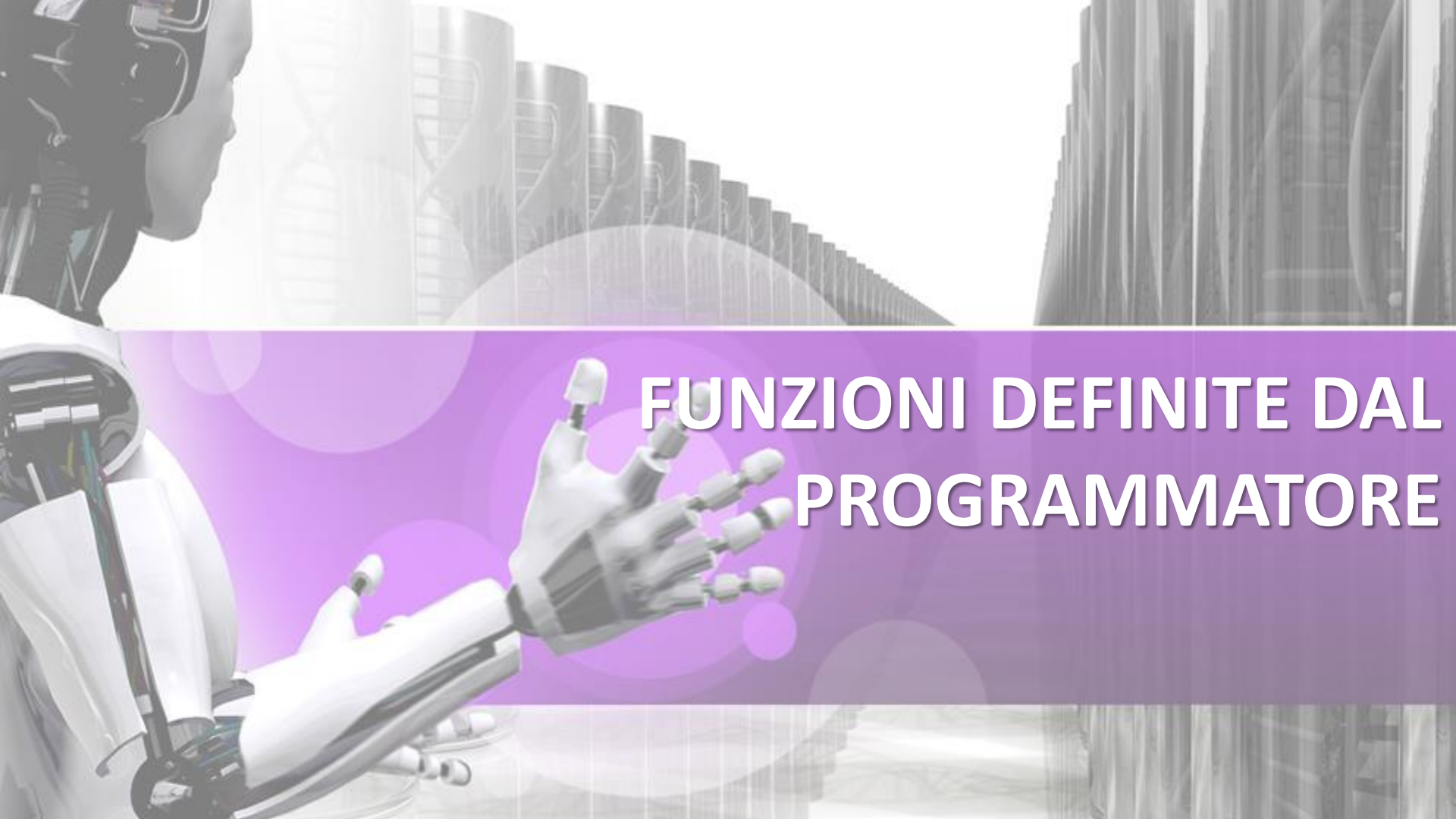


PYTHON

FUNZIONI

- Una funzione è un blocco organizzato di codice che è utilizzato per eseguire un preciso compito
- L'utilizzo di funzioni garantisce maggiore modularità al codice e una maggiore riutilizzabilità dello stesso

