Topic 8

String Manipulation

Introduction

- Strings are any characters contained in single quotes.

```
''
'c'
'bot'
'Hello world'
'791'
```

- Every element is a single character.
- Many vector operations and functions also work with strings.
- MATLAB also has many built-in functions specifically to manipulate strings.

Creating Variables

String can be created by:

- assigning to a variable
- using input function

```
>> unit='kNm';
>> name=input('Type your name :','s')
Type your name :Ali
name =
Ali
```

```
>> age=input('How old are you : ')
How old are you : '24 yrs'
age =
24 yrs
```

Leading and trailing blanks are will be stored in in the string

String as Vectors

- String are treated as vectors of characters.
- Many vector operations and functions also work with strings.

```
% …continue …
>> length(unit)
ans =
      3

>> unit(2)
ans =
N

>> unit'
ans =
k
N
m
```

String as Vectors - cont'd

- A character matrix can be created that consists of strings in every row.

```
>> strmat=['hello';'Ahmad']
strmat =
hello
Ahmad

>> size(strmat)
ans =
     2     5

>> strmat(2,3)
ans =
M

>> strmat(:,4)
ans =
l
a
```

Rows within a matrix must have the same length. Shorter strings must be padded with blanks so that the strings have the same length.

String as Vectors - cont'd

```
>> strmat=['hello';'Ali']
Error using vertcat
Dimensions of matrices being concatenated
are not consistent.
```

2 spaces

```
>> strmat=['hello';'Ali  ']
strmat =
hello
Ali

>> size(strmat)
ans =
     2     5
```

Rows within a matrix must have the same length. Shorter strings must be padded with blanks so that the strings have the same length.

String Concatenation

To create a long string from two short strings

Use either **[ ]** or **`strcat`** function

***a) Using* [ ]**

```
>> a='long';
>> b='house';
>> [a b]
ans =
longhouse
```

```
>> s1='xxx   ';
>> s2='   kkk';
>> [s1 s2]
ans =
xxx      kkk
>> length(ans)
ans =
    12
```

Leading and trailing spaces included

String Concatenation – cont'd

b) Using `strcat` function

```
>> s1='xxx   ';
>> s2='   kkk';
>> s3=strcat(s1,s2)
s3 =
xxx   kkk
>> length(s3)
ans =
     9
```

```
>> s4=strcat(s2,s1)
s4 =
   kkkxxx
>> length(s4)
ans =
     9
```

`Strcat` *removes trailing spaces but not leading spaces*

String – `char` function

`char` function can also be used to **concatenate vertically** (ie. create matrix of characters).

```
>> strmat=char('Hello','Ali')
strmat =
Hello
Ali

>> size(strmat)
ans =
     2     5
```

`char` *function will automatically pad the strings in the rows with trailing blanks as necessary so that they are all the same length*

String – `blanks` function

`blanks` function creates a string containing **n** blank characters.

```
>> a='long';
>> b='house';
>> [a blanks(5) b]
ans =
long     house
>>
>> disp(blanks(4)')




>>
```

Transposing blanks will insert 4 blank lines. Equivalent to printing newline character 4 times .

String – sprintf function

sprints function works exactly like the **fprintf** function, but instead of printing it creates a **string**.

```
>> txt1=sprintf('Grav. Acc. is %.2f',9.806)
txt1 =
Grav. Acc. is 9.81

>> length(txt1)
ans =
    18

>> txt2=sprintf('The lucky number is %i', ...
   randi([1,5]))
txt2 =
The lucky number is 5
```

All formating used in fprintf can also be used in sprintf.

String – sprintf function

sprints function works exactly like the **fprintf** function, but instead of printing it creates a **string**.

```
% Loads a file that stores experiment no. and data
% then plots the data with the expr. no in the title
% sprintf

load data.dat
expermtNo=data(1);
speed=data(2:end);
plot(speed,'-kd')
xlabel('time'),ylabel('Speed')
title(sprintf('Experiment No. %d',expermtNo))
```

