

Programming

Introduction

This chapter describes how to program using the HP 39G/40G. In this chapter you'll learn about:

- using the Program catalog to create and edit programs
- programming commands
- storing and retrieving variables in programs
- programming variables.



More information on programming, including examples and special tools can be found at HP's calculators web site:

www.hp.com/calculators

The Contents of a Program

An HP 39G/40G program contains a sequence of numbers, mathematical expressions, and commands that execute automatically to perform a task.

These items are separated by a colon (:). Commands that take multiple arguments have those arguments separated by a semicolon (;). For example,

`PIXON xposition,yposition:`

Structured Programming

Inside a program you can use branching structures to control the execution flow. You can take advantage of structured programming by creating building-block programs. Each building-block program stands alone—and it can be called from other programs. *Note: If a program has a space in its name then you have to put quotes around it when you want to run it.*

Example

```
RUN GETVALUE: RUN CALCULATE: RUN
"SHOW ANSWER":
```

This program is separated into three main tasks, each an individual program. Within each program, the task can be simple—or it can be divided further into other programs that perform smaller tasks.

Program catalog

The Program catalog is where you create, edit, delete, send, receive, or run programs. This section describes how to

- open the Program catalog
- create a new program
- enter commands from the Commands menu
- enter functions from the MATH menu
- edit a program
- run and debug a program
- stop a program
- copy a program
- send and receive a program
- delete a program or its contents
- customize an applet

Open Program catalog

1. Press **[SHIFT]PROGRAM**.

The Program catalog displays a list of program names. If you haven't created any programs, the only name you'll see is *Editline*.

Editline contains the last expression that you entered from the edit line in HOME, or the last data you entered in an input form. (If you press **[ENTER]** from HOME without entering any data, the HP 39G/40G runs the contents of Editline.)



Editline is a built-in function.



Program catalog menu

Before starting to work with programs, you should take a few minutes to become familiar with the Program catalog menu keys. You can use any of the following keys (both menu and keyboard), to perform tasks in the Program catalog.

Program catalog keys

Key	Meaning
EDIT	Opens the highlighted program for editing.
NEW	Prompts for a new program name, then opens an empty program.
SEND	Transmits the highlighted program to another HP 39G/40G or to a disk drive.
RCV	Receives the highlighted program from another HP 39G/40G or from a disk drive.
RUN	Runs the highlighted program.
SHIFT  or SHIFT 	Moves to the beginning or end of the Program catalog.
DEL	Deletes the highlighted program.
SHIFT CLEAR	Deletes all programs in the program catalog.

Creating and editing programs

Create a new program

1. Press **SHIFT** **PROGRAM** to open the Program catalog.
2. Press **NEW**.

The HP 39G/40G prompts you for a name.



A program name can contain special characters, such as a space. However, if you use special characters and then run the program by typing it in HOME, you must enclose the program name in double quotes (" "). Don't use the " symbol within your program name.

3. Type your program name, then press **OK**.
When you press **OK**, the Program Editor opens.



4. Enter your program.
When done, start any other activity. Your work is saved automatically.

Enter commands

Until you become familiar with the HP 39G/40G commands, the easiest way to enter commands is to use the Commands menu from the Program editor. You can always type in commands using alpha characters.

1. From the Program editor, press **[SHIFT] CMDS** to open the Program Commands menu.

[SHIFT] CMDS



2. On the left, use **[↓]** or **[↑]** to highlight a command category, then press **[▶]** to access the commands in the category. Select the command that you want.

[↓] [↓] [▶] [↓]



3. Press **OK** to paste the command into the program editor.

OK

To enter functions (more to come)















Edit a program

1. Press **[SHIFT] PROGRAM** to open the Program catalog.



2. Use the arrow keys to highlight the program you want to edit, and press **EDIT**. The HP 39G/40G opens the Program Editor. The name of your program appears in the title bar of the display. You can use the following keys to edit your program.

Editing keys

Key	Meaning
STO►	Inserts the STO► character at the editing point.
SPACE	Inserts space into text.
▲PAGE	Displays previous page of the program.
PAGE▼	Displays next page of the program.
 	Moves up or down one line.
 	Moves right or left one character.
A...Z	Alpha-lock for letter entry. Press  A...Z to lock lower case.
BKSP	Backspaces cursor and deletes character.
	Deletes current character.
	Starts a new line.
 CLEAR	Erases the entire program.
 	Menus for entering variable names, contents of variables, math functions, and program constants.
 CMDS	Menus for entering program commands.
 CHARS	Displays all characters. To type one, highlight it and press OK . To enter several characters in a row, use the ECHO menu key while in the CHARS menu.

Using programs

Run a program

From HOME, type RUN *program_name*.

or

From the Program catalog, highlight the program you want to run and press RUN.

Regardless of where you start the program, all programs run in HOME. What you see will differ slightly depending on where you started the program. If you start the program from HOME, the HP 39G/40G displays the contents of *Ans* (Home variable containing the last result), when the program has finished. If you start the program from the Program catalog, the HP 39G/40G returns you to the Program catalog when the program ends.

Debug a program

If you run a program that contains errors, the program will stop and you will see an error message.



To debug the program:

1. Choose **YES** to edit the program.
The insert cursor appears in the program at the point where the error occurred.
2. Edit the program to fix the error.
3. Re-start the program.
4. Repeat the process until you find and correct all errors.

Stop a program

You can stop the execution of a program at any time by pressing **CANCEL** (the **ON** key). *Note: You may have to press it a couple of times.*

Working with programs

Copy a program

You can use the following procedure if you want to make a copy of your work before editing—or if you want to use one program as a template for another.

1. Press **[SHIFT]PROGRAM** to open the Program catalog.
2. Press **NEW**.
3. Type a new file name, then choose **OK**.

The Program Editor opens with an empty new program.

4. Press **[VAR]** to open the Variable menu.
5. Press **P** to quickly scroll to Program.
6. Press **[▶]**, then highlight the program you want to copy.
7. Press **VALUE**, then press **OK**.

The contents of the highlighted program are copied into the current program at the cursor location.

Hint

If you use a programming routine often, save the routine under a separate program name, then use the above method to copy it into your programs.

Transmit a program

You can send and receive programs to/from other calculators just as you can send and receive aplets, matrices, lists, and notes.

