Programmazione in Python

Strutture controllo: while

Dario Pescini - Mirko Cesarini

Università degli Studi di Milano-Bicocca

Dipartimento di Statistica e Metodi Quantitativi

dario.pescini@unimib.it

Outline

Strutture di controllo

Sequenza

Selezione
- if
- if ... else
- if ... elif ... else

Iterazione
- while
- for

Strutture di controllo

Sequenza

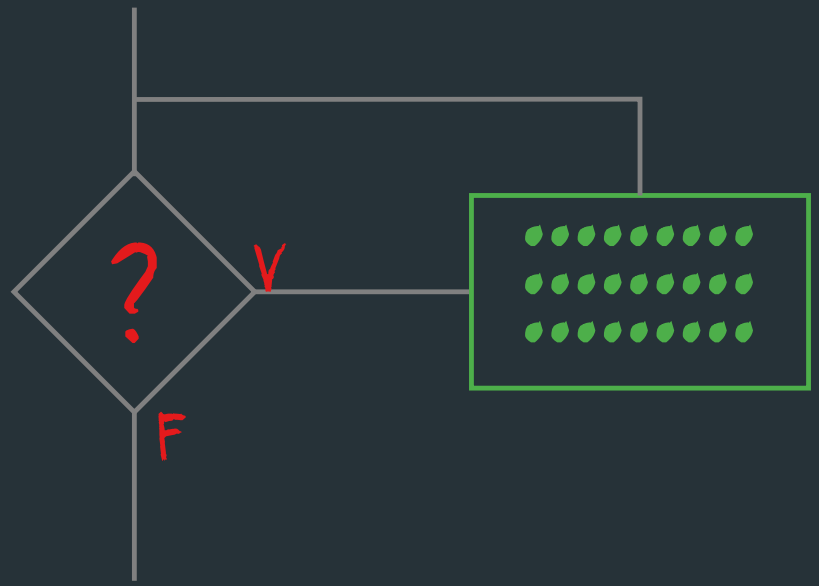
Selezione
- if
- if ... else
- if ... elif ... else

Iterazione
- while
- for

!! in Python i blocchi di istruzioni si isolano tramite indentazione
. . . . (4 spazi) !!

Strutture Controllo - Iterazione: WHILE

```
while condizione :
    lista comandi
```



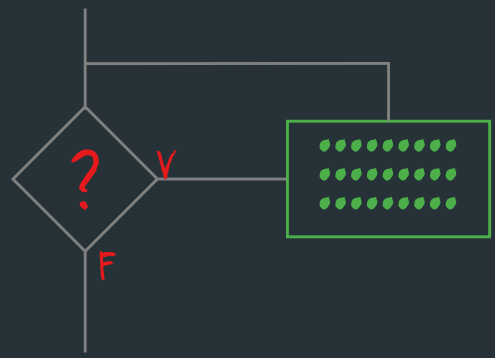
Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2

    print "y = ", y, "\tx = ", x
    x = x + 1.65
```



Strutture Controllo - Iterazione: WHILE

preverifica

Nel ciclo while la condizione viene verificata sempre prima di eseguire il blocco di istruzioni.

controllare che la condizione sia soddisfatta prima della prima iterazione

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0
5	0	2

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0
5	0	2

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0
5	0	2
7	1.65	2

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0
5	0	2
7	1.65	2

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0
5	0	2
7	1.65	2
9	1.65	2.825

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0
5	0	2
7	1.65	2
9	1.65	2.825

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0
5	0	2
7	1.65	2
9	1.65	2.825
11	3.3	2.825

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0
5	0	2
7	1.65	2
9	1.65	2.825
11	3.3	2.825

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0
5	0	2
7	1.65	2
9	1.65	2.825
11	3.3	2.825
…		
45	16.5	10.25

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0
5	0	2
7	1.65	2
9	1.65	2.825
11	3.3	2.825
...		
45	16.5	10.25

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0
5	0	2
7	1.65	2
9	1.65	2.825
11	3.3	2.825
…		
45	16.5	10.25
47	18.25	10.25

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0
5	0	2
7	1.65	2
9	1.65	2.825
11	3.3	2.825
...		
45	16.5	10.25
47	18.25	10.25

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

passo	x	y
1	–	0
2	0	0
5	0	2
7	1.65	2
9	1.65	2.825
11	3.3	2.825
...		
45	16.5	10.25
47	18.25	10.25

Strutture Controllo - Iterazione: WHILE

