

MATLAB 2024

8. gyakorlat

Cellatömbök, struktúrák, fájlműveletek



[illegible]

- $$C = \{1, 2; 3, 4\}$$



Cellatömb létrehozás, és ábrázolás

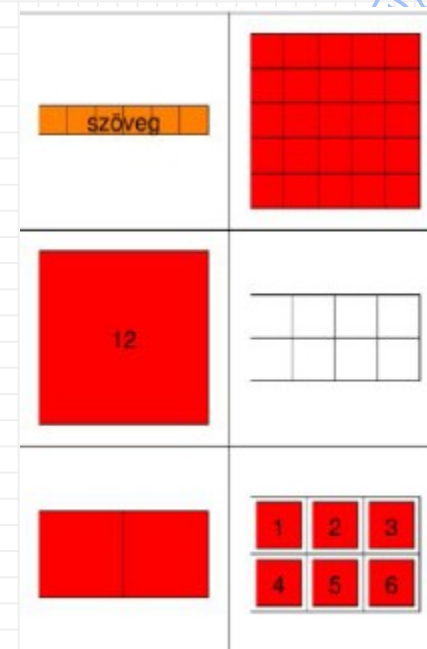
- ✗ Hozzunk létre egy 3×2 méretű üres cellatömböt!
- ✗ Az egyes cellákat oszlopfolytonosan töltjük fel az alábbi elemekkel:
- ✗ 'szöveg' , 12, [8.13, 14.86], magic(5), cell(2,4),
{1, 2, 3; 4, 5, 6}
- ✗ Jelenítsük meg a cellatömb tartalmát a cellplot parancs használatával!
- ✗ Próbálgassuk az egyes elemek indexelését!

```
% cellatömb létrehozása
C1 = cell(3,2);

% feltöltés oszlopfolytonosan
C1{1,1} = 'szöveg';
C1{2,1} = 12;
C1{3,1} = [8.13, 14.86];
C1{1,2} = magic(5);
C1{2,2} = cell(2,4);
C1{3,2} = {1, 2, 3; 4, 5, 6};

% másik lehetőség
C2 = {'szöveg', magic(5); 12, cell(2,4); [8.13 14.86], {1, 2, 3; 4, 5, 6}};

% megjelenítés
cellplot(C1);
title('C1 tartalma');
figure;
cellplot(C2);
title('C2 tartalma');
```



Struktúrák

- ✗ A cellatömbökhöz hasonlóan **különböző típusú** és/vagy **méretű** változók tárolására használható **adattípus** azzal a különbséggel, hogy a tárolt adatok **névvel ellátott mezőkbe** kerülnek (hasonlóan pl a C nyelv `struct` típusához).
- ✗ Az adatok címzése a `változónév.mezőnév` minta alapján történik

Struktúra létrehozása

```
hallgato.nev = 'Kis Pista';  
hallgato.szul_datum = '1992.03.24.';  
hallgato.evfolyam = 3;  
hallgato.osztalyzatok = [2 4 3 5 3 4 4 3];
```

