Learning Outcomes

When you have completed this module, you will be able to:

1. Describe and apply the if and progn looping functions.
2. Describe and apply the 1+ and 1- operators.

AutoLISP Function: if

The if {if - then - else} function is used to evaluate a test-expression, and depending on what the test-expression returns, if will evaluate either one of the two expressions contained within it.

The if function first evaluates the test-expression, and if it returns anything but nil, it evaluates the then-expression, but if it evaluates nil, it evaluates the else-expression. The else-expression is optional.

Returns: The value of the selected expression. If the else-expression is selected and it is missing, it returns nil.

Format: (if (test-expression) (then-expression) [(else-expression)])

Examples:

1

Figure Example 1A
The `if` function will test to see if the text height entered by the user is less than or equal to 1.0. If it is, it sets the TEXTSIZE system variable to the number entered by the user. If it is greater than 1, it prints the error message to the user.

Study this `if` function and see if you can understand how it works.

**Author's Comments:** Study this `if` function and see if you can understand how it works.
AutoLISP Function: **progn**

The **progn** function is used to evaluate more than one expression where only one expression was expected.

This function is normally used inside the **if** function. It allows **if** to evaluate more than one expression in either the then-expression or the else-expression.

**Returns:** The value of the last expression evaluated.

**Format:**

```
(progn
  (expression)
  (...)
)
```

**Examples:**

1

![Figure Example 1A](image1)

```auto-lisp
(defun C:EX13-3 (/ pnt1 txt1)
  (setq pnt1 (getpoint "\select the text start location: "))
  (if (= pnt1 nil)
      (princ "You must specify the text start location: ")
      (progn
        (setq txt1 (getstring "\enter the text: "))
        (command "TEXT" pnt1 "" "" txt1)
      ))
  (setqvar "cmdecho" 1)
  (princ)
)
```

![Figure Step 1B](image2)

**Author's Comments:** This example demonstrates how the **progn** function allows the **if** function to evaluate more than one expression in the else area.
Figure Example 2A

Figure Example 2B

**Author's Comments:** This example shows how the else-expression of the `if` function can evaluate more than one expression by using the `progn` function. This program works the same as the LINE command. When the user presses Enter to the 'Enter next point' prompt, the program ends.

The `if` function is best read 'if - then - else'.

It first evaluates the test-expression, and if it returns anything but nil, it evaluates the then-expression, but if it evaluates nil, it evaluates the else-expression.
A good way to exit the while loop is create and set a variable. In the following example, the variable is named stp1 and is set to 0 before the while loop. When you want to exit the loop, set the stp1 variable to 1. i.e.

```
(setq stp1 0)
(while (= stp1 0)
  (setq pnt2 (getpoint pnt1 "\nEnter next point: "))
  (if (= pnt2 nil)
   (setq stp1 1)
   (progn
     (command "LINE" pnt1 pnt2 "")
     (setq pnt1 pnt2))
  )
)
```

The **progn** function is used to evaluate more than one expression where only one expression was expected. It is normally used inside the if function. i.e.

```
(if (= pnt2 nil)
  (setq stp1 1)
  (progn
    (command "LINE" pnt1 pnt2 "")
    (setq pnt1 pnt2))
)
```

**AutoLISP Operator:** **1+**

The **1+** *{increment by 1}* operator is used to increase a number by one increment.

The number can be an integer or a real.

**Returns:** A number. If the number is a real, it returns a real and if the number is an integer, it returns an integer.

**Format:** (1+ number)

**Examples:**

1. Command: *(setq cou1 1)*
   
   1

2. Command: *(setq cou1 (1+ cou1))*
   
   2

**Author's Comments:** This example shows how the 1+ function is used to count.
**AutoLISP Operator: 1-**

The 1- \{decrease by 1\} operator is used to decrease a number by one increment.

The number can be an integer or a real.

**Returns:** A number. If the number is a real, it returns a real and if the number is an integer, it returns an integer.

**Format:** (1- number)

**Examples:**

```
1 Command: (setq cou1 10)
10
Command: (setq cou1 (1- cou1))
9
Command:
```

**Author's Comments:** This expression shows how to count backwards.

---

**Using the if and progn Functions**

**Purpose:** This program allows the user to create a new layer and draws one line. Its purpose is to practice using the if and progn functions to check to see if the user presses the Enter key to get functions and terminate the program when required.

