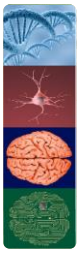
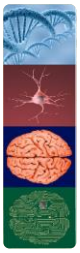


Introduction to Programming 2017/2018

Tirgul 8: String formats & XLS access

Michal Israelashvili
Yocheved Loewenstern



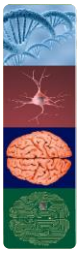
Lesson Outline

- Reading & writing .xls files.
- String formats.
- Reading & writing text files.

Reading Excel Files

- You can import data from a Microsoft excel worksheet (.xls file) into Matlab.
- The imported data is read into a Matlab variable.

- Syntax:
  data=xlsread('fileName')
  data=xlsread('fileName','sheet')
  data=xlsread('fileName','sheet','range')
  [dataNum dataTxt] = xlsread ('fileName',…)
  [dataNum dataTxt dataRaw] = xlsread ('fileName',…)

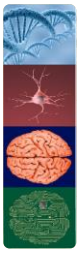


Writing Excel Files

- You can export data from Matlab into an Excel file.
- Data can be exported into an existing file or a new file.

- Syntax:
  xlswrite('fileName',*data*)
  xlswrite('fileName',*data*,'sheet','range')
  ** data = is a Matlab variable (numeric, character or cell array)*

- *Example code: xlsExample.m*



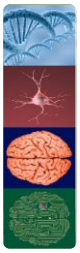
String formats I

- We can create complex strings constructed of text and combined values of variables.
- Functions: sprintf
- Syntax: str = sprtinf('format', varA, varB, ….)
- Format syntax: "*%type*" to indicate the place of the variable value inside the string.

- Example:
  nStudents = 8;
  str = sprintf('Number of students: %d', nStudents)
  →
  str-> Number of students: 8

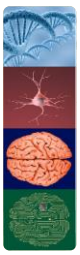
String formats II

- The 'type' is constructed of:
  - Conversion:
    - d, i, o, u, x, X, f, e, E, g, G, c, and s.
  - Precision and width
    - digit – width of variable
    - .digit – precision after the decimal point
  - Flags:
    - +, 0, …
- Special characters: \n, \t, \\, etc.



String formats example

- Example: *stringFormats.m*
- Examples for uses of formatted strings:
  fileNames.m
  funcPlot.m

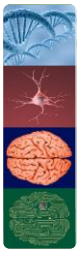


Reading & Writing Text files

- String formats can be used to read and write text files.
- Steps in dealing with files:
  - Open - fopen
  - Read or write – fprintf, fscanf, fgets, fgetl.
  - Close – fclose
- We will not cover this in the tirgul, but the example file ***textFilesFnc.m*** can be found in the course's web-site.

Practice

1. Create a 5X6 matix of random numbers between 0-100.
2. Use the *xlswrite* command to write this matrix into an excel file, with these specifications: file name: ***xlsPractice***, worksheet name: ***dataMat***.
3. Write a function called **myXls** which receives one input argument – an integer between 1-5. This function should read the numerical data from the excel file you created in part 2 of the exercise. The function should read only the data in the line whose number is given by its input argument (use the *xlsread* command to read the data). Tip: you can use the *sprintf* command to generate the string containing the range of cells to read.
4. Next, your function should calculate the mean value of the numbers in the given line and write this number to a new worksheet in the same xls file (the *xlsPractice* file). The name of this new worksheet should be: ***meanLineX*** (where *X* is the line number). Use the *sprintf* command to generate the string containing the worksheet name.



Functions/Commands List

- xlsread, xlswrite
- disp
- sprintf
- Functions dealing with text-files:
  fopen, fclose, fprintf, fscanf, fgets, fgetl