



UNIVERSITÀ
DEGLI STUDI DELLA
TUSCIA

PROGRAMMAZIONE

Linguaggio Python
Flusso di esecuzione

Dott. Franco Liberati
franco.liberati@unitus.it

LINGUAGGIO PYTHON

Argomenti del corso





PYTHON

FLUSSO DI ESECUZIONE

- ❑ Le strutture di controllo, ovvero l'insieme dei costrutti che servono a gestire il flusso di esecuzione di un programma, cioè a specificare quando e come eseguire le istruzioni del codice, sono classificati in Python (come negli altri linguaggi) in:
 - ❑ **Costrutti condizionali**, l'esecuzione dipende dal raggiungimento di certe condizioni
 - ❑ **Costrutti iterativi**, una o più istruzioni consecutive sono eseguite zero o più volte
 - ❑ **Costrutti fondamentali**, come **funzioni** e **metodi**
 - ❑ **Costrutti non locali**, gestione delle eccezioni



PYTHON

FLUSSO DI ESECUZIONE

- ❑ Le strutture di controllo rimandano all'esecuzione di un insieme di istruzioni
- ❑ Nei tradizionali linguaggi esiste il costrutto BEGIN e END o { }
- ❑ In Python si usa la tabulazione

```
if(a>b) {  
    massimo=a  
}  
else {  
    massimo=b  
}
```

```
if(a>b):  
    massimo=a  
else:  
    massimo=b
```

PYTHON

FLUSSO DI ESECUZIONE: IF ELSE

- ❑ Il **costrutto condizionale** serve a valutare una condizione e se essa è soddisfatta è eseguito il blocco di istruzioni corrispondente (if) altrimenti si eseguono le istruzioni di un altro blocco (else)
- ❑ La sintassi è:

if condizione:

istruzione1
istruzione2
...
istruzionen

else:

istruzionea
istruzioneb
...
istruzionez

ESEMPIO

```
a=56
b=67
if a>b:
    massimo=a
else:
    massimo=b
print (massimo)
```

67

PYTHON

FLUSSO DI ESECUZIONE: IF ELIF ELSE

- ❑ Il costrutto condizionale può valutare più condizioni usando **elif** (elif è la versione abbreviata di else if. Il costrutto elif è opzionale)
- ❑ Il blocco if può avere più blocchi **elif**
- ❑ **else** è elaborato qualora non siano verificate le condizioni dello **if** e degli **elif**; inoltre **else** è opzionale
- ❑ La sintassi è:

```
if condizione1 :  
    istruzione1  
    ...  
    istruzionen  
elif condizione2 :  
    istruzione$  
    ...  
    istruzione?  
else:  
    istruzionea  
    ...  
    istruzionez
```

ESEMPIO

```
x=2000  
if x<10:  
    print ("Piccolo")  
elif x<100:  
    print ("Medio")  
else:  
    print ("Grande")
```

Grande

PYTHON

FLUSSO DI ESECUZIONE: WHILE

- ❑ Il **costrutto iterativo con sentinella** permette di eseguire un blocco di istruzioni un numero indefinito di volte finché una particolare condizione risulta veritiera
- ❑ La sintassi generale è:

while (condition):

istruzione1

...

istruzionen

ESEMPIO

```
import random
number=int(random.randint(0,10))
guess=False
print(number)
while(guess==False):
    valore=input("Inserisci un intero: ")
    val=int(valore)
    if val==number:
        print ("Complimenti! Numero segreto indovinato")
        guess=True
    elif val < number:
        print ("Numero deve essere più grande")
        guess=False
    elif val > number:
        print ("Numero deve essere più piccolo")
        guess=False
```



PYTHON

FLUSSO DI ESECUZIONE: FOR

❑ Il **costrutto iterativo** FOR è utilizzato per ripetere un insieme di istruzioni un numero definito di volte

❑ La sintassi è:

```
for <variable> in <sequence>:
```

```
    istruzione1
```

```
    ...
```

```
    istruzionen
```

❑ Python può eseguire dei loop su qualsiasi tipo di sequenza di dati: tuple, liste, stringhe

ESEMPIO

```
vett = (1,56,45,33,55)
```

```
for i in vett:
```

```
    print (i)
```

```
1
```

```
56
```

```
45
```

```
33
```

```
55
```

PYTHON

FLUSSO DI ESECUZIONE: FOR

- ❑ Per iterare su una sequenza di interi, Python dispone della funzione

`range([start,] stop[, step])`

che restituisce una lista di interi

Attenzione RANGE parte dal valore start e arriva fino a stop-1

ESEMPIO

```
Val=range(1,10,3)
```

```
#valori 1,4,7
```

```
print(Val[0])
```

```
1
```

```
print(Val[1])
```

```
4
```

```
print(Val[2])
```

```
7
```

```
print(Val[3])
```

```
Not found
```

PYTHON

FLUSSO DI ESECUZIONE: istruzione PASS

- ❑ **Pass** è un'istruzione generica che 'non fa niente' e può essere inserita in qualsiasi parte del codice
- ❑ È utile in situazioni in cui è necessario inserire almeno una istruzione per ragioni sintattiche
- ❑ Esempio il corpo di una funzione non può essere omissivo.

```
def f(a,b):
```

```
    pass
```

```
a=3
```

```
b=4
```

