Matlab

Operators
Float and Integer
Text in String and Character Arrays
Conditional statements

Arimatic Operator

Symbol	Role
+	Addition
+	Unary plus
-	Subtraction
-	Unary minus
.*	Element-wise multiplication
*	Matrix multiplication
./	Element-wise right division
/	Matrix right division
.\	Element-wise left division
\	Matrix left division (also known as *backslash*)
.^	Element-wise power
^	Matrix power
.'	Transpose
'	Complex conjugate transpose

Logical Operators

Symbol	Role
&	Find logical AND
|	Find logical OR
&&	Find logical AND (with short-circuiting)
||	Find logical OR (with short-circuiting)
~	Find logical NOT

```
a = [true|true false|false false|true true|false]
```

```
a =

  1×4 logical array

   1   0   1   1
```

```
a = [true&true false&false false&true true&false]
```

```
a =

  1×4 logical array

   1   0   0   0
```

Relational Operators

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

```
fhandle = @(x,y) x.^2 + y.^2;
fhandle(2,3)
```

```
ans =

    13
```

```
A = [7 1 7 7 4];
B = [7 0 4 4 0];
C = intersect(A,B)
```

```
C =

     4     7
```

```
A = [7 1 7 7 4];
B = [7 0 4 4 0];
C = union(A,B)
```

```
C =

     0     1     4     7
```

```
x = 25.783;
whos x
```

```
  Name        Size                Bytes  Class     Attributes

  x           1x1                     8  double
```

```
isfloat(x)
```

```
ans =

  logical

   1
```

```
x = 4/3
```

```
x =

    1.3333
```

```
format long
x
```

```
x =

   1.333333333333333
```

Example 1 — Round-Off or What You Get Is Not What You Expect

```
e = 1 - 3*(4/3 - 1)
```

```
e =

    2.2204e-16
```

```
a = 0.0;
for i = 1:10
   a = a + 0.1;
end
a == 1
```

```
ans =

  logical

   0
```

```
b = 1e-16 + 1 - 1e-16;
c = 1e-16 - 1e-16 + 1;
b == c
```

```
ans =

  logical

   0
```

Integer

Class	Range of Values	Conversion Functio
Signed 8-bit integer	-2^7 to 2^7-1	`int8`
Signed 16-bit integer	-2^{15} to $2^{15}-1$	`int16`
Signed 32-bit integer	-2^{31} to $2^{31}-1$	`int32`
Signed 64-bit integer	-2^{63} to $2^{63}-1$	`int64`
Unsigned 8-bit integer	0 to 2^8-1	`uint8`
Unsigned 16-bit integer	0 to $2^{16}-1$	`uint16`
Unsigned 32-bit integer	0 to $2^{32}-1$	`uint32`
Unsigned 64-bit integer	0 to $2^{64}-1$	`uint64`

```
x = 325.499;
int16(x)
```

```
ans =

  int16

   325
```

```
x = x + .001;
int16(x)
```

```
ans =

  int16

   326
```

```
street1 = imread('street1.jpg'); % Load image data
street2 = imread('street2.jpg');
whos street1 street2
```

```
Name              Size                    Bytes  Class     Attributes

street1       480x640x3                 921600  uint8
street2       480x640x3                 921600  uint8
```

```
cla;
image(street1); % Display image
axis equal
axis off
```

```
image(street2); % Display image
axis equal
axis off
```

```
duller = 0.5 * street2; % Scale image with a double
constant but create an integer
whos duller
```

```
  Name           Size                     Bytes  Class     Attributes

  duller      480x640x3                  921600  uint8
```

```
subplot(1,2,1);
image(street2);
axis off equal tight
title('Original');  % Display image

subplot(1,2,2);
image(duller);
axis off equal tight
title('Duller');     % Display image
```

Original

Duller

```
combined = street1 + duller; % Add |uint8| images
subplot(1,1,1)
cla;
image(combined); % Display image
title('Combined');
axis equal
axis off
```

Data in Character Arrays

```
>> t = "Hello, world"

t =

    "Hello, world"

>> q = "Something ""quoted"" and something else."

q =

    "Something "quoted" and something else."
```

```
>> whos t
  Name      Size            Bytes  Class     Attributes

  t         1x1               166  string
```

```
f = 71;
c = (f-32)/1.8;
tempText = "Temperature is " + c + "C"
```

```
tempText =

    "Temperature is 21.6667C"
```

A = ["a","bb","ccc"; "dddd","eeeeee","fffffff"]

```
A =

  2×3 string array

    "a"       "bb"        "ccc"
    "dddd"    "eeeeee"    "fffffff"
```

```
strlength(A)
```

```
ans =

     1     2     3
     4     6     7
```

```
seq = 'GCTAGAATCC';
whos seq
```

```
  Name        Size                Bytes  Class     Attributes

  seq         1x10                   20  char
```

```
seq(4)
```

```
ans =

    'A'
```

```
seq2 = [seq 'ATTAGAAACC']
```

```
seq2 =

    'GCTAGAATCCATTAGAAACC'
```

MATLAB Language Syntax

`if, elseif, else`	Execute statements if condition is true
`switch, case, otherwise`	Execute one of several groups of statements
`for`	`for` loop to repeat specified number of times
`while`	`while` loop to repeat when condition is true
`try, catch`	Execute statements and catch resulting errors
`break`	Terminate execution of for or while loop
`return`	Return control to invoking script or function
`continue`	Pass control to next iteration of `for` or `while` loop
`pause`	Stop MATLAB execution temporarily
`parfor`	Parallel for loop
`end`	Terminate block of code or indicate last array index

Conditional Statements

```
% Generate a random number
a = randi(100, 1)

% If it is even, divide by 2
if rem(a, 2) == 0
    disp('a is even')
    b = a/2;
end
```

```
a =

    91
```

```
a =

    92

a is even
```

```
a = randi(100, 1)

if a < 30
    disp('small')
elseif a < 80
    disp('medium')
else
    disp('large')
end
```

```
a =

    10

small
```

```
a =

    55

medium
```

```
yourNumber = input('Enter a number: ');

if yourNumber < 0
    disp('Negative')
elseif yourNumber > 0
    disp('Positive')
else
    disp('Zero')
end
```

```
Enter a number: 4
Positive
```

```
Enter a number: -4
Negative
```

Language Fundamentals

Syntax, array indexing and manipulation, data types, operators

MATLAB is an abbreviation for "matrix laboratory." While other programming languages usually work with numbe include basic operations, such as creating variables, array indexing, arithmetic, and data types.

Entering Commands
Build and run MATLAB statements

Matrices and Arrays
Array creation, combining, reshaping, rearranging, and indexing

Data Types
Numeric arrays, characters and strings, tables, structures, and cell arrays; data type conversion

Operators and Elementary Operations
Arithmetic, relational, and logical operators, special characters, rounding, set functions

Loops and Conditional Statements
Control flow and branching using keywords, such as `if`, `for`, and `while`