Basic String Operations, Display, Search, Join and Split

back to Fan's Intro Math for Econ, Matlab Examples, or MEconTools Repositories

Date String

Generate a string based on date and time as file suffix.

```
st_file_suffix = ['_d' datestr(now,'yymmdd_tHHMMSS')];
disp(st_file_suffix);
```

```
_d210701_t094001
```

Combine String, Numeric values etc, Single and Double Quotes

Convert a string array into a single string, note the double quotes, and the auto space between:

```
st_a = "another string";
ar_st = ["abc", num2str(2), "opq", st_a];
disp(strjoin(ar_st));
```

```
abc 2 opq another string
```

If we do not want to have spaces between words, the second parameter for strjoin allows for string connectors:

```
st_a = "another string";
ar_st = ["abc", num2str(2), "opq", st_a];
disp(strjoin(ar_st, ""));
```

```
abc2opqanother string
```

With single quotes, the str element is not an array, so does not need strjoin, but not need to have spaces:

```
st_a = 'another string';
str = ['abc ', num2str(2), ' opq ', st_a];
disp((str));
```

```
abc 2 opq another string
```

Construct String Print Statement with String and Numeric Elements

In the example below, we have a number of strings we want to put inside a string array, then join with strjoin, but two of the strings need to be constructed as strings first. Note below that double quoates are own strings, single quotes in brackets constructing additional strings.

```
st_a = "another string";
ar_st = ["abc", "efg"];
ar_fl_vals = rand([1,3]);
st_print = strjoin(...
    ["Completed SNW_DS_MAIN:", ...
    ['SNW_MP_PARAM=' num2str(123)], ...
    ['SNW_MP_CONTROL=' num2str(678)], ...
    ['STR_VAR1=' char('string')], ...
```

```
    ['STR_VAR2=' char(strjoin(ar_st, "-"))], ...
    ['STR_VAR3:;' char(strjoin(ar_st, ";"))], ...
    ['ar_st:;' char(strjoin(strcat( ...
    num2str((1:length(ar_st))', '%0.3d'), '=', ar_st' ...
    ), ";"))], ...
    ['ar_fl_vals:;' strjoin(string(strcat( ...
    num2str((1:length(ar_fl_vals))', '%0.3d'), '=', num2str(ar_fl_vals', '%3.2f'))), ";")], ...
    ], ";");
% Ways to print
st_out = st_print;
ar_ch_out = char(strsplit(st_print,";")');
ar_st_out = strsplit(st_print,";")';
% Print
disp(st_out);
```

```
Completed SNW_DS_MAIN:;SNW_MP_PARAM=123;SNW_MP_CONTROL=678;STR_VAR1=string;STR_VAR2=abc-efg;STR_VAR3:;abc;efg;ar_st:
```

```
disp(ar_ch_out);
```

```
Completed SNW_DS_MAIN:
SNW_MP_PARAM=123
SNW_MP_CONTROL=678
STR_VAR1=string
STR_VAR2=abc-efg
STR_VAR3:
abc
efg
ar_st:
001=abc
002=efg
ar_fl_vals:
001=0.87
002=0.25
003=0.48
```

```
disp(ar_st_out);
```

```
    "Completed SNW_DS_MAIN:"
    "SNW_MP_PARAM=123"
    "SNW_MP_CONTROL=678"
    "STR_VAR1=string"
    "STR_VAR2=abc-efg"
    "STR_VAR3:"
    "abc"
    "efg"
    "ar_st:"
    "001=abc"
    "002=efg"
    "ar_fl_vals:"
    "001=0.87"
    "002=0.25"
    "003=0.48"
```

Paste Join Strings Together with Separator

Join strings together with separator, this is similar to the paste0 function in R.

```
ar_st = ["abc", "efg", "opq"];
disp(strjoin(ar_st, '-'));
```

```
abc-efg-opq
```

Combine Char with Numeric Value

Compose a string with words and numerical values

```
st_title = strcat("Figure Title ", ...
    "(", ...
    "threedeci=%.3f,", ...
    "twodeci=%.2f,", ...
    "int=%.0f", ...
    ")");
ar_params = 123.4567 + zeros(1,3);
st_combo = compose(st_title, ar_params);
disp(st_combo);
```

```
Figure Title (threedeci=123.457,twodeci=123.46,int=123)
```

Search if String Contains Substring

Does string contain substring?

```
st_long1 = 'simu_dense';
st_long2 = 'simu_denser';
st_long3 = 'simuverydense';
st_long4 = 'simu_medium';
st_long5 = 'simuverysmall';
disp([contains(st_long1, 'dense'), contains(st_long2, 'dense'), contains(st_long3, 'dense'), ..
    contains(st_long4, 'dense'), contains(st_long5, 'dense')]);
```

```
   1   1   1   0   0
```

Find Elements of an Array that Matches Any Elements of Another Array

There is an array of strings, check if each element equals values specified in another string, using the match function.

```
% String array from strings
ar_st_long = string({st_long1, st_long2, st_long3, st_long4, st_long5});
% If matches simu_dense
disp(matches(ar_st_long, 'simu_dense'));
```

```
   1   0   0   0   0
```

```
% If matches simu_dense or simu_denser
disp(matches(ar_st_long, ["simu_dense", "simu_denser"]));
```

```
   1   1   0   0   0
```

Change File Name MLX to M

```
st_file_name_mlx = 'continuous_differentiable.mlx';
at_st_split_file_name = split(st_file_name_mlx, ".");
```

```
st_file_name_m = strcat(at_st_split_file_name{1}, '_m.m');
disp(st_file_name_m);
```

```
continuous_differentiable_m.m
```