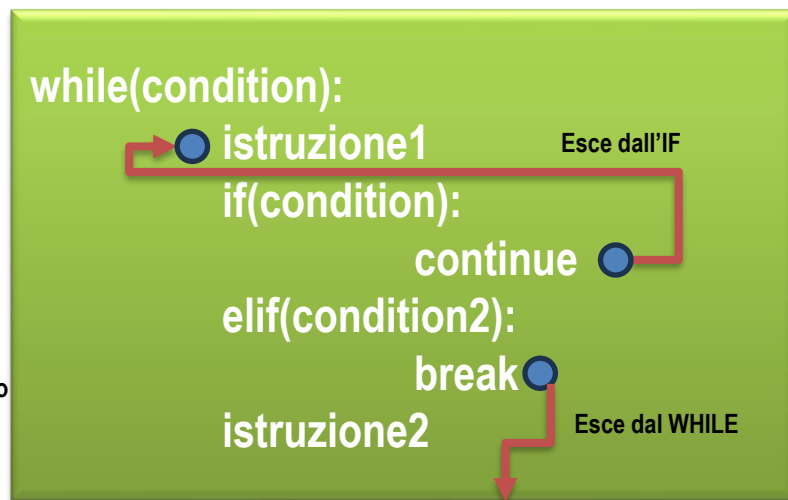
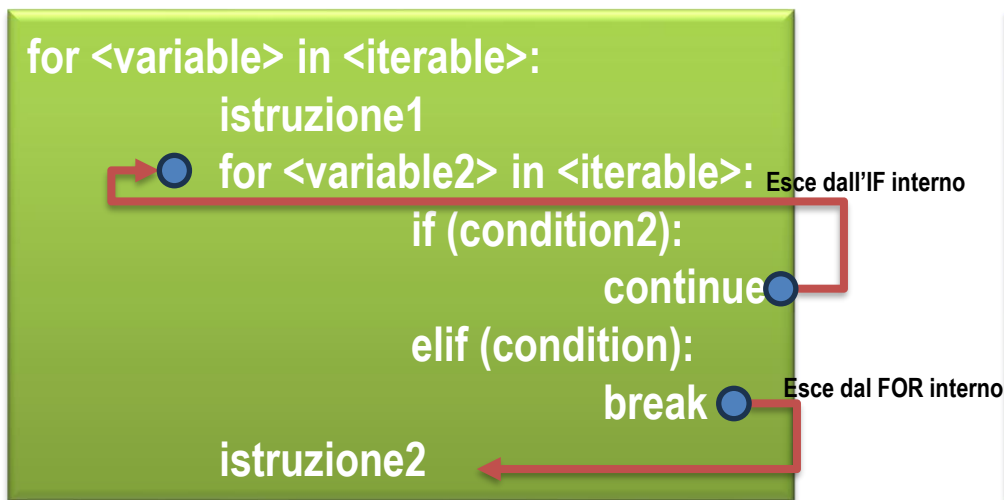
```
f(a,b)
```

```
#non restituisce niente
```

PYTHON

FLUSSO DI ESECUZIONE: istruzioni CONTINUE-BREAK

- ❑ L'esecuzione di un loop può essere interrotta qualora siano verificate condizioni eccezionali.
- ❑ L'istruzione **break** interrompe l'esecuzione del loop. Per i cicli annidati, break interrompe esclusivamente l'esecuzione del loop più interno.
- ❑ L'istruzione **continue** salta all'iterazione successiva di un ciclo. Nel caso di cicli annidati, continue agisce sul ciclo più interno.



PYTHON

FLUSSO DI ESECUZIONE: istruzioni FOR-ELSE/WHILE-ELSE

- ❑ Ai cicli condizionali for e while può essere associato un **else**, con la seguente sintassi:

```
for <variable> in <iterable>:  
    istruzione1  
    ...  
    istruzionen  
else:  
    istruzioneA  
    ...  
    IstruzioneZ
```

```
while(condition):  
    istruzione1  
    ...  
    istruzionen  
else:  
    istruzioneA  
    ...  
    IstruzioneZ
```

- ❑ Il blocco associato all'**else** è eseguito al termine dell'esecuzione dei rispettivi cicli a patto che questi non siano terminati con l'istruzione break



PYTHON

FLUSSO DI ESECUZIONE: istruzioni FOR-ELSE/WHILE-ELSE

ESEMPIO

```
n2=25
step=5
print ("L'intervallo va da  %d a %d con passo %d"%(0,n2,step))

number=int(input("Inserire un numero: "))

trovato=False
for i in range(0,n2,step):
    if i==number:
        trovato=True
        break
    else:
        pass

if trovato==True:
    print ("Il numero E' nell'intervallo con passo %d"%(step))
else:
    print ("Il numero NON E' nell'intervallo con passo %d"%(step))
```



PYTHON

ESERCIZIO

Trovare le soluzioni di una equazione di secondo grado del tipi

$$A^2x+Bx+C$$



PYTHON

ESERCIZIO

```
import math
a=int(input("Inserire il coefficiente A: "))
b=int(input("Inserire il coefficiente B: "))
c=int(input("Inserire il coefficiente C: "))
if a==0:
    print("x=", -c/b)
elif (b**2-4*a*c<0):
    print("Il sistema non ammette soluzioni nell'insieme dei Numeri Reali")
else:
    x1=(-b-sqrt(b**2-4*a*c))/2*a
    x2=(-b+sqrt(b**2-4*a*c))/2*a
    print("x1=%f x2=%f",x1,x2)
```



Fine