



# Osnove računarstva II

MATLAB / Octave - Uvod

# Šta je MATLAB?

- MATLAB (**MAT**rix **LAB**oratory) je **okruženje za numeričke proračune** i **programski jezik** koji proizvodi firma MathWorks.
- MATLAB omogućava:
  - lako manipulisanje matricama,
  - grafički prikaz funkcija (2D i 3D grafika) i fitovanje,
  - implementaciju algoritama,
  - kreiranje grafičkog korisničkog interfejsa,
  - povezivanje sa programima pisanim u drugim jezicima, uključujući C, C++, Java, Fortran.
- Posjeduje visoke performanse u numeričkim proračunima, i omogućava znatno elegantnije proračune nego što to čine programski jezici poput Fortran-a, Pascal-a, C-a.
- MATLAB je interpreter – izvršava komande jednu za drugom bez prevođenja (kompajliranja).

# Open source alternativa?

- Nedostatak MATLAB-a je visoka cijena licence.
- Alternativa MATLAB-u je Octave
  - Open source (GNU Licenca)
  - Aktuelna Windows verzija je 6.2.0.
  - Može se naći na <http://www.gnu.org/software/octave/>
- Prednost Octave je dobra kompatibilnost sa MATLAB-om.
- Nedostaci su mu siromašniji korisnički interfejs, te manja brzina izvršavanja.

# Šta je MATLAB?

- Radno okruženje MATLAB-a ima više prozora. Naš fokus će biti na „Command window“ prozoru.
- Komande ćemo unositi u komandnom prozoru, nakon komandnog prompta >>
- Operacije i iskazi se pišu, kada god je to moguće, na prirodan način – kao na papiru.
- Jednostavna izračunavanja vršimo tako što otkucamo izraz i pritisnemo taster ENTER.
- Ne sadrži izraze za određivanje dimenzija ili tipa matrice, već se tip i dimenzije određuju automatski prilikom dodjele vrijednosti nekoj promjenljivoj.
- Svi specijalni slučajevi pravougaone matrice slijede pravila za opšti slučaj, osim u nekim operacijama (biće naglašeno).

# MATLAB promjenljive

- Naziv promjenljive čine slova, brojevi i karakter `_` (mora početi slovom)  
`Studenti2013`, `S2`, `s5`, `mtf2013`, `mala_slova`
- Razlikuju se mala i velika slova (npr. `a` i `A` su dvije promjenljive).

# MATLAB/Octave izraz

- Oblik: `promjenljiva = izraz`  
`abc = 1234;` => cijeli broj 1234 se dodjeljuje promjenljivoj `abc`
- Ako se izostavi promjenljiva sa lijeve strane znaka `=`, privremena promjenljiva `ans` poprima vrijednost izraza koji se izračunava. `ans` pamti posljednju dodijeljenu vrijednost.
- Tačka-zarez na kraju izraza sprečava ispis vrijednosti promjenljive na ekranu. Zarez omogućava pisanje više iskaza u jednom redu, uz prikaz rezultata.

# Tipovi podataka u MATLAB-u/Octave

- Osnovni tip podataka je matrica (brojeva).
- Skalar je matrica dimenzije 1x1
  - `A = 2.3;` => realan broj (za odvajanje cijelog i decimalnog dijela broja se koristi tačka, **nikako zarez**)
  - `B = 3e-5;` => realan broj u eksponencijalnom zapisu  $3 \cdot 10^{-5}$
  - `C = 2+3i;` => kompleksan broj
  - `T = 'tekst';` => tekstualna promjenljiva – string
- Numerički podaci su podrazumijevano tipa `double` (skalar `double`, kompleksan broj tipa `double+i*double`).
- Nakon `%` se unosi komentar, tj. taj dio koda se ne izvršava.

