



# Naučni softverski alati

Naredbe vezane za logičke operacije.  
Funkcijski fajlovi

# Naredbe vezane za logičke operacije

- **any(A)** – vraća 1 za svaku kolonu matrice A, takvu da u njoj postoji makar jedna vrijednost različita od nule

```
>> A = [1, 2, 0, 0; 2, 3, 0, 1]
```

```
A =
```

```
    1    2    0    0
    2    3    0    1
```

```
>> any(A)
```

```
ans =
```

```
    1    1    0    1
```

```
>> x = [1 2 0 8 4 0 6]
```

```
x =
```

```
    1    2    0    8    4    0    6
```

```
>> any(x)
```

```
ans =
```

```
    1
```

# Naredbe vezane za logičke operacije

- **all(A)** – vraća 1 za svaku kolonu matrice A, takvu da su u njoj sve vrijednosti različite od nule

```
>> A = [1, 2, 0, 0; 2, 3, 0, 1]
```

```
A =
```

```
    1    2    0    0
    2    3    0    1
```

```
>> all(A)
```

```
ans =
```

```
    1    1    0    0
```

```
>> x = [1 2 0 8 4 0 6]
```

```
x =
```

```
    1    2    0    8    4    0    6
```

```
>> all(x)
```

```
ans =
```

```
    0
```

# Naredbe vezane za logičke operacije

- **find(A)** – za vektor vraća poziciju, a za matricu redni broj elemenata (broji po kolonama) koji su različiti od nule.

```
>> A = [1, 2, 0, 0; 2, 3, 0, 1]
```

```
A =
```

```
    1    2    0    0
    2    3    0    1
```

```
>> find(A)
```

```
ans =
```

```
1
2
3
4
8
```

```
>> x = [1 2 0 8 4 0 6]
```

```
x =
```

```
    1    2    0    8    4    0    6
```

```
>> find(x)
```

```
ans =
```

```
    1    2    4    5    7
```

# Naredbe vezane za logičke operacije

- $[m, n] = \text{find}(A)$  – za matricu vraća indekse elemenata koji su različiti od nule.  $m$  – broj vrste,  $n$  – broj kolone

```
>> A = [1, 2, 0, 0; 2, 3, 0, 1]
```

```
A =
```

```
    1    2    0    0
    2    3    0    1
```

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

```
m =
```

```
1
2
1
2
2
```

```
n =
```

```
1
1
2
2
4
```

Petom elementu koji je veći od nule bismo pristupili na sljedeći način:

```
>> A(m(5),n(5))
```

```
ans =
```

```
    1
```

# Naredbe vezane za logičke operacije

- Logički operatori se mogu koristiti za utvrđivanje elemenata matrice/vektora koji zadovoljavaju željeni logički uslov

```
>> A = [1, 2, 0, 0; 2, 3, 0, 1]
```

```
A =  
    1    2    0    0  
    2    3    0    1
```

```
>> A > 2
```

```
ans =  
    0    0    0    0  
    0    1    0    0
```

```
>> [m, n] = find(A > 2)
```

```
m =  
    2  
  
n =  
    2
```

```
>> [m, n] = find(A == 2)
```

```
m =  
    2  
    1
```

```
n =  
    1  
    2
```

```
>> x = [1 2 0 8 4 0 6];
```

```
>> find(x > 2)
```

```
ans =  
    4    5    7
```

```
>> x(x > 2)
```

```
ans =  
    8    4    6
```

# Naredbe vezane za logičke operacije

- **isnan(x)** – Vraća 1 za elemente matrice koji su NaN.
- **isfinite(x)** – Vraća 1 za konačne elemente matrice (niju NaN ili  $\infty$ ).
- **isempty(x)** – Vraća jedinicu ako je  $x$  prazna matrica,  $x=[]$ .
- **isstr(x)** – Vraća jedinicu ako je matrica  $x$  string.
- Primjer: Napisati m-fajl *broj2* kojim se od korisnika traži unos matrice  $A$  i određuje i ispisuje koliko ima elemenata većih od 2.

```
A = input('Unesi elemente matrice A');  
broj_2 = sum(sum(A == 2)); % S = sum(sum(A .* (A > 2)))  
disp(broj_2) % protumačute kod u gornjem komentaru
```

```
>> broj2
```

Jedno izvršenje

```
Unesi elemente matrice A[1 3 -4 2 5;1 3 4 2 6; 2 2 3 4 2]
```

```
5
```

# Funkcijski fajlovi

- **m-funkcije** su skript-fajlovi određenog formata zaglavlja:

```
function [iz1, iz2, ..., izN] = ime(ul1, ul2, ..., ulM)  
naredbe
```

