

Programming

If you have an active Internet Explorer® connection, you can watch the Writing a MATLAB Program video demo for an overview of the major functionality.

- “Flow Control” on page 4-2
- “Other Data Structures” on page 4-9
- “Scripts and Functions” on page 4-20
- “Object-Oriented Programming” on page 4-33

Flow Control

In this section...
“Conditional Control — if, else, switch” on page 4-2
“Loop Control — for, while, continue, break” on page 4-5
“Error Control — try, catch” on page 4-7
“Program Termination — return” on page 4-8

Conditional Control — if, else, switch

Conditional statements enable you to select at run time which block of code to execute. The simplest conditional statement is an if statement. For example:

```
% Generate a random number
a = randi(100, 1);

% If it is even, divide by 2
if rem(a, 2) == 0
    disp('a is even')
    b = a/2;
end
```

if statements can include alternate choices, using the optional keywords elseif or else. For example:

```
a = randi(100, 1);

if a < 30
    disp('small')
elseif a < 80
    disp('medium')
else
    disp('large')
end
```

Alternatively, when you want to test for equality against a set of known values, use a switch statement. For example:

```
[dayNum, dayString] = weekday(date, 'long', 'en_US');

switch dayString
    case 'Monday'
        disp('Start of the work week')
    case 'Tuesday'
        disp('Day 2')
    case 'Wednesday'
        disp('Day 3')
    case 'Thursday'
        disp('Day 4')
    case 'Friday'
        disp('Last day of the work week')
    otherwise
        disp('Weekend!')
end
```

For both `if` and `switch`, MATLAB executes the code corresponding to the first true condition, and then exits the code block. Each conditional statement requires the `end` keyword.

In general, when you have many possible discrete, known values, `switch` statements are easier to read than `if` statements. However, you cannot test for inequality between `switch` and `case` values. For example, you cannot implement this type of condition with a `switch`:

```
yourNumber = input('Enter a number: ');

if yourNumber < 0
    disp('Negative')
elseif yourNumber > 0
    disp('Positive')
else
    disp('Zero')
end
```

Array Comparisons in Conditional Statements

It is important to understand how relational operators and `if` statements work with matrices. When you want to check for equality between two variables, you might use