

Преобразование ячеечного массива в матрицу выполняется функцией **cell2mat(C)**. Аргументом данной функции является ячеечный массив. При этом элементами ячеечного массива могут являться численные, логические матрицы, или массив символов (в том числе строки). При этом происходит конкатенация объектов (функция **cat ()**). Если конкатенация невозможна, то невозможно преобразование.

Примеры:

```
c = {1,2,3,4;5,6,7,8}
cell2mat(c)
c = {[-0.6;7], [-2;6], [0;3.4]}
cell2mat(c)
c = {[-0.6;7], logical([-2;0]), [0;3.4]}
cell2mat(c)
cell2mat(c')
```

Обратное преобразование можно осуществить функцией **num2cell(M)**. При этом каждый элемент матрицы заносится в отдельную ячейку.

```
num2cell([-0.23, 0; 20 5-2/7i])
s='abcde'
s(3,:)=['w' 'v' 'x' 'y' 'z']
s(2,[1 3 5])='<=>'
d = num2cell(s)
cell2mat(d([1 3],:))
```

Более сложное преобразование матрицы в ячеечный массив осуществляется функцией **mat2cell(M, dim1, dim2, ..., dimi, ..., dimn)**. Первым аргументом является матрица. Указанные далее аргументы показывают, как разбивать исходную матрицу на подматрицы, которые заносятся в ячейки. Каждый из аргументов (*dim1, dim2, ...*) представляет собой целочисленный вектор, указывающий, на какие блоки по соответствующей размерности разбивается матрица. При этом обязательным условием является требование, чтобы сумма элементов соответствующего вектора была равна соответствующей размерности ($\text{sum}(\text{dimi}) == \text{size}(M, i)$).

Для двумерных матриц можно указывать только один аргумент размерности (**mat2cell(M, dim1)**). В таком случае матрица разбивается на соответствующие блоки по строкам.

Примеры:

```
m = 1:16; m = reshape(m, [4 4])
mat2cell(m, [1, 1, 2], [3, 1])
mat2cell(m, [2,2])
c = mat2cell(s,[1 1 1])
cell2mat(c')
```

Функция **cellstr(S)** преобразует массив строк в ячеечный массив, в каждую ячейку помещается строка.

Большинство функций преобразования на М имеют вид `typeone2typetwo`.

Перевод числа в число по другому основанию:

base2dec (*s*, *base*)

Преобразование строки, состоящей из цифр по основанию *base* в десятичное число.

bin2dec (*s*)

Преобразует строку *s*, являющуюся записью двоичного числа в десятичное.

hex2dec (*s*)

Преобразует строку *s*, являющуюся записью шестнадцатеричного числа в десятичное.

dec2base (*d*, *base*)

Преобразование десятичного числа в символьный массив по основанию *base*.

dec2bin (*d*, *len*)

Преобразование десятичного числа в двоичный символьный массив.

dec2hex (*d*, *len*)

Преобразование десятичного числа в шестнадцатеричный символьный массив.

hex2num (*s*)

Преобразование строки, представляющей число в шестнадцатеричном формате IEEE в десятичное число.

num2hex (*n*)

Преобразование десятичного числа к шестнадцатеричному формату IEEE.

Преобразование численных и строковых типов:

char (*x*)

Преобразование массива, содержащего ASCII коды в символьный.

int2str (*n*)

Преобразование действительного числа в строку.

num2str (*x*)

num2str (*x*, *precision*)

num2str (*x*, *format*)

Преобразование числа в строку с возможностью задания точности и формата.

mat2str (*x*, *n*)

Преобразование действительных, комплексных и логических массивов в символьные с заданной точностью.

x = **str2num** (*s*)

[*x*, *state*] = **str2num** (*s*)

Преобразование строки *s* в число или массив. Дополнительный аргумент является 1, если преобразование успешно, или 0 в противном случае.

str2double (*s*)

Преобразование строки в действительное или комплексное число.

Преобразование ячеек массивов и структур данных:

c = **struct2cell** (*s*)

Создание ячеек массива из полей структуры (объекта).

cell2struct (*cell*, *fields*)

Преобразование ячеек массива в структуру. Вторым аргументом указываются имена полей структуры.