- Python dispone di **funzioni built-in**, p.e. print(), help(), dir() etc, ma è possibile definire nuove funzioni: **user-defined-function**



FUNZIONI DEFINITE DAL PROGRAMMATORE

PYTHON

FUNZIONI: definizione

- La definizione di una funzione in python ha la seguente sintassi:

def *NomeFunzione*(arg1,arg2,…,argN):

La parola chiave **def** è seguita dal nome della funzione e dalla lista dei suoi argomenti

Al prototipo della funzione segue il corpo della definizione

def *NomeFunzione*(arg1,arg2,…,argN):
    istruzione1
    …
    istruzionen

ESEMPIO

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

a = int(input('Inserisci il primo numero: '))
b = int(input('Inserisci il secondo numero: '))
result=sum(a, b)

print('La somma di {a} con {b} è {result}')
```

PYTHON

FUNZIONI: definizione

- In Python è importante distinguere tra oggetti mutabili e immutabili per capire il passaggio di argomenti ad una funzione.

- Gli oggetti immutabili (stringhe, tuple, numeri) sono **passati per valore**: non si modifica il valore degli oggetti immutabili, ma si preleva solo il loro valore che sono manipolati dalla funzione
- Gli oggetti mutabili (liste, dictionary) sono **passati per referenza**: si modifica il valore degli oggetti mutabili (una modifica nella funzione comporta il cambiamento anche dopo la chiamata a funzione)

- Inoltre le variabili definite nel corpo di una funzione sono utilizzate solo all'interno (*local scope*).

ESEMPIO PASSAGGI PER VALORE (STRINGA)

```
def PrimiQuattroCaratteri(stringa):
        return(stringa[0:4])

str="Richard Benson"
print(str)
Richard Benson
result=PrimiQuattroCaratteri(str)
print(result)
Rich
print(str)
Richard Benson
```

PYTHON

FUNZIONI: definizione

- In Python è importante distinguere tra oggetti mutabili e immutabili per capire il passaggio di argomenti ad una funzione.

- Gli oggetti immutabili (stringhe, tuple, numeri) sono **passati per valore**: non si modifica il valore degli oggetti immutabili, ma si preleva solo il loro valore che sono manipolati dalla funzione
- Gli oggetti mutabili (liste, dictionary) sono **passati per referenza**: si modifica il valore degli oggetti mutabili (una modifica nella funzione comporta il cambiamento anche dopo la chiamata a funzione)

- Inoltre le variabili definite nel corpo di una funzione sono utilizzate solo all'interno (*local scope*).

ESEMPIO PASSAGGI PER REFERENZA (LISTA

```
def RimuoviDuplicati(l1):
        for i in l1:
            c=l1.count(i)
            if c>1:
                l1.remove(i)

lista=[1,2,3,3,4,4,7,8]
print ("Lista prima della chiamata a funzione: ", lista)
[1,2,3,3,4,4,7,8]
RimuoviDuplicati(lista)
print ("Lista dopo della chiamata a funzione: ", lista)
[1,2,3,4,7,8]
```

PYTHON

FUNZIONI: parametri

- Il numero di parametri che devono essere passati ad una funzione deve essere conforme al prototipo della funzione (*required parameters*)

ESEMPIO NUMERO DI PARAMETRI NON CORRETTO

```
import math
def CalcoloEquazioneSecondoGrado(a,b,c):
    x1=0
    x2=0
    x3=False
    if (a!=0):
        if (b**2-4*a*c)>=0:
            x1=(-b+math.sqrt(b**2-4*a*c))/2*a
            x2=(-b-math.sqrt(b**2-4*a*c))/2*a
            x3=True
    return x1,x2,x3

val1=int(input("Valore A: "))
val2=int(input("Valore B: "))
val3=int(input("Valore C: "))
ris1,ris2,ris3=CalcoloEquazioneSecondoGrado(val1,val2,val3)
if (ris3==True):
    print ("X1: ", ris1," X2: ",ris2)
else:
    print("soluzioni non ammissibili nei reali")
ris1,ris2,ris3=CalcoloEquazioneSecondoGrado(val1,val2)
ERRORE
TypeError: CalcoloEquazioneSecondoGrado() missing 1 required positional argument: 'c'
```

PYTHON

FUNZIONI: parametri

- È possibile definire degli argomenti di default per una funzione, associando al nome del parametro un valore di default
- I parametri di default seguono quelli obbligatori
- Specificando dei parametri di default la chiamata a funzione può essere eseguita specificando meno parametri di quelli presenti nel prototipo

ESEMPIO PARAMETRI DI DEFAULT

