MATLAB Fast Automation

Abbreviations

abbreviation	*meaning*	*abbreviation*	*meaning*
arg	argument	fdr	folder
c	column	ls	list
char	character	m	matrix
dbt	dialogBoxTitle	num	number
desc	description	o	offset
dest	destination	out	output
dlg	dialog	p	path
elts	elements	pmt	prompt
ext	extension	r	row
f	formatted	sel	selection
fct	fonction	sz	size
fl	file	nm	name

1. The Basic Tools for Automation

Cells

description	*command*
define a cell	`cell = {x}`
access cell content	`cell{:}`
define a list	`list = [{'firstWord'}; {'secondWord'}]`
access a cell	`list{i}`
number of cells	`numel(list)`

Matrices

description	*command*
define matrix	`m = [1 2; 3 4]`
number of rows	`size(m, 1)`
number of columns	`size(m, 2)`
remove second column	`m(:, 2) = []`
remove second row	`m(2, :) = []`
inverse matrix	`inv(m)`
transpose of a matrix	`m'`
multiplying matrices	`A*B`
element by element	`A.*B`

Useful Functions

description	*command*
separate data	`[token, remain] = strtok(variable, char)`
split data	`C = strsplit(variable, delimiter)`
compare strings	`strcmp(text1, text2)`
number to string	`num2str(numberToConvertToString)`
string to number	`str2num(stringToConvertToNumber)`
check for content	`isempty(variable)`
compare lists	`[newElts oldElts] = newOld(oldLs, newLs)`

Standard Example:

strsplit
```
>> C = strsplit('w1 w2', ' ')
C =
  1x2 cell array
    'w1'    'w2'
```

strtok
```
>> [tkn, rmn] = strtok('fl.txt', '.')
tkn =
fl
rmn =
.txt
```

2. Extract and Generate Excel Files

description	*command*
read an Excel file	`[num txt txtAndNum] = xlsread(flNm)`
specify sheet name	`xlsread(flNm, sheetName)`
specify sheet number	`xlsread(flNm, sheetNumber)`
write an Excel file	`xlswrite(flNm, data, sheetNumber)`
store a matrix	`store_matrix(flNm, shtNum, m, p, r0, c0)`
store a column	`storeColumn(flNm, data, sheetNumber)`
get a column	`c = getSelectedColumns(flNm, shtNum, cSel)`
get column 1	`c = getFirstExcelColumn(flNm, shtNum)`

Standard Example:

read text in an Excel file
```
fl = 'excelFile.xlsx';
sht = 1;
[~, txt, ~] = xlsread(fl, sht)
```

write text in an Excel file
```
fl = 'excelFile.xlsx';
sht = 1;
data = [{'firstWord'}; {'sceondWord'}];
xlswrite(fl, data, sht);
```

3. Manipulate Files

Files and Folders

description	*command*
move a file	`movefile(flNm, fullfile(dest, flNm))`
copy a file	`copyfile(flNm, fullfile(dest, flNm))`
create a folder	`mkdir(fdrNm)`
delete a folder	`rmdir(fdrNm, 's')`
move a folder	`movefile(fdrNm, fullfile(dest, fdrNm))`
use the clock	`c = clock % [year month day hour min]`

Read Files in a Folder

description	*command*
list folder content	`FilesAndFolders = dir`
list .m files	`files = dir('*.m')`
list specific extension	`files = dir(['*' extension])`

Read and Write Files

description	*command*
open a file to write	`fid = fopen('textFileName.txt', 'at')`
open a file to read	`fid = fopen('textFileName.txt')`
write content	`fprintf(fid, '%s\n', 'Text')`
read a line	`fgetl(fid)`
close a file	`fclose(fid)`
replace words	`strrep(textLine, oldText, newText)`
check keyword	`~isempty(strfind(textLine, keyword))`
detect errors	`assert(booleanValue, 'Error!')`

