**Summary of MATLAB Onramp**

**Basic syntax**

| **Example** | **Description** |
| --- | --- |
| [x = pi](https://www.mathworks.com/help/matlab/matlab_env/create-and-edit-variables.html) | Create variables with the equal sign (=). The left-side (x) is the variable name containing the value on the right-side (pi). |
| [y = sin(-5)](https://www.mathworks.com/help/matlab/learn_matlab/calling-functions.html) | You can provide inputs to a function using parentheses. |

**Desktop management**

| **Function** | **Example** | **Description** |
| --- | --- | --- |
| [save](https://www.mathworks.com/help/matlab/ref/save.html) | save data.mat | Save your current workspace to a MAT-file. |
| [load](https://www.mathworks.com/help/matlab/ref/load.html) | load data.mat | Load the variables in a MAT-file to the Workspace. |
| [clear](https://www.mathworks.com/help/matlab/ref/clear.html) | clear | Clear all variables from the Workspace. |
| [clc](https://www.mathworks.com/help/matlab/ref/clc.html) | clc | Clear all text from the Command Window. |
| [format](https://www.mathworks.com/help/matlab/ref/format.html) | format long | Change how numeric output is displayed. |

**Array types**

| **Example** | **Description** |
| --- | --- |
| 4 | scalar |
| [3 5] | row vector |
| [1;3] | column vector |
| [3 4 5;6 7 8] | matrix |

**Evenly-spaced vectors**

| **Example** | **Description** |
| --- | --- |
| 1:4 | Create a vector from 1 to 4, spaced by 1, using the [colon (:)](https://www.mathworks.com/help/matlab/ref/colon.html) operator. |
| 1:0.5:4 | Create a vector from 1 to 4, spaced by 0.5. |
| [linspace](https://www.mathworks.com/help/matlab/ref/linspace.html)(1,10,5) | Create a vector with 5 elements. The values are evenly spaced from 1 to 10. |

**Creating matrices**

| **Example** | **Description** |
| --- | --- |
| [rand](https://www.mathworks.com/help/matlab/ref/rand.html)(2) | Create a square matrix with 2 rows and 2 columns. |
| [zeros](https://www.mathworks.com/help/matlab/ref/zeros.html)(2,3) | Create a rectangular matrix with 2 rows and 3 columns. |

**Indexing**

| **Example** | **Description** |
| --- | --- |
| A([end](https://www.mathworks.com/help/matlab/ref/end.html),2) | Access the element in the second column of the last row. |
| A(2,:) | Access the entire second row |
| A(1:3,:) | Access all columns of the first three rows. |
| A(2) = 11 | Change the value of the second element an array to 11. |

**Array operations**

| **Example** | **Description** |
| --- | --- |
| [1 1; 1 1]\*[2 2;2 2]  ans =  4 4  4 4 | Perform [matrix multiplication](https://www.mathworks.com/help/matlab/matlab_prog/array-vs-matrix-operations.html#btyv9yp-4). |
| [1 1; 1 1].\*[2 2;2 2]  ans =  2 2  2 2 | Perform [element-wise multiplication](https://www.mathworks.com/help/matlab/matlab_prog/array-vs-matrix-operations.html#bu90xxy-1). |

**Multiple outputs**

| **Example** | **Description** |
| --- | --- |
| [xrow,xcol] = [size](https://www.mathworks.com/help/matlab/ref/size.html#bvfgzsm-6)(x) | Save the number of rows and columns in x to two different variables. |
| [xMax,idx] = [max](https://www.mathworks.com/help/matlab/ref/max.html)(x) | Calculate the maximum value of x and its corresponding index value. |

**Documentation**

| **Example** | **Description** |
| --- | --- |
| [doc](https://www.mathworks.com/help/matlab/ref/doc.html) randi | Open the documentation page for the randi function. |

**Plotting**

| **Example** | **Description** |
| --- | --- |
| [plot](https://www.mathworks.com/help/matlab/ref/plot.html)(x,y,"ro-","LineWidth",5) | Plot a red (r) dashed (--) line with a circle (o) marker, with a heavy line width. |
| [hold](https://www.mathworks.com/help/matlab/ref/hold.html) on | Add the next line to existing plot. |
| hold off | Create a new axes for the next plotted line. |
| [title](https://www.mathworks.com/help/matlab/creating_plots/add-title-axis-labels-and-legend-to-graph.html)("My Title") | Add a label to a plot. |

**Using tables**

| **Example** | **Description** |
| --- | --- |
| [data.HeightYards](https://www.mathworks.com/help/matlab/matlab_prog/access-data-in-a-table.html) | Extract the variable HeightYards from the table data. |
| data.HeightMeters = data.HeightYards\*0.9144 | Derive a table variable from existing data. |