Struktúra/mezők lekérdezése

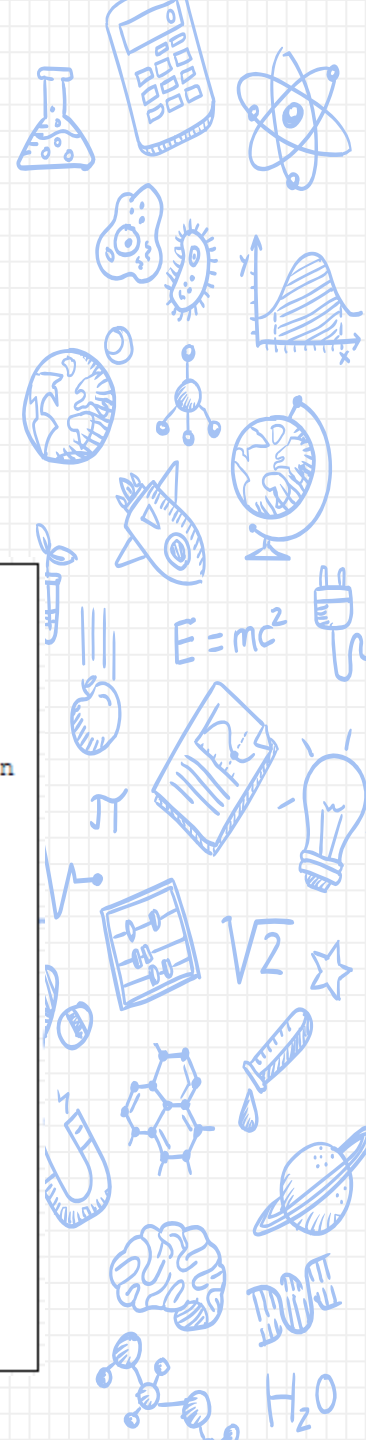
```
>> hallgato  
hallgato =  
    nev: 'Kis Pista'  
    szul_datum: '1992.03.24.'  
    evfolyam: 3  
    osztalyzatok: [2 4 3 5 3 4 4 3]  
  
>> hallgato.nev  
ans =  
Kis Pista
```

Fájlkezelés

- ✗ Többféle módon lehet MATLAB-ban fájlokat írni/olvasni, melyek közül az alábbi három módszerről lesz szó:
- ✗ fájl írás/olvasás **formázott szöveg**ként: `fprintf`, `fscanf`, `textscan`
 - ✗ Használható logok készítésére, vagy pedig egyszerű adatok olvasható tárolására, látszik a formátum
- ✗ fájl írás/olvasás **binárisan**: `fwrite`, `fread`, `save`, `load`
 - ✗ Adatok kompakt tárolása, csak programmal olvasható, a formátum nem feltétlenül derül ki a fájlból
 - ✗ A MATLAB beépített tárolási módja is bináris fájlokat generál/olvas be (.mat)
- ✗ fájl írás/olvasás **táblázatként**: `writetable`, `readtable`
 - ✗ **1 hét múlva**
 - ✗ Táblázatok hatékony kezeléséhez, nagyobb mennyiségű, rendezett adathoz praktikus
 - ✗ Többféle formátumot

Elérési útvonalak

- ✗ A kiíráshoz kell lenni fog a célfájl elérési útvonala stringként megadva:
- ✗ Az egyes platformokon más a fájlleválasztó karakter ('\' vagy '/'), ami csapatmunka esetén problémás lehet. Erre nyújt megoldást a **filesep** parancs, amely megadja az aktuális platformon használt elválasztó karaktert.
- ✗ Windows alatt pl.: `DIR_PATH = 'D:\myfolder\file_op\'`;
- ✗ Linux alatt pl.: `DIR_PATH = '/home/myfolder/file_op/'`;
- ✗ Aktuális könyvtár lekérdezése: `pwd`
- ✗ Fájlok listázása: `dir`
 - ✗ Szűrők pl.: `matFilesInfo = dir([pwd filesep '*.mat'])`;
- ✗ Fájl nyitás és **zárás**: `fopen`, `fclose`