- ul1, ul2, ..., ulM su ulazni parametri, a iz1, iz2, ..., izN izlazni parametri, koji negdje u tijelu funkcije poprimaju vrijednosti.
- Funkcija se izvršava pozivom u MATLAB/Octave komandnom prozoru  

```
>> [iz1, iz2, ..., izN] = ime(ul1, ul2, ..., ulM)
```
- Kod poziva funkcije se ne moraju navoditi svi parametri, ukoliko je samom realizacijom omogućen promjenljiv broj parametara.
- Broj prosljeđenih ulaznih parametara se u funkciji dobija **nargin**, a izlaznih **nargout** komandom.
- Sve promjenljive definisane unutar funkcije su lokalnog karaktera, tj. nijesu vidljive van funkcije. Funkcije mogu pristupati samo promjenljivim koje su im prosljeđene kao argumenti ili definisane u tijelu funkcije!!!
- Funkcijski fajl komunicira sa radnim prostorom samo preko promjenljivih ulaza i izlaza. Nazivi promjenljivih za koje se vrši poziv ne moraju odgovarati nazivima promjenljivih u zaglavlju funkcije.



# Funkcijski fajlovi - nastavak

- Funkcijski fajlovi omogućavaju izvršavanje koda i vraćanje rezultata.
- Obavezno u prvoj liniji (zaglavlju) imaju ključnu riječ **function**
- Uobičajeno je da ime funkcije bude jednako imenu fajla u kojem se snima funkcija.
- Na primjer, funkcija koja za argument ima broj **x** i vraća kvadrat tog broja izgleda:

```
function y = kvadrat(x)
y = x ^ 2
```

- U funkciji kvadrat, **x** predstavlja **ulazni parametar** funkcije, a **y** **izlazni**.
- Funkcija kvadrat se poziva na sledeći način:

```
>> a = kvadrat(4.5)
```

```
a =
```

```
20.2500
```

Da nijesmo naveli promjenjivu kojoj se dodjeljuje rezultat, isti bi bio upisan u *ans*.

Ovdje **y** i **x** nijesu vidljivi – lokalna promjenljiva

# Funkcijski fajlovi - primjer

- Napisati f-jski fajl pod nazivom suma\_reda kojim se za ulazni argument  $x$  vraća kao rezultat suma reda

$$f(x, N) = \sum_{k=1}^N x^k / k^2,$$

- pri čemu se sumiranje vrši sve dok je opšti član veći od  $10^{-5}$ , ukoliko se funkcija pozove sa jednom promjenljivom, odnosno dok je opšti član veći od drugog ulaznog argumenta, ako se pozove sa dva ulazna argumenta.
- Ukoliko se fajl pozove sa dva izlazna argumenta, kao drugi izlazni argument dati broj sumiranih elemenata reda.
- Javiti grešku ukoliko je  $x$  po apsolutnoj vrijednosti veće od 1. (Da bi red konvergirao, mora biti  $|x^k / k^2| < 1$ )

# Funkcijski fajlovi – primjer predlog rješenja

```
function [s, br] = suma_reda(x,e)
% funkcijski fajl za racunanje sume reda
if(abs(x) > 1)
    error('|x| mora biti manje od jedan')
end
if(nargin == 1)
    e = 1e-5;
end
k = 1;
s = 0;
while(abs(x ^ k / k ^ 2) > e)
    s = s + x ^ k / k ^ 2;
    k = k + 1;
end
if(nargout == 2)
    br = k - 1;
end
```

```
help uz naziv funkcije daje tekst koji
smo ispisali nakon zaglavlja, posle %
>> help suma_reda
    funkcijski fajl za racunanje sume reda
```

# Funkcijski fajlovi – Nekoliko načina pozivanja

```
>> suma_reda(0.3)
```

```
ans =
```

```
0.3261
```

```
>> suma = suma_reda(0.5)
```

```
suma =
```

```
0.5822
```

```
>> [suma, broj] = suma_reda(0.5)
```

```
suma =
```

```
0.5822
```

```
broj =
```

```
9
```

```
>> [suma, broj] = suma_reda(0.5, 1e-7)
```

```
suma =
```

```
0.5822
```

```
broj =
```

```
15
```

# Funkcijski fajlovi - primjer

- Napisati funkcijski fajl za crtanje funkcije  $\sin(x)/x$ . Grafik se crta u 100 tačaka, u intervalu  $-2\pi \leq x \leq 2\pi$ , odnosno u granicama zadatim vektorom koji čini ulazna promjenljiva, ukoliko se ista proslijedi pri pozivu.

```
function si(x)
% funkcija sin(x)/x, crta se
% u intervali -2pi do 2pi, ukoliko se
% ne zada drugacije
if(nargin == 1)
    x = linspace(x(1), x(2));
else
    x = linspace(-2*pi, 2*pi);
end
x(x == 0) = eps;
y = sin(x) ./ x;
plot(x,y)
```

Dva moguća načina pozivanja,  
sa i bez argumenata

```
>> si
>> si([-4 * pi, 4 * pi])
>> help si
    funkcija sin(x)/x, crta se
    u intervali -2pi do 2pi, ukoliko se
    ne zada drugacije
```