

```
% Introduction to MatLab
% Script file is a *.m file
% Comments are entered using '%'

clc % clears command window
clear variables % clears workspace variables
close all % closes open figures

% scalar assignment
a = 1; % ; suppresses printing the result in command window

% vector assignment
b = [1;2;3];
c = [4 5 6];
d = zeros(3,2);
e = ones(2,3);
f = 1:1:10;

% get size of a matrix
g = size(d);

% concatenation (join matrices of permissible size)
h = [b';c];

% access element of a matrix
i = h(2,2);

% matrix calculations are done using normal calculator symbols
j = h * b;

% element-wise calculations are done using . followed by operation symbol
k = h.^2;

% if-else construct
if i < 5
    disp([num2str(i) ' is less than 5']);
elseif i == 5
    disp([num2str(i) ' is equal to 5']);
else
    disp([num2str(i) ' is greater than 5']);
end

% for iterations
for m = 1:10
    disp(m);
    if m >= 5
        break
    end
end

% while loop
n = 0;
p = 10;
while n < 5 || p > 5
    disp([n p]);
```

```
n = n + 1; % required to stop while loop in finite time
p = p - 2; % required to stop while loop in finite time
end

% : operation (operation over entire row or column)
h(1,:) = h(1,:) + 3;

% user-defined function
q = example_fun(b);

% Use function name as an input by prefixing it with @
fun = @example_fun;
r = fun(b);

% plot
x = 1:10;
y = x.^0.5 + 1;
plot(x,y); hold on;
plot(x,x);
legend('sqrt x vs x', 'x vs x')

% use help <function> to get info about the function

% define global variables to assign system constants which are transferred
% to functions

% save variables for later use
save mydata.mat x y
clear x y
% load saved data
load mydata.mat x
x

% time to run your code
t_start = cputime;
u = h * h';
v = inv(u^2);
t_end = cputime;
t = t_end - t_start;
```