

MATLAB Marina: Variables

Student Learning Objectives

After completing this module, one should:

1. Be able to explain the purpose of storing results in variables.
2. Be able to create and use MATLAB variables.

Terms

variable, assignment, data type, weakly typed

MATLAB Functions, Keywords, and Operators

=, ans, clear, clc, who, whos, real, imag, abs, angle

MATLAB Variables

Variables are names referring to data stored in a computer's memory. Variables allow one to store values and refer to the values using meaningful names. MATLAB variables must start with a letter and can consist of up to 31 characters (letters, digits, underscores). MATLAB variable names are case sensitive.

The variable data type indicates what of data is being stored. MATLAB is a weakly typed language. Variable names and types do not need to be defined before they are used. A variable's type (single, double, uint8, int64, logical etc.) can be changed and operations can be performed on data of different types. Unless specifically defined, all MATLAB numeric data is treated as a double precision real number (double). In a strongly typed language such as Python and Java, variable types must be defined and types must match when performing operations.

MATLAB being a scripted language and weakly typed makes it a good choice for quickly developing solutions to problems. Weakly typing also generally allows for easier code reuse. Weakly typed languages do have drawbacks. Unexpected user input and operations on mixed types can produce unexpected or erroneous results.

MATLAB variables are allocated in memory when they are first used in an expression. The equal sign (=) allows one to assign the result of the right-hand side of an expression to a variable on the left hand side of the expression. Once a variable has a value, it can then be used in other expressions. Variables can be redefined by assigning a new value or set of values to them. Figure 1 shows an example of creating two variables named `len` and `ht` and then computing the area of rectangle using the values held by the two variables.

```
>> len = 5.0;  
>> ht = 2.7;  
>> area = len*ht;
```

Figure 1. Creating and Using Variables

When a variable is assigned the result of an expression, the right-hand side of the expression is evaluated and the resulting value is assigned to the variable on the left hand side. If no variable is given for the result of an expression, MATLAB places the result in the default variable `ans`. Variables can be cleared (deleted) from the workspace with the `clear` command and the command window can be cleared and the cursor homed with the `clc` command. The first command in Figure 2 will clear only the `ht` variable and the second command will clear all variables in the workspace.

```
>> clear ht;
>> clear;
```

Figure 2. Clearing Workspace Variables

The MATLAB command `who` returns a list of all currently defined variables. The command `whos` returns the information from the `who` command plus the variable sizes, numbers of bytes used, and class. You can see the value stored by a defined variable by looking in the Workspace window or typing the variable name at the MATLAB command line.

Special Variables and Reserved Words

MATLAB has some built in variables and predefined reserved words. Reserved words cannot be used as variables but the built-in variables can be reassigned values. Generally redefining built in MATLAB variables is not a good idea.

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

Table 1. MATLAB Reserved Words

Variable Name	Description
<code>ans</code>	Default variable name for results
<code>beep</code>	Make a sound
<code>pi</code>	Mathematical constant
<code>eps</code>	Smallest positive that can be subtracted from 0 to get a negative value
<code>inf</code>	Infinity
<code>NaN</code>	Not a number
<code>i, j</code>	Imaginary number, $\sqrt{-1}$
<code>realmin, realmax</code>	Smallest and largest positive real numbers
<code>intmax</code>	Largest positive integer
<code>nargin, nargout</code>	Number of function in or out variables
<code>varargin</code>	Variable number of function in arguments
<code>varargout</code>	Variable number of function out arguments

Table 2. MATLAB Built in Variables

Complex Numbers

MATLAB supports complex numbers and MATLAB's arithmetic operators and built in functions will operate on complex numbers if appropriate. The imaginary number $\sqrt{-1}$ is represented by the lower case characters `i` and `j`. The result of operations on complex numbers is typically also a complex number. The real and imaginary parts of a complex number can be extracted using the MATLAB commands `real` and `imag`. Also the magnitude and phase of a complex number can be found using the MATLAB commands `abs` and `angle`. The phase will be in radians. Figure 3 shows examples of entering and using complex numbers.

```
>> c1 = 2 + 3j;
>> c2 = 3 - 2j;
>> c3 = c1 + c2
c3 = 5.0000 + 1.0000i
>> real(c3)
ans = 5
>> imag(c3)
ans = 1
>> c4 = c1 * c2
c4 = 12 + 5i
>> abs(c4)
ans = 13
>> angle(c4)
ans = 0.3948
```

Figure 3. Complex Numbers in MATLAB

Either `i` or `j` can be used to represent the complex number although MATLAB will display the imaginary number using `i`. Complex numbers can also be entered using the form `a+b*i` or `a+b*j`.

Last modified Friday, September 18, 2020

 [MATLAB Marina](https://creativecommons.org/licenses/by-nc-sa/4.0/) is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.