```
def SempliceProdotto(a=7,b=8):
    return a*b

ris1=SempliceProdotto()
print(ris1)
56
ris2=SempliceProdotto(4)
print(ris2)
32
ris3=SempliceProdotto(4,5)
print(ris3)
20
```

PYTHON

FUNZIONI: parametri

- È possibile chiamare una funzione utilizzando delle parole chiavi che definiscono gli argomenti (*keyword arguments*) con la sintassi

  keyword=value

- L'utilizzo dei keyword arguments permette di specificare i parametri senza seguire l'ordine posizionale

ESEMPIO PARAMETRI DI DEFAULT

```
def SempliceProdotto(pippo=7,pluto=8, paperino=10):
    return pippo*pluto*paperino


ris1=SempliceProdotto(pluto=10,pippo=4)
print(ris1)
400
ris2=SempliceProdotto(paperino=1,pippo=2,pluto=4)
print(ris2)
8
ris3=SempliceProdotto(paperino=6)
print(ris3)
336
```

PYTHON

FUNZIONI: parametri

- Python consente di utilizzare come parametro di funzione una lista di lunghezza variabile di argomenti.
- Un asterisco * preposto al nome del parametro indica una lista di argomenti di lunghezza variabile
- La sintassi è:

```
def func_var_arg(a,b,c,…,*args)
```

gli argomenti sono mantenuti in una tupla.

ESEMPIO NUMERO PARAMETRI VARIABILE

```
def lista_di_argomenti(*args):
    for el in args:
        print(el)

lista_di_argomenti(1)
1
lista_di_argomenti(1,2,3,4)
1
2
3
4
lista_di_argomenti(1,2,3,4,5,6,7,8)
1
2
3
4
5
6
7
8
```

PYTHON

FUNZIONI: Esercizio

Definire una funzione che prenda un numero di valori non precisato e ne calcoli, minimo, massimo e media

PYTHON

FUNZIONI: soluzione

```
def min_max_average(*args):
    media =0
    n=0
    for el in args:
        n=n+1
        media=media+el
    media=media/float(n)
    minimo=min(args)
    massimo=max(args)
    print ("Media ", media)
    print ("Min ", minimo)
    print ("Max ", massimo)

min_max_average(3,2,1)
Media  2.0
Min  1
Max  3
min_max_average(2)
Media  2.0
Min  2
Max  2
```

PYTHON

FUNZIONI: parametri

- Python permette di utilizzare una funzione come argomento di un'altra funzione

ESEMPIO FUNZIONE COME PARAMETRO

```
from math import *


def DistanzaLineare(x1,y1,x2,y2):
    return(x1-x2,y2-y1)

def Distanza(DistanzaLineare,x1,y1,x2,y2):
    ris1,ris2=DistanzaLineare(x1,y1,x2,y2)
    return(sqrt(ris1**2+ris2**2))

d=Distanza(DistanzaLineare,1,1,3,1)
print(d)
2
```

PYTHON

FUNZIONI: parametri

- Un uso della funzione come argomento è ben visibile nella

Approssimazione della derivata seconda in un punto

Derivata seconda

- Ricaviamo per la derivata seconda:

$$f''(x_i) = \frac{f(x_i - h) - 2f(x_i) + f(x_i + h)}{h^2} - \frac{h^2}{12} f^{(4)}(\xi)$$

- Cioè

$$f''(x_i) \approx \frac{f(x_{i-1}) - 2f(x_i) + f(x_{i+1})}{h^2}$$

con l'errore del second'ordine in h.

Tratto da Lucio Demeio
https://www.dipmat.univpm.it/~demeio/public/Analisi_Numerica/Lezioni%20stampabili/4%20-%20Derivate.pdf

ESEMPIO FUNZIONE COME PARAMETRO

