Основные типы данных MATLAB

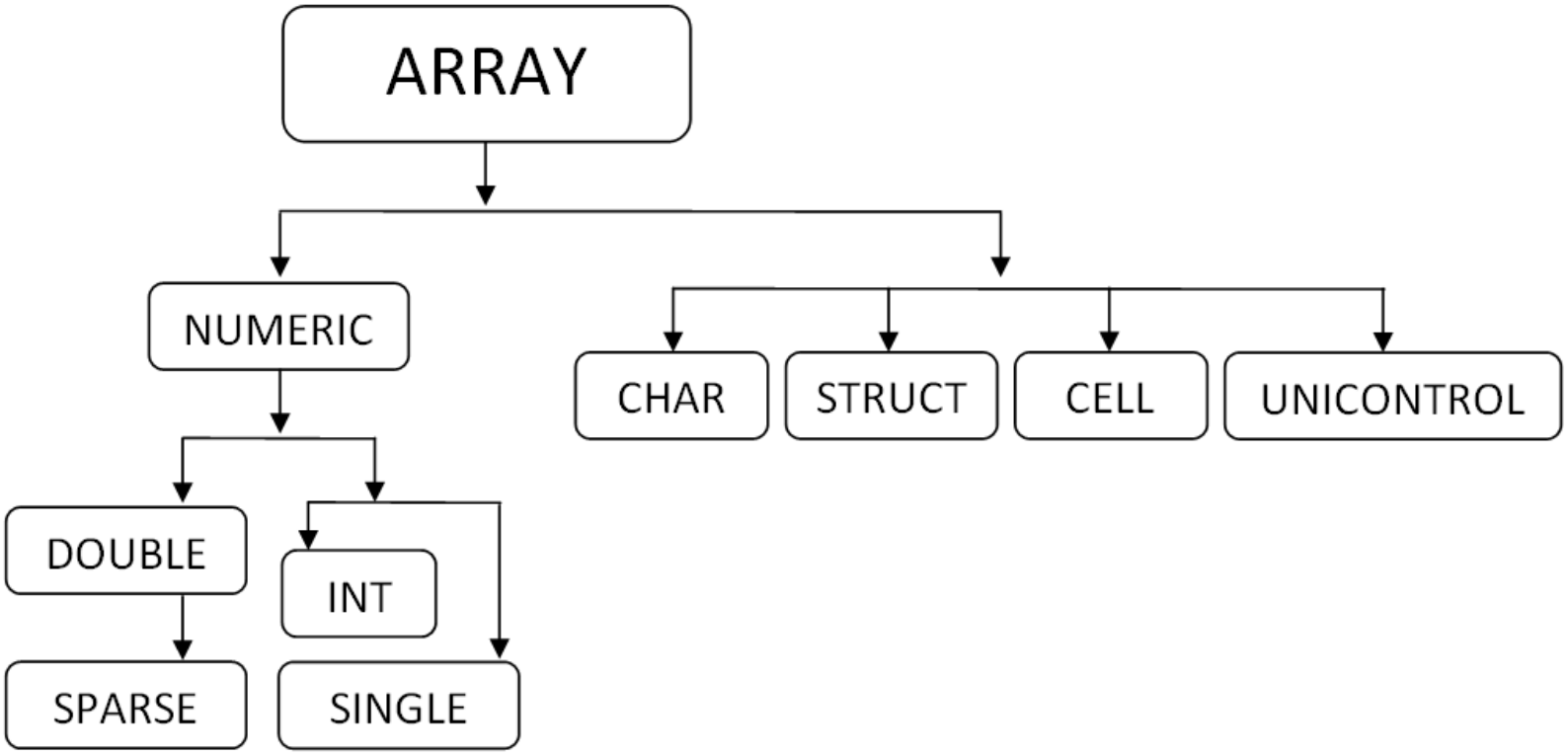
Лекция #3

Пустовалова О.Г.
доцент. каф. мат.мод.
ИММиКН ЮФУ

Содержание

Основные типы данных

Любой объект в ML, в том числе скаляр, является массивом.



Класс ARRAY является родительским для всех потомков.

Основные типы данных

Хранение массивов в ML осуществляется в векторной форме последовательно по столбцам, поэтому поддерживается как двойная (матричная) нумерация, так одинарная (векторная).

