

# MATLAB - DATA TYPES

[http://www.tutorialspoint.com/matlab/matlab\\_data\\_types.htm](http://www.tutorialspoint.com/matlab/matlab_data_types.htm)

Copyright © tutorialspoint.com

MATLAB does not require any type declaration or dimension statements. Whenever MATLAB encounters a new variable name, it creates the variable and allocates appropriate memory space.

If the variable already exists, then MATLAB replaces the original content with new content and allocates new storage space, where necessary.

For example,

```
Total = 42
```

The above statement creates a 1-by-1 matrix named 'Total' and stores the value 42 in it.

## Data Types Available in MATLAB

MATLAB provides 15 fundamental data types. Every data type stores data that is in the form of a matrix or array. The size of this matrix or array is a minimum of 0-by-0 and this can grow up to a matrix or array of any size.

The following table shows the most commonly used data types in MATLAB –

| Data Type       | Description                                                                                                              |
|-----------------|--------------------------------------------------------------------------------------------------------------------------|
| int8            | 8-bit signed integer                                                                                                     |
| uint8           | 8-bit unsigned integer                                                                                                   |
| int16           | 16-bit signed integer                                                                                                    |
| uint16          | 16-bit unsigned integer                                                                                                  |
| int32           | 32-bit signed integer                                                                                                    |
| uint32          | 32-bit unsigned integer                                                                                                  |
| int64           | 64-bit signed integer                                                                                                    |
| uint64          | 64-bit unsigned integer                                                                                                  |
| single          | single precision numerical data                                                                                          |
| double          | double precision numerical data                                                                                          |
| logical         | logical values of 1 or 0, represent true and false respectively                                                          |
| char            | character data <i>strings are stored as vector of characters</i>                                                         |
| cell array      | array of indexed cells, each capable of storing an array of a different dimension and data type                          |
| structure       | C-like structures, each structure having named fields capable of storing an array of a different dimension and data type |
| function handle | pointer to a function                                                                                                    |
| user classes    | objects constructed from a user-defined class                                                                            |
| java classes    | objects constructed from a Java class                                                                                    |

## Example

Create a script file with the following code –

```
str = 'Hello World!'
n = 2345
d = double(n)
un = uint32(789.50)
rn = 5678.92347
c = int32(rn)
```

When the above code is compiled and executed, it produces the following result –

```
str = Hello World!
n = 2345
d = 2345
un = 790
rn = 5678.9
c = 5679
```

## Data Type Conversion

MATLAB provides various functions for converting, a value from one data type to another. The following table shows the data type conversion functions –

| Function       | Purpose                                                            |
|----------------|--------------------------------------------------------------------|
| char           | Convert to character array <i>string</i>                           |
| int2str        | Convert integer data to string                                     |
| mat2str        | Convert matrix to string                                           |
| num2str        | Convert number to string                                           |
| str2double     | Convert string to double-precision value                           |
| str2num        | Convert string to number                                           |
| native2unicode | Convert numeric bytes to Unicode characters                        |
| unicode2native | Convert Unicode characters to numeric bytes                        |
| base2dec       | Convert base N number string to decimal number                     |
| bin2dec        | Convert binary number string to decimal number                     |
| dec2base       | Convert decimal to base N number in string                         |
| dec2bin        | Convert decimal to binary number in string                         |
| dec2hex        | Convert decimal to hexadecimal number in string                    |
| hex2dec        | Convert hexadecimal number string to decimal number                |
| hex2num        | Convert hexadecimal number string to double-precision number       |
| num2hex        | Convert singles and doubles to IEEE hexadecimal strings            |
| cell2mat       | Convert cell array to numeric array                                |
| cell2struct    | Convert cell array to structure array                              |
| cellstr        | Create cell array of strings from character array                  |
| mat2cell       | Convert array to cell array with potentially different sized cells |
| num2cell       | Convert array to cell array with consistently sized cells          |

## Determination of Data Types

MATLAB provides various functions for identifying data type of a variable.

Following table provides the functions for determining the data type of a variable –

| Function           | Purpose                                           |
|--------------------|---------------------------------------------------|
| is                 | Detect state                                      |
| isa                | Determine if input is object of specified class   |
| iscell             | Determine whether input is cell array             |
| iscellstr          | Determine whether input is cell array of strings  |
| ischar             | Determine whether item is character array         |
| isfield            | Determine whether input is structure array field  |
| isfloat            | Determine if input is floating-point array        |
| ishandle           | True for Handle Graphics object handles           |
| isinteger          | Determine if input is integer array               |
| isjava             | Determine if input is Java object                 |
| islogical          | Determine if input is logical array               |
| isnumeric          | Determine if input is numeric array               |
| isObject           | Determine if input is MATLAB object               |
| isreal             | Check if input is real array                      |
| isscalar           | Determine whether input is scalar                 |
| isstr              | Determine whether input is character array        |
| isstruct           | Determine whether input is structure array        |
| isvector           | Determine whether input is vector                 |
| class              | Determine class of object                         |
| validateattributes | Check validity of array                           |
| whos               | List variables in workspace, with sizes and types |

## Example

Create a script file with the following code –

```
x = 3
isinteger(x)
isfloat(x)
isvector(x)
isscalar(x)
isnumeric(x)
```

```

x = 23.54
isinteger(x)
isfloat(x)
isvector(x)
isscalar(x)
isnumeric(x)

x = [1 2 3]
isinteger(x)
isfloat(x)
isvector(x)
isscalar(x)

x = 'Hello'
isinteger(x)
isfloat(x)
isvector(x)
isscalar(x)
isnumeric(x)

```

When you run the file, it produces the following result –

```

x = 3
ans = 0
ans = 1
ans = 1
ans = 1
ans = 1
x = 1177/50
ans = 0
ans = 1
ans = 1
ans = 1
ans = 1
x =
      1      2      3

ans = 0
ans = 1
ans = 1
ans = 0
x = Hello
ans = 0
ans = 0
ans = 1
ans = 0
ans = 0

```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js