After aligning the calculators' infrared ports, open the Program catalogs on both calculators. Highlight the program to send, then press **SEND** on the sending calculator and **RCV** on the receiving calculator. See "Sending and receiving aplets" on page 16-242.

You can also send (copy) and receive programs to and from a remote storage device (aplet disk drive or computer). This takes place via a cable connection and requires an aplet disk drive or specialized software running on a PC or Mac (such as a connectivity kit).

Delete a program

You can delete any program except Editline.

1. Press **[SHIFT]PROGRAM** to open the Program catalog.
2. Highlight a program to delete, then press **[DEL]**.

Delete all programs

You can delete all programs at once.

1. In the Program catalog, press **[SHIFT]CLEAR**.
2. Press **YES**.

Delete the contents of a program

You can clear the contents of a program without deleting the program itself.

1. Press **[SHIFT]PROGRAM** to open the Program catalog.
2. Highlight a program, then press **EDIT**.
3. Press **[SHIFT]CLEAR**, then press **YES**.
4. The contents of the program are deleted, but the program name remains.

About customizing an aplet

There are two levels to the process of customizing an aplet. The simplest requires no programming skills at all.

The first and simplest way to customize an aplet is to save it under a new name, perhaps with pre-loaded data and/or functions and pre-set axes. See “Creating new aplets based on existing aplets” on page 16-239.

The second way to customize an aplet involves programming. This method, used to produce the teaching aplets available from Hewlett-Packard’s web site, uses the **SETVIEWS** command to create a custom **VIEWS** menu which links specially written programs to the new aplet. This method is discussed in detail later.

See “*SETVIEWS*” on page 15-211 for information on the command.

To customize an aplet to work with a set of programs, use the following method:

1. Decide on the aplet type that you want to use, for example the Function aplet or the Statistics aplet. The copied aplet inherits all the abilities of the parent aplet. Save the standard aplet under a new name.
2. Configure the new aplet if you need to, for example by presetting axes or angle measures.
3. Develop the programs to work with your aplet. When you develop the aplet’s programs, use the standard aplet naming convention. This allows you to keep track of the programs in the Program catalog that belong to each aplet.
4. Develop a program that uses the **SETVIEWS** command to modify the aplet’s **VIEWS** menu. The menu options provide links to associated programs. You can specify

any other programs that you want transferred with the applet. See “*SETVIEWS*” on page 15-211 for information on the command.

5. Ensure that the new applet is selected, then run the menu configuration program to configure the applet’s VIEWS menu.
6. Test the applet and debug the associated programs.(Refer to “Debug a program” on page 15-204).

Aplet naming convention

To assist users to keep track of applets and associated programs, use the following naming convention when setting up an applet’s programs:

- Start all program names with an abbreviation of the applet name. We will use APL in this example.
- Name programs called by menu entries in the VIEWS menu number, after the entry, for example:
.APL.ME1 for the program called by menu option 1.
.APL.ME2 for the program called by menu option 2.
- Name the program that configures the new VIEWS menu option as .APL.SV where SV stands for SETVIEWS.

For example, a customised applet called ‘Differentiation’ might call programs called .DIFF.ME1, .DIFF.ME2, .DIFF.SV.

Customizing an applet

This example applet is designed to demonstrate the process of configuring an applet. The new applet is based on the Function applet. *Note: This applet is NOT intended to serve a serious use, merely to illustrate the process.*

Save the applet

1. Open the Function applet and save it as “Experiment”. The new applet appears in the Aplet library.

[APLET] Select
Function SAVE
[ALPHA] EXPERIMENT
OK



Configuring the Setviews menu option programs

In this section we will begin by configuring the VIEWS menu by using the SETVIEWS command. We will then create the “helper” programs called by the VIEWS menu which will do the actual work.

2. Open the Program catalog and create a program named “.EXP.SV”. Include the following code in the program. (Text shown in *italics* below are comments only.)

Each entry line after the command SETVIEWS is a trio that consists of a VIEWS menu text line (a space indicates none), a program name, and a number that defines the view to go to after the program has run its course. All programs listed here will transfer with an aplet when the aplet is transferred.



```
SETVIEWS " "; " ";18;
```

Sets the first menu option to be "Auto scale". This is the fourth standard Function aplet view menu option and the 18 (14+4) "Auto scale", specifies that it is to be included in the new menu. The empty quotes will ensure that the old name of "Auto scale" appears on the new menu. See "SETVIEWS" on page 15-211.

```
"My Entry1";".EXP.ME1";1;
```

Sets the second menu option. This option runs program .EXP.ME1, then returns to view 1, Plot view.

```
"My Entry2";".EXP.ME2";3;
```

Sets the third menu option. This option runs the program .EXP.ME2, then returns to view 3, the NUM view

```
" "; ".EXP.SV";0;
```

This line specifies that the program to set the View menu (this program) is transferred with the aplet. The space character between the first set of quotes in the trio specifies that no menu option appears for the entry. You do not need to transfer this program with the aplet, but it allows users to modify the aplet's menu if they want to.

```
" "; ".EXP.ANG";0;
```

The program EXP.ANG is a small routine that is called by other programs that the aplet uses. This entry specifies that the program .EXP.ANG is transferred when the aplet is transferred, but the space in the first quotes ensures that no entry appears on the menu.

```
"Start";".EXP.S";7;
```

This specifies the Start menu option. The program that is associated with this entry, .EXP.S, runs automatically

when you start the applet. Because this menu option specifies view 7, the VIEWS menu opens when you start the applet.

You only need to run this program once to configure your applet's VIEWS menu. Once the applet's VIEWS menu is configured, it remains that way until you run SETVIEWS again.

You do not need to include this program for your applet to work, but it is useful to specify that the program is attached to the applet, and transmitted when the applet is transmitted.

