

# 1. BRIEF GLOSSARY

---

The following definitions will acquaint you with general terms used throughout this primer. You will probably want to read through them now, and use this chapter as a reference while you read through the rest of the primer.

advising	A Medley facility for specifying function modifications without necessarily knowing how a particular function works or even what it does. Even system functions can be changed with advising.
argument	A piece of information given to a Lisp function so that it can execute successfully. When a function is explained in the primer, the arguments that it requires will also be given. Arguments are also called Parameters.
atom	The smallest structure in Lisp; like a variable in other programming languages, but can also have a property list and a function definition.
Background Menu	The menu that appears when the mouse is not in any window and the right mouse button is pressed.
binding	The value of a variable. It could be either a local or a global variable. See unbound.
bitmap	A rectangular array of "pixels," each of which is on or off representing one point in the bitmap image.
BREAK	An Lisp function that causes a function to stop executing, open a Break window, and allows you to find out what is happening while the function is halted.
Break Window	A window that opens when an error is encountered while running your program (i.e., when your program has broken). There are tools to help you debug your program from this window. This is explained further in Chapter 14.
browse	To examine a data structure by use of a display that allows you to "move" around within the data structure.
button	(1) (n.) A key on a mouse. (2) (v.t.) To press one of the mouse keys when making a selection.
CAR	A function that returns the head or first element of a list. See CDR.
caret	The small blinking arrowhead that marks where text will appear when it is typed in from the keyboard.
CDR	A function that returns the tail (that is, everything but the first element) of a list. See CAR.

CLISP	A mechanism for augmenting the standard Lisp syntax. One such augmentation included in Interlisp is the iterative statement. See Chapter 9.
cr	Press your Return key.
datatype	(1) The kind of a datum. In Interlisp, there are many system-defined datatypes, e.g., Floating-Point, Integer, Atom, etc.  (2) A datatype can also be user-defined. In this case, it is like a record made up from system types and other user-defined datatypes.
DWIM	"Do-what-I-mean." Many errors made by Medley users could be corrected without any information about the purpose of the program or expression in question (e.g., misspellings, certain kinds of parenthesis errors). The DWIM facility is called automatically whenever an error occurs in the evaluation of an Interlisp expression. If DWIM is able to make a correction, the computation continues as though no error had occurred; otherwise, the standard error mechanism is invoked.
error	Occasionally, while a program is running, an error may occur which will stop the computation. Interlisp provides extensive facilities for detecting and handling error conditions, to enable the testing, debugging, and revising of imperfect programs.
evaluate or EVAL	To find the value of a form. For example, if the variable <i>x</i> is bound to 5, we get 5 by evaluating <i>x</i> . Evaluation of a Lisp function involves evaluating the arguments and then applying the function.
Executive Window	This is your main window, where you will run functions and develop your programs. This is the window that the caret is in when you turn on your machine and load Medley.
file package	A set of functions and conventions that facilitate the bookkeeping involved with working in a large system consisting of many source code files and their compiled counterparts. Essentially, the file package keeps track of where things are and what things have changed. It also keeps track of which files have been modified and need to be updated and recompiled.
form	Another way of saying s-expression. A Lisp expression that can be evaluated.
function	A piece of Lisp code that executes and returns a value.
history	The programmer's assistant is built around a memory structure called the history list. The history functions (e.g. FIX, UNDO, REDO) are part of this assistant. These operations allow you to conveniently rework previously specified operations.
History List	As you type on the screen, you will notice a number followed by a slash, followed by another number. The first number is the exec number, the second is the event number. Each number, and the information on that line, is stored sequentially as the History List Using the History List, you

---

	can easily reexecute lines typed earlier in a work session. See Chapter 2.
icon	A pictorial representation, usually of a shrunken window.
inspector	An interactive display program for examining and changing the parts of a data structure. Medley has inspectors for lists and other data types.
iterative statement	(also called i.s.) A statement in Interlisp that repetitively executes a body of code. For example, <code>(for x from 1 to 5 do (PRINT x))</code> is an i.s.
iterative variable	(also called i.v.) Usually, an iterative statement is controlled by the value that the i.v. takes on. In the iterative statement example above, <code>x</code> is the iterative variable because its value is being changed by each cycle through the loop. All iterative variables are local to the iterative statement where they are defined.
Lisp	Family of languages invented for "list processing." These languages have in common a set of basic primitives for creating and manipulating symbol structures. Interlisp-D is an implementation of the Lisp language together with an environment (set of tools) for programming, and a set of packages that extend the functionality of the system.
list	A collection of atoms and lists; a list is denoted by surrounding its contents with a pair of parentheses.
Masterscope	A program analysis tool. When told to analyze a program, Masterscope creates a database of information about the program. In particular, Masterscope knows which functions call other functions and which functions use which variables. Masterscope can then answer questions about the program and display the information with a browser.
menu	A way of graphically presenting you with a set of options. There are two kinds of menus: pop-up menus are created when needed and disappear after an item has been selected; permanent menus remain on the screen after use until deliberately closed.
mouse	The mouse is the box attached to your keyboard. It controls the movement of the cursor on your screen. As you become familiar with the mouse, you will find it much quicker to use the mouse than the keyboard.
Mouse Cursor	The small arrow on the screen that points to the northwest.
Mouse Cursor Icons	Four types of mouse cursor icons are shown below.
	Wait. The processor is busy.
	The Mouse Confirm Cursor. It appears when you have to confirm that the choice you just made was correct. If it was, press the left button. If the choice was not correct, press the right button to abort.