**Step 1** Start a new drawing using the template AutoLISP English.

**Step 2** Create a program naming it AutoLISP Work 13-1.lsp

**Step 3** Name the function LINE1. Have it run like an AutoCAD command.

**Step 4** Declare all variables as local variables.

**Step 5** Save the current value of the system variables CMDECHO, and CLAYER.

**Step 6** Set the value of the system variables CMDECHO to 0.

**Step 7** Ask the user to enter the layer name and allow spaces in the string.

**Step 8** Using the if function, test to ensure that the user has entered a name or pressed the Enter key. If the user presses the Enter key, do nothing. If the user entered a string, then do the for the following:

A. Ask the user to enter the layer color.
B. Create a layer with the name and color entered by the user and make it the current layer.

**Step 9** Ask the user to specify the first point of a line.
Step 10  Using the *if* function, check to see if the user has specified a location.

A  If the user specifies a location, ask the user to specify the next point and draw a line between the two points.

B  If the user press the Enter key instead of specifying a point, print a message 'Program Terminated' and end the program.

Step 11  Reset the original value of the system variables CMDECHO, and CLAYER.

Step 12  Load and run the program in AutoCAD.  (Figure Step 12A and 12B)

**Author's Comments:**  Do your best to complete the AutoLISP program on your own.  If you get stuck and cannot complete it on your own, check my answer on page 13-13.  If you complete the program on your own and it returns the correct results, it is correct even if it does not match my program exactly.  There are many ways to write the same program.

Using the *if*, *progn* and *while* Functions

**Purpose:**  This program allows the user to create a new layer and then work just like the LINE command to draw lines until the users presses the Enter key.  Its purpose is to practice using the *if*, *progn*, and *while* functions to check to see if the user presses the Enter key to terminate the program.

**Step 1**  Open the program *AutoLISP Work 13-1.lsp*.

**Step 2**  Using SAVEAS, save the program with the name *AutoLISP Work 13-2.lsp*.

**Step 3**  Rename the function LINE2.
Step 4  Edit the code as follows:
A  Add a while function to the user input for the layer name.  If the user presses an Enter rather then entering a layer name, the program should loop and ask the user to enter the layer name.  It should only exit the while after the user enters a name.
B  Add a while function to the user input for the first point.  If the user presses an Enter rather then specifying a location, the program should end.
C  If the user specifies the first point, ask the user to specify the next point.  if the user specifies the next point, the program draws a line from point 1 to point 2.  It then asks the user to enter the next point in a continuous loop, drawing lines until the user presses the Enter key.

Step 5  Load and run the program in AutoCAD.  (Figure Step 5A and 5B)

Author's Comments:  Do your best to complete the AutoLISP program on your own.  If you get stuck and cannot complete it on your own, check my answer on page 13-14.  If you complete the program on your own and it returns the correct results, it is correct even if it does not match my program exactly.  There are many ways to write the same program.

The Key Principles in Module 13
1 The if function first evaluates the test-expression, and if it returns anything but nil, it evaluates the then-expression, but if it evaluates nil, it evaluates the else-expression.
2 The progn function is used to evaluate more than one expression where only one expression was expected.  This function is normally used inside the if function.
Lab Exercise 13-1

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoLISP Lab 13-1.lsp</td>
<td>AutoLISP English</td>
</tr>
</tbody>
</table>

**Purpose:** This program allows the user to insert several lines of text in one command. Very similar to the DTEXT command except it aligns the text in a column. The program terminates when the user presses Enter to the 'Enter text:' prompt.

**Step 1** Write a program and name the function LTEXT. Have it run like an AutoCAD command.

**Step 2** Declare all variables as local variables.

**Step 3** Save the current value of the required system variables.

**Step 4** Set the required system variables to the required values.

**Step 5** Ask the user to enter the layer name and the layer color. Save them as strings.

- **A** If the user presses Enter to either the layer name or the layer color, do nothing.
- **B** If the user enters a name and color, create the layer with the color and set it as the current layer.

**Step 6** Ask the user to enter the following:

- **A** Specify start point:
- **B** Enter or specify the text height:
- **C** Enter or specify the text spacing:

**Step 7** Using a while loop, ask user to the 'Enter text:' prompt.

- **A** If text is entered, insert it on the drawing and ask to enter text again.
- **B** When the user presses the Enter key to the 'Enter text:' prompt, end the program.

**Step 8** Using the spacing distance entered by the user, calculate the location for each line of the text. Insert the lines of text as single line text.