Fájlba írás formázott szöveggént

X `fprintf(fileID, formatString, data)`

```
% kiirando jel generalasa
% mintaveteli frekvencia
Fs=1000; % Hz
% idovektor
t=0:1/Fs:10; % s
% a szinusz frekvenciaja
f=5; % Hz
% szinusz jelalak
s=sin(2*pi*f*t);

% az adott helyhez kepest vett relativ cimzes
% (lehet abszolut is, teljes utvonallal)
fs = filesep;
DIR_PATH=['.' fs 'files' fs];

% a kiirando fajl neve
filename='szinusz_fprintf.txt';

% a fajl eleresi utvonala (ha csak a
fajlnevet
% adom meg, az aktualis könyvtarba menti)
file_path=[DIR_PATH filename];
```

```
% fajl megnyitasa irasra, ez utan az fid-val
% hivatkozok erre a fajlra
fid=fopen(file_path, 'w');

% header irasa, hogy tudjuk, mi van a fajlban
% pl.: Szinusz (Fs = 1000)
fprintf(fid, ['Szinusz (Fs = ' num2str(Fs) ' )
\n']);
fprintf(fid, 't sin \n');

% az iras utan nyitva marad a fajl
% ezert irhatok meg bele formazott adatot

% irjuk bele a kiszamolt szinusz fuggvenyt
fprintf(fid, '%5.2f %4.4f\n', [t;s]);

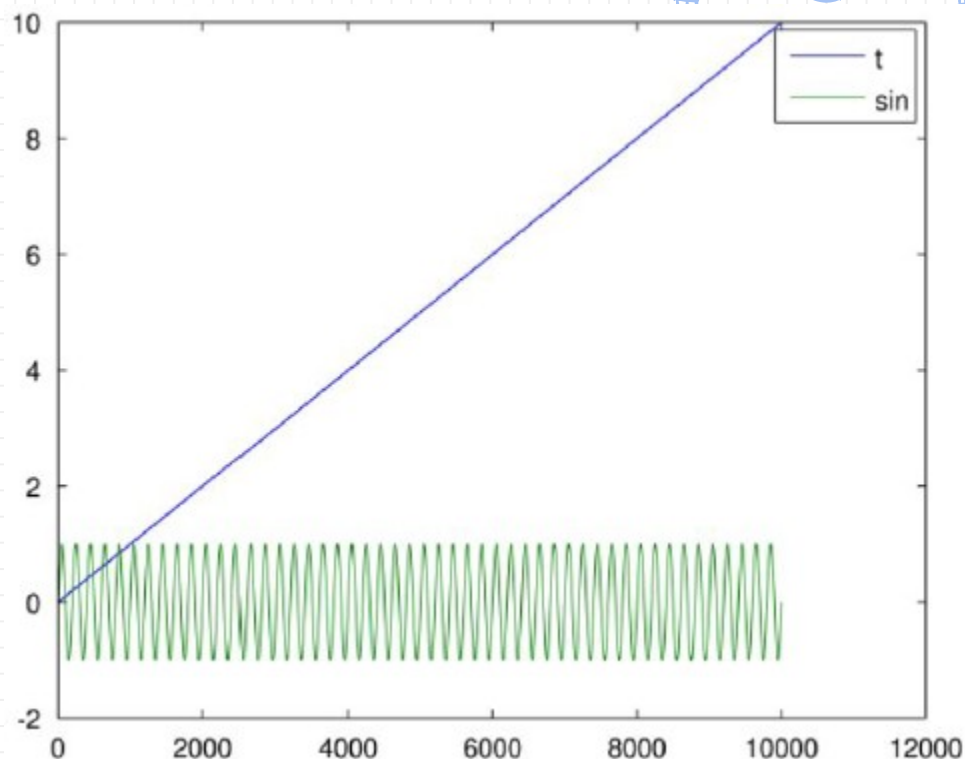
% fajl bezarasa - FONTOS, ne felejtsek el!
fclose(fid);
```

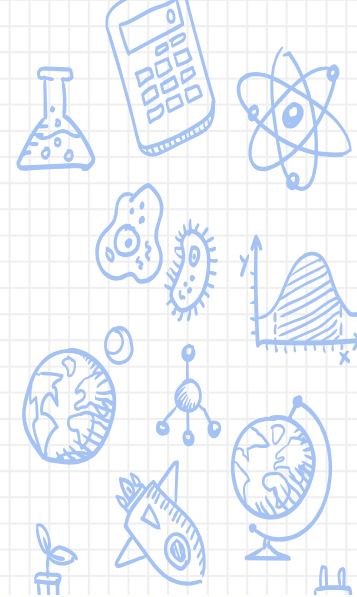



Fájlból olvasás formázott szövegeként (fscanf)