3. Set up the program for the "My Entry1" menu option as shown. This program configures the plot ranges, then runs a program that allows you to configure the angle format. You will create this program later.

```
.EXP.ME1 PROGRAM
-10Xmin:
10Xmax:
-6Ymin:
6Ymax:
RUN ".EXP.ANG":
STD SPACE  Am2 BKSP
```

4. Set up the program for the "My Entry2" menu option as shown. This program sets the numeric view options for the applet, and runs the program that you can use to configure the angle mode.

```
.EXP.ME2 PROGRAM
10NumStart:
2NumStep:
MSGBOX "Numeric
values set.":
RUN ".EXP.ANG":
STD SPACE  Am2 BKSP
```

5. Set up the EXP.ANG program that the previous two programs call.

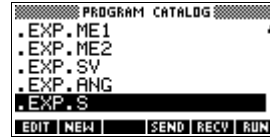
```
.EXP.ANG PROGRAM
C:
CHOOSE C:
"ANGLE MEASURE";
"Degrees";
"Radians";
"Grads":
STD SPACE PAGE  Am2 BKSP
```

Configure the applet's SETVIEW menu

6. Set up the program that runs when you start the applet, as follows: This program sets the angle mode to degrees, and sets up the initial function that the applet plots.

```
.EXP.S PROGRAM
Angle:
X2-2F1(X):
CHECK 1:
STD SPACE  Am2 BKSP
```

7. Return to the program catalog. The programs that you created should appear as follows:
8. You must now RUN the program .EXP.SV to execute the SETVIEWS command and create the modified VIEWS menu. Check that the name of the new aplet is highlighted in the APLET view.
9. You can now return to the APLET library and press START to run your new aplet.



Programming commands

This section describes the commands for programming with the HP 39G/40G. You can enter these commands in your program by typing them or by accessing them from either the Math menu or the Commands menu.

To include programming commands in programs:

1. Press $\boxed{\text{SHIFT}}$ PROGRAM to open the Program catalog.
2. Press NEW or EDIT to open the Program editor.
3. Move the cursor to the position where you want to include a command.
4. From the Program editor, press $\boxed{\text{SHIFT}}$ CMDS.

The HP 39G/40G displays the Program Commands menu. The left side of the screen shows the *command categories* and the right side of the screen shows the available *commands*.



5. Use the arrow keys to select the category and command that you want, and press OK to paste the command at the cursor position.

Aplet commands

These commands control aplets.

CHECK

Checks (selects) the corresponding function in the current aplet. For example, Check 3 would check F3 if the current aplet is Function. Then a checkmark would appear next to F3

in Symbolic view, F3 would be plotted in Plot view, and evaluated in Numeric view.

CHECK *n*

SELECT

Selects the named applet and makes it the current applet. *Note: Quotes are needed if the name contains spaces or other special characters.*

SELECT *appletname*

SETVIEWS

The SETVIEWS command is used to define entries in the VIEWS menu for applets that you customize. The SETVIEWS command works only with a customized applet, that is, a standard applet that you have saved under a new name. See “About customizing an applet” on page 15-206 for an example of using the SETVIEWS command.

When you use the SETVIEWS command, the applet’s standard VIEWS menu is deleted and the customized menu is used in its place. You only need to apply the command to an applet once. The View menu changes remain unless you apply the command again.

Typically, you develop a program that uses the SETVIEWS command only. The command contains a separate trio of arguments for each menu option to create, or program to attach. Keep the following points in mind when using this command:

- You cannot use the SETVIEWS command on a standard applet. You must save a standard applet under a new name before you invoke the command.
- The SETVIEWS command deletes an applet’s standard Views menu options. If you want to use any of the standard options on your reconfigured VIEWS menu, you must include them in the configuration.
- When you invoke the SETVIEWS command, the changes to an applet’s VIEWS menu remain with the applet. You need to invoke the command on the applet again to change the VIEWS menu.
- All the programs that are called from the VIEWS menu are transferred when the applet is transferred, for example to another calculator or to a PC.
- As part of the VIEWS menu configuration, you can specify programs that you want transferred with the

applet, but are not called as menu options. For example, these can be sub-programs that menu options use, or the program that defines the applet's VIEWS menu.

- You can include a "Start" option in the VIEWS menu to specify a program that you want to run automatically when the applet starts. This program typically sets up the applet's initial configuration. The Start option on the menu is also useful for resetting the applet.

Command syntax

The syntax for the command is as follows:

```
SETVIEWS  
"Prompt1"; "ProgramName1"; ViewNumber1;  
"Prompt2"; "ProgramName2"; ViewNumber2;  
(You can repeat as many Prompt/ProgramName/  
ViewNumber trios of arguments as you like.)
```

Within each *Prompt/ProgramName/ViewNumber* trio, you separate each item with a semi-colon.

Prompt

Prompt is the text that is displayed for the corresponding entry in the Views menu. Enclose the prompt text in double quotes.

Associating programs with your applet

If *Prompt* consists of a single space, then no entry appears in the view menu. The program specified in the *ProgramName* item is associated with the applet, and transferred whenever the applet is transmitted. Typically, you do this if you want to transfer the Setviews program with the applet, or you want to transfer a sub-program that other menu programs use.

Auto-run programs

If the *Prompt* item is "Start", then the *ProgramName* program runs whenever you start the applet. This is useful for setting up a program to configure the applet. Users can select the Start item from the Views menu to reset the applet if they change configurations.

You can also define a menu item called "Reset" which is autorun if the user chooses the RESET button in the APLET view.

ProgramName

ProgramName is the name of the program that runs when the corresponding menu entry is selected. All programs that are identified in the applet's SETVIEWS command are transferred when the applet is transmitted.

ViewNumber

ViewNumber is the number of a view to start after the program finishes running. For example, if you want the menu option to display the Plot view when the associated program finishes, you would use 1 as the *ViewNumber* value.

Including standard menu options

To include one of an applet's standard View menu options in your customized menu, set up the quotes trio as follows:

- For the first set of quotes that specifies the menu item name:
 - Leave the quotes empty to use the standard Views menu name for the item, *or*
 - Enter a menu item name to replace the standard name.
- For the second pair of quotes, specifying the program to run:
 - Leave the quotes empty to run the standard menu option.
 - Insert a program name to run the program before the standard menu option is selected.
- For the third pair of quotes, specifying the view, specify the menu number for the item. Determine the menu number from the View numbers table.

View numbers

The views are numbered as follows:

0	Go to HOME	11	List Catalog
1	Plot	12	Matrix Catalog
2	Symbolic	13	Notepad Catalog
3	Numeric	14	Programs Catalog
4	Plot-Setup	15	Plot-Detail
5	Symbolic-Setup	16	Plot-Table
6	Numeric-Setup	17	Overlay Plot
7	Views	18	Auto scale
9	Note	19	Decimal
10	Sketch view	20	Integer
		21	Trig

UNCHECK

Unchecks (unselects) the corresponding function in the current applet. For example, Uncheck 3 would uncheck F3 if the current applet is Function.

