

MATLAB Cheat Sheet: B(E)0B17MTB & A8B17CAS Courses

Useful Commands

<code>clc</code>	clear command window
<code>clear</code>	clear workspace
<code>clear x</code>	clear x from workspace
<code>close all</code>	close all figures
<code>help function</code>	help for function
<code>doc function</code>	documentation for function
<code>quit, exit</code>	terminates MATLAB
<code>edit</code>	opens MATLAB editor
<code>functionName</code>	
<code>format</code>	sets command window output format
<code>short/long/compact</code>	

Built in Functions/Constants

<code>pi</code>	constant π
<code>exp()</code>	exponential function
<code>eps()</code>	floating point accuracy
<code>1e6</code>	1×10^6
<code>Inf</code>	infinity value
<code>NaN</code>	not-a-number value
<code>1i, 1j</code>	imaginary unit

Vectors/Matrices Creation

<code>x = [1 2 3]</code>	row vector 1×3
<code>x = [1; 2; 3]</code>	column vector 3×1
<code>x = [1 2; 3 4]</code>	matrix 2×2
<code>a:b</code>	row vector $[a, a+1, \dots, b]$
<code>a:x:b</code>	row vector $[a, a+x, \dots, b]$
<code>linspace(a, b, n)</code>	n evenly spaced points between a and b
<code>logspace(a, b, n)</code>	n logarithmically spaced points between 10^a and 10^b
<code>ones(n, m)</code>	matrix of values 1, $[n \times m]$
<code>zeros(n, m)</code>	matrix of values 0, $[n \times m]$
<code>eye(n)</code>	identity matrix, $[n \times n]$
<code>NaN(n, m)</code>	matrix of values NaN, $[n \times m]$
<code>Inf(n, m)</code>	matrix of values true, $[n \times m]$
<code>true(n, m)</code>	matrix of logical ones, $[n \times m]$
<code>false(n, m)</code>	matrix of logical zeros, $[n \times m]$
<code>rand(n, m)</code>	matrix of random uniformly distributed values, $[n \times m]$ (see also <code>randi</code> and <code>randn</code>)
<code>diag()</code>	diagonal elements of matrix, vector -> square matrix

Vectors/Matrices Operations

<code>a + b</code>	element-wise addition
<code>a - b</code>	element-wise subtraction
<code>a * b</code>	matrix multiplication
<code>a .* b</code>	element-wise multiplication
<code>a / b</code>	matrix division, <code>mrdivide</code>
<code>a \ b</code>	<code>ldivide</code>
<code>a ./ b</code>	element-wise division
<code>a ^ b</code>	matrix power
<code>a .^ b</code>	element-wise power
<code>a.'</code>	transpose
<code>a'</code>	transpose + complex conjugate (Hermitian transpose)
<code>[A B]</code>	concatenates 2 matrices horizontally
<code>[A; B]</code>	concatenates 2 matrices vertically
<code>cat(dim, A, B)</code>	concatenates 2 matrices
<code>length(A)</code>	size of longest dimension of A
<code>size(A)</code>	size of A
<code>ndims(A)</code>	number of dimensions of A
<code>numel(A)</code>	number of elements of A
<code>tril(A)</code>	lower triangular part of matrix
<code>triu(A)</code>	upper triangular part of matrix
<code>repmat(A, x, y)</code>	repeats copy of matrix
<code>reshape(A, x, y)</code>	changes dimensions of matrix
<code>squeeze(A)</code>	removes dimensions of size 1
<code>flip(A, dim)</code>	flips array (see also <code>fliplr</code> and <code>flipud</code>)
<code>circshift(A, K, dim)</code>	circularly shifts array
<code>det(A)</code>	matrix determinant
<code>inv(A)</code>	matrix inversion
<code>roots(vec)</code>	polynomial roots
<code>round(A)</code>	round to nearest decimal integer (see also <code>ceil</code> , <code>floor</code> , and <code>fix</code>)
<code>mod(A, n)</code>	modulo operation
<code>rem(A, n)</code>	remainder after division

Logical Operators

<code>&</code>	logical AND
<code>&&</code>	short-circuit logical AND
<code> </code>	logical OR
<code> </code>	short-circuit logical OR
<code>~</code>	not
<code>all</code>	function all
<code>any</code>	function any
<code>is*</code>	several function with logical output

Relation Operators

<code>></code>	greater then
<code>>=</code>	greater then or equal to
<code><</code>	less then
<code><=</code>	less then or equal to
<code>==</code>	equal to
<code>~=</code>	not equal to

