



Loop in Matlab

Informatica B

Ripetere le istruzioni più volte...

- ❑ Trasformiamo diverse velocità da km/h in m/s

```
%Prima conversione
```

```
v = input('Inserire vel (km/h): ');
```

```
disp(['Velocità convertita: ' num2str(v*1000/3600)]);
```

```
%Seconda conversione
```

```
v = input('Inserire vel (km/h): ');
```

```
disp(['Velocità convertita: ' num2str(v*1000/3600)]);
```

```
%Terza conversione
```

```
v = input('Inserire vel (km/h): ');
```

```
disp(['Velocità convertita: ' num2str(v*1000/3600)]);
```

È proprio necessario ripetere più volte lo stesso pezzo di codice?

Il ciclo for



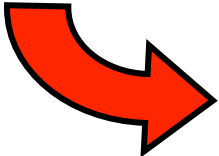
Variabile
contatore

```
for i = start:end
    statement1
    ...
    statementN
end
```

Corpo del
ciclo

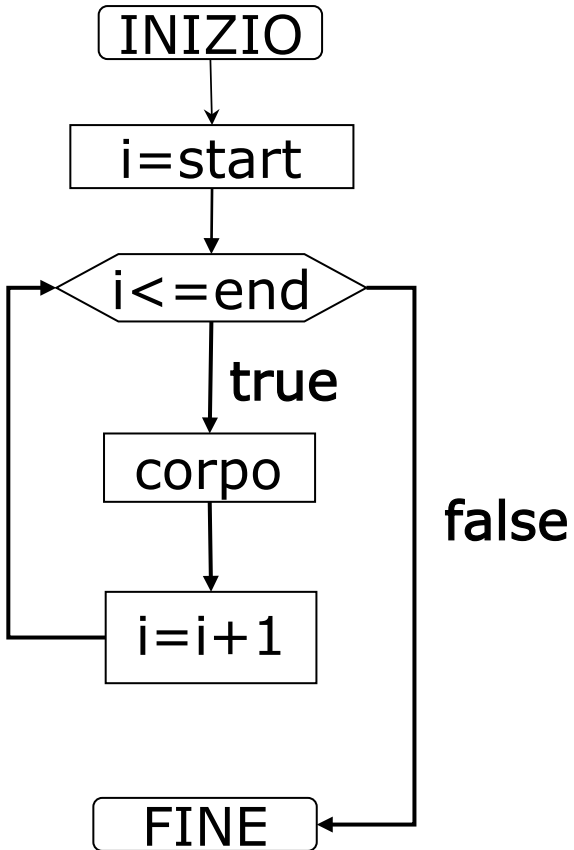
Il ciclo for

```
for i = start:end  
    corpo  
end
```



❑ Come lo uso nell'esempio precedente?

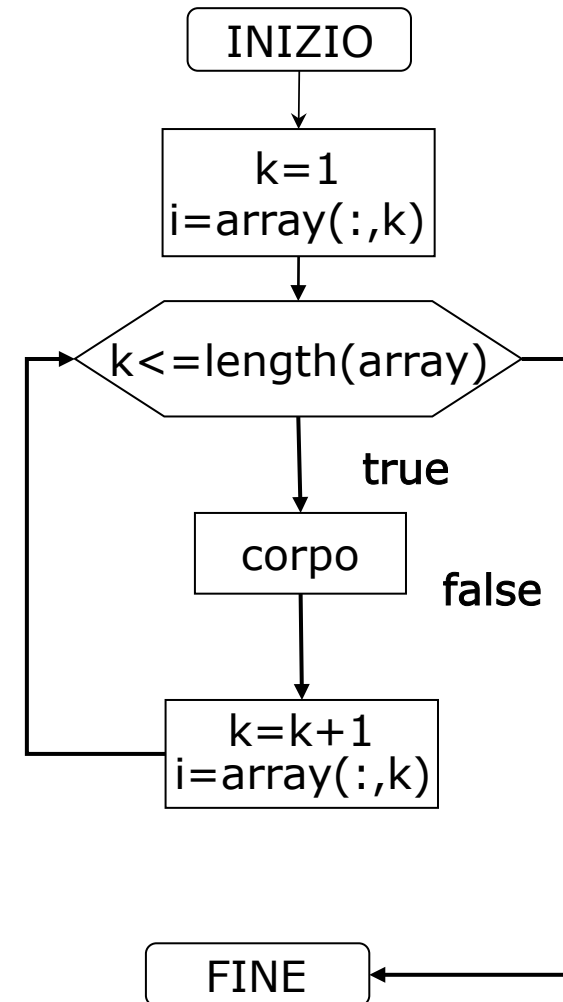
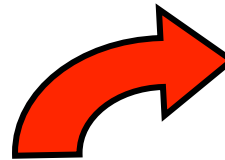
```
%Esegue tre conversioni  
for i=1:3  
    v = input('Inserire vel (km/h): ');  
    disp(['Vel (m/s): ' num2str(v*1000/3600)]);  
end
```



Il ciclo for... rendiamolo più flessibile

- ❑ La variabile contatore può essere fatta *variare* all'interno di un array arbitrario!
- ❑ Ad ogni *iterazione* del ciclo il contatore assumerà il valore di un elemento di tale array.

```
for i = array
    corpo
end
```



Il ciclo for: esempi

□ Conto alla rovescia

```
time = input('durata? ');  
for count = time:-1:1  
    pause(1)  
    disp([num2str(count) 'secondi rimanenti'])  
end  
disp('done')
```