Filename : data.dat

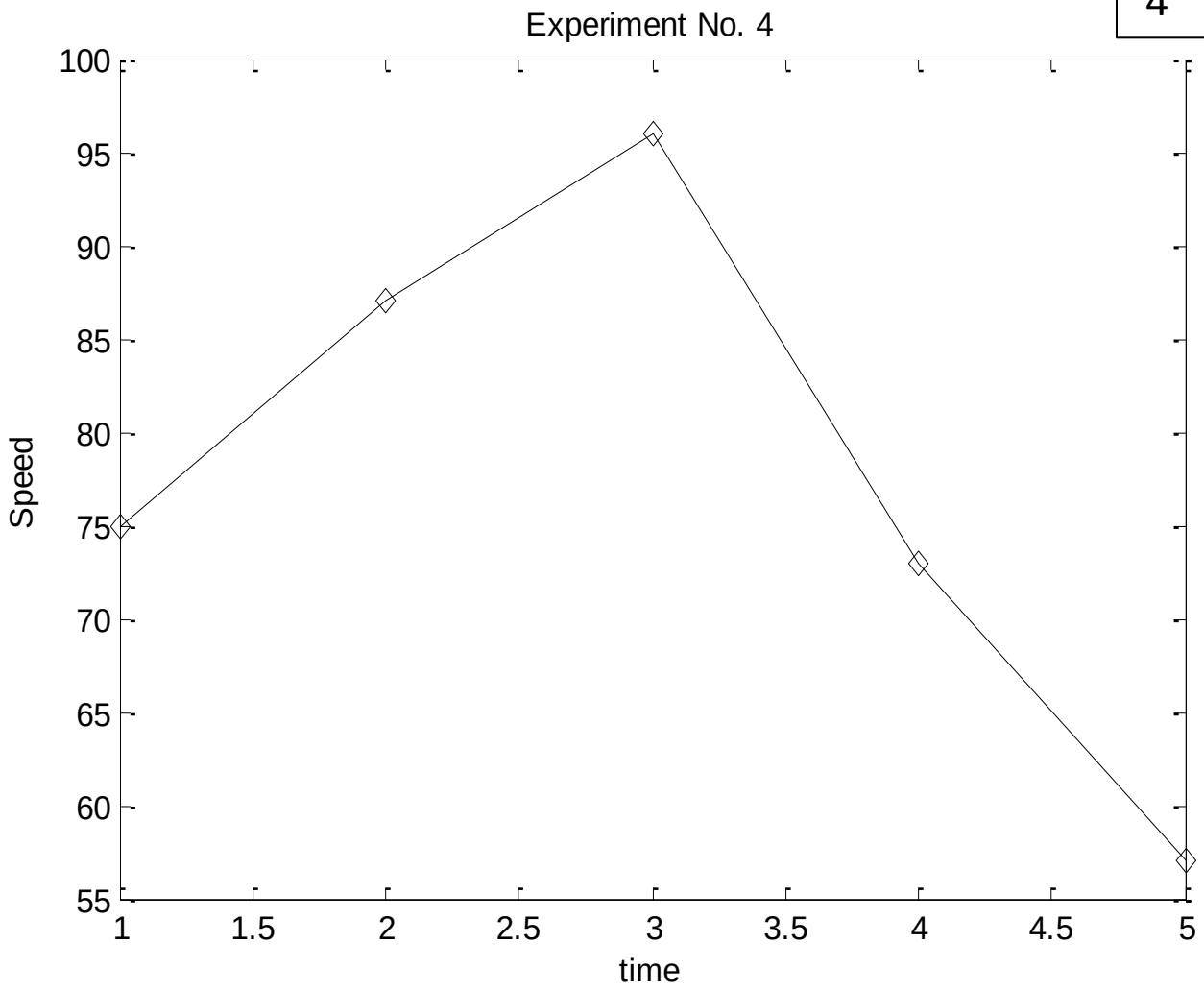
```
4  75  87  96  73  57
```

String – `sprintf` function (cont'd)

Script file executed

Filename : data.dat

```
4  75  87  96  73  57
```



String – sprintf & fprint in input function

```
% To demonstrate the use of sprintf in input function

username=input('Enter your name : ','s');
txtprompt=sprintf('%s, Enter your ID # : ',username);
ID_No=input(txtprompt)

%another way is by using fprintf
fprintf('%s, Enter your id no : ',username);
id_no=input('')
```

```
>> input_sprintf
Enter your name : Ali
Ali, Enter your ID # : 123
ID_No =
   123
Ali, Enter your id no : 23
id_no =
    23
```