UNCHECK *n*

Branch commands

Branch structures let a program make a decision based on the result of one or more tests. Unlike the other programming commands, the control structure commands work in logical groups. Therefore, the commands are described together rather than each independently.

IF...THEN...END or IF... THEN... ELSE... END

Executes a sequence of commands in the *test-clause* only if the *test-clause* evaluates to true. Its syntax is:

IF *test-clause*

THEN *true-clause* ELSE *false-clause* END

IF...THEN...ELSE...END executes the *true-clause* sequence of commands if the *test-clause* is true, or the *false-clause* sequence of commands if the *test-clause* is false.

Example

```
1 ► A :  
IF A = 1
```



```

THEN MSGBOX A " EQUALS 1" :
ELSE MSGBOX A " IS NOT EQUAL TO 1" :
END

```

CASE...END

Executes a series of test-clause commands that execute the appropriate *true-clause* sequence of commands. Its syntax is:

```

CASE
IF test-clause1 THEN true-clause1 END
IF test-clause2 THEN true-clause2 END
.
.
IF test-clausen THEN true-clausen END
END

```

When CASE is executed, *test-clause*₁ is evaluated. If the test is true, *true-clause*₁ is executed, and execution skips to END. If *test-clause*₁ is false, execution proceeds to *test-clause*₂. Execution with the CASE structure continues until a true-clause is executed (or until all the test-clauses evaluate to false).

IFERR... THEN... END...

Many conditions are automatically recognized by the HP 39G/40G as *error conditions*—and they are automatically treated as errors in programs.

IFERR...THEN...END allows a program to intercept error conditions that otherwise would cause the program to abort. Its syntax is:

```

IFERR trap-clause
THEN error-clause END

```

RUN

Runs the named program. If your program name contains special characters, such as a space, then you must enclose the file name in double quotes (" ").

```

RUN "program name" or RUN programname

```

STOP

Stops the current program.

```

STOP

```

Drawing commands

The Drawing commands act on the display. The scale of the display depends on the current applet's Xmin, Xmax, Ymin, and Ymax values. *The following examples assume the HP 39G/40G default settings with the Function applet as the current applet.*

ARC

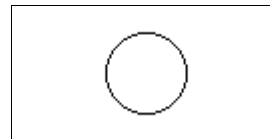
Draws a circular arc, of given radians, whose centre is at (x,y). The arc is drawn from *start_angle_measurement*, and *end_angle_measurement*.

ARC *x*;*y*;*radius*;*start_angle_measurement*;
end_angle_measurement

Example

```
ARC 0;0;2;0;360:  
FREEZE
```

Draws a circle centered at (0,0) of radius 2. The FREEZE command causes the circle to remain displayed on the screen until you press a key.



BOX

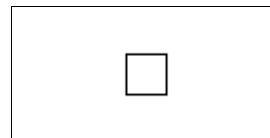
Draws a box with opposite corners (*x1*,*y1*) and (*x2*,*y2*).

BOX *x1*;*y1*;*x2*;*y2*

Example

```
BOX -1;-1;1;1:  
FREEZE
```

Draws a box, lower corner at (-1,-1), upper corner at (1,1)



ERASE

Clears the display

```
ERASE
```

FREEZE

Halts the program, freezing the current display. Execution resumes when any key is pressed.

LINE

Draws a line from (*x1*, *y1*) to (*x2*, *y2*).

```
LINE x1;y1;x2;y2
```

PIXOFF

Turns off the pixel at the specified coordinates (*x*,*y*).

```
PIXOFF x;y
```

PIXON	Turns on the pixel at the specified coordinates (x,y). <code>PIXON x ; y</code>
TLINE	Toggles the pixels along the line from (x1, y1) to (x2, y2) on and off. Any pixel that was turned off, will be turned on; any pixel that was turned on, will be turned off. TLINE can be used to erase a line. <code>TLINE x1 ; y1 ; x2 ; y2</code>
Example	<code>TLINE 0 ; 0 ; 3 ; 3</code> Erases previously drawn 45 degree line from (0,0) to (3,3), or draws that line if it doesn't already exist.

Graphic commands

The Graphic commands use the graphics variables G0 through G9—or the Page variable from Sketch—as *graphicname* arguments. The *position* argument takes the form (x,y). Position coordinates depend on the current applet's scale, which is specified by Xmin, Xmax, Ymin, and Ymax. The upper left corner of the target graphic (*graphic2*) is at (Xmin,Ymax).

You can capture the current display and store it in G0 by simultaneously pressing `[ON]`+`[PLOT]`.

DISPLAY→	Stores the current display in <i>graphicname</i> . <code>DISPLAY→ graphicname</code>
→DISPLAY	Displays graphic from <i>graphicname</i> in the display. <code>→DISPLAY graphicname</code>
→GROB	Creates a graphic from <i>expression</i> , using <i>font_size</i> , and stores the resulting graphic in <i>graphicname</i> . Font sizes are 1, 2, or 3. If the <i>fontsize</i> argument is 0, the HP 39G/40G creates a graphic display like that created by the SHOW operation. <code>→GROB graphicname ; expression ; fontsize</code>
GROBNOT	Replaces graphic in <i>graphicname</i> with bitwise-inverted graphic. <code>GROBNOT graphicname</code>

GROBOR	<p>Using the logical OR, superimposes <i>graphicname2</i> onto <i>graphicname1</i>. The upper left corner of <i>graphicname2</i> is placed at <i>position</i>.</p> <p>GROBOR <i>graphicname1 ; position ; graphicname2</i></p>
GROBXOR	<p>Using the logical XOR, superimposes <i>graphicname2</i> onto <i>graphicname1</i>. The upper left corner of <i>graphicname2</i> is placed at <i>position</i>.</p> <p>GROBXOR <i>graphicname1 ; position ; graphicname2</i></p>
MAKEGROB	<p>Creates graphic with given width, height, and hexadecimal data, and stores it in <i>graphicname</i>.</p> <p>MAKEGROB <i>graphicname ; width height hexdata</i></p>
PLOT→	<p>Stores the Plot view display as a graphic in <i>graphicname</i>.</p> <p>PLOT→ <i>graphicname</i></p> <p>PLOT→ and DISPLAY→ can be used to transfer a copy of the current PLOT view into the sketch view of the applet for later use and editing.</p>
Example	<p>1 ►PageNum: PLOT→Page:</p> <p>FREEZE</p> <p>This program stores the current PLOT view to the first page in the sketch view of the current applet and then displays the sketch as a graphic object until any key is pressed.</p>
→PLOT	<p>Puts graph from <i>graphicname</i> into the Plot view display.</p> <p>→PLOT <i>graphicname</i></p>
REPLACE	<p>Replaces portion of graphic in <i>graphicname1</i> with <i>graphicname2</i>, starting at <i>position</i>. REPLACE also works for lists and matrices.</p> <p>REPLACE <i>graphicname1 ; (position) ; graphicname2</i></p>
SUB	<p>Extracts a portion of the named graphic (or list or matrix), and stores it in a new variable, <i>name</i>. The portion is specified by <i>position</i> and <i>positions</i>.</p> <p>SUB <i>name ; graphicname ; (position) ; (positions)</i></p>