X `fscanf(fileID, formatString, outputSize)`

```
% a beolvasando fajl neve (amit az elobb  
kiirtunk)  
filename='szinusz_fprintf.txt';  
  
% a fajl eleresi utvonala  
file_path=[DIR_PATH filename];  
  
% fajl megnyitasa olvasasra  
fid=fopen(file_path);  
  
% fajl beolvasasa formazott szovegkent  
% eloszor a header beolvasasa stringkent  
Sheader=fscanf(fid,'%s',6);  
  
% majd az adatok beolvasasa lebegopontos szamkent  
Sdata=fscanf(fid,'%f %f',[2 inf]);  
  
% fajl bezarasa  
fclose(fid);  
  
% beolvasott adatok kirajzolasa  
figure;  
plot(Sdata');  
legend('t','sin');
```





Fájlból olvasás formázott szöveggként (textscan)

X `textscan(fileID, formatSpec, nTimes)`

```
% a textscan-nek lehet regularis kifejezest is adni (lásd
HELP)
% a beolvasando fajl neve (meg mindig ugyanaz)
filename='szinusz_fprintf.txt';

% a fajl eleresi utvonala
file_path=[DIR_PATH filename];

% fajl megnyitasa olvasasra
fid=fopen(file_path);

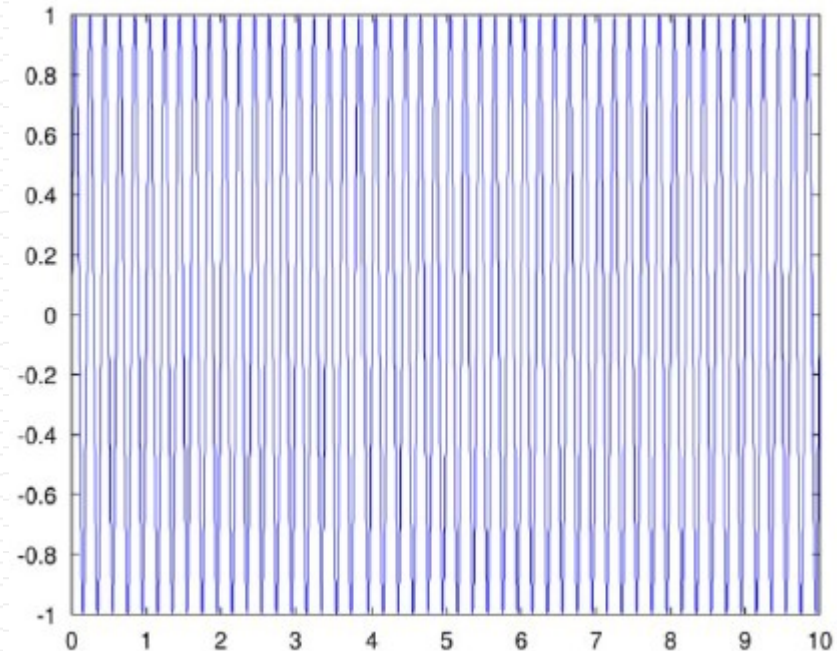
% fajl beolvasasa cellatombbe header beolvasasa
Sheader_ts=textscan(fid, '%s', 6);

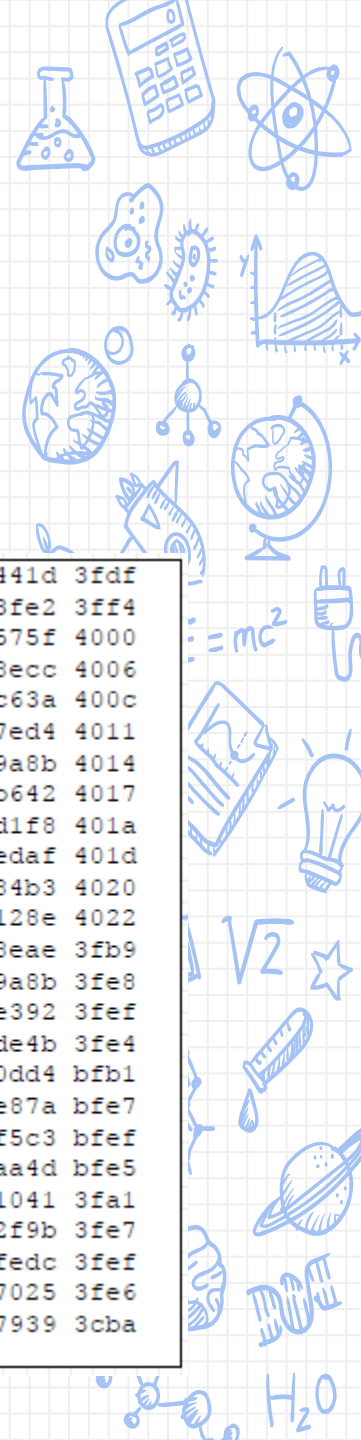
% adatok beolvasasa
Sdata_ts=textscan(fid, '%f %f');

% fajl bezarasa
fclose(fid);

% beolvasott adatok kirajzolasa
tt = Sdata_ts{1};
data = Sdata_ts{2};

figure;
plot(tt, data);
```