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

M =

    1.00    2.00    3.00
    4.00    5.00    6.00
    7.00    8.00    9.00

>> M(4)
ans = 2.00
```

Функции isa и class

```
class(pi)

class('hello')
```

```
ans = char

ans = double
```

```
isa(pi, 'double')

isa('hello', 'char')
```

```
ans = 1

ans = 1
```

Методы класса Array. diag

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

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

```
diag(A)
```

→ 1 5 9

```
diag(A,1)
```

→ 2 6

```
diag(A,-1)
```

→ 4 8

```
diag(diag(A))
```

→

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 9 \end{matrix}$$

Методы класса Array. ones / eye / zeros / linspace

```
ones(2,3)
```

```
1     1     1
1     1     1
```

```
eye(2)
```

```
1     0
0     1
```

```
zeros(2)
```

```
0     0
0     0
```

```
% Вектор из 100 компонент
c1=linspace(1,100)


% Вектор из 20 компонент
c2=linspace(1,100,20)
```

Методы класса Array. rand

```
rand([2 3])
```

```
0.2785    0.9575    0.1576
0.5469    0.9649    0.9706
```

```
round(rand([2,3])*10)
```

```
10     8     4
 5     1     9
```

```
round(rand([2,3])*10)-5
```

```
 2    -2    -4
-5    -5     3
```

Методы класса Array. cat

```
A=[1 2]
B=[3 4]
```

```
cat(1,A,B)
```

```
1     2
3     4
```

```
cat(2,A,B)
```

```
 1     2     3     4
```

```
cat(3,A,B)
```

```
ans(:,:,1) = 1     2

ans(:,:,2) = 3     4
```

Методы класса Array. sum / prod

```
A =   1     2
      3     4
```

```
sum(A)          →   4   6
sum(A,2)        →   3   7

prod(A)         →   3   8
prod(prod(A))   →   24
```

Методы класса Array. reshape

```
d=1:12
size(d)
```

→

```
1    12
```

```
d=reshape(d,3,4)
size(d)
```

→

```
1    4    7    10
2    5    8    11
3    6    9    12
```

```
d=reshape(d,4,[])
size(d)
```

→

```
1    5    9
2    6    10
3    7    11
4    8    12
```

Методы класса Array. reshape

```
d=1:12

d=reshape(d,5,[])
```

```
Error using reshape
Product of known dimensions, 5, not divisible into total
number of elements, 12.
```

Методы класса Array. repmat

```
d=[1 2]
```

```
1     2
```

```
d1=repmat(d,2)
```

```
1     2     1     2
1     2     1     2
```

```
d2=repmat(d,3,1)
```

```
1     2
1     2
1     2
```

Методы класса Array. fliplr / rot90

```
d=[1 2 3]
```

```
1     2     3
```

```
fliplr(d)
```

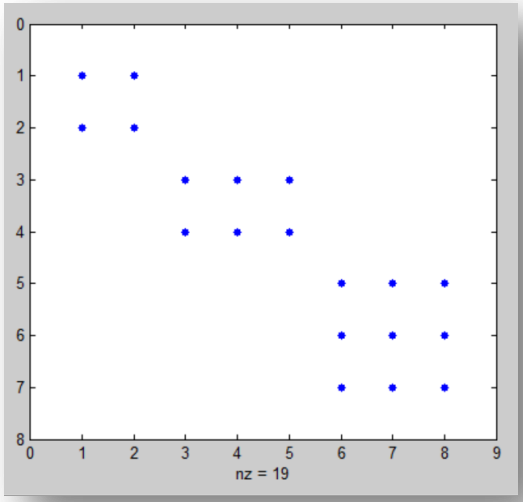
```
3     2     1
```

```
rot90(d)
```

```
3
2
1
```

Методы класса Array. blkdiag

```
m1=[1 2; 3 4]
m2=[10 20 30; 40 50 60]
m3=[2 4 6; 1 3 7; 5 4 3]

m=blkdiag(m1,m2,m3)
spy(m)
```



1	2	0	0	0	0	0	0
3	4	0	0	0	0	0	0
0	0	10	20	30	0	0	0
0	0	40	50	60	0	0	0
0	0	0	0	0	2	4	6
0	0	0	0	0	1	3	7
0	0	0	0	0	5	4	3

Методы класса Array. size