ZEROGROB

Creates a blank graphic with given *width* and *height*, and stores it in *graphicname*.

```
ZEROGROB graphicname ; width ; height
```

Loop commands

Loop structures allow a program to execute a routine repeatedly. The HP 39G/40G has three loop structures. The example programs illustrate each of these structures incrementing the variable A from 1 to 12.

DO...UNTIL ...END

Do ... Until ... End is a loop structure that executes the *loop-clause* repeatedly until *test-clause* returns a true (nonzero) result. Because the test is executed *after* the loop-clause, the loop-clause is always executed at least once. Its syntax is:

```
DO loop-clause UNTIL test-clause END

1 ► A:
DO A + 1 ► A
UNTIL A = 12
END
```

**WHILE...
REPEAT...
END**

While ... Repeat ... End is a loop structure that repeatedly evaluates *test-clause* and executes *loop-clause* sequence if the test is true. Because the test-clause is executed before the loop-clause, the loop-clause is not executed if the test is initially false. Its syntax is:

```
WHILE test-clause REPEAT loop-clause END

1 ► A:
WHILE A < 12
REPEAT A+1 ► A

END
```

**FOR ...TO ...STEP
...END**

```
FOR name=start-expression TO end-expression STEP  
INCREMENT;  
  
loop-clause END  
  
FOR A=1 TO 12 STEP 1;  
  
DISP 3;A:  
  
END:
```

BREAK

Terminates loop.

BREAK

Matrix commands

The Matrix commands use the matrix variables M0–M9 as arguments.

ADDCOL

Add Column. Inserts *values* into a column before *column_number* in the specified matrix. You enter the *values* as a vector. The values must be separated by commas and the number of values must be the same as the number of rows in the matrix *name*.

ADDCOL *name* ; [*value1*, ..., *valuen*] ; *column_number*

ADDROW

Add Row. Inserts *values* into a row before *row_number* in the specified matrix. You enter the values as a vector. The values must be separated by commas and the number of values must be the same as the number of columns in the matrix *name*.

ADDROW *name* ; [*value1*, . . . , *valuen*] ; *row_number*

DELCOL

Delete Column. Deletes the specified column from the specified matrix.

DELCOL *name* ; *column_number*

DELROW

Delete Row. Deletes the specified row from the specified matrix.

DELROW *name* ; *row_number*

EDITMAT

Starts the Matrix Editor and displays the specified matrix. If used in programming, returns to the program when user presses **OK**.

EDITMAT *name*

RANDMAT

Creates random matrix with a specified number of rows and columns and stores the result in *name* (name must be M0 . . . M9). The entries will be integers ranging from -9 to 9.

RANDMAT *name* ; *rows* ; *columns*

REDIM

Redimensions the specified matrix or vector to *size*. For a matrix, *size* is a list of two integers {*n1*,*n2*}. For a vector, *size* is a list containing one integer {*n*}.

REDIM *name ; size*

REPLACE

Replaces portion of matrix or vector stored in *name* with an object starting at position *start*. *Start* for a matrix is a list containing two numbers; for a vector, it is a single number. Replace also works for lists and graphics.

REPLACE *name ; start ; object*

SCALE

Multiplies the specified *row_number* of the specified matrix by *value*.

SCALE *name ; value ; rownumber*

SCALEADD

Multiplies the row of the matrix *name* by *value*, then adds this result to the second specified row.

SCALEADD *name ; value ; row1 ; row2*

SUB

Extracts a *sub-object*—a portion of a list, matrix, or graphic from *object* and stores it into *name*. *Start* and *end* are each specified using a list with two numbers for matrix, a number for vector or lists, or an ordered pair, (X,Y), for graphics.

SUB *name ; object ; start ; end*

SWAPCOL

Swaps Columns. Exchanges *column1* and *column2* of the specified matrix.

SWAPCOL *name ; column1 ; column2*

SWAPROW

Swap Rows. Exchanges *row1* and *row2* in the specified matrix.

SWAPROW *name ; row1 ; row2*

Print commands

These commands print to an HP infrared printer.

PRDISPLAY

Prints the contents of the display.

PRDISPLAY

PRHISTORY

Prints all objects in the history.

PRHISTORY

PRVAR

Prints name and contents of *variablename*.

```
PRVAR variablename
```

You can also use the PRVAR command to print the contents of a program or a note.

```
PRVAR programname ; PROG
```

```
PRVAR notename ; NOTE
```

Prompt commands

You can use the following commands to prompt users for input during your program—or to provide information to users.

BEEP

Beeps at the frequency and time you specify.

```
BEEP frequency ; seconds
```

CHOOSE

Creates a Choose Box, which is a box containing a list of options, from which the user chooses one. Each option is numbered, 1 through *n*, with 0 meaning the Choose Box was cancelled. The result of the choose command is to store the number of the option chosen in a variable. The syntax is


```
CHOOSE name ; prompt , item-1 ... ; item-n
```

Where *name* is the variable to which the option # will be stored, *prompt* is the text displayed in the choose box title bar, and *item-1* through *item-n* are the text items displayed in the Choose Box list. When the Choose Box is created one of the options may be highlighted. The contents of the variable *name* at the time when the box opens determines which item is highlighted and the choose list is read-only.

Example

```
3 STO► A : CHOOSE A ;  
"COMIC STRIPS" ;  
"DILBERT" ;  
"CALVIN&HOBBES" ;  
"BLONDIE" ;
```



Press 

to highlight "CALVIN&HOBBES" , then press **OK**.
This will store 3 in A.

