

Cicli

- Cicli:
 - For
 - While
 - Controllo

1

Cicli

Comandi per eseguire ripetutamente porzioni di codice

- Due tipi disponibili in Matlab:
 - **for**
 - **while**

2

Il ciclo “for”

Usato per un numero definito di ripetizioni di un gruppo di comandi:

```
for indice = vettore
    { comandi da ripetere }
end
```

Regole:

- Una ripetizione per ogni colonna del *vettore*
- *indice* assume il valore della colonna considerata

3

Esempio – raccogli 7 cifre per un numero telefonico e mettile in un vettore:

```
for digit = 1:7
    number(digit) = input('enter value ');
end
```

7 ripetizioni,
il vettore è [1 2 3 4 5 6 7]

digit cicla sui 7 valori,
Per creare il vettore “*number*” (1x7)

4

Command Window

```

enter value 8
enter value 7
enter value 4
enter value 5
enter value 8
enter value 0
enter value 2
>> number

number =

     8     7     4     5     8     0     2

>>

```

5

Esempio – calcolo degli interessi per 10 anni:

```

value = 1000;
for year = 1:10
    value = value * 1.08;
    fprintf(1,$ %g after %g years\n',value,year);
end

```

6

Command Window

```

$ 1080.00 after 1 years
$ 1166.40 after 2 years
$ 1259.71 after 3 years
$ 1360.49 after 4 years
$ 1469.33 after 5 years
$ 1586.87 after 6 years
$ 1713.82 after 7 years
$ 1850.93 after 8 years
$ 1999.00 after 9 years
$ 2158.92 after 10 years
>>

```

7

Esempio – implementare un conto alla rovescia (in secondi):

```

time = input('how long? ');
for count = time:-1:1
    pause(1)
    fprintf(1,'%g seconds left \n',count)
end
disp('done')

```

8

Command Window

```

how long? 5
5 seconds left
4 seconds left
3 seconds left
2 seconds left
1 seconds left
done
>> |

```

9

Esempio – con un vettore non ordinato:

```

for x = [ 8 7 4 5 8 0 2 ]
    fprintf(1,'dial %g \n', x)
    pause(1)
end

```

Command Window

```

dial 8
dial 7
dial 4
dial 5
dial 8
dial 0
dial 2
>>
>>

```

10

Esempio – ciclo per una matrice:

```

board = [ 1 1 0 ; 1 1 -1 ; 0 1 0 ]
for x = board
    x
end

```

Command Window

board =

```

1    1    0
1    1   -1
0    1    0

```

x =

```

1
1
0

```

x =

```

1
1
1

```

x =

```

0
-1
0

```

>>

11

Esempio – ciclo su una stringa:

```

for x = 'EGR106'
    disp(x)
end

```

Command Window

```

E
G
R
1
0
6
>>

```

12

Controllo dei cicli

I cicli contengono una serie di istruzioni che vogliamo ripetere.

Però potremmo aver bisogno di:

- Saltare all'iterazione successiva
- Terminare il ciclo

Perché continuare se abbiamo già trovato quello che cercavamo???

13

Saltare alla successiva: Continue

Continue – Salta all'iterazione successiva:

```
for k = 1:25000
    if x(k) > 0
        continue
    end
    { altri comandi }
end
```

Salta alla
successiva

14

Uscire dal ciclo: Break

Break – esce dal ciclo:

```
for variable = {Vettore lungo n}
    .....
    break
end
```

Vai ai comandi dopo il ciclo

15

Esempio – Calcoliamo gli interessi fino al raddoppio del capitale:

```
value = 1000;
for year = 1:1000
    value = value * 1.08;
    fprintf(1, '$ %g after %g years\n', value, year);
    if value >= 2000
        break
    end
end
```

Calcolerà fino a 1000 anni,
Se necessario

Questo "if" decide se
terminare il ciclo!

16

Command Window

```
$ 1080.00 after 1 years
$ 1166.40 after 2 years
$ 1259.71 after 3 years
$ 1360.49 after 4 years
$ 1469.33 after 5 years
$ 1586.87 after 6 years
$ 1713.82 after 7 years
$ 1850.93 after 8 years
$ 1999.00 after 9 years
$ 2158.92 after 10 years
>>
```

Bastano 10 anni!