```
% size - определяет вектор размерностей массива

a=1;
size(a)                    ⟶      1    1

b=[1 2];
size(b)                    ⟶      1    2

c=[1 2 3; 4 5 6];
size(c)                    ⟶      2    3
```

Методы класса Array. length

```
% Длина вдоль большей размерности

a=1;
length(a)                    ⟶      1

b=[1 2];
length(b)                    ⟶      2

c=[1 2 3; 4 5 6];
length(c)                    ⟶      3

%определяет длину массива, записанного вектором;

length(c(:))                 ⟶      6
```

Методы класса Array. numel

```
% numel(a) -  количество элементов массива;

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

numel(a)
```

```
6
```

Методы класса Array. ndims

```
% ndims – количество размерностей многомерного массива

a=1;
ndims(a)                    ⟶    2

b=[1 2; 3 4];
ndims(b)                    ⟶    2

c=zeros(2,3,4);
ndims(c)                    ⟶    3
```

Методы класса Array. disp display

```
% display -  визуализация объектов,
            не подавляется знаком (;)


display('Thinking vectorized');

disp([1 2]*[3 4]');
```

```
11
```

Логические операции с матрицами

Операция	Знак операции
Равно	==
Не равно	~=
Больше	>
Больше или равно	>=
Меньше	<
Меньше или равно	<=
Логическое И	&
Логическое ИЛИ	|

Логические операции с матрицами. Пример

```
A =

     1     2     1
     3     1     5


A==1


ans =

     1     0     1
     0     1     0
```

Логические операции с матрицами. Примеры

```
A=[1 2 1; 3 1 5]
B=[1 1 1; 6 7 8]

sum(sum(A==1))

sum(sum(A==B))

sum(sum((A==1)&(B==1)))
```

?

?

?

Типы данных Numeric и Double

Все объекты в ML делятся на числовые и нечисловые

```
a=21;
isnumeric(a)        → 1

b='abc'
isnumeric(b)        → 0

islogical(2>3)      → 1
```

Обработка строк

- ✓ **blanks - сформировать строку пробелов**
- ✓ **cellstr - преобразовать массив символов в массив ячеек для строк**
- ✓ **char - сформировать массив символов**
- ✓ **deblank - удалить пробелы в конце строки**
- ✓ **double - преобразовать символы строки в числовые коды**

Проверка строк

- ✓ ischar - истинно, если это массив символов (строка)
- ✓ iscellstr - истинно, если это массив ячеек для строк
- ✓ isletter - истинно, если это символ алфавита
- ✓ isspace - истинно, если это пробел

Операции над строками

- ✓ strcat - горизонтальное объединение строк
- ✓ strvcat - вертикальное объединение строк
- ✓ strcmp - сравнить строки
- ✓ strncmp - сравнить n символов строк
- ✓ findstr - найти заданную строку в составе другой строки
- ✓ strjust - выравнять массив символов

Операции над строками

- ✓ strmatch - найти все совпадения
- ✓ strrep - заменить одну строку другой
- ✓ strtok - найти часть строки, ограниченную разделителями
- ✓ upper - перевести все символы строки в верхний регистр
- ✓ lower - перевести все символы строки в нижний регистр

Преобразование строк

- ✓ num2str - преобразование числа в строку
- ✓ int2str - преобразование целого в строку
- ✓ mat2str - преобразование матрицы в строку
- ✓ str2mat - объединение строк в матрицу
- ✓ str2num - преобразование строки в арифметическое выражение и его вычисление
- ✓ sprintf - записать форматированные данные в виде строки
- ✓ sscanf - прочитать строку с учетом формата

Преобразование систем счисления

- ✓ **hex2num – преобразовать шестнадцатеричное число в число удвоенной точности**
- ✓ **hex2dec – преобразовать шестнадцатеричное число в десятичное число**
- ✓ **dec2hex – преобразовать десятичное число в шестнадцатеричную число**
- ✓ **bin2dec – преобразовать двоичную строку в десятичное число**
- ✓ **dec2bin – преобразовать десятичное число в двоичную строку**
- ✓ **base2dec – преобразовать B-строку в десятичное число**
- ✓ **dec2base – преобразовать десятичное число в B-строку**

Массивы ячеек. cell

Массив ячеек – это массив, элементами которого являются ячейки, содержащие массивы любого типа, в том числе и массивы ячеек.

