ic Operations

lementary operations

```
> 5+6          > 3-2        > 5*8        > 1/2        > 2^6
ans = 11       ans = 1      ans = 40     ans = 0.500  ans = 64
```

ogical operations

```
> 1==2         > 1 ~= 2 (not)   > 1 && 0 (and)   > 1 || 0 (or)   > xor(1,0)
ans = 0 false  ans = 1 true     ans = 0          ans = 1          ans = 1
```

ariable

```
> a = 3   if we do a=3; it won't print
> b = 'hi';
> c = (3>=1)
c = 1
> disp(a)
> disp(sprintf('2 decimals: %.0.2f', a))   } printing
```

no. of decimal places

Vectors & Matrices

```
A = [1 2; 3 4; 5 6]

  1 2
  3 4
  5 6        ← Matrix 3 x 2

> V = [1; 2; 3]   or even V = [1 2 3]
V = 1 2 3   column/row
  1
  2
  3          ← Vector (Only has one ~~row~~)

> V = 1 : 0.1 : 2

1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2

> V = 1:6
1 2 3 4 5 6
```

1st element

```
> size(A)        > size(A,1)
ans =            ans =
  3 2              3
```

```
> ones(2,3)
ans = 1 1 1
      1 1 1

> zeros(1,3)
ans = 0 0 0

> length(V)
ans = 4
```

```
> rand(1,3)        random numbers
ans = 0.91477  0.14359  0.84560
> randn(1,3)   - will contain negative number
                 gaussian
```

```
> eye(3)
ans = 1 0 0
      0 1 0
      0 0 1
```

copying elements

```
V = [1 2 3 4]
> b = V(1:3)
b = 1 2 3
```

~~b = V(1:3)~~
~~b = 1 2~~

Moving Data Around

- Current directory
  pwd
- Change directory
  cd
- List elements in dir
  ls
- Loading data
  load featurex.dat
- Displaying variables in memory
  who use whos for more detail
- Removing variable from memory
  clear featuresX use clear to clear all variables
- Saving data to memory
  save hello.mat v; % binary form
  save hello.txt v -ascii % save as text

Handling Matrix

```
> A = [1 2; 3 4; 5 6]
A =  1 2
     3 4
     5 6
```

```
> A(3,2)
ans = 6
```

```
> A(2,:)
ans = 3 4
```

```
> A(:,2)
ans = 2
      4
      6
```

Get all elements of row 1 & 3

```
> A([1 3],:)
     1 2
     5 6
```

- Appending

```
> A = [A, [100; 101; 102]]
A =  1 2 100
     3 4 101
     5 6 102
```

You can even append two matrices

C = [A B] or [A; B]

(A) (B)

(A)
(B)

- Matrix to vector

```
> A(:)
ans = 1
      3
      5
      2
      4
      6
      100
      101
      102
```

Computing on Data

```
A = [1 2; 3 4; 5 6]

A =  1 2
     3 4
     5 6
```

```
>V = [1; 2; 3]
V =  1
     2
     3
```

```
>B = [11 12; 13 14; 15 16]
B =  11 12
     13 14
     15 16
```

```
>C = [11, 22]
C =  11 22
```

A*C - normal matrix multiplication

```
ans =  5  5
      11 11
      17 17
```

A.*B - elementary multiplication

```
ans =  11 24
       39 56
       75 96
```

1×11 2×12
3×13 4×14
5×15 6×16

A.^2 - element wise squaring

```
ans =  1  4
       9 16
      25 36
```

1./V - 1/V for every element of V

```
ans =  1.0000
       0.5000
       0.3333
```

Elementary division

log(V) - elementary logarithm

```
ans =  0.0000
       0.0931
       1.09861
```

exp(x) - e^x for every element

-V

```
ans = -1
      -2
      -3
```

abs(V) - absolute value

V + ones(length(V), 1) or V+1 - incrementing by 1

```
ans =  2
       3
       4
```

```
>a = [1 15 2 0.5]
>[val ind] = max(a)
val = 15
ind = 2
```

To find maximum element

A' - transpose

```
ans =  1 3 5
       2 4 6
```

```
>a < 3
ans =  1 0 1 1
```

```
>find(a<3)
ans =  1 3 4
```

indexes

```
>[r, c] = find(A>=7)
```

↑ for matrices

adds up elements
sum(a)

product of all elements
prod(a)

Round off down
floor(a)

Round off up
ceil(a)

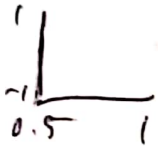
flipud(A) - mirror image

pinv(A) - invert matrix

Plotting Data

- plot(x,y);

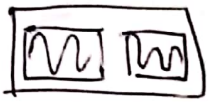
- plot(x1, y1);
  hold on;
  plot(x2, y2, 'r') — red colour
  — To make 2 more graphs appear in 1 plot

- xlabel('time');
  ylabel('value');
  legend('sin', 'cos');
  title('my plot');
  axis([0.5 1 -1 1]); — changing axis
  

  — attributes

- print -dpng 'myPlot.png' — Save plot in png
- close — close plot

- figure(1): plot(t, y1);
  figure(2): plot(t, y2); — getting 2 plot windows

- subplot(1,2,1); — (no. of rows, no. of columns, index)
  plot(t, y1);
  subplot(1,2,2);
  plot(t, y2);
  

- imagesc(A) — colour mapping matrix
  imagesc(A), colorbar, colormap gray;

~~fprintf("Name: %d\n")~~

fprintf('Age: %d\n', age);

Control Statements

- for

```
for i = 1:10,
    v(i) = 2^i;
end;
```

- while

```
while i<=5,
    v(i) = 100;
    i = i+1;
end;
```

- if

```
if i==6,
    a=0;
end;
```

- ~~Nested~~ if, elseif, else

```
if v(1)==1,
    a=0;
elseif v(1)==2,
    a=1;
else
    a=2;
end;
```

- function

- function y = squarenum(x)

```
y = x^2;
```

- function [y1,y2] = sqtcube(x)

```
y1 = x^2;
y2 = x^3;
```