**Logicals**

| **Example** | **Description** |
| --- | --- |
| [[5 10 15] > 12](https://www.mathworks.com/help/matlab/matlab_prog/array-comparison-with-relational-operators.html) | Compare a vector to the value 12. |
| [v1(v1 > 6)](https://www.mathworks.com/help/matlab/matlab_prog/find-array-elements-that-meet-a-condition.html) | Extract all elements in v1 that are greater than 6. |
| x(x==999) = 1 | Replace all values in x that are equal to 999 with the value 1. |

**Programming**

| **Example** | **Description** |
| --- | --- |
| [if](https://www.mathworks.com/help/matlab/ref/if.html) x > 0.5  y = 3  else  y = 4  end | If x is greater than 0.5, set the value of y to 3.  Otherwise, set the value of y to 4. |
| [for](https://www.mathworks.com/help/matlab/ref/for.html) c = 1:3  disp(c)  end | The loop counter (c) progresses through the values 1:3 (1, 2, and 3).  The loop body displays each value of c. |

>> save datafile.mat

Task 2 ✔

>> clear

Task 3 ✔

>> load datafile.mat

Task 4 ✔

>> data

data =

3.0000 0.5300 4.0753 NaN

18.0000 1.7800 6.6678 2.1328

19.0000 0.8600 1.5177 3.6852

20.0000 1.6000 3.6375 8.5389

21.0000 3.0000 4.7243 10.1570

23.0000 6.1100 9.0698 2.8739

38.0000 2.5400 5.3002 4.4508

Task 5

>> clc

u = sin(x)

y = pi/2

z = sqrt(-2)

In script

r = 3

x = pi\*r^2

Creating Vectors

Task 1

x = 4

Task 2

x = [7 9]

Task 3

x = [7;9]

Task 4

x = [3 10 5]

Create column vector

Task 5

x = [8;2;-4]

Create Matrix

Task 6

x = [5 6 7; 8 9 10]

Task 7

x = [sqrt(10) pi^2]

Creating Evenly-Spaced Vectors

Task 1

x = [1 2 3]

Row vector of 1 2 3 4

Task 2

x = 1:4

Task 3

x = 1:0.5:5

Row vector with values between 3 and 13 with difference as 2

Task 4

x = 3:2:13

Row vector with values 1 to 10 and number of values be 5

Task 5

x = linspace(1, 10, 5)

Transposing row vector to column vector and vice versa

Task 6

x = x'

Task 7

x = (5:2:9)'

Array Creation Functions

Create Matrix of random numbers 5 X 5

Task 1

x = rand(5)

Create Matrix of random numbers 5 X 1

Task 2

x = rand(5, 1)

Create Matrix of 0's 6 X 3

Task 3

x = zeros(6, 3)

Create Matrix of 1's 2 X 4

x = ones(2, 4)

rand(size(x))

Indexing into Arrays

This code sets up the interaction. data is a matrix of 9 X 9

load datafile

data

Getting value from matrix

Task 1

x = data(6, 3)

end is the last row here

Task 2

x = data(end, 3)

colon : refers to all rows here and 3rd column

Task 3

x = data(:, 3)

Extracting Multiple Elements

Instructions are in the task pane to the left. Complete and submit each task one at a time.

This code sets up the interaction.

data = [3 0.53 4.0753 NaN;18 1.78 6.6678 2.1328;19 0.86 1.5177 3.6852;20 1.6 3.6375 8.5389;21 3 4.7243 10.157;23 6.11 9.0698 2.8739;38 2.54 5.30023 4.4508]

Task 1

density = data(:,2)

Extract 3rd and 4th column all rows

Task 2

volumes = data(:,3:4)

Extract 6th value from vector starting with index as 1

Task 3

p = density(6)

Extracting values from vector with index 2 to 5

Task 4

p = density(2:5)

Changing Values in Arrays

Instructions are in the task pane to the left. Complete and submit each task one at a time.

This code sets up the interaction.

load datafile

data

Task 1

v2 = data(:,end)

Changing value in vector

Task 2

v2(1) = 0.5

Changing value in matrix

Task 3

data(1, end) = 0.5

Performing Array Operations on Vectors

Instructions are in the task pane to the left. Complete and submit each task one at a time.

This code sets up the interaction.

load datafile

density = data(:,2);

v1 = data(:,3);

v2 = data(:,4);

Task 1

r = v1 +1

Task 2

vs = v1 + v2

Task 3

va = vs/2

Task 4

vm = max(va)

Task 5

vr = round(va)

Task 6

mass = density .\* va

Obtaining Multiple Outputs

Instructions are in the task pane to the left. Complete and submit each task one at a time.

