

Elementi di MATLAB

Matrici, funzioni con le matrici
Funzioni per il caricamento e la
visualizzazione di immagini



Definizione di matrice

```
>>A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

fine riga

A =

```
16  3  2 13
 5 10 11  8
 9  6  7 12
 4 15 14  1
```

risultato

Le variabili si definiscono in maniera dinamica

Operazioni sulle matrici



```
>>sum(A)
```

```
ans =
```

```
34 34 34 34
```

E' possibile realizzare operazioni direttamente sulle matrici

```
>>A'
```

```
ans =
```

```
16 5 9 4  
3 10 6 15  
2 11 7 14  
13 8 12 1
```

```
>>sum(sum(A))
```

```
ans =
```

```
136
```

Operazioni sulle matrici



```
>> X=A;
```

```
>> X
```

E' possibile fare assegnazioni tra matrici

```
X =
```

```
16 3 2 13  
5 10 11 8  
9 6 7 12  
4 15 14 1
```

non visualizza il risultato

```
>> size(X)
```

dimensione di una matrice

```
ans =
```

```
4 4
```

Elementi della matrice



L'elemento in riga i e colonna j si indica con $A(i,j)$

```
>>A(1,4) + A(2,4) + A(3,4) + A(4,4)
```

```
ans =  
    34
```

Se si cerca di accedere al di fuori dei limiti della matrice si genera un errore

```
>>t = A(4,5)  
Index exceeds matrix dimensions.
```

Elementi della matrice



```
>>X = A;  
>>X(4,5) = 17  
X =
```

```
    16     3     2    13     0  
     5    10    11     8     0  
     9     6     7    12     0  
     4    15    14     1    17
```

E' possibile modificare dinamicamente
le dimensioni di una matrice



L'operatore :

```
>> 1:10
```

```
ans =
```

```
1 2 3 4 5 6 7 8 9 10
```

```
>> 1:2:10
```

```
ans =
```

```
1 3 5 7 9
```

Serve a specificare un intervallo di valori:

```
>> 100:-5:40
```

```
ans =
```

```
100 95 90 85 80 75 70 65 60 55 50 45 40
```



L'operatore :

```
>> A(1:3,4)
```

```
ans =
```

```
13
```

```
8
```

```
12
```

E' utile per identificare sottoparti di una matrice

```
>> A(1:3,:)
```

```
ans =
```

```
16 3 2 13
```

```
5 10 11 8
```

```
9 6 7 12
```

```
>> A(1:3,2:end)
```

```
ans =
```

```
3 2 13
```

```
10 11 8
```

```
6 7 12
```

Eliminazione di parti di matrici



```
>>X(:,2) = []
```

```
X =
```

```
16  2 13
 5 11  8
 9  7 12
 4 14  1
```

Concatenazioni di matrici



```
>>B = [A A+32; A+48 A+16]
```

```
B =
```

```
16  3  2 13 48 35 34 45
 5 10 11  8 37 42 43 40
 9  6  7 12 41 38 39 44
 4 15 14  1 36 47 46 33
64 51 50 61 32 19 18 29
53 58 59 56 21 26 27 24
57 54 55 60 25 22 23 28
52 63 62 49 20 31 30 17
```

Costrutti selettivi



```
if rem(n,2) ~= 0
    M = odd_magic(n)
elseif rem(n,4) ~= 0
    M = single_even_magic(n)
else
    M = double_even_magic(n)
end
```

```
switch (rem(n,4)==0) + (rem(n,2)==0)
    case 0
        M = odd_magic(n)
    case 1
        M = single_even_magic(n)
    case 2
        M = double_even_magic(n)
    otherwise
        error('This is impossible')
end
```

Costrutti ciclici



```
for i = 1:m
    for j = 1:n
        H(i,j) = 1/(i+j);
    end
end
```

```
a = 0; fa = -Inf;
b = 3; fb = Inf;
while b-a > eps*b
    x = (a+b)/2;
    fx = x^3-2*x-5;
    if sign(fx) == sign(fa)
        a = x; fa = fx;
    else
        b = x; fb = fx;
    end
end
```

Funzioni

