

Dialog Box

A dialog box, created by the INFORM command is an input tool that allows the user to input several values at once. It requires 5 items on the stack.

- On Level 5: A string naming the dialog box
- On Level 4: A list of the variables to be entered. Each item in the list can be a string giving the name of the variable or a list of two strings, the first the name of the variable and the second an explanation of the variable.
- On Level 3 A number or a list with two numbers. If just a number it is the number of columns in the dialog box. If a list of two numbers, the first number is the number of columns and the second is the number of spaces between the variable name and the field in which the value will be typed. The default value for the number of space is three if no value is provided. There is only room for 4 rows in the dialog box, so the number of columns must be large enough so that all of the variables will fit in four rows. That is, the number of variables divided by the number of columns must be at most 4.
- On Level 2 A list of values to be inserted if the user uses the RESET option. The list may be left empty, in which case RESET leave the fields blank. If some of the variables are to be reset to values and others are to be left blank, put NOVAL in the place of those to be left blank.
- On Level 1 A list of values to be inserted when the box is initially activated. The same rules apply as for the list of reset values.

The dialog box is activated with the command INFORM, which can be found by pressing `LS PRG NXT F5-IN F1-INFORM`. Notice that NOVAL is in F2 in this same menu. Once the box is activated the user can use the arrow keys to navigate about the box and fill in values the same as with the dialog boxes found in many of the HP applications. If the user exits the box by pressing `F6-OK`, a list of the values the user has entered, in the same order as the variables are listed, will be left on level 2 of the stack and a one will be on level 1. If the user has left any of the fields in the box blank, NOVAL will show in its location in the list. If the user exits the box by pressing `F5-CANCL`, a zero will be left on level 1 of the stack. A typical program that uses a dialog box for input has the following structure:

Steps to set up the stack for the dialog box

INFORM

IF THEN

Steps to use the input for whatever the program is supposed to do.

ELSE

Steps to do whatever the program should do if there is no input.

END

Example: Suppose we have a program that requires 5 inputs, a , b , c , d , and e . The following is a possible solution to create an appropriate dialog box.

Level 5: "Dialog Box Test"

Level 4: { "a:" { "b:" "How many bananas?" } "c:" "d:" "e:" }

Level 3: { 2 1 }

Level 2: { 2 NOVAL NOVAL NOVAL "B Split" }

Level 1: { 2 NOVAL NOVAL NOVAL "B Split" }

INFORM

Try this. You will find that when the dialog box opens there will be a 2 preset for a and "B Split" will be preset for e . The message for e should be "B Split", but the last " does not show because the field is not long enough even though there is no field to the right of it. When the cursor is moved to e and F1-EDIT is pressed the whole message shows. The fields for the other three variables will be blank. When the cursor is moved to b the message "How many bananas?" will show in the lower left corner of the screen.

In this case the reset list and initial value list are the same. The most common usage is to leave both of these lists empty, but when they are used, they are usually the same.

Choose Box

A choose box, created by the CHOOSE command is an input tool that allows the user to select an input from a list. It requires 3 items on the stack.

On Level 3: A string naming the choose box.

On Level 2: A list of the choices. Each item in the list can be an object or a list of two objects. If is just a list of objects, the object will show in the choose box and will be placed on level two of the stack if it is chosen. If a list of two objects the first will show in the choose box and the second will be left on level 2 of the stack if chosen.

On Level 1: A positive integer indicating which item in the choose box is to be selected when it is activated.

The choose box is activated with the command CHOOSE, which can be found by pressing LS PRG NXT F5-IN F3-CHOOSE. Once the box is activated the user can use the up and down arrow keys to select the desired option in the box. If the user exits the box by pressing F6-OK, the chosen object will be left on level 2 of the stack and a one will be on level 1. If the user exits the box by pressing F5-CANCL, a zero will be left on level 1 of the stack.

Example: Suppose the user must make a choice from a list of three items at some point in a program. The following is a possible way to implement this requirement.

```
Level 3:      "Choose Box Test."
Level 2:      { 1 { TWO 2 } "Forget it." }
Level 1:      2
```

When the CHOOSE command is issued, a choose box will appear with 1 as the first choice, TWO as the second, and "Forget it." as the third. The second choice will be selected. If 1 is chosen, a 1 will be on level 2 of the stack; if TWO is chosen, a 2 will be on level 2 of the stack; and if "Forget it." is chosen, "Forget it." will be on level 2 of the stack.