Fájlba írás bináris formában

X `fwrite(fileID, data, precision)`

```
% a kiirando fájl neve  
filename='szinusz_fwrite.bin'; % nem kotelezo a  
bin
```

```
fs = filesep;  
dir_path = ['. ', fs];  
file_path=[dir_path, filename];
```

```
% fájl megnyitása írásra, ez után az  
% fid-val hivatkozok erre a fájlra  
fid=fopen(file_path, 'w');
```

```
t = linspace(0.1, 3*pi, 25);  
y = sin(t);
```

```
% adatok kiírása binárisan, double-ként  
fwrite(fid, [t' y'], 'double');  
% fájl bezárása, változók törlése  
fclose(fid);
```

```
0000000 999a 9999 9999 3fb9 4f3a 7666 441d 3fdf  
0000010 1c07 4333 10ea 3fec 8839 e599 3fe2 3ff4  
0000020 826e a999 7750 3ffa be51 36cc 575f 4000  
0000030 bb6c 18cc 7316 4003 b886 facc 8ecc 4006  
0000040 b5a1 dccc aa83 4009 b2bc becc c63a 400c  
0000050 afd6 a0cc elf1 400f 5678 4166 7ed4 4011  
0000060 5505 b266 0caf 4013 5392 2366 9a8b 4014  
0000070 521f 9466 2866 4016 50ad 0566 b642 4017  
0000080 4f3a 7666 441d 4019 4dc7 e766 d1f8 401a  
0000090 4c55 5866 5fd4 401c 4ae2 c966 edaf 401d  
00000a0 496f 3a66 7b8b 401f 23fe 55b3 84b3 4020  
00000b0 2345 0e33 4ba1 4021 228c c6b3 128e 4022  
00000c0 21d2 7f33 d97c 4022 cb2c cb8b 8eae 3fb9  
00000d0 e198 b947 0982 3fde b2a5 a45f 9a8b 3fe8  
00000e0 b0e0 a17f 856c 3fee ccdf 6f6b e392 3fef  
00000f0 312e ee58 80ca 3fec 42f2 a097 de4b 3fe4  
0000100 3fc3 bd52 3edd 3fd4 40f8 bf0b 0dd4 bfb1  
0000110 1dfd 8f70 2314 bfdc 73cb 2a3e e87a bfe7  
0000120 5fd7 17a0 2f0c bfee 9258 1b28 f5c3 bfef  
0000130 fdb9 a6a0 f8d6 bfec 2bd3 4813 aa4d bfe5  
0000140 a12e 92e3 41fa bfd6 8ad1 f3b9 1041 3fa1  
0000150 9144 b5db 34a5 3fda 7c8f e9d4 2f9b 3fe7  
0000160 45bc d771 d015 3fed 53c3 bc52 fedc 3fef  
0000170 eaa2 e338 68a4 3fed 319b 7793 7025 3fe6  
0000180 243f ccb8 3ec2 3fd8 8a0a 4c9e 7939 3cba  
0000190
```