**Step 9** Reset the original system variables values as necessary.
Step 10  Load and run the program in AutoCAD. (Figure Step 10A and 10B)

![AutoCAD Text Window - Drawing5.dwg](image)

Figure Step 10A

**NOTES**

1. MATERIAL STEEL  
2. DIMENSIONS ARE IN INCHES

Figure Step 10B

Author's Comments:  Do your best to complete the AutoLISP program on your own. If you get stuck and cannot complete it on your own, check my answer in Module 28. If you complete the program on your own and it returns the correct results, it is correct even if it does not match my program exactly. There are many ways to write the same program.
# Lab Exercise 13-2

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoLISP Lab 13-2.lsp</td>
<td>AutoLISP English</td>
</tr>
</tbody>
</table>

**Purpose:** This program allows the user to insert incremented numbers on the drawing at the specified locations. The program will ask the user for the starting number and the increment.

**Step 1** Write a program and name the function INCNUM. Have it run like an AutoCAD command.

**Step 2** Declare all variables as local variables.

**Step 3** Save the current value of the required system variables.

**Step 4** Set the required system variables to the required values.

**Step 5** Ask the user to ' Specify the text height: '  
A. If the user enters a text height, set system variable TEXTSIZE to that value.  
B. If the user presses Enter, use the current text height set in the system variable TEXTSIZE.

**Step 6** Ask the user to enter the following:  
A. Enter starting number:  
B. Enter increment:

**Step 7** In a **while** loop, ask the user to ' Select start point of text: '  
A. If the user presses Enter, exit the loop.  
B. If the user specifies a start point, insert the first number.

**Step 8** Count how many numbers are inserted and print it on the screen when the program ends.

**Step 9** Reset the original system variables values as necessary.
**Step 10**  Load and run the program in AutoCAD.  (Figure Step 10A and 10B)

![AutoCAD Text Window - Drawing3.dwg](image)

**Figure Step 10A**

```
Command: INCNUM
Specify text height: 0.5
Enter starting number: 2
Enter the increment: 3
Select start point of text:
Select start point of text:
Select start point of text:
Select start point of text:
Select start point of text:
Select start point of text:
You inserted 5 numbers
```

**Figure Step 10B**

```
5 11
2 8
```

**Author’s Comments:**  Do your best to complete the AutoLISP program on your own.  If you get stuck and cannot complete it on your own, check my answer in Module 28.  If you complete the program on your own and it returns the correct results, it is correct even if it does not match my program exactly.  There are many ways to write the same program.
Answers to Workalongs

(defun c:LINE1 (/ svcm svcl layn layc pnt1 pnt2)
  (setq svcm (getvar "cmdecho"))
  (setq svcl (getvar "clayer"))
  (setq "cmdecho" 0)
  (setq layn (getstring T "\nEnter the name for the new layer: "))
  (if (= layn "")
    (progn
      (setq layc (getstring nil "\nEnter layer color: ")
          (command "-LAYER" "m" layn "c" layc layn "")
      )
    )
  (setq pnt1 (getpoint "\Specify first point: ")
  (if (= pnt1 nil)
    (progn
      (print "Program terminated")
      (progn
        (setq pnt2 (getpoint pnt1 "\Specify next point: ")
        (command "LINE" pnt1 pnt2 "")
      )
    )
  (setq "cmdecho" svcm)
  (setq "clayer" svcl)
  (print"
;; The End
")

AutoLISP Work 13-1.lsp
AutoLISP Work 13-2.lsp

;***********************************************************************
; AutoLISP Program
;***********************************************************************
(defun C::LINE2 (/ svcm svc1 layn layc pnt1 pnt2 stp1 stp2 stp3)
  (setq svcm (getvar "cmdecho"))
  (setq svc1 (getvar "clayer"))
  (setqvar "cmdecho" 0)
  (setq stp1 0)
  (while (= stp1 0)
    (setq layn (getstring T "\nEnter the name for the new layer: ")
     (if (= layn"
          (prog
            (setq layc (getstring n11 "Enter layer color: ")
            (command "-LAYER" m" layn "c" layc layn "))
            (setq stp1 1)
          (setq stp1 1))
    )
  )
  )
  (setq stp2 0)
  (setq stp3 0)
  (while (= stp2 0)
    (setq pnt1 (getpoint "\nEnter the first point: ")
     (if (= pnt1 n11)
          (setq stp2 1)
          (prog
            (while (= stp3 0)
              (setq pnt2 (getpoint pnt1 "\nEnter next point: ")
              (if (= pnt2 n11)
                (prog
                  (setq stp2 1)
                  (setq stp3 1)
                )
                (prog
                  (command "LINE" pnt1 pnt2 ")
                  (setq pnt1 pnt2)
                )
              )
            )
          )
        )
    )
  )
  )
  (setqvar "cmdecho" svcm)
  (setqvar "clayer" svc1)
  (princ)

;***********************************************************************
The End
;***********************************************************************

AutoLISP Work 13-2.lsp