Text Manipulation

Topics:

- Characters, Character vectors, and string arrays
- Operations on text
- "is" function for text
- Conversion between text and numbers

Book Sections: 7.1 - 7.4

1. Characters, Character Vectors, and String Arrays

Files often contain characters and numbers. When reading a file, it is often necessary to parse these elements line by line.

Definition: Individual characters, **char**, are stored using single quotation marks. Characters include: alphabet, digits, punctuation marks, white space (e.g., space, tab), and control characters (e.g., tab).

1.1 Character vectors

A character vector consists of any number of characters (possibly none) enclosed by single quotes. Effectively, each character is indexed in an array. For example,

```
% 1 x 5 character array (vector)
charArray='Hello';

% Here are more character arrays
arr1=''
arr2='x'
arr3='123'

% Check the size of these arrays
size(charArray);

% Try to transpose
charArray'
```

1.2 String array

Introducted in 2016b, **string** type can store word(s). A string is enclosed with double quoation marks. For example,

```
% ALWAYS
help string

% A string
x="Hello";

% Note the size is 1 x 1
size(x);

% To find the length, use strlength() function.
strlength(x);

% You can extract individual characters by using curly brackets followed by
% parenthesis.

x{1}(3);    % returns 'l'
x{1}(1);    % returns 'H'

% Create a vector of strings
cities = ["Boston", "New York", "Chicago"]

cities(1)   % returns string
cities{2}   % returns character array
```

2. Operations on Text

```
% Concatenate strings:  +
"Hello"+"World"

% Create blanks: blanks function
blanks(5);  % 5 blank spaces

% Create string from character array
string('hello')
```

```
%
```

2.1 Creating and Concatenating

```
% Horizontal concatenate
strcat('Red','Yellow')

% Append string arrays, +
"Red "+ "Yellow"
```

2.2 Removing Characters

Functions that will remove trailing and/or leadning spaces (blanks) from strings and character arrays: deblank, strtrim, strip, erase.

```
% Remove trailing spaces (not leading blanks)
deblank("   Hello   ")

% Remove both leading and trailing blanks (not middle)
strtrim("    Hello  World    ")

% Remove unwanted characters
strip("xxxHello World!x","x");    % 2nd argument is what to remove


% Remove all characters from string
erase("xxxabcxdefgxkjkjdxkjkjsk", "x");  % removes x from string
```

2.3 Change case

Make string all upper or lower case

```
x="Hello";

upper(x);   % all upper case
lower(x);   % all lower case
```

2.4 Comparing Strings

MATLAB has several functions to compare strings and character arrays. These functions return a logical value.

```
% Equality
strcmp('cat', 'car')

% Compare first n characters
strncmp('cat', 'car', 2);
```

2.5 IS Function

```
% IS functions are common in nearly all programming languages.
% They return a boolean value based on the results of a test, typically
% the 'one' appended to the IS function.

% For example: isnumeric, islogical, isjava

% In particular, ischar, isstring.

x="Hello";
y=1;

isstring(x) % Is x a string type?  Yes
isstring(y) % Is y a string type?  No
```

2.6 Converting between types (string to numeric)

```
x="3";

% String to Double
y=str2double("3")

% Double to String
z=num2str(y)

% compare
```

```
strcmp(x,z)
```

2.7 Finding Substrings

```
% findstr and strfind
findstr('abcde','d')

findstr('cd','abcde')

findstr('ac','abcde')

% Note: findstr always searches for shorter string in longer string
```