DISP

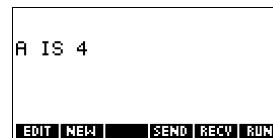
Displays *textitem* in a row of the display at the *line_number*. A text item consists of any number of expressions and quoted strings of text. The expressions are evaluated and turned into strings. Lines are numbered from the top of the screen, 1 being the top and 7 being the bottom.

`DISP line_number;textitem`

Example

`DISP 3;"A is" 2+2`

Result: A is 4
(displayed on line 3)



DISPTIME

Displays the current date and time.

`DISPTIME`

To set the date and time, simply store the correct settings in the date and time variables. Use the following formats: M.DDYYYY for the date and H.MMSS for the time.

Examples:

`5.152000 ► DATE` (sets the date to May 15, 2000).

`10.1500 ► TIME` (sets the time to 10:15 am).

EDITMAT

Matrix Editor. Opens the Matrix editor for the specified matrix. Returns to the program when user presses **OK**.

`EDITMAT matrixname`

FREEZE

This command prevents the display from being updated after the program runs. This allows you to view the graphics created by the program. Cancel **FREEZE** by pressing any key.

`FREEZE`

GETKEY

Waits for a key, then stores the keycode rc.p in *name*, where r is row number, c is column number, and p is key-plane number. The key-planes are designated 1 for unshifted, 2 for shifted, 4 for alpha-shifted, and 5 for both alpha-shifted and shifted.

`GETKEY name`

INPUT

Creates an input form with a title bar and one field. The field has a label and a default value. There is text help at the bottom of the form. The user enters a value and presses the **OK** menu

key. The value that the user enters is stored in the variable *name*. The *title*, *label*, and *help* items are text strings and need to be enclosed in double quotes.

Use `[SHIFT]CHARS` to type the quote marks " ".

```
INPUT name ; title , label ; help ; default
```

Example

```
INPUT R; "Circular Area";  
"Radius";  
"Enter Number";1:
```

MSGBOX

Displays a message box containing *textitem*. A text item consists of any number of expressions and quoted strings of text. The expressions are evaluated and turned into strings of text. For example,

"AREA IS: " 2+2 becomes "AREA IS: 4. Use

`[SHIFT]CHARS` to type the quote marks " ".

```
MSGBOX textitem :
```

Example:

```
1 ► A:
```

```
MSGBOX "AREA IS: "  $\pi * A^2$  :
```

You can also use the `NoteText` variable to provide text arguments. This can be used to insert line breaks. For example, press `[SHIFT]NOTE` and type AREA IS `[ENTER]`.

Now the position line

```
MSGBOX NoteText " "  $\pi * A^2$  :
```

will display the same message box as the previous example.

WAIT

Halts program execution for the specified number of seconds.

```
WAIT seconds
```

Stat-One and Stat-Two commands

The following commands are used for analysis of one-variable and two-variable statistical data. Refer to “Computing statistics (1VAR and 2VAR)” on page 8-106 for complete definitions of these variables.

Stat-One commands

DO1VSTATS

Calculates STATS using *datasetname* and stores the results in the corresponding variables: $N\Sigma$, $Tot\Sigma$, $Mean\Sigma$, $PVar\Sigma$, $SVar\Sigma$, $PSDev$, $SSDev$, $Min\Sigma$, $Q1$, Median, $Q3$, and $Max\Sigma$. *Datasetname* can be H1, H2, ..., or H5. *Datasetname* must define at least two data points.

DO1VSTATS *datasetname*

SETFREQ

Defines *datasetname* frequency according to *column* or value. *Datasetname* can be H1, H2,..., or H5, *column* can be C0-C9 and value can be any positive integer

SETFREQ *datasetname* ; *column*

or

SETFREQ *definition* ; *value*

SETSAMPLE

Defines *datasetname* sample according to *column*. *Datasetname* can be H1-H5, and *column* can be C0-C9.

SETSAMPLE *datasetname* ; *column*

Stat-Two commands

DO2VSTATS

Calculates STATS using *datasetname* and stores the results in corresponding variables: $MeanX$, ΣX , ΣX^2 , $MeanY$, ΣY , ΣY^2 , ΣXY , Corr, PCov, SCov, and Fit. *Datasetname* can be S1, S2,..., or S5. *Datasetname* must define at least four pairs of data points.

DO2VSTATS *datasetname*

SETDEPEND

Defines *datasetname* dependent *column*. *Datasetname* can be S1, S2, ..., or S5 and *column* can be C0 - C9.

SETDEPEND *datasetname* ; *column*

SETINDEP

Defines *datasetname* independent *column*. *Datasetname* can be S1, S2, ..., or S5 and *column* can be C0-C9.

SETINDEP *datasetname* ; *column*

Storing and retrieving variables in programs

The HP 39G/40G has both *Home* variables and *Aplet* variables. Home variables are used for real numbers, complex numbers, graphics, lists, and matrices. Home variables keep the same values in HOME and in aplets.

Aplet variables are those whose values depend on the current aplet. The aplet variables are used in programming to emulate the definitions and settings you make when working with aplets interactively.

You use the Variable menu (**[VARS]**) to retrieve either Home variables or aplet variables. See “The VARS menu” on page 11-159. Not all variables are available in every aplet. S1fit-S5fit, for example, are only available in the Statistics aplet.

Under each variable name is a list of the aplets where the variable can be used.

Plot-view variables

The following aplet variables control the Plot view.

Area **Function**

Contains the last value found by the Area function in Plot-FCN menu.

Axes **All Aplets**

Turns axes on or off.

From Plot Setup, check (or uncheck) **__AXES**.

or

In a program, type:

- 1 ► **AXES**—to turn axes on (default).
- 0 ► **AXES**—to turn axes off.

Connect **Function** **Parametric** **Polar** **Solve** **Statistics**

Draws lines between successively plotted points.

From Plot Setup, check (or uncheck) **__CONNECT**.

or

In a program, type

- 1 ► **CONNECT**—to connect plotted points (default, except in Statistics where the default is off).
- 0 ► **CONNECT**—not to connect plotted points.

Coord
Function
Parametric
Polar
Sequence
Solve
Statistics

Turns the coordinate-display mode in Plot view on or off.

From Plot view, use the Menu mean key to toggle coordinate display on an off.

In a program, type

- 1 ► Coord—to turn coordinate display on (default).
- 0 ► Coord—to turn coordinate display off.

Extremum
Function

Contains the last value found by the Extremum operation in the Plot-FCN menu.