Vector Indexing

<code>a(1)</code>	the first element in vector
<code>a([1, 2, 5])</code>	the first, second and fifth element in vector
<code>a(1:3)</code>	the first three elements of an vector
<code>a([1:3 5:6])</code>	selected elements of a vector
<code>a(end)</code>	the last element of the vector
<code>a(5:end)</code>	from the fifth to the last element in vector
<code>a(5:end-1)</code>	from the fifth to the penultimate element in vector
<code>a(1) = 10</code>	replace the first element in vector by 10
<code>a([1 3 5]) = [2 5 7]</code>	replace elements in vector by values 2, 5, and 7
<code>a([1 3 5]) = []</code>	discard elements on position 1, 3, and 5

Matrix Indexing

<code>A(1, 1)</code>	an element in the first row and first column of matrix
<code>A([1 5], 2:4)</code>	elements of matrix in the first and fifth rows and in columns two, three and four
<code>A(n, :)</code>	the n th row in array
<code>A(:, n)</code>	the n th column in array
<code>A(n, :) = []</code>	discard n th row in array
<code>A(:, n) = []</code>	discard n th column in array
<code>A(1)</code>	the first element in array (linear indexing)
<code>A(1:3)</code>	the first three elements of an array (linear indexing)

Logical Indexing

<code>A(A == 3)</code>	extract elements equal to 3
<code>A(A > 5)</code>	extract elements greater than 5
<code>A(A > 5 & A < 30)</code>	extract elements greater than 5 and at the same time less than 30

Debugging and Time Measurement

<code>keyboard</code>	pauses execution
<code>dbcont</code>	resume execution (end debug mode)
<code>dbclear all</code>	removes all breakpoints
<code>tic, toc</code>	start/stop time measurement
<code>profile</code>	profile code execution
<code>on/off/clear/viewer</code>	

Loops and Branching

<pre> if expression ... elseif expression ... else ... end </pre>	program branching using if-else
<pre> switch variable case value1 ... case {value2, value3} ... otherwise ... end </pre>	program branching using switch
<pre> for n = 1:10 ... end </pre>	for cycle
<pre> while expression ... end </pre>	while cycle
<pre> break continue return </pre>	terminates execution of loop pass control to next iteration return to invoking function

Functions

<code>function</code>	function definition
<code>[out1, out2] = foo(in1, in2)</code>	
<code>@sin</code>	handle function definition
<code>@(x) x^2 + sin(x)</code>	anonymous function definition
<code>nargin</code>	returns number of inputs
<code>nargout</code>	returns number of outputs
<code>varargin</code>	input variable allowing multiple inputs
<code>varargout</code>	output variable allowing multiple outputs

Set operations

<code>intersect(A, B)</code>	set intersection of two arrays
<code>union(A, B)</code>	set union of two arrays
<code>setdiff(A, B)</code>	set difference of two arrays
<code>setxor(A, B)</code>	set exclusive or of two arrays
<code>unique(A)</code>	unique values in array
<code>sort(X)</code>	sort array elements
<code>sortrows(X)</code>	sorts rows of matrix
<code>issorted(X)</code>	determines if the array is sorted
<code>ismember(A, B)</code>	array elements that are members of set array

Basic Visualization

figure	opens empty figure
axes	creates Cartesian axes
plot(x, y)	2D line plot
hold on/off	retain current plot with new line
grid	display/hide grid lines
on/off/minor	
title(txt)	adds title to figure
xlabel(txt)	adds x-axis label (same for y and z)
xlim([min, max])	sets limits of x-axis (same for y and z)
colormap	view colormap
stem(x, y)	discrete stem plot
pcolor(A)	displays array data as colored cells
surf(X, Y, Z)	displays surface plot
semilogx(X)	semi-logarithmic plot (same for y)
image(X)	Show image from array
doc LineSpec	plot parameters to customize curves
set(ref, Name, Value)	sets graphics object property
get(ref, Name)	query graphics object property value
ref.Name = Value	dot notation, sets graphics object property
Value = ref.Name	dot notation, query graphics object property value
view(az, el)	camera line of sight

Character arrays and Strings

<code>"Hello world!"</code>	character vector is created using single quotation marks
<code>A = char(B)</code>	convert another data type to character array
<code>strfind(str, pattern)</code>	find string within other string
<code>strcmp(str1, str2)</code>	compare strings
<code>strjoin(str, delimiter)</code>	join strings in array
<code>strtok(str, delimiter)</code>	split string into parts
<code>regex(str, reg)</code>	match regular expression
<code>"Hello world!"</code>	string is created using double quotation marks
<code>A = string(B)</code>	convert another data type to string

