

MATLAB Marina: Conditional Statements, switch

Student Learning Objectives

After completing this module, one should:

1. Be able to use switch-case statements to selectively execute blocks of code.
2. Be able to use switch-case statements to selectively perform operations from a menu.

Terms

logical true, logical false, enumerated type, condition, menu

MATLAB Functions, Keywords, and Operators

switch, case, otherwise, end

Switch Statements

The `switch` statement (`switch-case`) is a conditional structure that is useful for handling multiple cases of the value of a single variable and enables a program to conditionally execute one block of statements from several choices or cases. Generally a `switch` statement is preferred over an `if-else` statement if there are four or more cases and if the cases are selected based on a single parameter. The `switch` statement has the general form shown in Figure 1a.

```
switch switch expression
    case first case expression
        statements to be executed if first case true
    case second case expression
        statements to be executed if second case true
    :
    otherwise
        statements to be executed if all cases false
end
```

Figure 1a. General Form of switch-case Statement

The result of the MATLAB switch expression must be a scalar or string. The result of the switch expression is compared to the result of each case expression until a match is obtained. The block of statements in the case with the first match is executed. Only one case of a `switch` statement should evaluate to true and that block of statements is executed. The program goes to the statement after the end of the `switch` statement after executing the block of statements.

If the result of the switch expression does not match any of the cases, the statements in the `otherwise` block are executed. The `otherwise` block is optional and without an `otherwise` block it is possible for no statements in the switch-case blocks to be executed. It is ok for no `switch` cases to be matched.

Multiple cases can be handled with the same block of statements using a list of cases as shown in Figure 1b. All the cases for the block of statements are provided as a comma separated list enclosed in braces (a cell array of items, cell arrays will be covered later in the MATLAB Marina Cell Array module).

```
case {case expression 1, case expression 2, ...}
    statements to be executed if any of cases in {} true
```

Figure 1b, case Block for Multiple Cases

Using switch Statement

A common use of the `switch` statement is to conditionally execute a block of statements based on a menu choice. The program of Figure 2 computes either the sine, cosine, or tangent of an angle based on a user menu choice.

```
% read in angle and user choice or trig function
angleDegrees = input('Enter angle in degrees: ');
choice = menu(' ', 'sine', 'cosine', 'tangent');
result = 0;
type = "";
% compute selected trigonometric function
switch (choice)
    case {1}
        result = sind(angleDegrees);
        type = "sin";
    case {2}
        result = cosd(angleDegrees);
        type = "cos";
    case {3}
        result = tand(angleDegrees);
        type = "tan";
end

fprintf('%s(%0.1f) = %0.2f\n', type, angleDegrees, result);
```

Figure 2. Program to Compute Sine, Cosine, or Tangent Based on Menu Choice

The `menu` function will return an integer corresponding to the number of the button selected. The variable `choice` will be assigned the value corresponding to the menu button pressed. In the example of Figure 2, the `switch` statement does not contain an `otherwise` block as the only options of the menu are the three cases.

Generally, `switch` statements should only be used when selecting a case based on an enumerated type such as an integer, character, or string. They should not be used when selecting a case based on a real number result as real numbers are rarely equal and matches for the cases may not occur. The case expression cannot include relational operators (result will be

a Boolean which is not a legal result for a case expression). Comparisons based on relational operators should be performed using `if-else` statements.

Variables should generally be defined before conditional structures such as `switch` and `if-else` statements although they may be modified inside a block of a conditional structure. In the example of Figure 2, the `result` and `type` are initialized before the conditional statement. This ensures that if the variables have values even in the conditional statement does not assign something to them or modify them.

Last modified Monday, September 28, 2020

 [MATLAB Marina](#) is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.