Creating MATLAB variables:

MATLAB variables are created with an assignment statement. The syntax of variable assignment is

variable name = a value (or an expression)

For example,

X = expression

where expression is a combination of numerical values, mathematical operators, variables, and function calls. On other words, expression can involve:

1. manual entry
2. built-in functions
3. user-defined functions

Overwriting variable:

Once a variable has been created, it can be reassigned. In addition, if you do not wish to see the intermediate results, you can suppress the numerical output by putting a semicolon (;) at the end of the line. Then the sequence of commands looks like this:

```
>> t = 5;

>> t = t+1

t =

6
```

Error messages:

If we enter an expression incorrectly, MATLAB will return an error message. For example, in the following, we left out the multiplication sign, *, in the following expression :

```
>> x = 10;
>> 5x
??? 5x
   |
Error: Unexpected MATLAB expression.
```

Making corrections:

To make corrections, we can, of course retype the expressions. But if the expression is lengthy, we make more mistakes by typing a second time. A previously typed command can be recalled with the up-arrow key ↑. When the command is displayed at the command prompt, it can be modified if needed and executed.

Controlling the hierarchy of operations or precedence:

Let's consider the previous arithmetic operation, but now we will include parentheses. For

example, 1 + 2 × 3 will become (1 + 2) × 3

```
>> (1+2)*3

ans =

     9
```

and, from previous example:

```
>> 1+2*3

ans =

     7
```

By adding parentheses, these two expressions give different results: 9 and 7.

The order in which MATLAB performs arithmetic operations is exactly that taught in high school algebra courses. Exponentiations are done first, followed by multiplications and divisions, and finally by additions and subtractions. However, the standard order of precedence of arithmetic operations can be changed by inserting parentheses. For example, the result of 1+2×3 is quite different than the similar expression with parentheses (1+2) ×3. The results are 7 and 9 respectively. Parentheses can always be used to overrule priority, and their use is recommended in some complex expressions to avoid ambiguity.

Therefore, to make the evaluation of expressions unambiguous, MATLAB has established a series of rules. The order in which the arithmetic operations are evaluated.

Precedence	Mathematical operations
First	The contents of all parentheses are evaluated first, starting from the innermost parentheses and working outward.
Second	All exponentials are evaluated, working from left to right
Third	All multiplications and divisions are evaluated, working from left to right
Fourth	All additions and subtractions are evaluated, starting from left to right

most computer programs. For operators of equal precedence, evaluation is from left to right.

another example:

$$\frac{1}{2+3^2}+\frac{4}{5}\times\frac{6}{7}$$

In MATLAB, it becomes

```
>> 1/(2+3^2)+4/5*6/7

ans =

0.7766
```

or, if parentheses are missing,

```
>> 1/2+3^2+4/5*6/7

ans =

10.1857
```

So here what we get: two different results. Therefore, we want to emphasize the importance of precedence rule in order to avoid ambiguity.

Managing the workspace:

The contents of the workspace persist between the executions of separate commands. Therefore, it is possible for the results of one problem to have an effect on the next one. To avoid this possibility, it is a good idea to issue a clear command at the start of each new independent calculation.

```
>> clear
```

The command clear or clear all removes all variables from the workspace. This frees up system memory.

```
>> who
```

while, who will give more details which include size, space allocation, and class of the variables.

Entering multiple statements per line:

It is possible to enter multiple statements per line. Use commas (,) or semicolons (;) to enter more than one statement at once. Commas (,) allow multiple statements per line without suppressing output.

```
>> a=7; b=cos(a), c=cosh(a)

b =

0.6570

c =

548.3170
```

Miscellaneous commands:

Here are few additional useful commands:

1. To clear the Command Window, type clc.
2. To abort a MATLAB computation, type ctrl-c.
3. To continue a line, type . . .