This means "sweep out" the shape of the window. To do this, move the mouse to a position where you want a corner. Press the left mouse button, and hold it down. Move the mouse diagonally to sketch a rectangle. When the rectangle is the desired size and shape, release the left button.



This is the "move window" prompt. Move the mouse so that the large "ghost" rectangle is in the position where you want the window. When you click the left mouse button, the window will appear at this new location.

NIL

NIL is the Lisp symbol for the empty list. It can also be represented by a left parenthesis followed by a right parenthesis (). It is the only expression in Lisp that is both an atom and a list.

pixel

Pixel stands for "picture element." The computer monitor screen is made up of a rectangular array of pixels. Each pixel corresponds to one bit. When a bit is turned on (i.e., set to 1), the pixel on the screen represented by this bit is black.

pretty printing

Pretty printing refers to the way Lisp functions are printed with special indentation, to make them easier to read. Functions are pretty printed in the structure editor, SEdit (see Chapter 7). You can pretty print uncompiled functions by calling the function `PP` with the function you would like to see as an argument, i.e. (`PP` function-name). For an example of this, see Figure 1.5.

```

Exec (INTERLISP)

105+ PP PP
FNS definition for PP:
(DEFINEQ
  (PP
    [NLAMBDA X (* --)
      (DECLARE (LOCALVARS . T))
      (MAPC (NLAMBDA (ARGS X)
              (FUNCTION (LAMBDA (NAME)
                        (for TYPE in (TYPESOF NAME NIL '(FIELDS)
                                      'CURRENT)
                          do (CL:FORMAT *TERMINAL-IO* "~A definition for
                                      ~S:~%" TYPE NAME)
                              (SHOWDEF NAME TYPE))
                        )
              )
            )
    )
  )
107+

```

Figure 1.5. Example of Pretty Printing Function `PP`

Programmer's  
Assistant

The programmer's assistant accesses the History List to allow you to `FIX`, `UNDO`, and/or `REDO` your previous expressions typed to the executive window (see Chapter 2).

Prompt Window

The narrow black window at the top of the screen. It displays system prompts, or prompts you have developed (see Figure 1.6).



Figure 1.6. Prompt Window

- property list** A list of the form ( <property-name1> <property-value1> <property-name2> <property-value2> ...) associated with an atom. It accessed by the functions `GETPROP` and `PUTPROP`.
- record** A record is a data structure that consists of named "fields". Accessing elements of a record can be separated from the details of how the data structure is actually stored. This eliminates many programming details. A record definition establishes a record template, describing the form of a record. A record instance is an actual record storing data according to a particular record template. (See `datatype`, second definition.)

**Right Button Default Window Menu**

This is the menu that appears when the mouse is in a window, and the right mouse button is pressed. It looks like the menu in Figure 1.7. If this menu does not appear when you press the right button of the mouse and the mouse is in the window, move the mouse so that it is pointing to the title bar of the window, and press the right button.

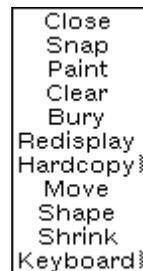


Figure 1.7. Right Button Default Window Menu

- s-expression** Short for "symbolic expression". In Lisp, this refers to any well-formed collection of left parentheses, atoms, and right parentheses.
- stack** A pushdown list. Whenever a function is entered, information about that specific function call is pushed onto (i.e., added to the front of) the stack. This information includes the variable names and their values associated with the function call. When the function is exited, that data is popped off the stack.
- sysout** A file containing a whole Lisp environment: namely, everything you defined or loaded into the environment, the windows that appeared on the screen, the amount of memory used, and so on. Everything is stored in the `sysout` file exactly as it was when the function `SYSOUT` was called.



- TRACE** A function that creates a trace of the execution of another function. Each time the traced function is called, it prints out the values of the arguments it was called with, and prints out the value it returns upon completion.
- unbound** Without value; an atom is unbound if a value has never been assigned to it.
- window** A rectangular area of the screen that acts as the main display area for some Lisp process,