```
C = {1 20 33 404}
```

```
B = cell(2,3)% пустой массив ячеек 2x3
```

```
% Можно построить массив ячеек, присваивая данные отдельным
% ячейкам
A(1, 1) = {[1 4 3; 0 5 8]}
A(1, 2) = {'Математика'}
A(2, 1) = {3+7i}
A(2, 2) = {-2:2:6}
```

Функции работы с массивами ячеек

{}, cell – создание массива ячеек;

```
A={[1,2] 'ABC' 3.1415}
```

```
A =
    [1x2 double]    'ABC'    [3.1415]
```

```
k = iscell(A)
```

```
1
```

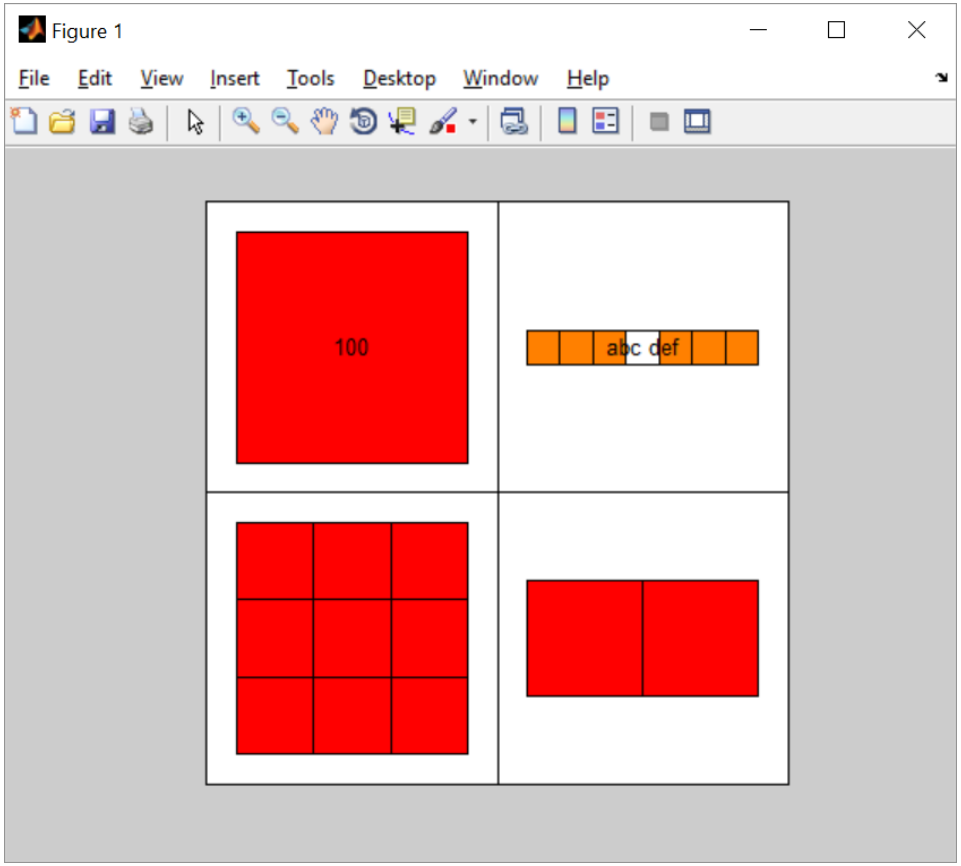
```
% пустой массив ячеек 2x3
Z=cell(2,3)
```

Функции работы с массивами ячеек

cellplot – отображения структуры массива ячеек

