

## Esercizio

Scrivere uno script che genera la matrice identità  $6 \times 6$ :

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

**Remark:** Matlab ha già un'istruzione che lo fa, `eye(6)`.

## Cicli for: primo esempio

È possibile automatizzare istruzioni ripetitive con una nuova costruzione:

```
I = zeros(6, 6);  
for k = 1:6  
    I(k,k) = 1;  
end
```

## Esercizio

Scrivere uno script che genera la matrice

$$I = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

## Visualizzare grafici

L'istruzione `plot(x,y)` prende due **vettori** della stessa lunghezza, e traccia sullo schermo la spezzata che si ottiene congiungendo i punti di coordinate  $(x(1), y(1)), (x(2), y(2)), \dots, (x(\text{end}), y(\text{end}))$ .

### Esercizio

Generare due vettori contenenti rispettivamente  $x = (1, 2, \dots, 10)$  e  $y = (1, 4, 9, 16, 25, \dots, 100)$ , e utilizzarli per tracciare sullo schermo il grafico della funzione  $y = x^2$ .

### Esercizio

Scrivete l'esercizio precedente con un codice strutturato in modo che basti cambiare il valore di una variabile  $n$  per cambiare il numero di punti in un generico  $(1, 2, \dots, n), (1, 4, \dots, n^2)$ .

## Script: visualizzare grafici

### Esercizio

Tracciare sullo schermo il grafico della funzione  $y(x) = x^3 - x + 1$ , disegnando la funzione 'per punti' su un numero sufficientemente alto di punti in  $[-2, 2]$ .

## Esercizio

Scrivere uno script che visualizza sullo schermo i quadrati di tutti i numeri da 1 a 20.

## Esercizio

Scrivere uno script che visualizza sullo schermo i quadrati di tutti i numeri dispari da 1 a 19:  $1^2, 3^2, 5^2, \dots, 19^2$ .

## Esercizio

Scrivere uno script che visualizza sullo schermo i quadrati di tutti i numeri da 1 a 20.

## Esercizio

Scrivere uno script che visualizza sullo schermo i quadrati di tutti i numeri dispari da 1 a 19:  $1^2, 3^2, 5^2, \dots, 19^2$ .

**Metodo 1:** come facciamo a far 'variare' il contatore di 2 per volta?

**a:t:b** diventa  $a, a + t, a + 2t, \dots, b$  (o l'ultimo minore di  $b$ ).

## Esercizio

Scrivere uno script che visualizza sullo schermo i quadrati di tutti i numeri da 1 a 20.

## Esercizio

Scrivere uno script che visualizza sullo schermo i quadrati di tutti i numeri dispari da 1 a 19:  $1^2, 3^2, 5^2, \dots, 19^2$ .

**Metodo 1:** come facciamo a far 'variare' il contatore di 2 per volta?

**a:t:b** diventa  $a, a + t, a + 2t, \dots, b$  (o l'ultimo minore di  $b$ ).

**Metodo 2:** solo con quello che già sappiamo:  $2*k + 1$



## Esercizio

Scrivere uno script che visualizza sullo schermo i quadrati di tutti i numeri da 1 a 20.

## Esercizio

Scrivere uno script che visualizza sullo schermo i quadrati di tutti i numeri dispari da 1 a 19:  $1^2, 3^2, 5^2, \dots, 19^2$ .

**Metodo 1:** come facciamo a far 'variare' il contatore di 2 per volta?

**a:t:b** diventa  $a, a + t, a + 2t, \dots, b$  (o l'ultimo minore di  $b$ ).

**Metodo 2:** solo con quello che già sappiamo:  $2*k + 1$

**Metodo 3:**  $z = z + 2;$

## Esercizio

Cosa viene visualizzato sullo schermo eseguendo questo script? Qual è il valore delle variabili alla fine dell'esecuzione?

```
a = 1;  
b = 1  
for k = 1:9  
a = a + 2;  
b = b + a  
end
```

# Somme di elementi

## Esercizio

Scrivere uno script che calcola la somma degli elementi di un vettore dato (per esempio  $v = [12 \ 3 \ 9 \ -1 \ 13 \ 18 \ -5 \ 7 \ 82 \ 9 \ 0 \ 15]$ ):

```
v = [12 3 9 -1 13 18 -5 7 82 9 0 15];
```

```
% ???
```

```
% ???
```

**Remark:** Matlab in realtà ha già una funzione che lo fa, `sum(v)`.

## Calcolo di somme

### Esercizio

Scrivere uno script che calcola la somma dei primi 100 quadrati:  
 $1^2 + 2^2 + 3^2 + \dots + 100^2$ . (Non dovete per forza memorizzarli in un vettore!)

