

## % Лекция 3. Часть 2

```
%% Обработка строк
% Основные функции
% blanks Сформировать строку пробелов
clear
clc
s1="a"+blanks(3)+"b"

%%
% cellstr Преобразовать массив символов в массив ячеек для строк
clear
clc
a1=["a","b","c"]
a2=cellstr(a1)

%%
%char Сформировать массив символов
clear
clc
a1=["a","b","c"]
a2=char('abc')
a3=char(a1)

%%
A = [77 65 84 76 65 66];
C = char(A)

%%
A1 = [65 66; 67 68];
A2 = 'abcd';
C = char(A1,A2)

%%
% deblank Удалить пробелы в конце строки
clear
clc
s1='abc '
s2=deblank(s1)

%%
% double Преобразовать символы строки в числовые коды
clear
clc
s=double(' 09AaZz')

%%
% Проверка строк
% ischar Истинно, если это массив символов (строка)
clear
clc
```

```

ischar(['a','b','c']) % 1
ischar('abc') % 1
ischar({'abc'}) % 0
ischar(123) % 0

%%
% iscellstr Истинно, если это массив ячеек для строк
clear
clc
iscellstr({'abc'}) % 1
iscellstr({'abc','123'}) % 1
iscellstr(['a','b','c']) % 0

%%
% isletter Истинно, если это символ алфавита
clear
clc
isletter('A') % 1
isletter('0') % 0
isletter('%') % 0

%%
% isspace Истинно, если это пробел
clear
clc
isspace(' ') % 1
isspace(' ') % 1 1 1 1
isspace('a b c') % 0 1 0 1 0

%%
% Операции над строками
% strcat Горизонтальное объединение строк
clear
clc
s1="abc" % попробовать одинарные кавычки 'abc'
s2="123" % попробовать одинарные кавычки '123'
s3=strcat(s1,s2)
s4=s1+s2
s5=char(s4)

%%
% strvcat Вертикальное объединение строк
clear
clc
s1="abc"
s2="123"
s3=strvcat(s1,s2)

%%
% strcmp Сравнить строки
clear

```

```

clc
strcmp("abc","abc") % 1
strcmp("abc","ab") % 0
strcmp("ab","abc") % 0

%%
% strncmp Сравнить n символов строк
clear
clc
strncmp("abc","abc",3) % 1
strncmp("abc","ab",2) % 1
strncmp("abc","ab",3) % 0

%%
% findstr Найти заданную строку в составе другой строки
% ищет вхождения в более длинном входном аргументе
% более короткого аргумента, возвращая начальные индексы вхождений
% в двойном массиве K. Если вхождений не найдено, findstr возвращает
% пустой массив, [].
% Рекомендуется использовать CONTAINS или STRFIND
clear
clc
s = 'How much wood would a woodchuck chuck?';
findstr(s,'a') % returns 21
findstr('a',s) % returns 21
findstr(s,'wood') % returns [10 23]
findstr(s,'Wood') % returns []
findstr(s,' ') % returns [4 9 14 20 22 32]

%%
% strjust Выравнивать массив символов
clear
clc
s="abc "
t1 = strjust(s)
t2 = strjust(s,"center")
t3 = strjust(t2,"left")

%%
% strmatch Найти все совпадения
clear
clc
strarray = ["max" "minimax" "maximum"];
x = strmatch("max",strarray,"exact") % 1
%
strarray = ["max" "minimax" "maximum"];
x = strmatch("max", strarray) % [1 3] - начинаются с "max"

% See Also
% matches | eq | strcmp | strcmpi | strncmp | strncmpi | strfind |
regexp | regexpi | regexprep | startsWith

```

```

%%
% strrep  Заменить одну строку другой
% Найти и заменить подстроку.
% MODIFIEDTEXT = strrep(ORIGTEXT,OLDSUBTEXT,NEWSUBTEXT) заменяет все
% вхождения текста OLDSUBTEXT в тексте ORIGTEXT на NEWSUBTEXT.
clear
clc
strrep("123 ab ab ** ab 789","ab","x")
% See also replace, erase, replaceBetween, contains, regexprep,
strfind

%%
% strtok  Найти часть строки, ограниченную разделителями
clear
clc
str = "A horse! A horse! My kingdom for a horse!"
token = strtok(str, '!') % '!' - разделитель. Вернет 'A horse'
%%
% upper  Перевести все символы строки в верхний регистр
clear
clc
upper('AbCd')
%%
% lower  Перевести все символы строки в нижний регистр
clear
clc
lower('AbCd')

%%
% Преобразования строк

% num2str  Преобразование числа в строку
clear
clc
num2str(3.14) % '3.14'

%%
% int2str  Преобразование целого в строку
clear
clc
int2str(97) % '97'

%%
% smat2str  Преобразование матрицы в строку
clear
clc
chr1 = mat2str([3.85 2.91; 7.74 8.99])
chr2 = mat2str([3.1416 2.7183],3) % Specify Precision

%%
% str2mat  Объединение строк в матрицу

