MATLAB Strings

Selim Aksoy
Bilkent University
Department of Computer Engineering
saksoy@cs.bilkent.edu.tr

Strings

- A string is an array of characters
  - s = 'abc'
    is equivalent to s = ['a' 'b' 'c']
- All operations that apply to vectors and arrays can be used together with strings as well
  - s(1) → 'a'
  - s([1 2]) = 'XX' → s = 'XXc'
  - s(end) → 'c'

String Conversion

- Conversion of strings to numerical arrays
  - double('abc xyz')
    ans =
    97 98 99 32 120 121 122
  - double('ABC XYZ')
    ans =
    65 66 67 32 88 89 90
- Conversion of numerical arrays to strings
  - char([72 101 108 108 111 33])
    ans =
    Hello!

Character Arrays

- 2-D character arrays
  - s = ['my first string'; 'my second string']
    ??? Error
  - s = char('my first string', 'my second string')
    s =
    my first string
    my second string
    } char function automatically pads strings
  - size(s) → [2 16]
  - size(deblank(s(1,:))) → [1 15]

String Tests

- ischar() : returns 1 for a character array
  - ischar ('CS 111')
    ans =
    1
- isletter() : returns 1 for letters of the alphabet
  - isletter('CS 111')
    ans =
    1 1 0 0 0 0
- isspace() : returns 1 for whitespace (blank, tab, new line)
  - isspace('CS 111')
    ans =
    0 0 1 0 0 0

String Comparison

- Comparing two characters
  - 'a' < 'e'
    ans =
    1
- Comparing two strings character by character
  - 'fate' == 'cake'
    ans =
    0 1 0 1
  - 'fate' > 'cake'
    ans =
    1 0 1 0

String Comparison

- strcmp() : returns 1 if two strings are identical
  - a = 'Bilkent';
  - strcmp(a, 'Bilkent')
    ans =
    1
  - strcmp('Hello', 'hello')
    ans =
    0
- strcmpi() : returns 1 if two strings are identical ignoring case
  - strcmpi('Hello', 'hello')
    ans =
    1

String Case Conversion

- Lowercase-to-uppercase
  - a = upper('This is test 1!')
    a =
    THIS IS TEST 1!
- Uppercase-to-lowercase
  - a = lower('This is test 1!')
    a =
    this is test 1!

Searching in Strings

- findstr() : finds one string within another one
  - test = 'This is a test!';
  - pos = findstr(test, 'is')
    pos =
    3 6
  - pos = findstr(test, ' ')
    pos =
    5 8 10

Searching in Strings

- strtok() : finds a token in a string
  - [token, remainder] = strtok('This is a test!', ' ')
    token =
    This
    remainder =
    is a test!
  - remainder = 'This is a test!';
    while (any(remainder)),
    [word, remainder] = strtok(remainder);
    disp(word);
    end

Replacing in Strings

- strrep() : replaces one string with another
  - s1 = 'This is a good example';
  - s2 = strrep(s1, 'good', 'great')
    s2 =
    This is a great example

String Conversion

- Recall num2str() for numeric-to-string conversion
  - str = ['Plot for x = ' num2str(10.3)]
    str =
    Plot for x = 10.3
- str2num() : converts strings containing numbers to numeric form
  - x = str2num('3.1415')
    x =
    3.1415

String Conversion

- sprintf() is identical to fprintf() but output is a string
  - str = sprintf('Plot for angle = %0.4f', pi)

    str =

    Plot for angle = 3.1416

String Comparison

- Example: Write a function that takes two strings and returns
  - -1 if the first string is lexicographically less than the second
  - 0 if they are equal
  - +1 if the first string is lexicographically greater than the second
- Pseudocode:
  - Get input strings
  - Pad strings to equal length
  - Compare characters from beginning to end and find the first difference
  - Return a value depending on the first distance

String Comparison

```
function result = c_strcmp(str1,str2)
%C_STRCMP Compare strings like C function "strcmp"
% Function C_STRCMP compares two strings, and returns
% a -1 of str1 < str2, a 0 if str1 == str2, and a
% +1 if str1 > str2.

%  Record of revisions:
%      Date       Programmer          Description of change
%      ====       ==========          =====================
%    10/18/98    S. J. Chapman        Original code

% Check to see if the arguments are strings
if ~(isstr(str1) & isstr(str2))
   error('Both str1 and str2 must both be strings!')
else
   % Pad strings
   strings = strvcat(str1,str2);
   % Compare strings
   diff = strings(1,:) ~= strings(2,:);
   if sum(diff) == 0
      % Strings match, so return a zero!
      result = 0;
   else
      % Find first difference between strings
      ival = find(diff);
      if strings(1,ival(1)) > strings(2,ival(1))
         result = 1;
      else
         result = -1;
      end
   end
end
```