# MATLAB Reference: MATLAB Help, Basic & Complex Calculations

## Help Commands

| Command & Syntax | Purpose |
|---|---|
| `help function_name` | Displays help text for the specified function_name |
| `doc` | Opens the MATLAB Help Browser |
| `helpwin topic` | Displays help information for the specified topic in a new window |

## Commands and Operators

| Operation | Expression | MATLAB Command |
|---|---|---|
| Addition | 3 + 3 | `3 + 3` |
| Subtraction | 3 − 3 | `3 - 3` |
| Negation | − 1 | `-1` |
| Multiplication | 3 x 3 | `3 * 3` |
| Division | 15 ÷ 3 | `15 / 3` |
| Exponentiation | $3^2$ | `3 ^ 2` |

## Order of Operations

| Operation | Operator |
|---|---|
| Parentheses | () |
| Transpose and exponentiation | ' or ^ |
| Negation | − |
| Multiplication or division | * or / |
| Addition or subtraction | + or − |
| Comparison | =, <, >, <=, or >= |

## MATLAB Variable Naming

| Rule for Naming Variables | Example |
|---|---|
| Must start with a letter | Correct: `time1`<br>Incorrect: `1time` |
| Must contain letters, numbers, or underscores | Correct: `travel_time`<br>Incorrect: `travel-time` |
| Are case sensitive | `Time ≠ time` |
| Must not be MATLAB keywords<br>*(type `iskeyword` in Command Window to view key words)* | Such as `break, if` |
| Do not use built-in function names | Such as `clc, sqrt, min, max` |

### Commands Related to Variables Names

| Command & Syntax | Purpose |
|---|---|
| `who` | List existing variables in the current workspace. |
| `clear` | Clear all variables from the workspace. |
| `clear variable_name` | Clear a single variable from the workspace |
| `help possible_variable_name` | Determine if `possible_variable_name` is a built-in function. |
| `exist possible_variable_name` | Check if variable names or functions are defined<br>• Returns “0” if `possible_variable_name` does not exist<br>• Returns “5” if it is a built-in function |
| `which possible_variable_name` | Identifies if `possible_variable_name` exists or is a built-in function. |

## Data Types & Definitions

**Array:** The fundamental form MATLAB uses to store and manipulate data. It is a list of data entries stored in rows and columns.

| Term | Definition | Example |
|---|---|---|
| **Scalar** | The most basic array; contains only one entry. | 5 |
| **Vector** | A one-dimensional array, composed of a single row or column of entries | [ 1 2 3 4 5 6] |
| **Matrix** | A two-dimensional array; collection of entries arranged in rows and columns | $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ |

## Vectors & Matrices

| Task | Command & Syntax | Explanation |
|---|---|---|
| **Create row vector** | `x = [0 1 2 3]` or<br>`x = [0, 1, 2, 3]` | In this example, the vector x has the values from 0 to 3 in increments of 1.<br>Values in a vector can be separated with spaces or commas. |
| **Create column vector** | `y = [0; 1; 2; 3]` | In this example, the vector y has the values from 0 to 3 in increments of 1.<br>The semicolon (;) indicates the start of a new row. |
| **Create evenly spaced vectors** | `xy = 1 : 5` | Creates a vector from 1 to 5 with increments of 1:<br>`xy = 1 2 3 4 5` |
| | `xy = 1 : 2 : 11` | Creates a vector from 1 to 11 with an increment of 2:<br>`xy = 1 3 5 7 9 11` |
| | `linspace (x1, x2)` | Generates a row vector of 100 equally spaced points between x1 and x2. |
| | `linspace (x1, x2, N)` | Generates a row vector of N equally spaced points between x1 and x2. |
| **Transpose vectors** | `c = [1 2 3 4]'` | Transposes the row vector c into a column vector. Notice the apostrophe at the end. |
| **Index vectors** | `c(x)`, where c is a vector | Extracts the $x^{th}$ element of the vector c |

| Task | Command & Syntax | Explanation |
|---|---|---|
| **Create matrix** | `z = [1 3; 9 2]` or<br>`z = [1, 3; 9, 2]` | Creates the matrix z = $\begin{bmatrix}1 & 3\\9 & 2\end{bmatrix}$.<br>Semicolons separate each row in the matrix. |
| **Index matrices** | `Z (x, y)`, where z is a matrix | Extracts the element from the matrix z in row x and column y |
| **Select rows or columns of matrices** | `B = z(:, j)` | Extracts the $j^{th}$ column of matrix z into vector B |
| | `B = z(i, :)` | Extracts the $i^{th}$ row of matrix z into vector B |
| **Determine largest element in a vector or matrix** | `max(X)` | When X is a vector: Returns the larges element.<br>When X is a matrix: Returns a row vector containing the maximum element from each column. |
| **Determine largest element and its index** | `[Y, I] = max(X)` | Returns the largest element in the vector (Y) and its index (I). |

## Matrix Multiplication

MATLAB performs **matrix multiplication** by default.