```
y=0.0
x=0.0

print "y = ", y, "\tx = ", x

while y <= 10 :
    y = 0.5 * x + 2
    print "y = ", y, "\tx = ", x
    x = x + 1.65
```

```
dario@vulcano: python while.py
y =  0.0        x =  0.0 <- inizializzazione
y =  2.0        x =  0.0
y =  2.825      x =  1.65
y =  3.65       x =  3.3
y =  4.475      x =  4.95
y =  5.3        x =  6.6
y =  6.125      x =  8.25
y =  6.95       x =  9.9
y =  7.775      x =  11.55
y =  8.6        x =  13.2
y =  9.425      x =  14.85
y =  10.25      x =  16.5
```

Trace table

É una tabella che viene costruita per tenere traccia della dinamica delle variabili del programma.

Per costruirla:

1. si identificano tutte le variabili dichiarate
2. si aggiunge una colonna per ogni variabile identificata
3. si aggiunge una riga ad ogni variazione del contenuto di una delle variabili
4. si modifica il valore nella cella della tabella associato alla variabile modificata
5. si riporta il valore delle altre variabili non modificate

Esempio

```
a=1
b=3
print "a = ",a, "b = ", b

temp = a;
a = b;
b = temp;

print "a = ",a, "b = ", b
```

passo	a	b	temp

Esempio

```
a=1
b=3
print "a = ",a, "b = ", b

temp = a;
a = b;
b = temp;

print "a = ",a, "b = ", b
```

passo	a	b	temp
1	1	–	–

Esempio

```
a=1
b=3
print "a = ",a, "b = ", b

temp = a;
a = b;
b = temp;

print "a = ",a, "b = ", b
```

passo	a	b	temp
1	1	–	–
2	1	3	–

Esempio

```
a=1
b=3
print "a = ",a, "b = ", b

temp = a;
a = b;
b = temp;

print "a = ",a, "b = ", b
```

passo	a	b	temp
1	1	–	–
2	1	3	–

Esempio

```
a=1
b=3
print "a = ",a, "b = ", b

temp = a;
a = b;
b = temp;

print "a = ",a, "b = ", b
```

passo	a	b	temp
1	1	–	–
2	1	3	–
4	1	3	1

Esempio

```
a=1
b=3
print "a = ",a, "b = ", b

temp = a;
a = b;
b = temp;

print "a = ",a, "b = ", b
```

passo	a	b	temp
1	1	–	–
2	1	3	–
4	1	3	1
5	3	3	1

Esempio

```
a=1
b=3
print "a = ",a, "b = ", b

temp = a;
a = b;
b = temp;

print "a = ",a, "b = ", b
```

passo	a	b	temp
1	1	–	–
2	1	3	–
4	1	3	1
5	3	3	1
6	3	1	1

Esempio

```
a=1
b=3
print "a = ",a, "b = ", b

temp = a;
a = b;
b = temp;

print "a = ",a, "b = ", b
```

passo	a	b	temp
1	1	–	–
2	1	3	–
4	1	3	1
5	3	3	1
6	3	1	1

WHILE: ciclo infinito

Nel ciclo `while` è fondamentale che la variabile coinvolta nella `condizione` di verifica venga `aggiornata durante l'iterazione` altrimenti il ciclo non termina.

Esempio:

```
x = True

while x == True:
    print 'iterazione'
```

per interrompere un programma in esecuzione: `ctrl + c`

Esercizio

Sfruttando il costrutto `while` si chieda all'utente di inserire un intero positivo. Il programma terminerà all'inserimento del primo numero negativo.