File Handling

<code>dir folderName</code>	lists folder content
<code>pwd</code>	identifies current folder
<code>exist name</code>	check existence of variable, function, folder, ...
<code>cd(path)</code>	changes current folder
<code>mkdir(path, folderName)</code>	makes new folder
<code>rmdir(path)</code>	remove folder

<code>writematrix(A,</code>	<code>writes matrix to file (see also</code>
<code>filename)</code>	<code>readmatrix)</code>
<code>imwrite(A,</code>	<code>writes image to graphics file (see also</code>
<code>filename)</code>	<code>imread)</code>
<code>fid =</code>	<code>open file</code>
<code>fopen(filename)</code>	
<code>fclose(fid)</code>	<code>close opened file</code>
<code>fgetl(fid)</code>	<code>read line from file</code>
<code>fread(fid)</code>	<code>read from binary file</code>
<code>fwrite(fid, A)</code>	<code>write to binary file</code>

Other Data Types

<code>A = cell(m, n)</code>	create cell array
<code>A = {x, y, z}</code>	create cell array
<code>A(1, 1)</code>	smooth parentheses indexing, creates subset cell array
<code>A{1, 1}</code>	curly brackets indexing, access to data in cell array
<code>B =</code>	create structure array
<code>struct(field, value)</code>	
<code>B.field = value</code>	create structure array
<code>double(A)</code>	converts array to double precision (see also <code>single</code>)
<code>int8(A)</code>	converts array to 8-bit signed integer (see also <code>int16</code> , <code>int32</code> , <code>uint8</code> , <code>int18</code> , ...)
<code>datetime()</code>	array representing points in time (see also years, days, hours, ...)
<code>sparse(A)</code>	creates sparse matrix

Other useful functions

<code>find(X)</code>	find indices of nonzero elements
<code>fprintf(fileID, format, A1)</code>	write data to text file
<code>sprintf(format, A1)</code>	format data into character vector
<code>eval(txt)</code>	executes MATLAB expression in text
<code>feval(fun, x1)</code>	evaluate function
<code>str2num(X)</code>	convert character array to numeric array
<code>str2double(X)</code>	convert characters to double precision numbers
<code>num2str(X)</code>	convert numeric array to character array
<code>realmax</code>	largest positive floating-point number (see also <code>realmin</code> , <code>intmax</code> , and <code>realmin</code>)
<code>interp1</code>	1-D interpolation (see also <code>interp2</code> , <code>interp3</code> , and <code>interpN</code>)
<code>fminbnd</code>	find minimum of single-variable function
<code>fft</code>	Fast Fourier transform (see also <code>fft2</code> , <code>fftn</code> , <code>ifft</code> , <code>fftshift</code> , ...)
<code>timer</code>	creates object of timer to schedule execution of MATLAB commands (see documentation)

Shortcuts Command Window

ENTER	line execution
ESC	delete whole line
DEL	delete one character (right to the cursor)
BACKSPACE	delete one character (left to the cursor)
HOME	moves cursor to the beginning of line
END	moves cursor to the end of line
CTRL + ↑	moves cursor to the beginning of next word
CTRL + ↓	moves cursor to the beginning of previous word
SHIFT + ENTER	sends cursor to the next line (without execution)
CTRL + K	deletes all to the right of cursor
CTRL + C	forces interruption of MATLAB (<i>e.g.</i> long/erroneous calculation)
CTRL + TAB	switching between windows of MATLAB environment
↓ and ↑	command history listing (searching in available CTRL + F)
F1	context help related to the word where the cursor is placed
TAB	function or variable name hint

Shortcuts MATLAB Editor

CTRL +	moves between tabs
PG.UP/PG.DOWN	
CTRL + R	comment lines
CTRL + T	uncomment lines
CTRL + D	open highlighted m-file (function/script)
F5	execute current script/function
CTRL + S	save current file (done automatically after pressing F5)
CTRL + HOME	jump to the beginning of file
CTRL + END	jump to the end of file
CTRL + →/←	jump word-by-word or expression-by-expression to the right/left
CTRL + W	close current file
CTRL + O	activates open file dialog box
CTRL + F	find/replace dialog box
CTRL + G	“go to”, jumps to the indicated line number
CTRL + I	indent of block of lines corresponding to key words
F1	open context help related to the function at position of the cursor
F9	run highlighted code

