

Table 1-5: Rounding functions (Continued)

Function	Description	Example
<code>fix(x)</code>	Round toward zero.	<pre>>> fix(13/5) ans = 2</pre>
<code>ceil(x)</code>	Round toward infinity.	<pre>>> ceil(11/5) ans = 3</pre>
<code>floor(x)</code>	Round toward minus infinity.	<pre>>> floor(-9/4) ans = -3</pre>
<code>rem(x,y)</code>	Returns the remainder after x is divided by y .	<pre>>> rem(13,5) ans = 3</pre>
<code>sign(x)</code>	Signum function. Returns 1 if $x > 0$, -1 if $x < 0$, and 0 if $x = 0$.	<pre>>> sign(5) ans = 1</pre>

1.6 DEFINING SCALAR VARIABLES

A variable is a name made of a letter or a combination of several letters (and digits) that is assigned a numerical value. Once a variable is assigned a numerical value, it can be used in mathematical expressions, in functions, and in any MATLAB statements and commands. A variable is actually a name of a memory location. When a new variable is defined, MATLAB allocates an appropriate memory space where the variable's assignment is stored. When the variable is used the stored data is used. If the variable is assigned a new value the content of the memory location is replaced. (In Chapter 1 we consider only variables that are assigned numerical values that are scalars. Assigning and addressing variables that are arrays is discussed in Chapter 2.)

1.6.1 The Assignment Operator

In MATLAB the `=` sign is called the assignment operator. The assignment operator assigns a value to a variable.

Variable_name = A numerical value, or a computable expression

- The left-hand side of the assignment operator can include only one variable name. The right-hand side can be a number, or a computable expression that can include numbers and/or variables that were previously assigned numerical values. When the **Enter** key is pressed the numerical value of the right-hand side is assigned to the variable, and MATLAB displays the variable and its assigned value in the next two lines.

The following shows how the assignment operator works.

```
>> x=15
x =
    15
>> x=3*x-12
x =
    33
>>
```

The number 15 is assigned to the variable x.

MATLAB displays the variable name and its assigned value.

A new value is assigned to x. The new value is 3 times the previous value of x minus 12.

The last statement ($x = 3x - 12$) illustrates the difference between the assignment operator and the equal sign. If in this statement the $=$ sign meant equal, the value of x would be 6 (solving the equation for x).

The use of previously defined variables to define a new variable is demonstrated next.

```
>> a=12
a =
    12
>> B=4
B =
     4
>> C=(a-B)+40-a/B*10
C =
    18
```

Assign 12 to a.

Assign 4 to B.

Assign the value of the expression on the right-hand side to the variable C.

- If a semicolon is typed at the end of the command, then when the **Enter** key is pressed, MATLAB does not display the variable with its assigned value (the variable still exists and is stored in memory).
- If a variable already exists, typing the variable's name and pressing the **Enter** key will display the variable and its value in the next two lines.

As an example, the last demonstration is repeated below using semicolons.

```
>> a=12;
>> B=4;
>> C=(a-B)+40-a/B*10;
>> C
C =
    18
```

The variables a, B, and C are defined but are not displayed, since a semicolon is typed at the end of each statement.

The value of the variable C is displayed by typing the name of the variable.

- Several assignments can be typed in the same line. The assignments must be separated with a comma (spaces can be added after the comma). When the **Enter** key is pressed, the assignments are executed from left to right and the variables

and their assignments are displayed. A variable is not displayed if a semicolon is typed instead of a comma. For example, the assignments of the variables `a`, `B`, and `C` above can all be done in the same line.

```
>> a=12, B=4; C=(a-B)+40-a/B*10
```

```
a =  
    12  
C =  
    18
```

The variable `B` is not displayed because a semicolon is typed at the end of the assignment.

- A variable that already exists can be reassigned a new value. For example:

```
>> ABB=72;
```

A value of 72 is assigned to the variable `ABB`.

```
>> ABB=9;
```

A new value of 9 is assigned to the variable `ABB`.

```
>> ABB
```

```
ABB =  
     9
```

The current value of the variable is displayed when the name of the variable is typed and the **Enter** key is pressed.

- Once a variable is defined it can be used as an argument in functions. For example:

```
>> x=0.75;
```

```
>> E=sin(x)^2+cos(x)^2
```

```
E =  
     1
```

```
>>
```

1.6.2 Rules About Variable Names

A variable can be named according to the following rules:

- Must begin with a letter.
- Can be up to 63 characters long.
- Can contain letters, digits, and the underscore character.
- Cannot contain punctuation characters (e.g., period, comma, semicolon).
- MATLAB is case-sensitive: it distinguishes between uppercase and lowercase letters. For example, `AA`, `Aa`, `aA`, and `aa` are the names of four different variables.
- No spaces are allowed between characters (use the underscore where a space is desired).
- Avoid using the name of a built-in function for a variable (i.e., avoid using `cos`, `sin`, `exp`, `sqrt`, etc.). Once a function name is used to for a variable name, the function cannot be used.