FOR STEP Loop

Besides the FOR NEXT loop, which increments the index one step at a time, there is also a FOR STEP loop that allows the user to increment the counter any fixed number of steps at a time. The general format is

```
Start Stop
FOR index
  Body of loop
  Step-size
STEP
```

This structure can be entered into a program by pressing RS F4-FOR from the PRG > F3-BRCH directory.

For example, the following program will output the squares of the odd integers from 1 to 7

```
<< 1. 7.
    FOR j
      j SQ
    2.
    STEP
>>
```

It is also possible to step backwards. For example, the following program will produce the same list of squares, but in decreasing order.

NOTE: The *Start*, *Stop* and *Step-size* must have the decimal to be sure the program works properly. If 1., 7. and -2. below are replaced with 1, 7 and 2 the program will go through the loop one extra time. A problem for HP engineers to worry about.

```
<<
  7. 1.
  FOR j
    j SQ
    -2.
  STEP
>>
```

Some List Commands

In the language of HP, a list is a group of objects enclosed in braces, $\{ \}$. A list can contain any type of objects. For example

$$\{ 23 \text{ 'X+1' "Hi there" } \{ 1 \ 2 \ 3 \} << + + + >> \}$$

is a list consisting of a number, an algebraic, a string, a list of numbers, and a program.

There are two primary places where a variety of list commands can be found. At this time we will only discuss a few of those commands. The first place to look is LS PRG F6-LIST. This menu has two important commands, F3-OBJ→ and F4→LIST. If there is a list with n elements on level 1 of the stack, F3-OBJ→ will leave those elements on levels $n+1$ to 2 on the stack and n will be on level 1. If there is an integer n on level 1 of the stack and at least n elements on the stack above it, pressing F4→LIST will put the elements from level $n+1$ to 2 in a list. Pressing F1-ELEM from this menu opens another menu with some important commands. If there is a list of length n on level 2 and a positive integer, i , less than or equal to n on level 1, pressing F1-GET will remove these two items and leave the i th item from the list on level 1. If there is a list of length n on level 3, a positive integer, i , less than or equal to n on level 2 and an object on level 3, pressing F3-PUT will remove the items from levels 1 and 2, and the list, with the i th item replaced with the object that was on level 1, will be left on level 1. If there is a list of length n on level 1 and F5-SIZE is pressed, the list will be deleted and n will be left on level 1.

The second place with several list commands can be found at LS MTH F3-LIST. If the list $\{a_1 \ a_2 \ \cdots \ a_n\}$ is on level 1 the command F1-ΔLIST will replace it with the list $\{b_1 \ b_2 \ \cdots \ b_{n-1}\}$ where $b_i = a_{i+1} - a_i$. If there is a list of numbers on level 1, the command F4-SORT will arrange the list in increasing order. If there is a list on level 1 of the stack, the command F5-REVLIST will reverse the order of the elements in the list.

On the keyboard, the + key is also important. If there is an object on level 2 and a list on level 1, + will insert the object from level 2 at the beginning of the list. If there is a list on level 2 and an object on level 1, + will add the object to the end of the list.

For more list commands see Chapter 8 of the User's Guide.

Enhanced INPUT Command

The simplest form of the INPUT command is to put a prompt as a string on level 2 of the stack and a pair of quotes on level 1, then issue the INPUT command. Program execution will halt with the prompt showing at the top of the screen and the cursor in the command line waiting for the user to enter data. When the user presses ENTER, whatever was typed in the command line will be left on level 1 of the stack as a string. The most common command to follow the INPUT command is OBJ→ to strip away the quotes from the string.

An alternative is to put the prompt on level 2 and on level 1 put a list of instructions telling the calculator how to handle the input. For example, if the list on level 1 is { α " " }, the keyboard will be locked in alpha mode when the INPUT command is executed. This makes it easier for the user if the expected input is to be alphabetic.

[Return to List of Lessons](#)

[Return to An Introduction to Programming
HP Graphing Calculators](#)