```

```

% str2mat отличается от strvcat тем,
% что пустые массивы символов создают пустые строки на выходе.
% В strvcat пустые массивы символов игнорируются ???
clear
clc
x = str2mat('1', '2', ' ', '4')
%y = strvcat('1', '2', ' ', '4')

%%
% str2num Преобразование строки в арифметическое выражение и его
вычисление
clear
clc
x=str2num("2*5-3")

%%
% sprintf Записать форматированные данные в виде строки
clear
clc
eps
A = 1/eps;
str_e = sprintf('%0.5e',A)
%%
str_f = sprintf('%0.5f',A)
%%
str_g = sprintf('%0.5g',A)
%%
formatSpec = 'The array is %dx%d.';
A1 = 2;
A2 = 3;
str = sprintf(formatSpec,A1,A2)

%%
formatSpec = "The current time is: %d:%d %s";
A1 = 11;
A2 = 20;
A3 = 'a.m.';
str = sprintf(formatSpec,A1,A2,A3)
%%
str = sprintf('%d',round(pi))

%%
% The 0 flag in the %025d format specifier requests leading zeros in
the output.
str = sprintf('%025d',[123456])
%%
% sscanf Прочитать строку с учетом формата
clear
clc
chr = '2.7183 3.1416 0.0073'
A = sscanf(chr,'%f')

```

```

%%
str = "2.7183 3.1416 0.0073"
A = sscanf(str, '%f', [1 3])
%%
A = sscanf(str, '%f', [2 2])

%%
% Преобразования систем счисления
%% hex2dec Преобразовать шестнадцатеричное число в десятичное число
hex2dec('f')
hex2dec('A')
%% dec2hex Преобразовать десятичное число в шестнадцатеричное число
dec2hex(256)
%% bin2dec Преобразовать двоичную строку в десятичное число
bin2dec('111')
%% dec2bin Преобразовать десятичное число в двоичную строку
dec2bin(127)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Массивы ячеек. cell 31 слайд в 3 презентации

% Массив ячеек – это массив, элементами которого являются ячейки,
% содержащие массивы любого типа, в том числе и массивы ячеек.
clear
clc
C = {1 20 33 404}
%%
clear
clc
B = cell(2,3) % пустой массив ячеек 2x3
%%
clear
clc

% 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}
%%
clear
clc
% {}, cell – создание массива ячеек;
A={ [1,2] 'ABC' 3.1415}
%%
k = iscell(A)
%%

% 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
clc
clear
A={ [1, 2; 3 55 ] 'ABC' 3.1415}
cellplot(A)
c = A{1, 2}

%%
% Извлечение элемента с индексами (2,2)
% из числового массива ячейки A{1,1}
d1 = A{1, 1}(2, 2)
d2 = A{1, 2}(1, 3)
d3 = A{1, 3}(1)
%%
% Доступ к подмножеству ячеек
V=A(1,:)
%% Удаление строки
clc
clear
A={ [1 2; 3 4 ] ; 'QWERTY' }
figure (1)
cellplot(A)

% Удалять можно либо целую строку, либо столбец
A(1, :)=[] % удаление первой строки
figure (2)
cellplot(A)
%% удаление второго столбца
clc
clear
A={ [1, 2; 3 4 ] 'QWERTY' [111, 222] }
figure (1)
cellplot(A)

% Удалять можно либо целую строку, либо столбец
A(:, 2)=[] % удаление второго столбца
figure (2)
cellplot(A)

```

```

%% удаление с третьего столбца до конца
clc
clear
A={1 'QWERTY' [101, 202; 33 44; 11 12] [33, 55, 77] 'ABCD'}
figure (1)
cellplot(A)

```

```

% Удалять можно либо целую строку, либо столбец
A(:, 3:end)=[] % удаление с третьего столбца до конца
figure (2)
cellplot(A)

```

```

%% Вложенные массивы ячеек
clc
clear
A(1, 1) = {magic(2)};
A(1, 2) = {
    {
        [5 2 8; 7 3 0; 6 7 3]
        'Test 1';
        [2-4i 5+7i]
        {17 []}
    }
}
cellplot(A)

```

```

%% Многомерные массивы ячеек
A{1, 1} = 'QWERTY';
A{1, 2} = [4 2; 1 5];
A{2, 1} = 2-4i;
A{2, 2} = 7;

```

```

%
B{1, 1} = 'ABCD';
B{1, 2} = [ 3 5 ]';
B{2, 1} = 0:1:3;
B{2, 2} = 'Z';

