Unit No : I
Module Name : Display Formats, Elementary Math Built-In Functions, Defining Scalar Variables: The Assignment Operator, Rules About Variable Names and Predefined Variables
Module No : 03

Display Formats

The user can control the format in which MATLAB displays output on the screen. The output format is fixed-point with 4 decimal digits (called short), which is the default format for numerical values. The format can be changed with the format command. Once the format command is entered, all the output that follows is displayed in the specified format.

Table 1: Display formats

Command	Description	Example
format short	Fixed-point with 4 decimal digits for: $0.001 \leq number \leq 1000$ Otherwise display format short e.	>> 44/7 ans = 6.2857
format long	Fixed-point with 14 decimal digits for: $0.001 \leq number \leq 100$ Otherwise display format long e.	>> 44/7 ans = 6.285714285714286
format short e	Scientific notation with 4 decimal digits.	>> 44/7 ans = 6.2857e+00
format long e	Scientific notation with 15 decimal digits.	>> 44/7 ans = 6.285714285714286e+00
format short g	Best of 5-digit fixed or floating point.	>> 44/7 ans = 6.2857
format long g	Best of 15-digit fixed or floating point.	>> 44/7 ans = 6.28571428571429
format bank	Two decimal digits.	>> 44/7 ans = 6.29
format compact	Eliminates empty lines to allow more lines with information displayed on the screen.	
format loose	Adds empty lines (opposite of compact).	

Elementary math built-In Functions

In addition to basic arithmetic operations, expressions in MATLAB can include functions. MATLAB has a very large library of built-in functions. A function has a name and

an argument in parentheses. For example, the function that calculates the square root of a number is sqrt(x). Its name is sqrt, and the argument is x. When the function is used, the argument can be a number, a variable that has been assigned a numerical value, or a computable expression that can be made up of numbers and/or variables. Functions can also be included in arguments, as well as in expressions.

Function	**Description**	**Example**
`sqrt(x)`	Square root	>> sqrt(121) ans = 11.00
`nthroot(x,n)`	Real *n*th root or a real number *x*.(If *x* is negative *n* must be an odd integer.)	>> nthroot(128,5) ans = 2.64
`exp(x)`	Exponential	exp(8) ans = 2.9810e+03
`abs(x)`	Absolute value	>> abs(-45) ans = 45
`log(x)`	Natural logarithm. Base e logarithm (ln).	>> log(2000) ans = 7.6009
`log10(x)`	Base 10 logarithm	>> log10(10000) ans = 4.0000
`factorial(x)`	The factorial function *x*! (*x* must be a positive integer.)	>> factorial(7) ans = 5040

Trignometric math functions:

Function	**Description**	**Example**
`sin(x)` `sind(x)`	Sine of angle *x* (*x* in radians). Sine of angle *x* (*x* in degrees).	>> sin(pi/4) ans = 0.7071
`cos(x)` `cosd(x)`	Cosine of angle *x* (*x* in radians). Cosine of angle *x* (*x* in degrees).	>> cos(pi/6) ans = 0.8660
`tan(x)` `tand(x)`	Tangent of angle *x* (*x* in radians). Tangent of angle *x* (*x* in degrees).	>> tan(pi/3) ans = 1.7321

`cot(x)` `cotd(x)`	Cotangent of angle *x* (*x* in radians). Cotangent of angle *x* (*x* in radians).	>> cot(pi/3) ans = 0.5774
Function	**Description**	**Example**
`round(x)`	Round to the nearest integer.	>> round(12/5) ans = 2
`fix(x)`	Round towards zero	>> fix(13/7) ans = 1
`ceil(x)`	Round towards infinity	>> ceil(13/7) ans = 2
`floor(x)`	Round towards minus infinity.	>> floor(-4/7) ans = -1
`rem(x,y)`	Returns the remainder after *x* is divided by *y*.	>> rem(18,4) ans = 2
`sign(x)`	Signum function. Returns 1 if x>0 , –1 if x<0 , and 0 if x=0.	>> sign(6) ans = 1

Defining scalar variables:

A variable is a name made of a letter or a combination of several letters (and digits) that is assigned a numerical value. Once a variable is assigned a numerical value, it can be used in mathematical expressions, in functions, and in any MATLAB statements and commands. A variable is actually a name of a memory location. When a new variable is defined, MATLAB allocates an appropriate memory space where the variable's assignment is stored. When the variable is used the stored data is used. If the variable is assigned a new value the content of the memory location is replaced.

The Assignment Operator

In MATLAB the = sign is called the assignment operator. The assignment operator assigns a value to a variable.

Variable_name = A numerical value, or a computable expression

The left-hand side of the assignment operator can include only one variable name. The right-hand side can be a number, or a computable expression that can include numbers and/or variables that were previously assigned numerical values. When the **Enter** key is pressed the numerical value of the right-hand side is assigned to the variable, and MATLAB displays the variable and its assigned value in the next two lines.
The following shows how the assignment operator works:

The number 13 is assigned to the variable 13.

```
>> a=13
a =
   13
```

MATLAB displays the variable and it's assigned value

Rules About Variable Names and Predefined Variables:

A variable can be named according to the following rules:
• Must begin with a letter.
• Can be up to 63 (in MATLAB 7) characters long (31 characters in MATLAB 6.0).
• Can contain letters, digits, and the underscore character.
• Cannot contain punctuation characters (e.g. period, comma, semicolon).
• MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. For example, AA, Aa, aA, and aa are the names of four different variables.
• No spaces are allowed between characters (use the underscore where a space is desired).
• Avoid using the names of a built-in function for a variable (i.e. avoid using: cos, sin, exp, sqrt, etc.). Once a function name is used to define a variable, the function cannot be used.

Predefined Variables and keywords

There are seventeen words, called keywords, that are reserved by MATLAB for various purposes, and cannot be used as variable names. These words are:

break case catch continue else elseif end for function global if otherwise persistent return switch try while