This code sets up the interaction.

load datafile

data

v1 = data(:,3);

v2 = data(:,4);

Task 1

dsize = size(data)

Task 2

[dr, dc] = size(data)

Task 3

[vMax, ivMax] = max(v2)

Further Practice

[~, ivMaximum] = max(v2)

Obtaining Help

Instructions are in the task pane to the left. Complete and submit each task one at a time.

Task 1

x = randi(20, 5, 7)

Further Practice

doc randi

Plotting Vectors

Instructions are in the task pane to the left. Complete and submit each task one at a time.

This code sets up the interaction.

load datafile

sample = data(:,1);

density = data(:,2);

v1 = data(:,3);

v2 = data(:,4);

mass1 = density.\*v1;

mass2 = density.\*v2;

Task 1

plot(sample, mass1)

Task 2

plot(sample, mass2, "r\*")

Task 3

hold on

plot(sample, mass1, "blacks")

Task 4

hold off

Task 5

plot(v1)

Task 6

plot(v1, "LineWidth", 3)

Task 7

plot(sample, v1, "r-o", "LineWidth", 4)

Further Practice

Plotting Vectors

Instructions are in the task pane to the left. Complete and submit each task one at a time.

This code creates data for the interaction.

load datafile

sample = data(:,1);

density = data(:,2);

v1 = data(:,3);

v2 = data(:,4);

mass1 = density.\*v1;

mass2 = density.\*v2;

This code creates the plot from the last activity.

plot(sample,mass1,"ks")

hold on

plot(sample,mass2,"r\*")

hold off

Task 1

title("Sample Mass")

Task 2

ylabel("Mass (g)")

Task 3

legend("Exp A", "Exp B")

Further Practice

Audio Frequency

Audio signals are usually comprised of many different frequencies. For example, in music, the note 'middle C' has a fundamental frequency of 261.6 Hz, and most music consists of several notes (or frequencies) being played at the same time.

In this project, you will analyze the frequency content of an organ playing the C chord.

The C chord consists of the C (261.6 Hz), E (329.6 Hz), and G (392.0 Hz) notes. The highlighted points in this frequency plot correspond to each note.

Audio Frequency

Instructions are in the task pane to the left. Complete and submit each task one at a time.

Task 1

load Cchord

n = numel(y)

t = 0:n-1

Task 2

t = t/fs

plot(t,y)

Task 3

yfft = abs(fft(y))

Task 4

f = 0:1:n-1

Task 5

f = f\* (fs/n)

plot(f, yfft)

xlim([0 1000])

Further Practice

Working with Tables

Instructions are in the task pane to the left. Complete and submit each task one at a time.

load datafile

elements

Task 1

d = elements.Density

Task 2 & 3

elements.Mass = elements.Density .\* elements.Volume1

elements = sortrows(elements, 'Mass')

Further Practice

top3 = elements(1:2, :)

Working with Tables

Instructions are in the task pane to the left. Complete and submit each task one at a time.

load datafile

elements

Task 1

d = elements.Density

Task 2 & 3

elements.Mass = elements.Density.\*elements.Volume1

elements = sortrows(elements,"Mass")

Further Practice

Logical Indexing

Instructions are in the task pane to the left. Complete and submit each task one at a time.

This code sets up the interaction.

load datafile

sample = data(:,1);

v1 = data(:,3);

Task 1

test = pi < 4

Task 2

test = v1 < 4

Task 3

v = v1 (v1 < 4)

Task 4

s = sample (v1 < 4)

Task 5

v1 (v1 < 4) = 0

Further Practice

x = v1(v1<4 & v1>2)

y = v1(v1>6 | v1<2)

z = sample (sample>10 & sample<20)

Decision Branching

Instructions are in the task pane to the left. Complete and submit each task one at a time.

Do not edit. This code creates a random number to test the if statement.

doPlot = randi([0 1])

This code loads the data.

load datafile

density = data(:,1);

Task 1 & 2

if doPlot == 1

plot(density)

title("Sample Densities")

xticklabels(element)

ylabel("Density (g/cm^3)")

else

disp("The density of " + element + " is " + density)

end

Further Practice

doDisplay = 0

if doPlot == 1

plot(density)

title("Sample Densities")

xticklabels(element)

ylabel("Density (g/cm^3)")

elseif 1 == 1

disp("The density of " + element + " is " + density)

else

disp("Never executed")

end

For Loops

Instructions are in the task pane to the left. Complete and submit each task one at a time.

This code loads the data

load datafile

density = data(:,1);

for idx = 1:length(density)

hold on

plot(idx,density(idx),"\*")

hold off

pause(0.2)

end