□ Iterare su una stringa

```
for x = 'EGR106'  
    disp(x)  
end
```



Stampa E G R 1 0 6

□ Iterare su una matrice

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

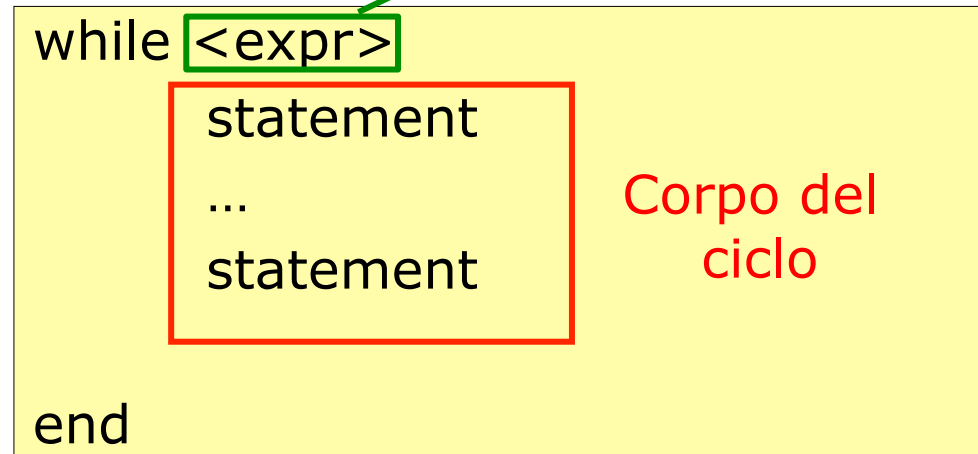


X vale all'iterazione
iesima board(:,i)

Il ciclo while



Condizione di
terminazione

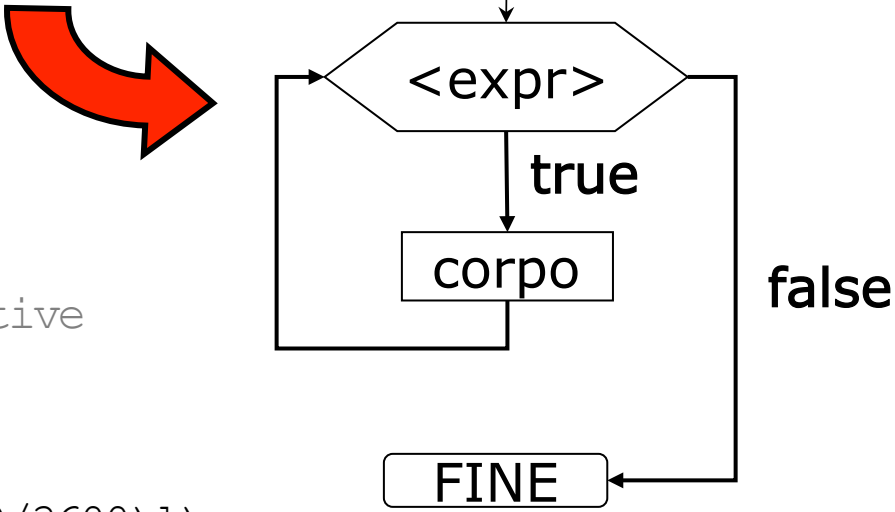


Il ciclo while

```
while <expr>  
    corpo  
end
```

❑ Esempio

```
%Converto velocità finchè sono positive  
v = input('Inserire vel (km/h): ');  
while v>0  
    disp(['Vel (m/s): \ num2str(v*1000/3600)]);  
    v = input('Inserire vel (km/h): ');  
end
```



Note sul while

- ❑ Il for è più pratico in alcune circostanze, ma **può sempre essere sostituito con un while**
- ❑ Per evitare **loop infiniti** o che il corpo non venga mai eseguito, nel while bisogna porre attenzione ad eventuali inizializzazioni e aggiornamento delle variabili usate nella condizione di terminazione
- ❑ Esempio (calcolo interessi annuali su un capitale)

```
value = 1000;  
year = 0;  
while value < 2000  
    value = value * 1.08  
    year = year + 1;  
    disp(['Anno ' num2str(year) ' : ' num2str(value)]);  
end
```

- ❑ I cicli contengono una serie di istruzioni che vogliamo ripetere, ma cosa succede se volessimo:
 - ▶ saltare all'iterazione successiva
 - ▶ terminare il ciclo
- ❑ Esistono due istruzioni che possono essere inserite all'interno di un ciclo (`for` o `while`) a questo scopo:
 - ▶ `continue` salta all'iterazione successiva
 - ▶ `break` interrompe l'esecuzione del ciclo

Esempio: break

- ❑ Acquisiamo numeri da tastiera finché non viene inserito un numero negativo. In ogni caso non accettiamo più di mille numeri.

```
for count = 1:1000 %Raccoglierà al max 1000 valori
    val = input('prossimo numero: ');
    if val < 0
        break %Se val negativo usciamo dal ciclo
    else
        numeri(count) = val;
    end
end
numeri %visualizza il contenuto del vettore
```

Esempio: continue

- ❑ Acquisiamo 100 numeri da tastiera e li memorizziamo in un vettore. Se il numero è positivo ne calcoliamo la radice quadrata e la memorizziamo in un altro vettore.

```
radici=[]; %radici è il vettore vuoto
for count = 1:100
    val = input('prossimo numero: ');
    numeri(count) = val;
    if val < 0
        continue
    end
    % aggiunge un elemento al vettore radici
    radici = [radici sqrt(val)];
end
radici %visualizza il contenuto del vettore
```