FastRes
Function
Solve

Toggles resolution between plotting in every other column (faster), or plotting in every column (more detail).

From Plot Setup, choose Faster or More Detail.

or

In a program, type

- 1 ► FastRes—for faster (default).
- 0 ► FastRes—for more detail.

Grid
All Aplets

Turns the background grid in Plot view on or off. From Plot setup, check (or uncheck) `_GRID`.

or

In a program, type

- 1 ► Grid to turn the grid on.
- 0 ► Grid to turn the grid off (default).

Hmin/Hmax
Statistics

Defines minimum and maximum values for histogram bars.

From Plot Setup for one-variable statistics, set values for HRNG.

or

In a program, type

- n_1 ► Hmin
- n_2 ► Hmax
- where $n_2 > n_1$

Hwidth
Statistics

Sets the width of histogram bars.

From Plot Setup in 1VAR stats set a value for Hwidth

	<p>In a program, type</p> $n \triangleright \text{Hwidth}$
Indep <i>All Aplets</i>	<p>Defines the value of the independent variable used in tracing mode.</p> <p>In a program, type</p> $n \triangleright \text{Indep}$
InvCross <i>All Aplets</i>	<p>Toggles between solid crosshairs or inverted crosshairs. (Inverted is useful if background is solid).</p> <p>From Plot Setup, check (or uncheck) <code>__InvCross</code></p> <p>or</p> <p>In a program, type:</p> $1 \triangleright \text{InvCross} \text{---to invert the crosshairs.}$ $0 \triangleright \text{InvCross} \text{---for solid crosshairs (default).}$
Isect <i>Function</i>	<p>Contains the last value found by the Intersection function in the Plot-FCN menu.</p>
Labels <i>All Aplets</i>	<p>Draws labels in Plot view showing X and Y ranges.</p> <p>From Plot Setup, check (or uncheck) <code>__Labels</code></p> <p>or</p> <p>In a program, type</p> $1 \triangleright \text{Labels} \text{---to turn labels on.}$ $0 \triangleright \text{Labels} \text{---to turn labels off (default).}$
Nmin / Nmax <i>Sequence</i>	<p>Defines the minimum and maximum independent variable values. Appears as the NRNG fields in the Plot Setup input form.</p> <p>From Plot Setup, enter values for NRNG.</p> <p>or</p> <p>In a program, type</p> $n_1 \triangleright \text{Nmin}$ $n_2 \triangleright \text{Nmax}$ <p>where</p> $n_2 > n_1$

Recenter
All Aplets

Recenters at the crosshairs locations when zooming.

From Plot-Zoom-Set Factors, check (or uncheck) `_Recenter`

or

In a program, type

- 1 ► `Recenter`—to turn recenter on (default).
- 0 ► `Recenter`—to turn recenter off.

Root
Function

Contains the last value found by the `Root` function in the Plot-FCN menu.

S1mark-S5mark
Statistics

Defines the mark to use for statistics 2-variable scatter plots.

From Plot Setup for two-variable statistics, `S1mark-S5mark`, then choose a mark.

or

In a program, type

- `n` ► `S1mark`
where `n` is 1,2,3,...7

SeqPlot
Sequence

Toggles type of sequence plot: Stairstep or Cobweb.

From Plot Setup, select `SeqPlot`, then choose `Stairstep` or `Cobweb`.

or

In a program, type

- 1 ► `SeqPlot` for statistics.
- 2 ► `SeqPlot` for cobweb.

Simult
Function
Parametric
Polar
Sequence

Toggles between simultaneous and sequential graphing of all selected expressions.

From Plot Setup, check (or uncheck) `_SIMULT`

or

In a program, type

- 1 ► `Simult`—for simultaneous graphing.
- 0 ► `Simult`—for sequential graphing.

Slope
Function

Contains the last value found by the `Slope` function in the Plot-FCN menu.

StatPlot
Statistics

Toggles type of 1-variable statistics plot between Histogram or BoxWhisker.

From Plot Setup, select StatPlot, then choose Histogram or BoxWhisker.

or

In a program, type

1 ► SeqPlot for Histogram.

2 ► SeqPlot for BoxWhisker.

Umin/Umax
Polar

Defines the minimum and maximum independent values. Appears as the URNG field in the Plot Setup input form.

From the Plot Setup input form, enter values for URNG.

or

In a program, type

n_1 ► Umin

n_2 ► Umax

where $n_2 > n_1$

Ustep
Polar

Defines the step size for an independent variable.

From the Plot Setup input form, enter values for USTEP.

or

In a program, type

n ► Ustep

where $n > 0$

Tmin / Tmax
Parametric

Defines the minimum and maximum independent variable values. Appears as the TRNG fields in the Plot Setup input form.

From Plot Setup, enter values for TRNG.

or

In a program, type

n_1 **STO►** Tmin

n_2 **STO►** Tmax

where $n_2 > n_1$

Tracing
All Aplets

Turns tracing mode on or off in Plot view.

In a program, type

- 1 ► Tracing to turn Tracing mode on (default).
- 0 ► Tracing to turn Tracing mode off.

Tstep
Parametric

Defines the step size for an independent variable.

From the Plot Setup input form, enter values for TSTEP.

or

In a program, type

- n ► Tstep
- where $n > 0$

Xcross
All Aplets

Defines the horizontal coordinate of crosshairs. Only works with TRACE off.

In a program, type

- n ► Xcross

Ycross
All Aplets

Defines the vertical coordinate of crosshairs. Only works with TRACE off.

In a program, type

- n ► Ycross

Xtick
All Aplets

Defines the distance between tick marks for the horizontal axis.

From the Plot Setup input form, enter a value for Xtick.

or

In a program, type

- n ► Xtick where $n > 0$

Ytick
All Aplets

Defines the distance between tick marks for the vertical axis.

From the Plot Setup input form, enter a value for Ytick.

or

In a program, type

$n \triangleright Ytick$ where $n > 0$

Xmin / Xmax
All Aplets

Defines the minimum and maximum horizontal values of the plot screen. Appears as the XRNG fields (horizontal range) in the Plot Setup input form.

From Plot Setup, enter values for XRNG.

or

In a program, type

$n_1 \triangleright Xmin$

$n_2 \triangleright Xmax$

where $n_2 > n_1$

Ymin / Ymax
All Aplets

Defines the minimum and maximum vertical values of the plot screen. Appears as the YRNG fields (vertical range) in the Plot Setup input form.

