

# String Manipulations with Arrays

back to [Fan's Reusable Matlab Repository](#) or [Dynamic Asset Repository](#).

## Duplicate String

```
it_duplicate_n = 10;  
disp(repmat({'String'}, [1, it_duplicate_n]));
```

'String' 'String' 'String' 'String' 'String' 'String' 'String' 'String' 'String' 'String' 'Str

## Combine Two Numeric Arrays to a Single String Array

```
ar_fl_abc1 = [0.4 0.1 0.25 0.3 0.4];  
disp([num2str(ar_fl_abc1', 'zw=%3.2f; '), num2str(ar_fl_abc1', 'zr=%3.2f')]);
```

```
zw=0.40;zr=0.40  
zw=0.10;zr=0.10  
zw=0.25;zr=0.25  
zw=0.30;zr=0.30  
zw=0.40;zr=0.40
```

## Convert Numeric Array to String, Apeend Prefix to all elements.

```
ar_fl_abc1 = [0.4 0.1 0.25 0.3 0.4];  
ar_st_wth_prefix = strcat('row=', string(ar_fl_abc1));  
disp(ar_st_wth_prefix);
```

"row=0.4" "row=0.1" "row=0.25" "row=0.3" "row=0.4"

```
% Does Array Exist in Longer Array as Subset  
ar_abc1 = [0.4 0.1 0.25 0.3 0.4];  
ar_abc2 = [0.4 0.1 0.2 0.3 0.4];  
ar_efg = [0.1 0.2 0.3 0.4 0.1 0.2 0.3 0.4 0.1 0.2 0.3 0.4 0.1 0.2 0.3 0.4];  
st_abc1 = strjoin(string(num2str(ar_abc1)));  
st_abc2 = strjoin(string(num2str(ar_abc2)));  
st_efg = strjoin(string(num2str(ar_efg)));  
contains(st_efg, st_abc1)
```

```
ans = logical  
0
```

```
contains(st_efg, st_abc2)
```

```
ans = logical  
1
```

```
% Display Convert to String  
fprintf('Display string [%s]', num2str([1,2,3]));
```

```
Display string [1 2 3]
```

```
fprintf('Display string [%s]', num2str(1.1));
```

```
Display string [1.1]
```

```
fprintf('Display string [%s]', 'abc');
```

```
Display string [abc]
```

## String Join to form Single Element

using char() is safe

```
st_var_name = "abc"
```

```
st_var_name =  
"abc"
```

```
st_var_name = [st_var_name ' percentile values']
```

```
st_var_name = 1x2 string array  
"abc"      " percentile values"
```

```
strjoin(st_var_name)
```

```
ans =  
"abc percentile values"
```

```
st_var_name = "abc"
```

```
st_var_name =  
"abc"
```

```
st_var_name = [char(st_var_name) ' percentile values']
```

```
st_var_name =  
'abc percentile values'
```

```
st_var_name = 'abc'
```

```
st_var_name =  
'abc'
```

```
st_var_name = [char(st_var_name) ' percentile values']
```

```
st_var_name =  
'abc percentile values'
```

## String Join dash

using char() is safe

```
st_var_name = "abc"
```

```
st_var_name =  
"abc"
```

```
st_var_name = [st_var_name 'efg']
```

```
st_var_name = 1×2 string array  
"abc"      "efg"
```

```
strjoin(st_var_name, "_")
```

```
ans =  
"abc_efg"
```

## Numeric Array to String without Space

String replace

```
ar_it_test_grp = [3, 8, 9];  
strrep(num2str(ar_it_test_grp), ' ', '_')
```

```
ans =  
'3_8_9'
```

## Substring replace in Cell Array

```
ar_st_cells = {'shock=0.35', 'shock=0.40', 'shock=0.46'};  
ar_st_updated_cells = strrep(ar_st_cells, 'shock', '$\epsilon$');
```

## Find position of String in String Cell

```
ls_st_param_key = {'fl_crra', 'fl_beta', ...  
                  'fl_w', 'fl_r_save', ...  
                  'fl_a_max', 'it_z_n', 'it_a_n'};  
st_param_key = 'fl_a_max';  
find(strcmp(ls_st_param_key, st_param_key))
```

```
ans = 5
```

## Find the positions of String Cells in Full String Cells

```
ls_st_param_key = {'fl_crra', 'fl_beta', ...  
                  'fl_w', 'fl_r_save', ...  
                  'fl_a_max', 'it_z_n', 'it_a_n'};  
cl_st_param_keys = {'fl_w', 'fl_beta', 'it_z_n'};  
cell2mat(cellfun(@(m) find(strcmp(ls_st_param_key, m)), ...  
                cl_st_param_keys, 'UniformOutput', false))
```

```
ans = 1×3  
      3      2      6
```

```
find(strcmp(ls_st_param_key, st_param_key))
```

```
ans = 5
```

## Cell to string Paste and Replace dash

```
cl_st_param_keys = {'fl_crra', 'fl_beta'};  
display(strrep(strjoin(cl_st_param_keys, '-'), '_', '\_'));
```

fl\\_crra-fl\\_beta