```
function P=allpermute(n)
% P=allpermute(n)
%
% Restituisce nella matrice P tutte le
% permutazioni possibili sugli interi 1..n

if(n==1)
    P=[1];
else
    Pprec=allpermute(n-1);
    rp=size(Pprec,1); cp=size(Pprec,2);
    P=zeros(rp*n,n);
    h=0;
    for i=1:rp
        row=Pprec(i,:);
        h=h+1;
        P(h,:)= [n row];
        for j=1:cp
            h=h+1;
            P(h,:)= [row(1:j) n row(j+1:end)];
        end
    end
end
end
```



Gestione delle immagini

- Matlab fornisce funzioni per la lettura, scrittura e visualizzazione di diversi formati di file immagine. I formati supportati sono:
 - BMP (Microsoft Windows Bitmap)
 - HDF (Hierarchical Data Format)
 - JPEG (Joint Photographic Experts Group)
 - PCX (Paintbrush)
 - PNG (Portable Network Graphics)
 - TIFF (Tagged Image File Format)
 - XWD (X Window Dump)



Gestione delle immagini



- MATLAB supporta tre classi diverse per la rappresentazione dei valori associati ai punti delle immagini:
 - 64-bit double-precision floating-point (double)
 - 16-bit unsigned integer (uint16)
 - 8-bit unsigned integer (uint8)

Le immagini in Matlab



- Matlab memorizza la maggior parte delle immagini come una array bidimensionale in cui ogni elemento corrisponde ad un pixel dell'immagine digitale.

Es.:

$A(12,15)$ restituisce il valore del pixel in riga 12 colonna 15.

Le immagini in Matlab



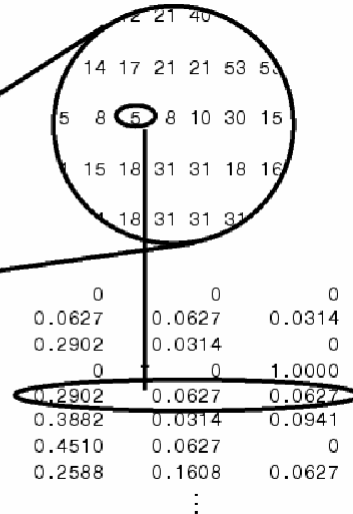
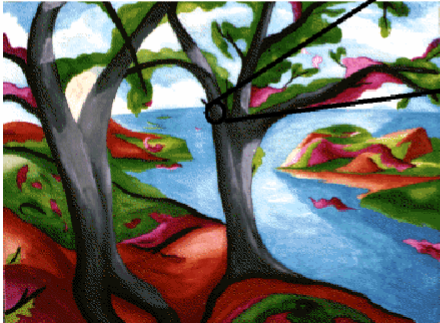
- Un'immagine consiste di una matrice di valori numerici ed eventualmente di una matrice contenente una palette (*colormap*).
- Esistono due tipi di immagine:
 - immagini indicizzate (*indexed images*)
 - immagini RGB (*truecolor*)

colormap

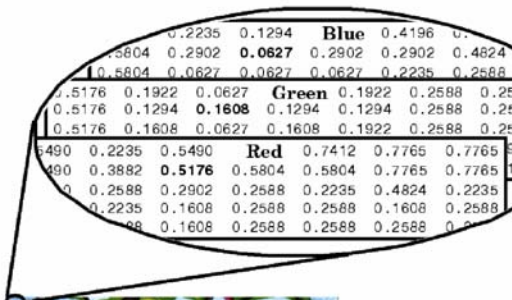


- Una *colormap* è una matrice $m \times 3$ di numeri float tra 0.0 e 1.0.
- Ogni riga è un vettore RGB che definisce un colore.
- La k -ma riga della colormap definisce il k -mo colore: I tre valori sulla riga specificano rispettivamente le intensità di rosso, verde e blu

Immagini indexed



Immagini RGB



Funzioni per la gestione delle immagini



| Funzione | Scopo | |
|----------|------------------------------------|----------|
| axis | Gestisce gli assi | Display |
| Image | Visualizza un'immagine | Display |
| imagesc | Scala i valori e visualizza | Display |
| imread | Legge da file grafico | File I/O |
| imwrite | Scrive su file grafico | File I/O |
| iminfo | Legge informazioni da file grafico | Utility |

Funzione per la visualizzazione



```
function immostra(I,tipomap)
% immostra(I)
%
% visualizza l'immagine I secondo la palette definita
% da tipomap
```

```
figure;
image(I);
colormap(tipomap);
axis image;
axis off;
```