# MATLAB 2017 3. gyakorlat

Formázott kiíratás, mátrixok, vezérlőszerkezetek



## Formázott kiíratás

- disp-szöveg vagy változótartalom egyszerű kiiratása (disp(ones(2)./[2 3; 4 5]) % → 0.5000 0.3333 % 0.2500 0.2000
- # fprintf szöveg kiiratása formázott-beágyazott mezőkkel (vagy fájlba, vagy a konzolra); különösen fontos: formatSpec rész a help-ben
- \$\mathbf{x} sprintf mint fprintf, csak string-változóba
  "nyomtat"



## Formázott kiíratás

## formatSpec

fprintf('Teszt: \n\tEgész: %2.0f\n\tTizedes: ...
%2.4f\n\tNormál: %e\n', 10\*pi, 10\*pi, 10\*pi)

sortörés, tabulálás

% - mező nyitása

2 - minimum mező szélesség

- . decimális pont
- 4 tizedesjegyek száma
- f fixpontos típus

változó hosszúságú paraméterlista

Teszt:

Egész: 31 Tizedes: 31.4159

Normál: 3.141593e+01



## Formázott kiiratás

format - lebegőpontos számok kijelzésének pontossága

```
format short; disp(pi/100) % \rightarrow 0.0314 format long; disp(pi/100) % \rightarrow 0.031415926535898 format shortE; disp(pi/100) % \rightarrow 3.1416e-02
```

X num2str-szám → string konverzió

```
strPi=num2str(pi,3);
disp(strPi); % \rightarrow 3.14
```

★ mat2str-mátrix → string konverzió

```
strVector=mat2str(1:5);
disp(strVector); % \rightarrow [1 2 3 4 5]
```



## Mátrixok, alapműveletek

korte = [77; 88];

[alma, korte; korte',

[alma, korte]



tmp2 = alma(2);alma(2, 2:3)teljes tartomány - pl összes oszlop: alma(2, :)

korte']

 $idx = [1 \ 3 \ 4];$ alma(idx) alma(idx2)

idx2 = logical([1 0 1; 1 0 0]);

2. oszlop 3. mélység

4. ...

## Mátrixok, alapműveletek

- x size, ones, zeros, rand, diag, eye
- 🗶 squeeze az 1 kiterjedésű (singleton) dimenziók eltávolítása

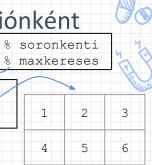
```
dio = ones(2, 1, 3); % 2 sor, 1 oszlop, 3 mélység squeeze(dio)
```

- reshape átméretezés

  B-nek legyen m sora, n oszlopa;

  B = reshape (A, m, n)

  A-ból az elemek oszlopirányban/
  oszlopfolytonosan választódnak ki
- 🗶 numel a mátrix összes elemének száma
- sum, prod, mean, min, max közös bennük: dimenziónként működnek (az első: oszlopirányban)
  alma = [1 2 3; 4 5 6]; % soronkenti max (alma, [], 2)
  % maxkereses
- logikai műveletek, logikai indexelés,
   elemenkénti műveletvégzés



alma = [1 2 3; 4 5 6];

9

-sum(alma)

5

1

1

## Vezérlőszerkezetek – elágazás

#### IF

if feltetel
 parancsok
end

#### **ELSE**

if feltetel
 parancsok1
else
 parancsok2
end

#### **ELSEIF**

if feltetel1
 parancsok1
elseif feltetel2
 parancsok2
else
 parancsok3
end

- nincs zárójelezés, az adott blokk végét az end jelzi
- a feltétel logikai értékét vizsgálja (ami nem 0, az igaz)
- ✗ relációs operátorok: ==, ~=, <, >, <=, >=
- X logikai operátorok: &, &&, |, ||, ~, xor, all, any



## Vezérlőszerkezetek – ciklusok

#### **FOR**

for n = 1:10
 parancsok
end

ismert számú iteráció elvégzésére

nincs zárójelezés, a blokk végét az end jelzi
 a ciklusváltozót sorvektorként definiáljuk, a

a **ciklusváltozó**t **sorvektor**ként definiáljuk, a ciklus törzsében értéke az aktuális elemnek megfelelő skalár

a ciklusváltozó tetszőleges sorvektor lehet (pl. [1 3 7 5 6])

#### WHILE

while feltetel parancsok end

ismeretlen számú iteráció elvégzésére, egy feltétel teljesüléséig

nincs zárójelezés, a blokk végét az end jelzi

nincs explicit módon megadott ciklusváltozó

**x** figyeljünk a végtelen ciklus elkerülésére!



## Vezérlőszerkezetek – switch

```
SWITCH
in = input('Irj be egy szamot: ');
switch in
    case 0
        disp('Nullat irtal be');
    case 1
        megoldas = 2*pi*exp(3.2);
        disp(megoldas);
    case {2, 5}
        disp('A bemenet 2 vagy 5');
    otherwise
        disp('Nem jo erteket adtal meg...');
end
```



### Feladatok

A feladatgyűjtemény 3.1 - 3.2 feladatai, egy-egy függvényben, melyeknek neve ez legyen, rendre:

```
gyak3_f31_[NEPTUN].m
gyak3_f32_[NEPTUN].m
```

(váz a weboldalon)

A diasorban 4-10 diákon ismertetett parancsok kikeresése és tanulmányozása a Help-ben

Amivel nem végzel / nem végzünk, azt otthon kell befejezni, ez a házi feladat is egyben. A határidő vasárnap (március 5.) éjfél.

Feltöltés: users.itk.ppke.hu/~zseta/matlab2017