

Files:

By the end of this class you should be able to:

- Prepare for EXAM 1.
- create an ASCII file
- describe the nature of an ASCII text
- Use and describe string variables
- import ASCII files into MATLAB
- save and restore MATLAB workspace variables

Sections 3.4, 3.5

Exam 1 - Questions?

Areas:

- MATLAB Basics - interface, calculations
- Polynomials
- Complex Numbers
- Vectors
- Script and Function Programming
- Problem Solving
- Matrices

Working with Text Variables

- Go to website www.engr.ipfw.edu/~moor/121
- Download: eng2PL.m to your working directory
- Run it with a text input
E.G., `>> eng2PL('Sally')`
- What does the program do?
- Look at the code
- How does it do this?

eng2pl.m

<code>function x = eng2pl(x)</code>	Text is passed in and out of the function in the variable x
<code>% Lots 'o' Comments</code>	Notice use of quotes on "ay" "ay" is data not a variable name
<code>x = [x(2:end), x(1),'ay'];</code>	Treats text as a vector and rearranges letters
<code>x=lower(x);</code>	Sets all letters to lower case
<code>x(1) = upper(x(1));</code>	Capitalizes first letter

Using a Variable for Text Data

You can assign text to a variable by

- passing text through a function command line
- using the assignment operator
- using an input statement
- using num2str() function to convert a number to a string variable.

A look at ASCII coding in MATLAB

Type the following, notice what you get:

```
>> a = 1
>> A = '1'
>> a,A
>> a*3, A*3
```

Notice that the two are not equivalent. The variable A, with a literal - results in the ascii value of 1 instead of 1.

You can convert between ascii (text) and numbers in MATLAB

```
>> double(A) → converts '1' to its ascii code (49)
>> char(49) → converts 49 to its ascii character.
>> num2str(a) → converts 1 to an ascii string '1'
>> str2num(A) → converts ascii '1' to numerical 1
```

American Standard Code for Information Interchange (ASCII)

Example Coding		
Character	ASCII Code	
	Decimal	Binary
A	65	01000001
a	97	01100001
:	58	00111010

More example ASCII code values

48	0	65	A	97	a
49	1	66	B	98	b
50	2	67	C	99	c
51	3	68	D	100	d
52	4	69	E	101	e
53	5	70	F	102	f
54	6	71	G	103	g
55	7	72	H	104	h
56	8	73	I	105	i
57	9	74	J	106	j
58	:	75	K	107	k
59	;	76	L	108	l
60	<	77	M	109	m

Creating an ASCII file

Text editor: a program that produces pure text (.txt) files

Possibilities: Notepad, WordPad, Matlab editor, Word (saving as a "plain text" file.)

Delimiters - character used to separate numbers on a line
Common delimiters: space, tab, comma, and semicolon
MATLAB will attempt to detect the above delimiters

Exercise: create an ASCII file

- open program (word, wordpad, notepad)
- type several lines of numbers with numbers on a line separated by spaces or tabs
- there must be the same number of numbers in each line
- file → save as: filename: test.txt
file type: plain text (ANSI usually)
- Read this file into MATLAB
 - >> load 'test.txt'
 - Look in workspace notice the array test
- Can also use a function form of this command
 - >> A = load('test.txt')
 - Look in workspace notice the array A

E.G., Drying Experiment Data Files

- Go to: www.engr.ipfw.edu/~moor
 - Click on ENGR 121: Drying Experiment Data
 - Click on ASCII text file
- notice the plain text is readable by your browser
- Click on back button
 - Download the text file to your MATLAB work area
 - Right click
 - Select "Save Link Target As ..."

Importing and Saving ASCII files

- Import Wizard: File → Import Data
 - Select the drying1.txt file you downloaded
 - You will need to change the number of "header lines" (to 5)
 - Notice you get three tabs on the right - look at each of these
 - Click Next button
 - This takes you to a screen with three files on the right
 - Click on them to see their contents. Right click and select rename to change the name of the variable used.
 - Click finish to transfer to the MATLAB workspace.
- to see what we have:
 - look at workspace or
 - use the who and size commands

Try plotting the data

- remember the basic plot command for plotting an x vs a y vector:
`>> plot(x,y, '.')` → will plot with a dot for every x, y pair
- what is being plotted: t vs m
- where is the x data → `data(:,1)`
- where is the y data → `data(:,2)`
- How would you Plot those two?
`>> plot(data(:,1),data(:,2), '.')`
- where is are the axis titles → `columnheaders(1) ...`
- how can you add these titles to the plot?
`>> xlabel columnheaders(1)`
`>> ylabel columnheaders(2)`

Creating a function to read in a file

```
function datagraph(fname)
% lots o' comments

% read in data file
data = load (fname);

% plot data
plot (data (:,1), data(:,2) , '.' )
```

`>> datagraph('drying2.txt')`

Other ways to read in an ASCII files

- Command Line: The load command (as before)
 - very important because it can be done in .m files
 - file must be a simple array (no header rows)
- as a command: `>> load filename`
- as a function: `>> A = load('filename')`
- Notice difference between a command and a function. Only a function output can be saved to another variable. The command will automatically save to a variable with the same name as the file.
- exercise: read `drying1.txt` into a text editor, remove the header row and save as an ASCII file `drying2.txt`. Replace `filename` in the commands above with `'drying2.txt'` to read the file in.