Fájlból olvasás bináris formában

X `fread(fileID, size, precision)`

```
% a beolvasando fajl neve
filename='szinusz_fwrite.bin';

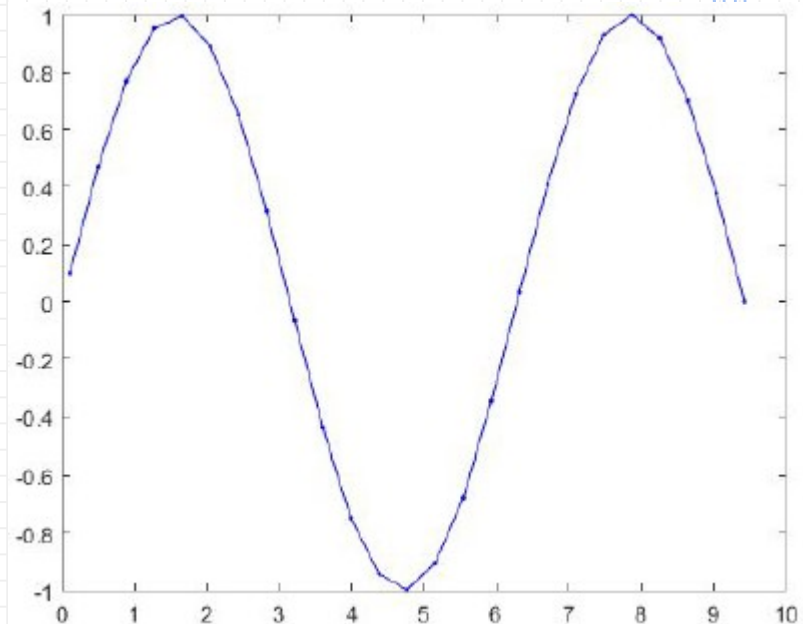
fs = filesep;
dir_path = ['.', fs];
file_path=[dir_path, filename];

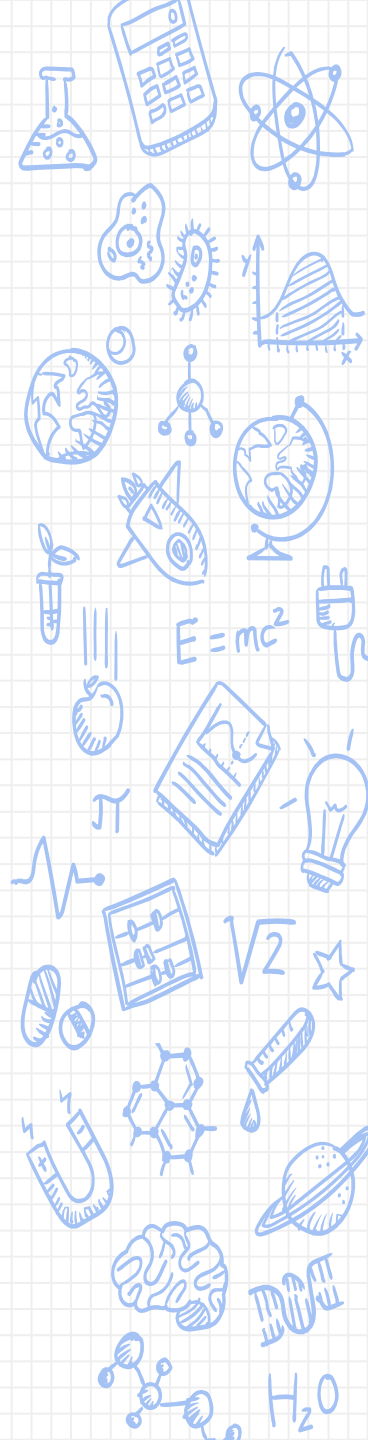
% nyitas, olvasas
fid=fopen(file_path, 'r'); % vagy siman
fid=fopen(file_path);

data = fread(fid, inf, 'double');
data = reshape(data, length(data)/2, 2);

fclose(fid);

figure;
plot(data(:, 1), data(:, 2), 'b.-');
```