17

Esempio – Prendiamo dei numeri in input, finché non viene inserito un negativo:

```
vector = [ ];
for count = 1:1000
    value = input('next number ');
    if value < 0
        break
    else
        vector(end+1) = value;
    end
end
vector
```

raccoglierà 1000
Valori, se necessario

18

Command Window

```
next number 5
next number 4
next number 3
next number 7
next number -2
```

```
vector =
```

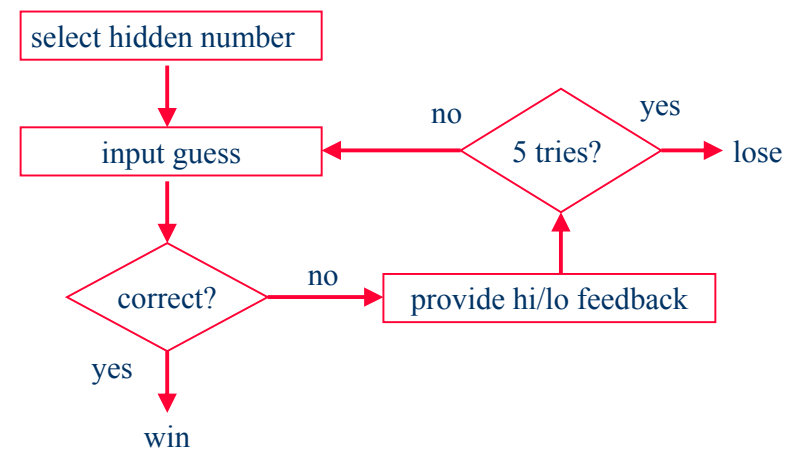
```
    5    4    3    7
```

```
>> |
```

Il valore negativo ferma l'input

19

Esempio – Hi-Lo: un gioco per indovinare i numeri



20

```
% HI-LO
numb = round ( 100 * rand (1) );
for count = 1:5
    guess = input('guess my number ');
    if guess == numb
        disp( 'You got it !!!' )
        break
    elseif guess > numb
        disp('you are too high')
    else
        disp('you are too low')
    end
    if count == 5
        disp('Sorry, you lose! ')
    end
end
```

```
guess my number 66
you are too low
guess my number 90
you are too high
guess my number 75
you are too low
guess my number 82
You got it !!!
```

21

Cicli nei cicli – Annidamento

```
for index1 = array1
    {Comandi del ciclo esterno}
    for index2 = array2
        {Comandi del ciclo interno}
    end
    {Altri comandi del ciclo esterno}
end
```

Possono esserci più di 2 livelli!!!

22

**Vediamo come
“girano” le variabili**

```
for index1 = 1:4
    for index2 = [ 3 6 9 ]
        [ index1 index2 ]
    end
end
```

index1	index2
1	3
1	6
1	9
2	3
2	6
2	9
3	3
3	6
3	9
4	3
4	6
4	9

23

**Esempio – Calcoliamo una tabella
 $z = x^2 + y^2$ per x e y uguali a 1, 2,...6:**

```
for x = 1:6
    for y = 1:6
        z(x,y) = x^2+y^2;
    end
end
z
```

Command Window					
z =					
2	5	10	17	26	37
5	8	13	20	29	40
10	13	18	25	34	45
17	20	25	32	41	52
26	29	34	41	50	61
37	40	45	52	61	72

>>

24

Esempio – Ottimizzare i ruoli delle persone:

Abbiamo:

- 4 lavori
- 4 persone con diverse abilità per ogni lavoro
- Tabelle dell'abilità

Obiettivo: assegnare i lavori per ottimizzare la somma

	Job 1	Job 2	Job 3	Job 4
Joe	7	4	4	2
Sue	6	8	5	2
Bob	4	7	1	3
Liz	6	5	2	1

Nell'esempio la somma è 20

25

Soluzione – Usare i cicli annidati per valutare tutte le soluzioni:

Innanzitutto inizializziamo:

```
skills = [ 7 4 4 2; 6 8 5 2; 4 7 1 3; 6 5 2 1 ];
best_assignment = [ ];
best_so_far = - Inf;
```

	Job			
	1	2	3	4
Joe	7	4	4	2
Sue	6	8	5	2
Bob	4	7	1	3
Liz	6	5	2	1

26

Poi cominciamo i cicli annidati

```
for Joe = 1:4
  for Sue = 1:4
    if Sue == Joe
      continue
    end
    for Bob = 1:4
      if Bob == Joe | Bob == Sue
        continue
      end
      for Liz = 1:4
        if Liz == Joe | Liz == Sue | Liz == Bob
          continue
        end
      end
    end
  end
end
```

