



Introduction to Programming 2017/2018


Tirgul 5: Strings & Data types

Michal Israelashvili
Yocheved Loewenstern




Lesson outline

- Strings
- Logical arrays and logical indexing
- Cell arrays
- Structures




Characters & strings

- Character = a printable/readable sign (text).
 - Creation in MATLAB: '<a letter>'
 - Examples: 'r' or '.' or ''
- String = an array of characters.
 - Creation in MATLAB: '<a few letters>'
 - Example: A = 'this is a MATLAB string'
- Strings and string manipulations:
- ***stringScript.m***



String functions


- Comparing strings:
 - strcmp
 - strcmpi – case insensitive.
- 'help strfun' - list of all string functions.
- More on strings – later in the course.



Logical expressions– a reminder


- There is no meaning to the numeric value – only to the truth value, which can be either 0 (false) or 1 (true). For example:


```
logical([0,1,2,5,0,-5]) =
[false, true, true, false, true] =
[0, 1, 1, 0, 1]
```
- We can use the logical operators (not, and, or, and their combinations) to create expressions with truth values (logical expressions).




Logical arrays

- The logical data type contains logical “true” or “false” values represented by the numbers 1 and 0, respectively.
- Logical arrays are most often used for indexing. “Logical indexing” is positional rather than numeric.
- “find” command
- Examples: *logicArrays.m*




Cell arrays

- A cell array provides a storage mechanism for dissimilar kinds of data.
- You can store arrays of different types and/or sizes within a single cell array.
- For Example: If you have vectors containing spike times of neurons (they are expected to be of different lengths) you can store them together in a cell array.
- Example: `cellArrays.m`




Structure arrays

- Structures are arrays with named "data containers" called fields.
- The different fields of a structure can contain different kinds of data.
- Structures are useful when you have different kinds of data that you want to save together in a single variable, while giving 'labels' to each type of data.
- Example: `structresFunc.m`



MATLAB data files (MAT-files)

- Matlab **variables** can be saved in files.
- Files with extension '.mat'.
- You can save as many variables as you want in one MAT-file.




Save

- Saving variables to disk - creating a mat-file with the variables.

Syntax: `save('fileName','parm1','parm2',...)`

- `fileName` – The name of the file (if not included, 'mat' extension will be added automatically).
- When omitting the variables names, all the variables in the workspace are saved.




Load

- Loading variables from disk - reading a mat-file that includes the variables and inserting them to the workspace.

Syntax: `load('fileName','parm1','parm2',...)`

- Overrides existing variables with the same name in the workspace.



Practice - Logical indexing

- Create a vector called **dataVec** of size 1x10, containing random numbers between 0-10. Create a new vector called **resVec**, containing values as follows:
 - In the locations `dataVec` has values smaller than 5 **resVec** should have the value -1.
 - In the locations `dataVec` has values bigger than 5 **resVec** should have the value 1.
- Use logical indexing to assign values to `resVec`.
- Use the 'find' command to assign values to `resVec`.



Practice - Structures and cell arrays

- The following table contains data about a few biblical personas – their name, gender and age:

| | | | |
|--------|---------|------|------|
| Name | Abraham | Sara | Adam |
| Gender | M | F | M |
| age | 175 | 127 | 930 |

- Create a 3x3 cell array called **cellBib** that contains the data in this table – each row should contain data of one person. Make sure you use the appropriate data types: names and genders are strings, ages are numbers.
- Create a 1x3 structure array called **structBib**, with the fields: name, gender, age. Assign the values in the table to **structBib**, such that each structure in the array will contain the data of one person.



Functions/Command List

- ''
- num2str
- str2num
- strcmp
- strcmpi
- Logical
- Find
- cell, cell2mat, num2cell, mat2cell.
- struct, rmfield, setfield, getfield, fieldnames, repmat
- save, load