# Matrice u MATLAB-u / Octave

- Promjenljiva (matrica) se može kreirati direktnim unošenjem elemenata unutar uglastih zagrada [] (eksplicitna lista elemenata).
- Elementi u jednoj vrsti se odvajaju zarezima ili razmacima. U sljedeću vrstu se prelazi korišćenjem tačke-zarez ili sa Enter:

```
>> A = [1, 2, 3; 4, 5, 6; 7, 8, 9]
```

```
A =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

- Vektor je specijalna matrica u kojoj je jedna dimenzija 1, pa imamo:

## Vektor vrstu

```
>> B = [1, 2, 3]
```

```
B =
```

```
1 2 3
```

## Vektor kolonu

```
>> C = [1; 2; 3]
```

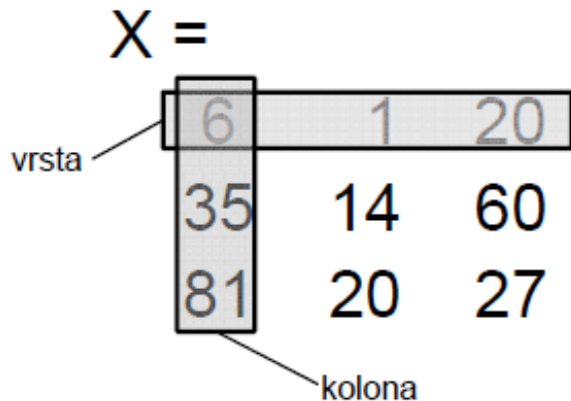
```
C =
```

```
1
```

```
2
```

```
3
```

- **Pristup elementima matrice**



Pristup elementu:

$X(m,n)$

$m$  – broj vrste

$n$  – broj kolone

<pre>&gt;&gt; X(1,1) ans = 6</pre>	<pre>&gt;&gt; X(1,2) ans = 1</pre>	<pre>&gt;&gt; X(1,3) ans = 20</pre>
<pre>&gt;&gt; X(2,1) ans = 35</pre>	<pre>&gt;&gt; X(2,2) ans = 14</pre>	<pre>&gt;&gt; X(2,3) ans = 60</pre>
<pre>&gt;&gt; X(3,1) ans = 81</pre>	<pre>&gt;&gt; X(3,2) ans = 20</pre>	<pre>&gt;&gt; X(3,3) ans = 27</pre>

**end** označava poslednji element  $X(\text{end}, \text{end})$  – element u posljednjoj vrsti i koloni,  $X(\text{end}, \text{end}-1)$  – element u posljednjoj vrsti i preposljednoj koloni

Za indeksiranje se **end** koristi samo unutar zagrada, ne samostalno



# Specijalne matrice u MATLAB-u/Octave

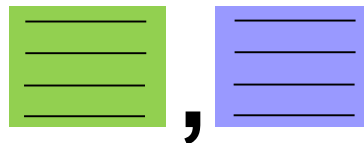
- MATLAB/Octave posjeduje i neke već unaprijed definisane matrice, kao što su:

<u><b>zeros(n)</b></u>	<u><b>ones(n)</b></u>	<u><b>eye(n)</b></u>
<pre>&gt;&gt;zeros(3) ans =   0  0  0   0  0  0   0  0  0</pre>	<pre>&gt;&gt;ones(3) ans =   1  1  1   1  1  1   1  1  1</pre>	<pre>&gt;&gt;eye(3) ans =   1  0  0   0  1  0   0  0  1</pre>
<pre>&gt;&gt;zeros(3,2) ans =   0  0   0  0   0  0</pre>	<pre>&gt;&gt;ones(3,4) ans =   1  1  1  1   1  1  1  1   1  1  1  1</pre>	<pre>&gt;&gt;eye(3,4) ans =   1  0  0  0   0  1  0  0   0  0  1  0</pre>

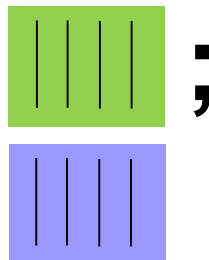
- **rand(N,M)** daje matricu sa slučajnim vrijednostima uniformno raspodijeljenim u intervalu od 0 do 1.
- **randn(N,M)** daje matricu sa slučajnim vrijednostima koje podliježu Gausovoj raspodjeli vjerovatnoća.
- **magic(N)** daje „magični kvadrat“ (konstantan zbir elemenata po vrstama i kolonama i glavnoj dijagonali)

# Kombinovanje matrica

- Jednostavnije matrice se mogu kombinovati u cilju dobijanja složenih matrica **bez direktnog unošenja elemenata**.
- Pravila za kombinovanje više matrica su jednostavna:
  - Složene matrice se zadaju u uglastim zagradama **[ ]**.
  - Ukoliko želimo nadovezati jednu matricu sa desne strane druge matrice, odvajamo ih zarezom. U ovom slučaju obje matrice moraju imati isti broj vrsta.



- Ukoliko želimo dodati matricu ispod druge matrice, odvajamo ih tačka zarezom. U ovom slučaju obje matrice moraju imati isti broj kolona.



# Formiranje složenih matrica

- **Primjer:** Bez direktnog unošenja elemenata, formirati matricu:

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 10 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

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