```
q{1, 1} = 100;
q{1, 2} = 'abc def';
q{2, 1} = eye(3);
q{2, 2} = [1 12];
```

```
cellplot(q)
```

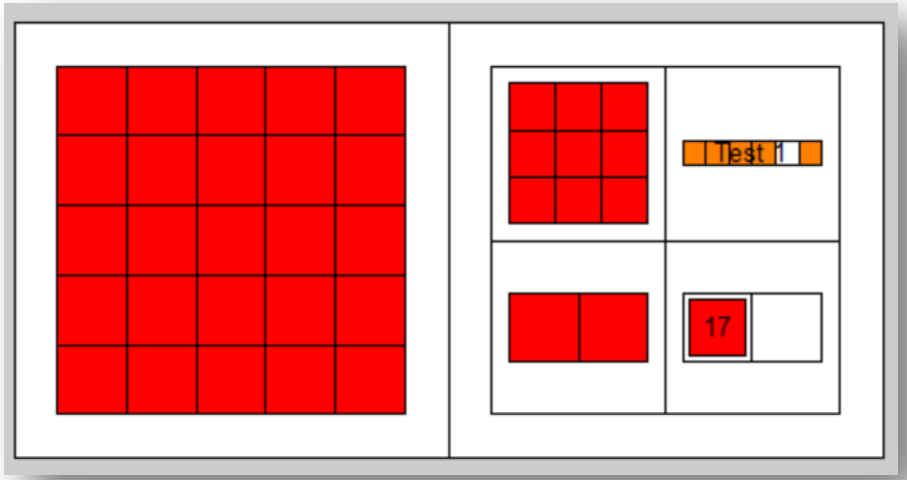


Обращение к ячейкам. Удаление ячеек

```
c = A{1, 2}


% Извлечение элемента с индексами (2,2)
% из числового массива ячейки A{1,1}
d = A{1, 1}(2, 2)

% Доступ к подмножеству ячеек
B=A(1,:)


% Удалять можно либо целую строку, либо столбец
A(1, :)=[] % удаление первой строки
```

Вложенные массивы ячеек

```
A(1, 1) = {magic(5)};

A(1, 2) = {
            {
              [5 2 8; 7 3 0; 6 7 3]
              'Test 1';
              [2-4i 5+7i]
              {17 []}
             }
            }
```



Многомерные массивы ячеек

```
A{1, 1} = 'Name';
A{1, 2} = [4 2; 1 5];
A{2, 1} = 2-4i;
A{2, 2} = 7;
```

```
B{1, 1} = 'Name2';
B{1, 2} = [ 3 5 ]';
B{2, 1} = 0:1:3;
B{2, 2} = 3;
```

```
C = cat(3, A, B);
C(2,2,:)
```

```
ans(:,:,1) =     [7]

ans(:,:,2) =     [3]
```

Функции работы с массивами ячеек. deal

С помощью функции deal возможно множественное присваивание входных данных выходным

```
[A,B,C,…]=deal(X,Y,Z,…)
```
→ A=X, B=Y, C=Z

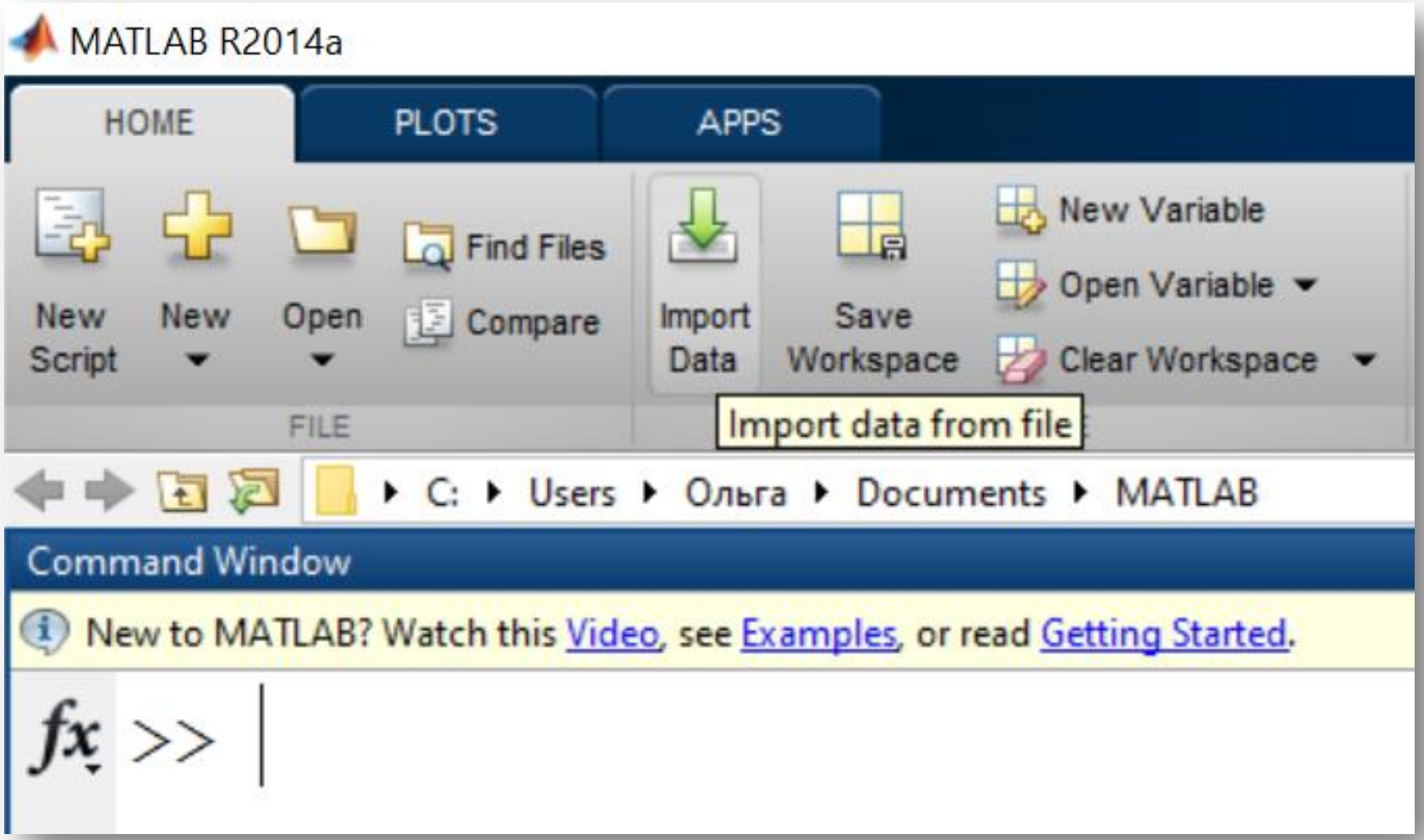
```
[A,B,C…]=deal(X)
```
→ A=X, B=X, C=X