For example:

| **Matrix multiplication** | |
|---|---|
| m1 = $\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$ m2 = $\begin{bmatrix}10 & 20\\30 & 40\end{bmatrix}$ | m1 * m2 = $\begin{matrix}1*10+2*30 & 1*20+2*40\\3*10+4*30 & 3*20+4*40\end{matrix}$ = $\begin{matrix}70 & 100\\150 & 220\end{matrix}$ |
| **Element-by-element multiplication** | |
| m1 = $\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$ m2 = $\begin{bmatrix}10 & 20\\30 & 40\end{bmatrix}$ | m1 .* m2 = $\begin{matrix}1*10+2*20\\3*30+4*40\end{matrix}$ = $\begin{matrix}10 & 40\\90 & 160\end{matrix}$ |

**To perform element-by-element multiplication**, use the dot operator. The dot operator consists of a period before the mathematical operator.

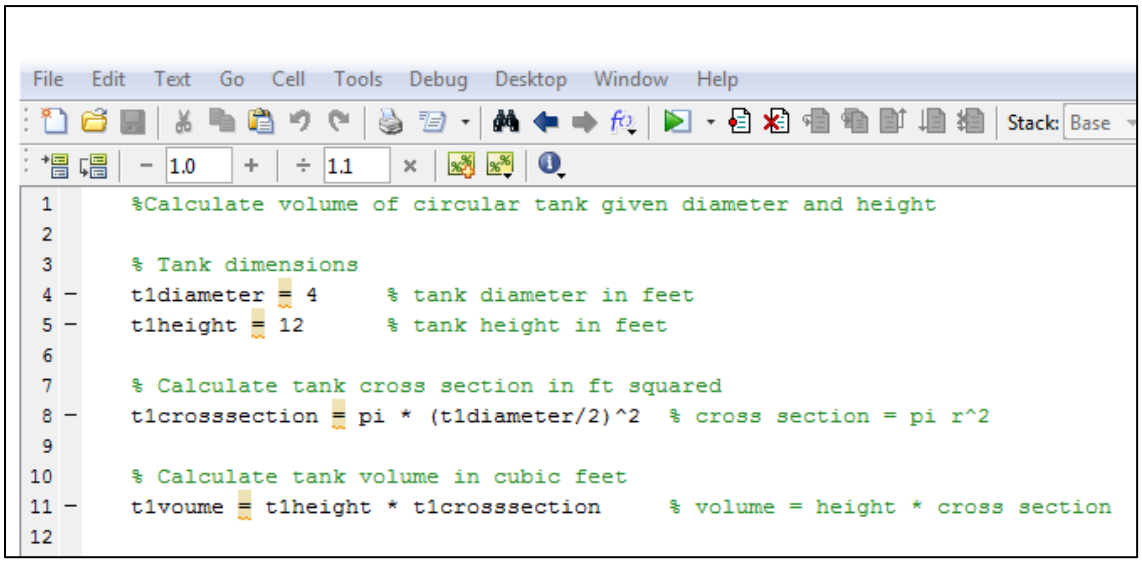
| Operation | Operator | Operator for element-by-element operation |
|---|---|---|
| Multiplication | * | **.*** |
| Division | / | **./** |
| Exponentiation | ^ | **.^** |

## MATLAB Comments

About comments:

- Add comments to MATLAB code using the percent (%) symbol.
- Comments allow others to understand your code, and can refresh your memory when you return to it later.
- Comment lines can appear anywhere in a program file.

**Example of appropriate comments in MATLAB code:**




***Remember: In this course, your MATLAB code must include comments!***

## MATLAB fprintf Command

| Command & Syntax | Purpose |
|---|---|
| `fprintf` | Formats data and displays information to the screen. |

### Special Characters

Special characters are a part of the text in the string. They cannot be entered as ordinary text, and they require a unique character sequence to represent them. Use any of the following character sequences to insert special characters into the output string.

| To Insert a . . . | Use . . . |
|---|---|
| Single quotation mark | ' ' |
| Percent character | %% |
| Backslash | \\ |
| Alarm | \a |
| Backspace | \b |
| Form feed | \f |
| New line | \n |
| Carriage return | \r |
| Horizontal tab | \t |
| Vertical tab | \v |
| Hexadecimal number, N | \xN |
| Octal number, N | \N |

### Conversion Characters

The conversion character specifies the notation of the output. It consists of a single character and appears last in the format specifier. It is the only required field of the format specifier other than the leading % character.

| Specifier | Description |
|---|---|
| c | Single character |
| d | Decimal notation (signed) |
| e | Exponential notation (using a lowercase e as in 3.1415e+00) |
| E | Exponential notation (using an uppercase E as in 3.1415E+00) |
| f | Fixed-point notation |
| g | The more compact of %e or %f. (Insignificant zeros do not print.) |
| G | Same as %g, but using an uppercase G |
| o | Octal notation (unsigned) |
| s | String of characters |

| Specifier | Description |
|---|---|
| u | Decimal notation (unsigned) |
| x | Hexadecimal notation (using lowercase letters a–f) |
| X | Hexadecimal notation (using uppercase letters A–F) |

### Flags

You can control the output using any of these optional flags.

| Character | Description | Example |
|---|---|---|
| A minus sign (-) | Left-justifies the converted argument in its field | %-5.2d |
| A plus sign (+) | Always prints a sign character (+ or -) | %+5.2d |
| Zero (0) | Pad with zeros rather than spaces | %05.2d |
| Digits (field width) | A digit string specifying the minimum number of digits to be printed | %6f |
| Digits (precision) | A digit string including a period (.) specifying the number of digits to be printed to the right of the decimal point | %6.2f |

### Example fprintf Statements