

## Characters, Strings and Text (From David Griffiths)

The ability to process text in numerical processing is useful for the input and output of data to the screen or to disk. In order to manage text, a new datatype of "character" is introduced. A piece of text is then simply a string (vector) or array of characters.

Help `strfun` will give you useful operations for manipulating strings.

The assignment,

```
>> t1 = 'A'
```

assigns the value A to the 1-by-1 character array t1.

The assignment,

```
>> t2 = 'BCDE'
```

assigns the value BCDE to the 1-by-4 character array t2.

Strings can be combined by using the operations for array manipulations.

The assignment,

```
>> t3 = [t1,t2]
```

assigns a value ABCDE to the 1-by-5 character array t3.

The assignment,

```
>> t4 = [t3,' are the first 5 characters in the alphabet.']
```

assigns the value

'ABCDE are the first 5 characters in the alphabet.' to the character array t4.

Sometimes it is necessary to convert a character to the corresponding number, or vice versa. These conversions are accomplished by the commands 'str2num' which converts a string to the corresponding number, and two functions, 'int2str' and 'num2str', which convert, respectively, an integer and a real number to the corresponding character string. These commands are useful for producing titles and strings, such as 'The value of pi is 3.1416'.

This can be generated by the command ['The value of pi is ', num2str(pi)].

```
>> N = 5; h = 1/N;
```

```
>> ['The value of N is ', int2str(N), ', h = ', num2str(h)]
```

```
ans = The value of N is 5, h = 0.2
```

# File I/O Overview (from matlab documentation)

The MATLAB product includes a set of low-level file I/O functions that are based on the I/O functions of the American National Standards Institute's ANSI® Standard C Library. If you know C, you are probably familiar with these routines.

To read or write data, perform these steps:

1. Open the file, using `fopen`. `fopen` returns a file identifier that you use with all the other low-level file I/O routines.
2. Operate on the file.
  1. Read binary data, using `fread`.
  2. Write binary data, using `fwrite`.
  3. Read text strings from a file line-by-line, using `fgets` or `fgetl`.
  4. Read formatted ASCII data, using `fscanf`.
  5. Write formatted ASCII data, using `fprintf`.
3. Close the file, using `fclose`.

**Exercise:** Download this file: [http://tamaraberg.com/teaching/Spring\\_09/Demos/text.txt](http://tamaraberg.com/teaching/Spring_09/Demos/text.txt)

How many times does the word 'text' appear in the file? How many times do the characters 'text' appear in the file (this could include other words that contain text such as texting context etc) Some useful functions that might be helpful are `strtok`, & `strcmp`, `strfind`. Type `help [command]` to learn more about using them.