MATLAB File Suffix (not all)

.m	script/function/class
.mlx	MATLAB live script
.p	protected MATLAB code
.mat	binary data file
.fig	MATLAB figure
.mdl, .slx	Simulink model
.mdl, .slxp	Simulink protected model
.mexa64,	mex libraries
.mexmaci64,	
.mexw32,	
.mexw64	
.mlappinstall	APP file – installer
.mlpkginstall	support package – installer
.mltbx	toolbox file – installer
.mn	MuPAD notebook
.mu	MuPAD code

Handle Graphics

```
fig = figure
set(ref, Name, Value)
get(ref, Name)
ref.Name = Value
Value = ref.Name
```

```
groot, get(0)
axes
axis(lims)
uimenu
uicontextmenu
uitoolbar
uipanel
uitabgroup
uitab
uitable
uibuttongroup
```

```
uicontrol
```

```
uicontrol('Style', style)
uicontrol('Style', 'Text')
uicontrol('Style', 'Edit')
uicontrol('Style', 'Pushbutton')
uicontrol('Style', 'Togglebutton')
uicontrol('Style', 'Radiobutton')
uicontrol('Style', 'Checkbox')
uicontrol('Style', 'Listbox')
uicontrol('Style', 'Popupmenu')
uicontrol('Style', 'Slider')
```

```
set(ref, 'Callback', @foo)
set(ref, 'Callback', @(src,
event)foo(in1))
set(ref, 'Callback', {@foo, in1})
```

```
gcf
```

```
gca
```

```
gco
```

```
findobj(property, value)
findall
allchild(ref)
copyobj(ref)
delete(ref)
```

```
setappdata(ref, field, data)
getappdata(ref, field)
```

```
msgbox(mess)
```

get figure reference
sets graphics object property
query graphics object property value
dot notation, sets graphics object property
dot notation, query graphics object property value

graphics root object (screen)
creates Cartesian axes
scales axes
creates menu or menu item
creates context menu
creates toolbar with icons
creates panel; container for other objects
creates tabgroup; container for other tabs
creates a tab; container for other objects
creates 2D table
creates superior object for group of buttons (*i.e.* radiobuttons)
creates fundamental element of GUI, behavior depends on style of the object
changes style of uicontrol
creates text field
creates editable text field
creates one-state button
creates two-state button
creates radio button
creates checkbox
creates list of items
creates popup menu with multiple choices
creates slider

set callback as function handle
set callback as anonymous function

set callback as function handle in cell array

returns reference of current figure (last active)
returns reference of current axes (last active)
returns reference of last "mouse-clicked" object
finds object(s) with required property
finds all graphics objects
finds all children objects of selected object
copies an object
deletes object

defines stored data inside the application
retrieves previously stored data

creates message dialog window

App Building (new graphics)

```
uifigure
uiaxes
uipanel
uitab
uigridlayout
```

```
uibuttongroup
```

```
uimenu
uitable
uilabel
uieditfield
uitextarea
uibutton
uiradiobutton
uicheckbox
uiliblistbox
uidropdown
```

```
uislider
uispinner
uiswitch(style)
```

```
uiknob(style)
```

```
uigauge(style)
```

```
uilamp
uitree
uidatepicker
uiimage
uihtml
```

```
ref.ButtonPushedFcn = @foo
ref.ButtonPushedFcn = @(src,
event)foo(in1))
ref.ButtonPushedFcn = {@foo, in1}
```

```
setappdata(ref, field, data)
getappdata(ref, field)
```

```
uialert
uiconfirm
```

creates graphics window
creates axes in uifigure
creates panel; container for other objects
creates tab; container for other objects
layout manager; positioning of graphics components
creates superior object for group of buttons (*i.e.* radiobuttons)
creates menu or menu item
creates 2D table
creates text field
creates editable text field
creates multiple line editable text
creates one-state or two-state button
creates radio button component
creates checkbox component
creates list of items
creates drop-down component (popup menu)
creates slider component
creates spinner component
creates switch object (possible styles: 'slider', 'rocker', and 'toggle')
creates knob component (possible styles: 'continuous' and 'discrete')
creates gauge component (possible styles: 'circular', 'linear', 'ninetydegree', and 'semicircular')
creates lamp component
creates tree component
creates calendar component
creates image component
creates HTML component

set callback as function handle
set callback as anonymous function

set callback as function handle in cell array

defines stored data inside the application
retrieves previously stored data

predefined alert window
predefined confirmation window