Mátrix fájlba írása

X `writematrix(matrix, filePath, ...)`

```
M = rand(8, 6)
```

M = 8×6

0.7819	0.1123	0.4326	0.7690	0.0594	0.4231
0.1006	0.7844	0.6555	0.3960	0.3158	0.6556
0.2941	0.2916	0.1098	0.2729	0.7727	0.7229
0.2374	0.6035	0.9338	0.0372	0.6964	0.5312
0.5309	0.9644	0.1875	0.6733	0.1253	0.1088
0.0915	0.4325	0.2662	0.4296	0.1302	0.6318
0.4053	0.6948	0.7978	0.4517	0.0924	0.1265
0.1048	0.7581	0.4876	0.6099	0.0078	0.1343

```
% több különböző fájltypust támogat
```

```
filepath = [pwd filesep 'pelda_matrix.txt'];
```

```
filepath2 = [pwd filesep 'pelda_matrix.xlsx'];
```

```
writematrix(M, filepath);
```

```
% A fájltypus megadása nem volna kötelező, azt kitalálná
```

```
% a kiterjesztésből
```

```
writematrix(M, filepath2, 'FileType', 'spreadsheet');
```

Mátrix olvasása fájlból

X `readmatrix(filePath, ...)`

```
readmatrix(filepath)
```

```
ans = 8×6
```

0.7819	0.1123	0.4326	0.7690	0.0594	0.4231
0.1006	0.7844	0.6555	0.3960	0.3158	0.6556
0.2941	0.2916	0.1098	0.2729	0.7727	0.7229
0.2374	0.6035	0.9338	0.0372	0.6964	0.5312
0.5309	0.9644	0.1875	0.6733	0.1253	0.1088
0.0915	0.4325	0.2662	0.4296	0.1302	0.6318
0.4053	0.6948	0.7978	0.4517	0.0924	0.1265
0.1048	0.7581	0.4876	0.6099	0.0078	0.1343





Mentés/betöltés MATLAB formátum esetén

- ✗ `save(filePath, variableNames, ...)`
- ✗ `load`

```
% csak hogy tiszta legyen a munkater
clear;
% a kiirando fajl neve
filename='sajat_matlab_archivum.mat';
fs = filesep;
dir_path = ['.', fs];
file_with_path=[dir_path, filename];
t = linspace(0.1, 3*pi, 25);
y = sin(t);
dummy1 = rand(20, 2);
% most mi van a munkaterben?
% a valtozok kimentese: save utasitassal
save(file_with_path, 't', 'y')
% valtozok torlese a munkaterbol clear('t', 'y', 'dummy1'); % lehetne meg a formaja: clear t, y;
% most mi van a munkaterben?
% valtozok szelektiv betoltese -- ha tudod elore mi van benne...
load(file_with_path, 't');
% most mi van a munkaterben?
load(file_with_path, 'y');
% most mi van a munkaterben? - lehetett volna egyszerre mindent: load(file_with_path);

figure;
plot(t, y, 'b.-');
```