```
from math import *
def cubo(x):
        return x**3

def diff2(f,x,h=1e-6):
        r=(f(x-h)-2*f(x)+f(x+h))/float(h*h)
        return r

print ("Diff2 di x^3 per x=3 ", diff2(cubo,3))
Diff2 di x^3 per x=3  18.001600210482138
print ("Diff2 di sin(x) per x=pi_greco ", diff2(sin,pi))
Diff2 di sin(x) per x=pi_greco  0.0
```

PYTHON

FUNZIONI: ritorno

- La keyword **return** permette di specificare dei valori di ritorno di una funzione al programma chiamante

- In Python è possibile specificare più parametri di ritorno utilizzando la virgola come separatore dei valori da riportare

ESEMPIO RITORNO DA FUNZIONE

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

ris1=SempliceSomma(3,4)
print(ris1)
7
def QuadratoCubo(a):
                quad=a*a;
                cub=a*a*a
                return quad,cub

ris1,ris2=SempliceProdotto(5)
print(ris1)
25
print(ris2)
125
```

PYTHON

SCOPO DI VARIABILI IN PYTHON (SCOPING)

- Lo scope di variabili definisce la visibilità di una variabile all'interno di un codice (un modulo).
- Con namespace si intende una collezione di nomi (funzioni, variabili, costanti) che sono definite in un codice e possono essere riutilizzati in un altro codice (un esempio sono le funzioni built-in, cioè quelle messe a disposizione da Python al programmatore, es.: sqrt).
- Ciascun modulo definisce un proprio namespace.
- In Python esistono tre livelli di namespace:
  - Built-in namespace
  - Module namespace definito in un modulo o in un file/codice
  - Local namespace, definito da una funzione o da una classe
- L'interprete durante l'esecuzione ricerca i nomi nell'ordine:
- Local namespace $\rightarrow$ Global namespace $\rightarrow$Built-in namespace

PYTHON

SCOPO DI VARIABILI IN PYTHON ESEMPIO

```
x=int(raw_input("Inserisci un numero: "))
def square(x):
        x=x*x
        print "Dentro la funzione il valore di x è: ",x

square(x)
print "Fuori dalla funzione il valore di x è: ", x
```

FUNZIONE BUILT IN

FUNZIONE MODULE

VARIABILE LOCALE

```
Inserisci un numero: 5
Dentro la funzione il valore di x è: 25
Fuori dalla funzione il valore di x è: 5
```

PYTHON

SCOPO DI VARIABILI IN PYTHON (SCOPING)

- Utilizzando la parola chiave global è possibile modificare una variabile globale (opzione spesso desiderata quando si vuole agire su una variabile all'interno di una funzione)

```
def stampa():
        gg="def"
        ll="ghi"
        print "Dentro stampa()", gg+ll

gg="abc"
print "Valore di gg prima della funzione stampa:", gg
stampa()
print "Valore di gg dopo la funzione stampa:", gg
```

```
Valore di  gg prima della la funzione stampa: abc
Dentro stampa() : defghi
Valore di  gg dopo la funzione stampa: abc
```

```
def stampa():
        global gg
        gg="def"
        ll="ghi"
        print "Dentro stampa()",gg+ll

print "Valore di gg prima della funzione stampa:", gg
gg="abc"
stampa()
print "Valore di gg prima della funzione stampa:", gg
```

```
Valore di  gg prima della la funzione stampa: abc
Dentro stampa() : defghi
Valore di  gg dopo la funzione stampa: def
```

PYTHON

FUNZIONI DI PYTHON

- Il caricamento di un modulo avviene tramite la keyword import. Esistono però due modalità differenti:

  import module

  from module **import** *|fun

- Nel primo caso viene importato il modulo ma non viene modificato il namespace corrente
- Nel secondo caso vengono importate funzioni e attributi da module al namespace corrente

ESEMPIO IMPORT MODULE

```
import os
os.path.basename(os.getcwd())
'Python25'
```

ESEMPIO FROM MODULE IMPORT

```
from os import *
path.basename(getcwd())
'Python25'
```


Fine