Saltare nel caso di ripetizioni!! (stesso lavoro a due persone)

27

Vediamo quanto vale la somma

```
quality = Skills(1,Joe) + Skills(2,Sue) + ...
          Skills(3,Bob) + Skills(4,Liz);
if quality > best_so_far
  best_assignment = [ Joe, Sue, Bob, Liz ];
  best_so_far = quality;
end
```

E poi terminiamo i 4 cicli for :

```
end
end
end
end
```

28

Otteniamo:

```
>> nested_loop_example
best_assignment =
    3    2    4    1
best_so_far =
    21
```

	Job 1	Job 2	Job 3	Job 4
Joe	7	4	4	2
Sue	6	8	5	2
Bob	4	7	1	3
Liz	6	5	2	1

29

E se non sappiamo quanti cicli fare?

Il ciclo `for` ripete le istruzioni un numero fissato di volte:

```
for variable = {array of length n}
    {commands}
end
```

E possiamo usare `break` per finire prima.

Domanda: Ma se noi dovessimo ripetere un ciclo “ad oltranza”? Senza un limite finito...

30

Risposta: il ciclo “while” di MATLAB

```
while espressione
    {Comandi da ripetere finché
    espressione è vera}
end
```

- *Espressione* deve essere inizializzata (avere un valore) prima dell’inizio del ciclo!
- Il valore di *espressione* deve cambiare nelle ripetizioni!

31

L’esempio di prima – Calcoliamo gli interessi fino al raddoppio del capitale:

```
value = 1000;
for year = 1:1000
    value = value * 1.08;
    fprintf(1, '%g years: $ %g \n', year, value);
    if value > 2000
        break
    end
end
```

Ciclo for

Usciamo con il break

32

Otteniamo:

```

1 years: $ 1080.00
2 years: $ 1166.40
3 years: $ 1259.71
4 years: $ 1360.49
5 years: $ 1469.33
6 years: $ 1586.87
7 years: $ 1713.82
8 years: $ 1850.93
9 years: $ 1999.00
10 years: $ 2158.92

```

33

Versione con il while

```

value = 1000;
year = 0;
while value < 2000
    value = value * 1.08;
    year = year + 1;
    fprintf(1,'%g years: $ %g \n', year,value);
end

```

Dobbiamo
Incrementare
year!!
(contatore)

34

Esempio – cronometrare:

```

time = input('how long to wait? ');
while time > 0
    fprintf(1,'%g seconds left \n',time)
    pause(1)
    time = time - 1;
end
disp('done')

```

```

Command Window
how long to wait? 6
6 seconds left
5 seconds left
4 seconds left
3 seconds left
2 seconds left
1 seconds left
done
>>

```

Ecco un altro contatore

35

**Esempio – Raccogliere i dati in un vettore
finché non immettiamo zero:**

```

x = [];
new = 1;
while new ~= 0
    new = input('enter value ');
    x = [ x, new ];
end
x = x(1:end-1)

```

```

Command Window
enter value 4
enter value 2
enter value 3
enter value 4
enter value 0

x =

     4     2     3     4

>>

```

Per togliere lo zero finale

36

Esempio – Input da tastiera validi:

Es. Dobbiamo accettare solo numeri tra 0 e 10:

```
x = -3;
while ( x < 0 ) | ( x > 10 )
    x = input( 'type a value ' );
end
```

37

oppure:

```
x = input('enter value ');
while (x<0)|(x>10)
    disp('invalid choice');
    x = input('enter value ');
end
disp('finally!');
```

```
Command Window
enter value 12
invalid choice
enter value 22
invalid choice
enter value -5
invalid choice
enter value 6
finally!
>>
```

38

Mettiamo tutto assieme – Hi-Lo:

```
numb = round (10*rand(1)); } inizializzazione
done = 0;
while ~done
    guess = input('guess');
    if guess == numb
        disp( 'You got it !!!' ); done = 1;
    elseif guess > numb
        disp('too high')
    else
        disp('too low')
    end
end
```

Controllo Di ciclo

Tentativo singolo

39

Annidare i cicli while

```
while espressione1
    {comandi del ciclo esterno}
    while espressione2
        {comandi del ciclo interno}
    end
    {altri comandi del ciclo esterno}
end
```

Possono esserci anche
multi livelli...

40