Standard Example:

read a file
```
fid = fopen('flNm.txt');
while ~feof(fid)
    textLine = fgetl(fid)
end
fclose(fid)
```

write a file
```
fid = fopen('flNm.txt', 'at');
fprintf(fid, '%s\n', 'Text.');
fclose(fid);
```

Know If a File or a Folder Exists

Standard Example:

7 if it's a folder
```
if exist('folderName')==7
    ...
end
```

2 if it's a file
```
if exist('fileName.ext')==2
    ...
end
```

Generate LaTeX Files

Standard Example:

```
fid = fopen('document.tex', 'at');
fprintf(fid, '%s', ['\documentclass[10pt]{article}' char(10)]);
fprintf(fid, '%s', ['\usepackage{mathpazo}' char(10) char(10)]);
fprintf(fid, '%s', ['%% comment' char(10)]);
fprintf(fid, '%s', ['\title{Generated Latex File}' char(10)]);
fprintf(fid, '%s', ['\begin{document}' char(10) char(10)]);

fprintf(fid, '%s', ['\maketitle' char(10)]);
fprintf(fid, '%s', ['\tableofcontents' char(10) char(10)]);
fprintf(fid, '%s', ['\section{Section}' char(10)]);
fprintf(fid, '%s', ['\begin{itemize}' char(10)]);
fprintf(fid, '%s', ['\item Bullet point' char(10)]);
fprintf(fid, '%s', ['\end{itemize}' char(10) char(10)]);

fprintf(fid, '%s', ['\end{document}' char(10)]);
fclose(fid);
```

Latex Functions

description	*command*
format text	`formattedStr = formatStrTex(str)`
format matrix	`fM = formatMatrixTex(matrix)`
write a table	`tableTex(fid, fM, h, colorH, forcedSz, fontSz)`

`fid`	output of the fopen MATLAB function
`fM`	formatted matrix of cells
`h`	header of the matrix
`colorH`	color of the first line of the table
`forcedSz`	size of each column relative to full page width
`fontSz`	font size of the content of the table

4. User Interface

description	*command*
workspace query	`inputArgument = input(prompt)`
find a file	`[flNm, pNm] = uigetfile(ext, dbt, fl)`
use GUI	`list = inputdlg(pmt, dlgTitle)`

Special Characters

description	*command*
single quotation mark	`char(39)`
go to next line	`char(10)`
tabulation	`char(9)`

5. Best Practices on MATLAB: Improve Your Programming Skills

description	*command*
move location	`cd(path)`
add a path	`addpath(path)`
create pcode	`pcode 'fileToConvert.m'`
open in windows	`winopen(file)`
open in MATLAB editor	`edit(file)`
display text	`display('text')`
clear workspace	`clear all; close all; bdclose all; clc`
delete a file	`delete('fileName.txt')`

Use Structures

description	*command*
define a structure	`structureNm.fieldNm = value`
access an element	`structNm.fieldNm`
access struct array element	`structNm(eltIndex).fieldNm`
check for field	`isfield(structureNm, fieldNm)`
find field names	`fieldnames(structureNm)`
check for structure	`isstruct(structureNm)`

Define Shortcuts

1. make sure you are in the right location using `cd(path)`
2. add your path in the search path using `addpath(path)`
3. define your arguments
4. call your function

Backup Shortcuts

1. use `winopen(prefdir)`
2. save/backup **shortcuts_2.xml**
3. save/backup **MATLABQuickAccess.xml**

Keyboard Shortcuts

description	*keyboard*
run a script	F5
indent the selected code	CTRL+I
comment code	CTRL+R
uncomment code	CTRL+T
open corresponding .m file	CTRL+D
run a piece of code in a script	F9
stop the execution of a code	CTRL+C
jump between variables	ALT+UP/DOWN
open new script/close script	CTRL+N/CTRL+W
run a cell delimited by %%	CTRL+ENTER