```
>> A = [eye(3), ones(3, 2); zeros(2, 5)]
```

```
A =
```

```
 1     0     0     1     1
 0     1     0     1     1
 0     0     1     1     1
 0     0     0     0     0
 0     0     0     0     0
```

```
>> A(3,4) = 10;
```

# Formiranje složenih matrica

- **Primjer:** Bez direktnog unošenja elemenata, formirati matricu:

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 2 & 2 \end{bmatrix}$$

$$A = \begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} \end{bmatrix}$$

```
>> A = [eye(3), [ones(1, 2); zeros(2)]; ones(2, 3), 2 * ones(2)]  
A =
```

```
1     0     0     1     1  
0     1     0     0     0  
0     0     1     0     0  
1     1     1     2     2  
1     1     1     2     2
```

# Operator :

- Osim unaprijed definisanih matrica, i matrica dobijenih njihovim kombinovanjem, bez direktnog navođenja elemenata se mogu zadavati i vektori koji predstavljaju aritmetički red, odnosno vektori kod kojih se svaka dva susjedna elementa razlikuju za istu vrijednost.
- Vektor sukcesivnih vrijednosti iz intervala  $[a, b]$  sa korakom 1:

```
>> x = a : b
```

čime dobijamo vektor  $[a, a+1, a+2 \dots b]$ . Dakle, kreće se od  $a$ , a svaki sljedeći element se dobija dodavanjem jedinice tekućem elementu. Najveći element mora biti manji ili jednak gornjoj granici intervala  $b$ .

- Vektor vrijednosti iz intervala  $[a, b]$  se može dobiti i sa definisanim korakom  $c$ :

```
>> x = a : c : b % [a, a+c, a+2*c ... a+n*c ], a+n*c <= b
```

- Matrica se može dobiti i kao rezultat izračunavanja iskaza, izvršavanja funkcija, kreiranjem u m-fajlovima i unošenjem iz spoljašnjih fajlova.

# Formiranje aritmetičkog reda – primjeri

>> **A = 1 : 7**

A =

1      2      3      4      5      6      7

>> **B = 0.2 : 5**

B =

0.2000      1.2000      2.2000      3.2000      4.2000

>> **C = 0.2 : 0.1 : 0.6**

C =

0.2000      0.3000      0.4000      0.5000      0.6000

>> **D = 9 : - 1 : 1** %može ići od veće ka manjoj vrijednosti

D =

9      8      7      6      5      4      3      2      1

# Kreiranje podmatrice i proširivanje matrice

- Navođenjem imena matrice iz koje izdvajamo elemente i u zagradi vektora rednih brojeva vrsta i kolona koje se žele izdvojiti, formira se podmatrica postojeće matrice sastavljena od elemenata koji se nalaze u presjeku navedenih vrsta i kolona.