### Esercizio

Scrivi un programma che calcola la somma dei primi 10 termini della serie  $1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$ . (Questa serie converge a  $\frac{\pi}{4}$ , ma molto piano). Poi cambia 10 in 100, 1000, 10000, ...

### Esercizio

Scrivi un programma che calcola il **prodotto** dei primi 20 numeri interi:  
 $1 \cdot 2 \cdot 3 \cdot \dots \cdot 20$ . Cosa bisogna cambiare?

## Esercizio

Una colonia di animali contiene inizialmente 2000 esemplari. Ogni anno, si aggiungono 1000 animali. Quanti animali conterrà la colonia dopo 15 anni?

Traccia anche un grafico dell'andamento della popolazione.

## Esercizio

Una colonia di animali contiene inizialmente 2 esemplari. Ogni anno, ogni animale produce due figli. Quanti animali conterrà la colonia dopo 15 anni?

Traccia anche un grafico dell'andamento della popolazione.

Per tracciare più grafici contemporaneamente: **Metodo 1**: `hold on`.

**Metodo 2**: `plot(x, Y)` dove  $Y$  è una matrice con più colonne.

## Esercizio

Una colonia di animali contiene inizialmente 200 esemplari. Ogni anno, ognuno di loro produce 3 figli, però c'è un predatore che mangia 150 animali ogni anno. Quanti animali conterrà la colonia dopo 15 anni? Traccia anche un grafico dell'andamento della popolazione.

## Esercizio

Una popolazione di animali cresce in base a queste regole:

- All'anno 0, la popolazione è composta da 100 animali.
- Ogni anno, ognuno di essi produce 2 figli.
- 180 individui all'anno muoiono (predatori, cause naturali...)

Quanti animali ci sono dopo 15 anni?

## Modello di Fibonacci [Pisa, circa 1200]

- Si dispone di una coppia di conigli appena nati (M+F).
- Dopo un mese, la coppia raggiunge l'età adulta, e dal secondo mese in poi ogni mese dà alla luce una nuova coppia di figli.
- Le coppie appena nate si comportano in modo analogo: dal secondo mese di vita in poi producono due figli al mese.

Mesi	Coppie neonate	Coppie adulte
0	1	0
1	0	1
2	1	1
3	1	2
4	2	3
5	3	5

Simulare l'andamento della popolazione per 15 mesi, memorizzando i numeri ottenuti in una matrice  $15 \times 2$ . Tracciare un grafico.



## Esercizio

Sapreste scrivere uno script che calcola il numero di conigli adulti al tempo  $n$ , ma questa volta senza memorizzare tutti i numeri ottenuti in una matrice?

## Ancora matrici

### Esercizio

Scrivere uno script che produce la matrice

$$A = \begin{bmatrix} 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 \\ 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 \\ 31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 \\ 41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 & 49 \\ 51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 \\ 61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 \\ 71 & 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 \\ 81 & 82 & 83 & 84 & 85 & 86 & 87 & 88 & 89 \\ 91 & 92 & 93 & 94 & 95 & 96 & 97 & 98 & 99 \end{bmatrix}$$

Servono due cicli. . .

## Cicli annidati

```
A = zeros(9, 9);  
for i = 1:9  
  for j = 1:9  
    A(i,j) = 10*i + j  
  end  
end
```

Le linee in rosso sono il corpo del loop più esterno. Come vengono trasformate queste righe in istruzioni?

**Indentazione** (allineamento verticale delle istruzioni tramite spazi) rende tutto molto più leggibile.

**Preallocare**  $A$  come una matrice di zeri rende il codice leggermente più veloce e assicura risultati coerenti (anche se c'era già una variabile  $A$  in giro).

## Altri esercizi

### Esercizio

Scrivere le 'tabelline':

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

## Altri esercizi

### Esercizio

Creare la matrice

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 \\ 1 & 2 & 3 & 0 & 0 \\ 1 & 2 & 3 & 4 & 0 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix} .$$

Riuscite a farlo riempiendo la matrice una riga per volta a partire dalla prima? Riuscite a farlo riempiendo la matrice una colonna per volta?

## Altri esercizi

### Esercizio

Creare la matrice  $n \times n$  (per esempio, per  $n = 10$ ) contenente elementi

$$C = \begin{bmatrix} 0 & 1 & & & & & & & & & \\ 1 & 0 & 1 & & & & & & & & \\ & 1 & 0 & 1 & & & & & & & \\ & & 1 & 0 & 1 & & & & & & \\ & & & \ddots & \ddots & \ddots & & & & & \\ & & & & 1 & 0 & 1 & & & & \\ & & & & & 1 & 0 & & & & \\ & & & & & & 1 & 0 & & & \end{bmatrix}.$$

Gli elementi non visualizzati sono tutti zeri.

(Vi servono davvero due `for` annidati qui?)