```
inserire un numero positivo:
1
inserire un numero positivo:
14
inserire un numero positivo:
-1
```

Esercizio

```
numero = 0
while numero >= 0 :
    numero = input("inserire un numero positivo: ")
```

WHILE: contatori

É possibile utilizzare una variabile per *contare il numero di iterazioni* svolte:

```
stringa = 'si'
contatore = 0

while stringa == 'si':
    stringa = raw_input("vuoi continuare l'iterazione? si/no")
    contatore = contatore + 1
print 'Hai eseguito ', contatore, ' iterazioni'
```

La variabile che 'conta' il numero di iterazioni si dice `contatore`

WHILE: contatori

É possibile sfruttare un contatore per eseguire un numero prefissato di iterazioni:

```
numIterazioni = input('inserisci il numero di iterazioni da
    eseguire ')

contatore = 0
while contatore < numIterazioni:
    print 'Hai eseguito ', contatore, ' iterazioni'
    contatore = contatore + 1

print 'il contatore vale: ', contatore
```

operatori di incremento

operatore	sintassi	funzione
`+=`	`var += val`	incrementa la `var` di `val`
`-=`	`var -= val`	decrementa la `var` di `val`
`*=`	`var *= val`	moltiplica la `var` `val` volte
`/=`	`var /= val`	divide la `var` per `val`
`%=`	`var %= val`	assegna a `var` il resto della sua divisione per `val`

variabili di accumulo: media

Spesso si utilizzano i cicli per “accumulare” quantità variabili in una variabile detta d’accumulo o accumulatore.

```
numIterazioni = 5
media = 0
contatore = 0
while contatore < numIterazioni:
    valore = float(input('inserisci un numero '))
    media += valore
    contatore += 1

media /= numIterazioni
print 'il valor medio è ', media
```

WHILE: valore sentinella

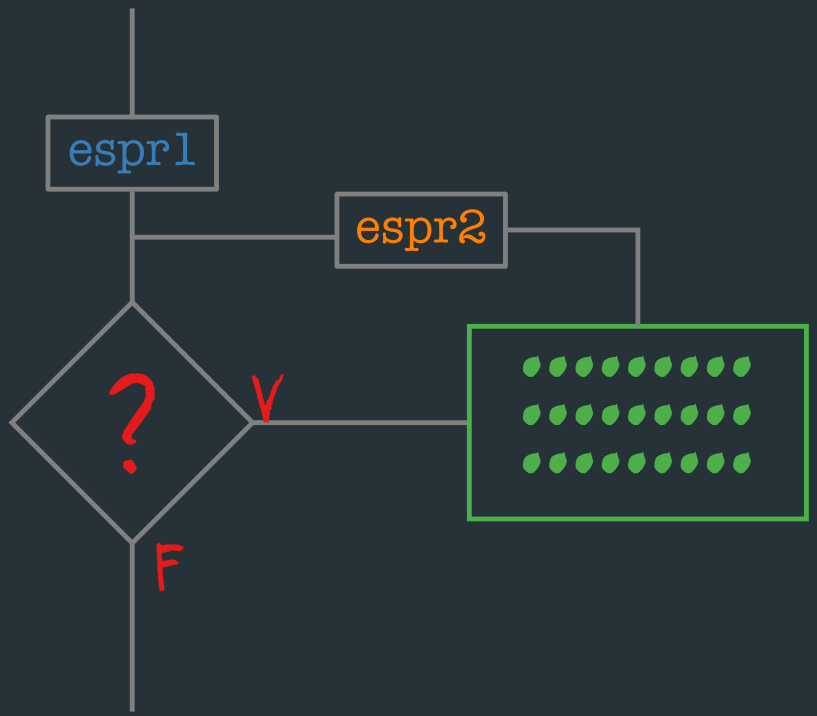
Sono valori dello stesso tipo dell'input richiesto all'utente ma sicuramente non parte dell'input atteso che vengono usati per interrompere il ciclo.

```
media = 0
valore = 0
iterazione = -1
while valore >= 0:
    media += valore
    iterazione +=1
    valore = float(input('inserisci un numero positivo (<0
    termina esecuzione)'))

media /= iterazione
print 'il valor medio è ', media
```

Strutture Controllo - Iterazione: FOR

```
for( espressione1; condizione; espressione2 ) :
   lista comandi
```



Strutture Controllo - Iterazione: FOR

Ne vedremo sintassi ed applicazione dopo le liste.

Algoritmo

Combinando i diagrammi di flusso ....