```
X =
    6    1   20
   35   14   60
   81   20   27

Y =
    1    2    3    4

Z = 8
     3
     4
```

Elementi prve vrste, prve i druge kolone

Elementi u koji pripadaju prvoj i drugoj vrsti i prvoj i drugoj koloni

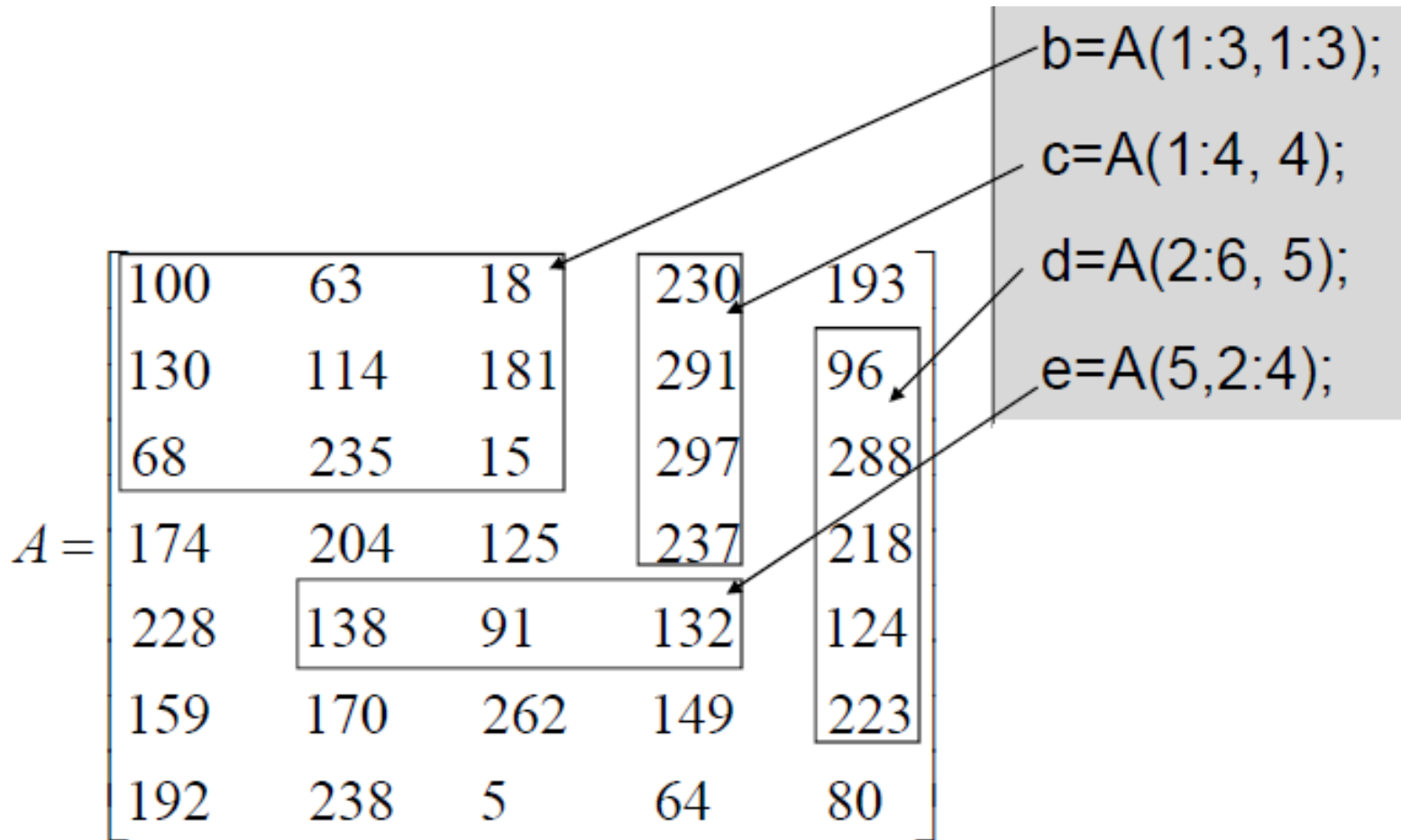
<pre>&gt;&gt; X1=X(1,[1:2]) X1 =     6    1</pre>	<pre>&gt;&gt; X2=X([1:2],[1:2]) X2 =     6    1    35   14</pre>	<pre>&gt;&gt; a=X(2:3,[1,3]) a =    35   60    81   27</pre>
<pre>&gt;&gt; b=X(:,[1,2]) b =     6    1    35   14    81   20</pre>	<pre>&gt;&gt; c=X(3,:) c =    81   20   27</pre>	<pre>&gt;&gt; A=[Y; [X, Z]] A =     1    2    3    4     6    1   20    8    35   14   60    3    81   20   27    4</pre>

Vektor sa rednim brojevima vrsta/kolona koje izdvajamo se zadaje direktno - u uglastim zagradama ili pomoću operatora :

