

MATLAB Cheatsheet

Arithmetic Operators

Addition: + Subtraction: - Multiplication: * Division: / Exponentiation: ^ Grouping: ()

Variables

Assignment: <varname> = <val> Suppress output: end with ; (e.g. <varname> = <val>;)

Variable names must not begin with a digit and cannot contain spaces or operators

Expressions and Values

Values: stored information (numbers, strings, or matrices) Expression: code that *evaluates* to a value

Comparison Operators and Boolean Expressions

Booleans: 1 (true) or 0 (false)

Comparison Operators compare two numbers with a given relationship and return a boolean describing if the relationship holds

Less than: < Less than or equal to: <= Greater than: > Greater than or equal to: >= Equal: == Not equal: ~=

Control Flow: if statements and for loops

If statement:

```
if <condition>
    <do this if condition is true>
else
    <do this if condition is false>
end
```

Range operator:

a:b gives the range of values from a to b inclusive (i.e. $a, a + 1, \dots, b - 1, b$).

a:d:b gives the range of values from a to b stepping by interval d (i.e. $a, a + d, \dots, b - d, b$). Won't include b if b is not an even multiple of d from a .

For loop:

```
for <loop_var> = <range of values>
    <do this for each value of
loop_var>
end
```

Calling Functions

Function call: <return_value> = <function>(<arguments>);

Function documentation: help <functionname>

Multiple argument functions use commas to separate arguments: <function>(<arg1>, <arg2>, ...)

If you don't use an = to assign output, the result will be stored as ans

Writing New Functions

```
function [ <output> ] = <function_name>( <input_args> )
    <write some code here that assigns a value to the output variable>
end
```

Reading and Writing Images

`im = imread('<filename>.jpg');` read an image from a file into variable `im`
 `imshow(im);` display the loaded image `im`
`imwrite(im, '<filename>.jpg');` save the image `im` to a file

Matrices

`zeros` create a matrix full of zeros (e.g. `zeros(2,3)` creates a matrix of zeros with 2 rows and 3 columns)
`ones` create a matrix full of ones
`M(i,j)` access the value stored in row i , column j of M
`size(M)` returns the list of sizes in each dimension of M (e.g. `size(zeros(2,3))` would return `2 3`)
Arithmetic operators work on each element individually (e.g. $N = M + 10$ implies $N(i,j) = M(i,j) + 10$)
:
 slicing operator, shorthand for “all entries in this dimension” (e.g. `M(:,j)` outputs only column j across all rows)
Images are three-dimensional matrixes: row, column, and color channel (in order: red, green, blue)
(E.g. to pull out the green channel only for every pixel in the image, one could run `im(:, :, 2);`)

Useful Functions

Math: `abs`, `sqrt`, `min`, `max`, `sum`, `mod`

Strings: `upper`, `strcmp`

Turtles: `Turtle`, `turtleForward`, `turtleTurn`, `turtleDown`, `turtleUp`

Matrices: `zeros`, `ones`, `size`, `circshift`, `cat`

Images: `imread`, `imshow`, `imwrite`, `imresize`

Logical Operations: `and`, `or`, `not`

Figure Management

`figure(1);` Create a new figure window to display an image
`title('Put Figure Title Here');`
`xlabel('Put Label Here');`, `ylabel('Put Label Here');`
`close all` Close all open figures
`clear all` Clear all variables in the workspace
`clc` Clear the command window