# 02-Variables

Text: Chapter 1.5-1.7

ECEGR 101
Engineering Problem Solving with Matlab
Professor Henry Louie

## Variables in MATLAB

- A MATLAB variable is a region of memory containing a value (or a set of values) that is known by a user-specified name.
- The contents of the variable may be used or modified at any time by including its name in an appropriate MATLAB command.
- Assignment operator: x = -10.
- Variables can be scalar or vector (matrix).

## Exercise

Suppose that u = 1 and v = 3. Evaluate the following expressions using MATLAB.

a) $\frac{4u}{3v}$

c) $\frac{2v^{-2}}{(u+v)^2}$

d) $\frac{v^3}{v^3 - u^3}$

e) $\frac{4}{3}\pi v^2$



## Exercise

```
>> u = 1;
>> v = 3;
>>
>> 4*u/(3*v)
ans =
        0.44
>>
>> (2*v^(-2))/(u+v)^2
ans =
        0.01
```

$\frac{4u}{3v}$

$\frac{2v^{-2}}{(u+v)^2}$

## Exercise

```
>> (v^3)/((v^3)-(u^3))
ans =
        1.04
>>
>> (4/3)*pi*v^2
ans =
       37.70
```

$$\frac{v^3}{v^3 - u^3}$$

$$\frac{4}{3}\pi v^2$$



## Exercise

Solve the following problems with paper and pencil. When you are finished, execute the code in the Command Window to verify your answers.

Part 1: What do the variables x and y contain after executing the following lines?

```
>> x = 4
>> y = x
>> x = 6
```

## Exercise

Solve the following problems with paper and pencil. When you are finished, execute the code in the Command Window to verify your answers.

Part 2: What does x contain after executing the following lines?

```
>> x = 2
>> x^2
```



## Exercise

Solve the following problems with paper and pencil. When you are finished, execute the code in the Command Window to verify your answers.

Part 3: What does x contain after executing the following lines?

```
>> x = 2
>> x = 2*x + 5
>> x = (x-3)/3
```

## Exercise

Part 1:
>> x = 4
>> y = x
>> x = 6
Answer: x = 6,
y = 4.

Part 2:
>> x = 2
>> x^2
Answer: x = 2.

Part 3:
>> x = 2
>> x = 2*x + 5
>> x = (x-3)/3
Answer: x = 2.



## Variable Names

- Restrictions:
  - Must be less than 63 characters long
  - Can contain letters, digits, and "_"
  - Must begin with a letter
  - Case sensitive
- Recommendations:
  - Do not use the names of built-in functions as variable names!
  - Give your variables descriptive and easy-to-remember names so your programs are easier to understand
    - i.e currency exchange rate could be given the name currency_exchange

## Exercise

Which of the following are legitimate variable names in MATLAB?

a) 3vars
b) global
c) help
d) My_var
e) sin
f) X + Y
g) _input
h) input
i) tax-rate
j) example1.1
k) example1  1



## Exercise

a) 3vars
   No, variable cannot start with a digit
b) global
   No, this is a keyword in MATLAB
c) help
   No, this is a function name
d) My_var
   Yes, this could be a variable name
e) sin
   Yes, but it is not recommend to use a function name as a variable name
f) X + Y
   No, variable name cannot have a "+" in it

## Exercise

g) _input
No, variable name cannot start with a "_"
h) input
No, this is a function name
i) tax-rate
Variable name cannot have "-" in it
j) example1.1
No, "." is not allowed in a variable name
k) example1_1
Yes, this could be a variable name



## Types of Variables in MATLAB

```
logical
char
int8
uint8
int16
uint16
int32
uint32
int64
uint64
single
double
```

The type of a variable is referred to as its class.

You can see the list of variables in the memory using the who and whos commands or by looking at the workspace window.

Useful commands:

- clear x, clear all

# Variable Types

- Selection of variable types is useful when managing large amounts of data

```
>> x=4

x =

     4

>> y=int8(x)

y =

    4

>> whos
  Name        Size              Bytes  Class     Attributes

  x           1x1                   8  double
  y           1x1                   1  int8
```

y uses one eighth the memory as x



# Variable Class in MATLAB

CLASS Create object or return object class.
C = CLASS(OBJ) returns the class of the object OBJ.
Possibilities are:

| | |
|---|---|
| double | -- Double precision floating point number array (this is the traditional MATLAB matrix or array) |
| single | -- Single precision floating point number array |
| logical | -- Logical array |
| char | -- Character array |
| cell | -- Cell array |
| struct | -- Structure array |
| function_handle | -- Function Handle |
| int8 | -- 8-bit signed integer array |
| uint8 | -- 8-bit unsigned integer array |
| int16 | -- 16-bit signed integer array |
| uint16 | -- 16-bit unsigned integer array |
| int32 | -- 32-bit signed integer array |
| uint32 | -- 32-bit unsigned integer array |
| int64 | -- 64-bit signed integer array |
| uint64 | -- 64-bit unsigned integer array |