```

```

%
C = cat(3, A, B);
C(2,2,:)
%
figure (1)
cellplot(C)
%
figure (2)
cellplot(C(:, :, 1))

```

```

%
figure (3)
cellplot(C(:, :, 2))

```



```

%% Функции работы с массивами ячеек. deal
% С помощью функции deal возможно множественное присваивание
% входных данных выходным
clc
clear
% [A,B,C,...]=deal(X,Y,Z,...) % конструктор
% [A,B,C,...]=deal(X)      % конструктор
X={1, 'Hello!', [7,13]} % пример
[A,B,C]=deal(X{:})      % пример

%%
% Функции работы с массивами ячеек

%% {}, cell – создание массив ячеек;
clc
clear
A=cell(2,3)
B={1, [10 20], "ABCD"}

%% cellstr – создание массива ячеек строк из символьного массива;
clc
clear
A = ["Past", "Present", "Future"]
C = cellstr(A)
%% cellfun – применение функции к каждому элементу в массиве ячеек;
clc
clear
C = {1:10, [2; 4; 6], []}
A = cellfun(@mean,C)

%% cellfun
clc
clear
X = {5:5:100, 10:10:100, 20:20:100};
Y = {rand(1,20), rand(1,10), rand(1,5)};
figure
hold on
p = cellfun(@plot,X,Y);
p(1).Marker = 'o';
p(2).Marker = '+';
p(3).Marker = 's';
hold off

%% celldisp – показ содержимого массива ячеек;
clc
clear
C = {'ABC', [1 2 3], 3+4i;
     'X*Y', [2 4; 1 3], {'qwerty', 1024}}
celldisp(C)
%% cellplot – показ графической структуры массива ячеек;
cellplot(C)

```

```

%% deal - обмен данными;
clc
clear
X={1, 'Hello!', [7,13]} % пример
[A,B,C]=deal (X{:}) % пример

%% num2cell - преобразование числового массива в массив ячеек;
clc
clear
a = magic(3)
c = num2cell(a)

%% cell2mat - преобразование массива ячеек в отдельную матрицу;
clc
clear
C = {[1], [2 3 4];
     [5; 9], [6 7 8; 10 11 12]}
cellplot(C)
A = cell2mat(C)

%% mat2cell - разбиение матрицы на массив ячеек матриц;
% Внимательно с размерностями!!!
clc
clear
A = reshape(1:20,5,4)'
C = mat2cell(A,[2 2],[3 2])
cellplot(C)
celldisp(C)

%% cell2struct - преобразование массива ячеек в структуру;
% help cell2struct
%% struct2cell - преобразование структуры в массив ячеек;
% help struct2cell
%% iscell - определяет, является ли введенная переменная массивом
ячеек.
clc
clear
A = {1,2,3;
     'text',rand(5,10,2),{11; 22; 33}}
iscell(A)
B = rand(2,5)
iscell(B)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Импортирование объектов. File→ImportData
clc
clear
% Правильно указать путь к файлу!
% Файл должен существовать!
t=readtable('d:\matr.csv')
t(2:3,1:2)

```

```

%%
clc
clear

% xlsread - чтение файла Excel
A=xlsread('d:\file01.xlsx')
sum(sum(A))
% Правильно указать путь к файлу!
% Файл должен существовать!
B=xlsread('d:\file01.xlsx','list2')
C=xlsread('d:\file01.xlsx','list3','B2:C3')
sum(sum(C))
%%

%% Импортирование объектов. Текстовые файлы
clc
clear
% Правильно указать путь к файлу!
% Файл должен существовать!
load 'd:\test.txt'
sum(test)
%% Импортирование объектов. fopen. Чтение из файла
clc
clear
% Правильно указать путь к файлу!
% Файл должен существовать!
f = fopen('d:\test.txt', 'r') % > 0 - хорошо
a = fscanf(f, '%d', [1, 9] )
%% 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 заранее
%%
clc
clear
f = fopen('d:\test2.txt', 'r')
a = fscanf(f, '%f %d', [2, inf] )

```

```

%% Запись в файл. fprintf. Форматированный вывод
clc
clear
b=[1 2]
c=[3 4]
f = fopen('d:\test4.txt', 'w')
fprintf(f, '%d %d\n', b);
fprintf(f, '%d %d', c);
%% Дозапись в файл
g=[5 6]
f = fopen('d:\test4.txt', 'a')
fprintf(f, '%d %d\n', g);

%%
% Запись в файл. fprintf. Форматированный вывод
f = fopen('d:\test5.txt', 'w')
for r = 1:10
    fprintf(f, '%d %f\n', round(rand()*5),rand()*10)
end

%% Дозапись в файл
f = fopen('d:\test5.txt', 'a')
for r = 1:4
    fprintf(f, '%d %f\n', round(rand()*5),rand()*100)
end
%%

```