From Plot Setup, enter the values for YRNG.

or

In a program, type

$n_1 \triangleright Ymin$

$n_2 \triangleright Ymax$

where $n_2 > n_1$

Xzoom
All Aplets

Sets the horizontal zoom factor.

From Plot-ZOOM-Set Factors, enter the value for XZOOM.

or

In a program, type

$n \triangleright XZOOM$

where $n > 0$

Yzoom
All Aplets

Sets the vertical zoom factor.

From Plot-ZOOM-Set Factors, enter the value for YZOOM.

or

In a program, type

$n \triangleright YZOOM$

Symbolic-view variables

The following applet variables contact the Symbolic view at the current applet.

Angle

All Applets

Sets the angle mode.

From Symbolic Setup, choose Degrees, Radians, or Grads for angle measure.

or

In a program, type

1 ► Angle for Degrees.

2 ► Angle for Radians.

3 ► Angle for Grads.

F1...F9, F0

Function

Can contain any expression in

Example

'SIN(X)' ► F1(X)

In the above example, you must put single quotes around the expression to keep it from being evaluated before it is stored. Use [SHIFT]CHARS to type the single quote mark.

X1, Y1...X9, Y9 X0, Y0

Parametric

Can contain any expression. Independent variable is T.

Example

'SIN(4*T)' ► Y1(T) : '2*SIN(6*T)' STO►
X1(T)

R1...R9, R0

Polar

Can contain any expression. Independent variable is θ .

Example

'2*SIN(2* θ)' ► R1(θ)

U1...U9, U0

Sequence

Can contain any expression. Independent variable is N.

Example

RECURSE (U, U(N-1)*N, 1, 2) ► U1(N)

E1...E9, E0

Solve

Can contain any equation or expression. *Independent variable is selected by highlighting it in Numeric View.*

Example

'X+Y*X-2=Y' ► E1

S1fit...S5fit
Statistics

Defines the type of fit to be used by the FIT operation to plot 2VAR statistics.

From Symbolic Setup view, specify the fit in the field for S1FIT, S2FIT, etc.

or

In a program, store one of the following constant names or # into a variable S1fit, S2fit, etc.

1. Linear
2. LogFit
3. ExpFit
4. Power
5. QuadFit
6. Cubic
7. Logist
8. User defined

Example

Cubic ► S2fit

or

6 ► S2fit

Numeric-view variables

The following applet variables control the Numeric view. The value of the variable applies to the current applet only.

C1...C9, C0
Statistics

C0 through C9, for columns of data. Can contain lists.

Enter data in the Numeric view or

In a program, type

LIST ► Cn

where $n = 0, 1, 2, 3 \dots 9$

Digits
All Applets

Number of decimal places to use for Number format.

From Solve's Numeric Setup view, enter a value in the second field of Number Format.

or

Format
All Aplets

In a program, type

$n \blacktriangleright$ Digits where $0 < n < 11$.

Except in Solve, the value of Digits takes effect only after the current aplet is saved with a new name. Until then, HDigit is in effect.

Defines the number display format.

From Solve's Numeric Setup view, choose Standard, Fixed, Scientific, or Engineering in the Number Format field.

or

In a program, store the constant name (or its number) into the variable Format.

Standard=1, Fixed=2, Scientific=3, Engineering=4.

Note: Fraction is not a valid mode in aplets.

Except in Solve, the value of Format takes effect only after the current aplet is saved with a new name. Until then, HFormat is in effect. Example

Scientific \blacktriangleright Format

or

3 \blacktriangleright Format

NumCol
All Aplets

All Aplets

Defines the highlighted column in Numeric view.

In a program, type

$n \blacktriangleright$ NumCol

where n can be 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

NumFont
Function
Parametric
Polar
Sequence
Statistics

Toggles the font size in Numeric view. Does not appear in the Num Setup input form. Corresponds to the BIG key in Numeric view.

In a program, type

0 \blacktriangleright NumFont for small (default).

1 \blacktriangleright NumFont for big.

NumIndep*Function**Parametric**Polar**Sequence*

List of independent values used by Build Your Own Table.

In a program, type

LIST ► NumIndep

NumRow*All Aplets*

Defines the highlighted row in Numeric view.

In a program, type

n ► NumRow

NumStart*Function**Parametric**Polar**Sequence*

Defines the starting value for table in Numeric view.

From Num Setup, enter a value for NUMSTART.

or

In a program, type

n ► NumStart

NumStep*Function**Parametric**Polar**Sequence*

Defines the step size (increment value) for an independent variable in Numeric view.

From Num Setup, enter a value for NUMSTEP.

or

In a program, type

n ► NumStep

where $n > 0$

NumType*Function**Parametric**Polar**Sequence*

Choose a table format.

From Num Setup, choose Automatic or Build Your Own.

or

In a program, type

0 ► NumType for Build Your Own.

1 ► NumType for Automatic (default).

NumZoom*Function**Parametric**Polar**Sequence*

Defines the Zoom factor in the Numeric view.

From Num Setup, type in a value for NUMZOOM.

or

In a program, type

$n \blacktriangleright \text{NumZoom}$
where $n > 0$

StatMode
Statistics

Toggles between 1-variable and 2-variable statistics in the Statistics applet. Does not appear in the Plot Setup input form. Corresponds to the 1VAR and 2VAR keys in Numeric View.

In a program, store the constant name (or its number) into the variable StatMode. 1VAR=1, 2VAR=2.

Example

1VAR \blacktriangleright StatMode
or
1 \blacktriangleright StatMode

Note variables

The following applet variable is available in Note view.

NoteText
All Aplets

Use NoteText to recall text previously entered in Note view.

Sketch variables

The following applet variables are available in Sketch view.

Page
All Aplets

Defines a *page* in a sketch set. A sketch set can contain up to 10 graphics. The graphics can be viewed one at a time using the \blacktriangle PAGE and PAGE \blacktriangledown keys.

The Page variable references the currently-displayed page of a sketch set.

In a program, type

graphicname \blacktriangleright Page

PageNum
All Aplets

Index for referring to a particular page of the sketch set (in Sketch view). Make the page numbered n .

In a program, type the page that is shown when $\boxed{\text{SHIFT}}\text{SKETCH}$ is pressed.

$n \blacktriangleright \text{PageNum}$