## Example

```
>> z = 10*pi
z =
   31.4159
>> class(z)
ans =
double
>> whos
  Name      Size            Bytes  Class     Attributes
  ans       1x6                12  char
  z         1x1                 8  double
```



## Exercise

Assuming x is a defined variable in the MATLAB workspace, are the following scalar operations valid in MATLAB? If not, why not?

Hint: Remember that "=" is the assignment operator in MATLAB.

| | |
|---|---|
| Part A | >> x + 2 = 4 |
| Part B | >> x = 4 – 2 |
| Part C | >> x = 4 – (-2) |
| Part D | >> 5 = x |
| Part E | >> 5 + 3 = 7 + i |

## Exercise

Part A

```
>> x + 2 = 4
??? x + 2 = 4
        |
```

```
Part A   >> x + 2 = 4
Part B   >> x = 4 – 2
Part C   >> x = 4 – (–2)
Part D   >> 5 = x
Part E   >> 5 + 3 = 7 + i
```

Error: The expression to the left of the equals sign is not a valid target for an assignment.

(Only a variable can be on the left side of the assignment; any calculations should be put on the right side of the equal sign.)



## Exercise

Part B

```
>> x = 4 – 2
```

This is a valid statement.

```
Part A   >> x + 2 = 4
Part B   >> x = 4 – 2
Part C   >> x = 4 – (–2)
Part D   >> 5 = x
Part E   >> 5 + 3 = 7 + i
```

Part C

```
>> x = 4 – (–2)
```

This is a valid statement.

## Exercise

Part D

```
>> 5 = x
>> 5=x
??? 5=x
     |
```

Error: The expression to the left of the equals sign is not a valid target for an assignment.

(same as in step a)

```
Part A   >> x + 2 = 4
Part B   >> x = 4 – 2
Part C   >> x = 4 – (–2)
Part D   >> 5 = x
Part E   >> 5 + 3 = 7 + i
```



## Exercise

Part E

```
>> 5 + 3 = 7 + i
>> 5 + 3 = 7 + i
??? 5 + 3 = 7 + i
         |
```

Error: The expression to the left of the equals sign is not a valid target for an assignment.

(same as above)

```
Part A   >> x + 2 = 4
Part B   >> x = 4 – 2
Part C   >> x = 4 – (–2)
Part D   >> 5 = x
Part E   >> 5 + 3 = 7 + i
```

## Exercise

Money saved in a bank will compound with the interest according to the following formula:

$$newBalance = oldBalance \times \left(1 + \frac{interestRate}{100}\right)$$

The interest rate is given in percentage.

a) In the MATLAB Command Window, set oldBalance to 1,000 and interestRate to 6 percent. Calculate newBalance. This is the account balance after one year.



## Exercise

b) Store the value of newBalance in the variable oldBalance and execute the command again to calculate newBalance. Store the value of newBalance in oldBalance again. Describe what is now stored in oldBalance.

c) Use the up arrow in the Command Window to calculate the balance after eight years.

## Exercise

a)

```
>> oldBalance = 1000;
>> interestRate = 6;
>> newBalance = oldBalance*(1+interestRate/100)
newBalance =
      1060.00
```



## Exercise

b)

```
>> oldBalance = newBalance;
>> newBalance = oldBalance*(1+interestRate/100)
newBalance =
      1123.60
```

(newBalance now holds the account balance after two years.)

Use the up arrow in the Command Window to calculate the balance after eight years.

## Exercise

```
>> oldBalance = newBalance;
>> newBalance = oldBalance*(1+interestRate/100)
newBalance =
      1191.02
>> oldBalance = newBalance;
>> newBalance = oldBalance*(1+interestRate/100)
newBalance =
      1262.48
>> oldBalance = newBalance;
>> newBalance = oldBalance*(1+interestRate/100)
newBalance =
      1338.23
```



## Exercise

```
>> oldBalance = newBalance;
>> newBalance = oldBalance*(1+interestRate/100)
newBalance =
      1418.52
>> oldBalance = newBalance;
>> newBalance = oldBalance*(1+interestRate/100)
newBalance =
      1503.63
>> oldBalance = newBalance;
>> newBalance = oldBalance*(1+interestRate/100)
newBalance =
      1593.85
```