- Navođenjem samo operatora **:** uzimaju se sve vrste ili kolone, zavisno od toga da li je prvi ili drugi argument, respektivno.

# Primjer:

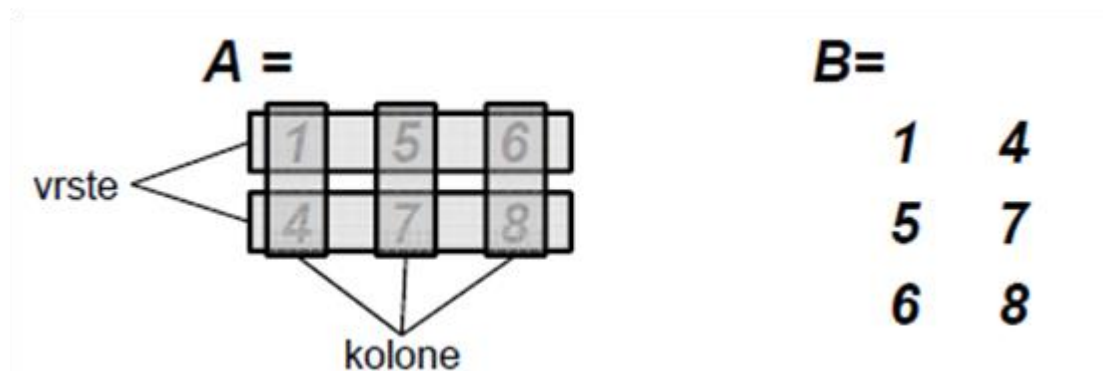
- Data je matrica  $A$ . Od zaokruženih djelova matrice formirati nove promjenljive  $b$ ,  $c$ ,  $d$  i  $e$





# Transponovana matrica

- Transponovana matrica neke matrice dobija se zamjenom vrsta sa kolonama.



- Operator transponovanja u MATLAB-u je apostrof:

$$B = A'$$

Ako je  $A = [1, 5, 6; 4, 7, 8]$  onda je  $B = [1, 4; 5, 7; 6, 8]$

# Aritmetičke operacije

- Sabiranje +
- Oduzimanje -
- Množenje \*
- Dijeljenje /
- Stepenuvanje ^

- Redosljed operacija je određen na osnovu prioriteta: unarni minus, stepenovanje, množenje/dijeljenje, sabiranje/oduzimanje...
- Upotrebom zagrada može se promijeniti redosljed izračunavanja.

- Primjeri:

```
>> 1 + 3 * 2 ^ 4 / 6 - 9 %1+3*16/6-9 – prvo se stepenuje
```

```
ans =
```

```
0
```

```
>> 1 + (3 * 2) ^ 4 / (6 - 9)
```

```
ans =
```

```
-431
```

# Promjena vrijednosti elemenata matrice

- Unesimo matricu

```
>> A=[1,2,3; 4,5,6; 7,8,9]
```

```
A =
```

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

- Možemo promijeniti znak elementu 6 (druga vrsta i treća kolona), dodati broj 2 elementu 7 (treća vrsta i prva kolona) i dodijeliti vrijednost 0.5 elementu 8 (treća vrsta i druga kolona):

```
>> A(2,3)=-A(2,3); A(3,1)=A(3,1)+2; A(3,2)=0.5
```

```
A =
```

```
1.0000    2.0000    3.0000
4.0000    5.0000   -6.0000
9.0000    0.5000    9.0000
```

Tačka-zarez odvaja više naredbi u jednoj liniji i ne ispisuje rezultat

# MATLAB konstante

- Ludolfov broj **pi** (3.1415...)
- Imaginarna jedinica **i** ili **j**
- Beskonačna vrijednost **Inf**
- Neodređena vrijednost (*not a number*) **NaN**
- Najmanji i najveći realni broj (po apsolutnoj vrijednosti) se mogu dobiti sa **realmin** i **realmax**, respektivno
- Dijeljenje nulom ne prekida izvršenje programa, kao u nekim jezicima, već, uz odgovarajuće upozorenje, nastaje vrijednost **Inf**
- **NaN** je rezultat neodređenih izraza **Inf/Inf** ili **0/0**.

# Matrice i funkcije

- Tip funkcije:
  - ugrađena (unutrašnja, najbrže se izvršava),
  - M-fajl iz MATLAB/Octave biblioteke (spoljašnja),
  - funkcija korisnika (spoljašnja, M-fajl).
- Funkcija se poziva njenim imenom, argumenti se navode u malim zagradama () i razdvojeni su zarezom.
- Funkcija za određivanje dimenzije matrice je **size(A)**; broj vrsta - **size(A, 1)**; broj kolona - **size(A, 2)**.
- Funkcija za određivanje dužine vektora je **length(C)**, ukoliko je C matrica, vraća veću dimenziju.
- Ako funkcija vraća više vrijednosti, promjenljive koje ih prihvataju se navode u uglastim zagradama [ ] razdvojene zarezom:

```
>> [m,n] = size(A)
```

# Neke od najviše korišćenih funkcija

- Parametri i povratne vrijednosti - rezultati funkcije su matrice.
- Korjenovanje: **sqrt(x)**
- Apsolutna vrijednost: **abs(x)**
- Prirodni logaritam: **log(x)**
- Logaritam sa osnovom 10: **log10(x)**, osnovom 2: **log2(x)**
- Eksponencijalna funkcija: **exp(x)**
- Trigonometrijske funkcije **sin(x)**, **cos(x)**, **tan(x)**, **asin(x)**, **acos(x)**, **atan(x)**

**Primjer:** Za vrijednost  $x=2$ , izračunati vrijednost izraza:

$$\sin^2 x + \sqrt[5]{x} + e^{2x^3} + \frac{\log(x+3)}{2x}$$

```
>> x = 2;
```

```
>> sin(x)^2 + x^(1/5) + exp(2*x^3) + log10(x+3)/(2*x)
```

```
ans =
```

```
8.8861e+006
```

# Neke od najviše korišćenih funkcija - nastavak

- Snimanje promjenljive u fajl se vrši pomoću naredbe **save** :
  - `save matrice a b` => snima promjenljive a i b u fajl `matrice.mat`
  - `save matrice a*` => snimaju se sve promjenljive koje počinju slovom a u fajl `matrice.mat`
  - `save matrice` => ako se ne navedu imena promjenljivih, snimaju se sve promjenljive iz radnog prostora (eng. *workspace*) u fajl `matrice.mat`
- Učitavanje promjenljivih iz fajla u radni prostor se vrši naredbom **load**:
  - `load matrice a b` => učitavaju se promjenljive a i b iz fajla `matrice.mat`
  - `load matrice a*` => učitavaju se sve promjenljive koje počinju slovom a iz fajl `matrice.mat`
  - `load matrice` => ako se ne navedu imena promjenljivih, učitavaju se sve promjenljive iz fajl `matrice.mat` u radni prostor

# Neke od najviše korišćenih funkcija - nastavak

- Brisanje promjenljive iz radnog prostora se vrši pomoću naredbe **clear**:  
**clear a b** => brisanje promjenljivih a i b  
**clear a\*** => brisanje svih promjenljivih koje počinju slovom a  
**clear** => brisanje svih promjenljivih iz radnog prostora
- Brisanje komandnog prozora se vrši naredbom **clc**.
- Snimanje sadržaja komandnog prozora u tekstualni fajl se vrši naredbom **diary**. Nakon te naredbe, sve što otkucamo u komandnom prozoru i rezultati koje dobijemo se snimaju u tekstualni fajl **diary.txt**.
- Izlistavanje svih promjenljivih koje se trenutno nalaze u radnom prostoru se vrši naredbom **who**, dok se detaljni prikaz dobija naredbom **whos**.