String – deblank function

deblank function **removes trailing blanks** from the end of a string.

```
>>place=char('Cukai','Kemaman','Trengganu')
place =
Cukai
Kemaman
Trengganu
>> town=place(1,:)
town =
Cukai
>> length(town)
ans =
     9
```

```
>> town=deblank(town)
town =
Cukai
>> length(town)
ans =
     5
```

String – `strtrim` function

`strtrim` function **removes both leading and trailing** blanks from a string, but **not in the middle of the string**.

```
>> txtstr=[blanks(3) 'mm' blanks(2) 'yy' blanks(4)]
txtstr =
   mm  yy
>> s1=length(txtstr)
s1 =
    13
>> s1trim=strtrim(txtstr)
s1trim =
mm  yy
>> length(s1trim)
ans =
     6
```

String – lower & upper function

lower function converts strings all to lowercase.

upper function converts strings all to uppercase.

```
>> faculty='CiVil Eng';
>> f1=lower(faculty)
f1 =
civil eng
>> f2=upper(f1)
f2 =
CIVIL ENG
>> f3=upper(faculty(1:5))
f3 =
CIVIL
>> sprintf([f3 ' UTM'])
ans =
CIVIL UTM
```

String – strcomp, strncmp & strcmpi function

```
>> word1='foot';
>> word2='feet';
>> word3='football';
>> word4='FeeT';
>> strcmp(word1,word3)
ans =
     0
>> strcmp(word1,'foot')
ans =
     1
>> strcmp(word2,word4)
ans =
     0
```

```
>> strncmp(word1,word3,4)
ans =
     1
>> strcmpi(word1,word4)
ans =
     0
>> strcmpi(word2,word4)
ans =
     1
>> strncmpi(word2,word4,4)
ans =
     1
```

String – strfind function

strfind function receives 2 strings (string and substring); it **finds all the occurrences of the substring** with the string and **returns the subscripts** of the beginning of the occurrences.

```
>> strfind ('kemaman','e')
ans =
     2
>> strfind ('kemaman','mam')
ans =
     3
>> strfind ('kemaman','m')
ans =
     3     5
>> strfind ('kemaman','ne')
ans =
     []
```

How to find the number of words in a string?

String – strrep function

strrep function finds all occurrences of a substring within a string and **replaces** them with a new substring.

```
>> newlevel=strrep('groundfloor','ground','first')
newlevel =
firstfloor
```

String – `is` functions

There several ***is*** functions for sting which return logical true or false.

`isletter` function returns a logical true for every character in a string if the character is a letter of the alphabet.

```
>> isletter('SA 34')
ans =
     1     1     0     0     0
```

`isspace` function returns logical true for every character that is a white space character.

```
>> isspace('SA 34')
ans =
     0     0     1     0     0
```

String – ischar functions

ischar function returns a logical true if the vector argument is a character vector (a string).

```
>> kod='sa 34';
>> ischar (kod)
ans =
     1
>> vek=3:6;
>> ischar (vek)
ans =
     0
```

String – conversion functions

int2str function converts integers to strings.

num2str function converts real numbers to strings.

```
>> yr=57
yr =
     57
>> syr=int2str(yr)
syr =
57
```

Strings are left justified

```
>> length(yr)
ans =
      1
```

57 is a scalar

```
>> length(syr)
ans =
      2
```

5 & 7 are two characters

String – conversion functions

```
>> str1=num2str(58.123456)
str1 =
58.1235
>> length(str1)
ans =
     7
```

String with 4 decimal places. Matlab default for displaying real numbers

```
>> str2=num2str(58.123456,5)
str2 =
58.123
```

Number of digits specified

```
>> str3=num2str(58.123456,'%6.1f')
str3 =
58.1
```

Leading blanks from the string removed

String – conversion functions

```
>> num=str2num('318.1957')
num =
   318.1957
```

String converted to double

```
>> mulnum=str2num('31 8 1957')
mulnum =
        31         8      1957
```

String converted to vector of numbers (default type double)

```
>> sum(num2)
ans =
         1996
```

Sum of the elements

Str2double function is a better function to use in general than sr2num, but it can only be used when a scalar is passed; it would ot work for the variable mulnum

```
>> str2double('11 3 123')
ans =
    NaN
```