```
X={1,'Hello!',[7,13]}
[A,B,C]=deal (X{:})
```
→ A=1, B='Hello!', C=7,13

Функции работы с массивами ячеек

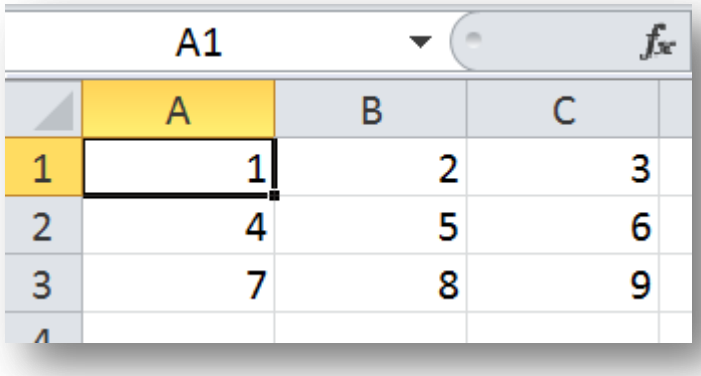
- {}, cell – создание массив ячеек;
- cellstr – создание массива ячеек строк из символьного массива;
- cellfun – применение функции к каждому элементу в массиве ячеек;
- celldisp – показ содержимого массива ячеек;
- cellplot – показ графической структуры массива ячеек;
- deal – обмен данными;
- num2cel – преобразование числового массива в массив ячеек;
- cell2mat – преобразование массива ячеек в отдельную матрицу;
- mat2cell – разбиение матрицы на массив ячеек матриц;
- cell2struct – преобразование массива ячеек в структуру;
- struct2cell – преобразование структуры в массив ячеек;
- iscell – определяет, является ли введенная переменная массивом ячеек.

Импортирование объектов. File →ImportData



Импортирование объектов. xlsread

xlsread – чтение файла Excel




```
A=xlsread('d:\file01.xlsx')
```

имена листов следует писать на английском языке!

Импортирование объектов. Текстовые файлы

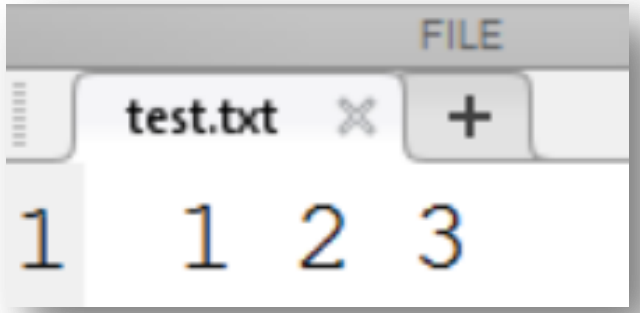
load file.txt – загрузка текстового файла

```
load 'test.txt'

test
```

```
test =

     1     2     3
```

Импортирование объектов. fopen. Чтение из файла

f>0, значит файл открыт




```
f = fopen('test.txt', 'r')
```

режим чтения

```
a = fscanf(f, '%d',  [3, 1] )
```

идентификатор

формат

размер

```
a = a.'
```

транспонирование

```
a =   1     2     3
```

Импортирование объектов. fscanf. Чтение из файла

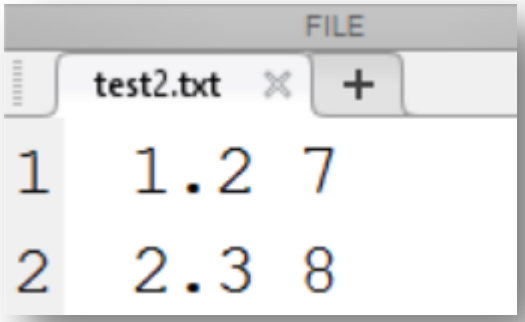
```
a = fscanf(f, format, size)
```

format	
%g	double с плавающей точкой машинного представления
%f	double с фиксированной точкой
%e	double с плавающей точкой
%s	строка, пробелы в которой не учитываются
%c	строка, учитываются пробелы
%d	целые десятичные

size – размер матрицы; столбцы матрицы, последовательно записаны в вектор-строку;

- [3,2] – 3 столбца, 2 строки
- [3,inf] – 3 столбца, количество строк неизвестно

Импортирование объектов. fscanf. Чтение из файла




```
f = fopen('test2.txt', 'r')

a = fscanf(f, '%f %d',  [2, inf] );

a=a'
```

2 столбца

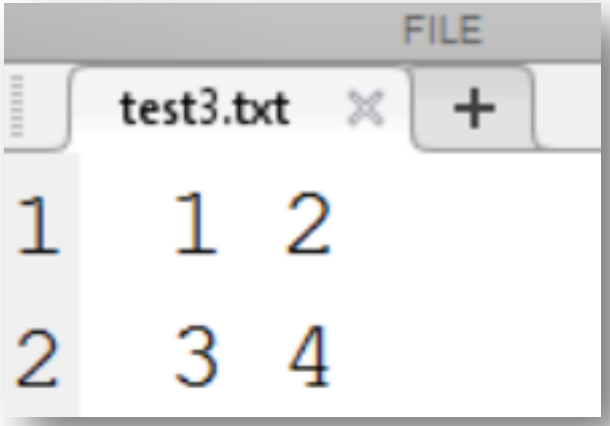
Количество строк неизвестно

```
a =
    1.2000    7.0000
    2.3000    8.0000
    4.5000    9.0000
```

Запись в файл. fprintf. Форматированный вывод

```
b=[1 2]
c=[3 4]

f = fopen('test3.txt', 'w')

fprintf(f, '%d %d\n',  b');

fprintf(f, '%d %d',  c');
```

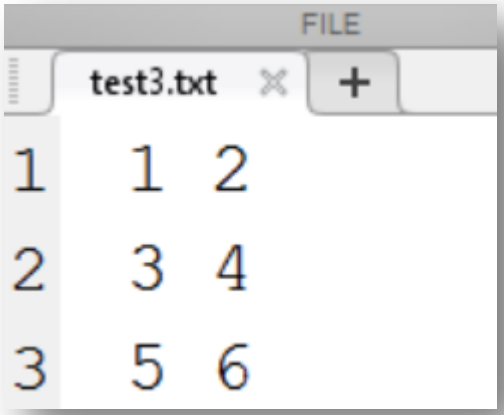
запись в файл



Запись в файл. fprintf. Форматированный вывод

```
g=[5 6]


f = fopen('test3.txt', 'a')

fprintf(f, '%d %d\n',  g');
```

ДОзапись в файл



Спасибо за внимание!