[9.5] Default values for variables in *Request*

Request is used in dialog boxes for prompt for user input. You can save your users a lot of time if you provide default values, and remember those defaults. However, there is a problem if the variables are numbers, since *Request* returns strings. This code gets around this problem:

```
string(x) → x

dialog
 request "Enter x",x
enddlog

expr(x) → x
```

This code assumes that *x* has a numeric value on entry, which is the default. The *string()* function converts *x* to a string for use in *Request*. When the dialog box exits, *expr()* converts the string to a number so you can use it in calculations.

If *x* is a local variable, you need to initialize it to the default first. If *x* is a global variable, the last-used value is the default the next time you run your program. While this is very convenient, it does take up RAM.

If you do use a global variable, it needs to be initialized the first time you run the program, otherwise the dialog box will look like this:

```
Enter x: x
```

The variable name will be displayed. This can be avoided like this:

```
if GetType(x)="NONE":0->x
```

This statement is executed before the *string()* function. The variable is initialized if it doesn't exist, otherwise it is unchanged.

If your program has a lot of inputs, consider saving them as a list. This creates a little more work for you, as the programmer, but it reduces the clutter of global variables in a folder. As an example, suppose that your program needs four inputs from the user. This example shows the basic idea.

```
local w,x,y,z                       © Define local variables for user inputs

if gettype(defaults)="NONE"         © If the user defaults variable doesn't exist,
 {1,2,3,4}→defaults                 © ... create and initialize it

string(defaults[1])→w               © Initialize local copies of variables
string(defaults[2])→x
string(defaults[3])→y
string(defaults[4])→z

Dialog                              © Prompt for new values
 Request "Enter w",w
 Request "Enter x",x
 Request "Enter y",y
 Request "Enter z",z
EndDlog

if ok=0:return                      © Just quit if user presses [ESC]

expr(w)→w : w→defaults[1]           © Convert string to expression; update global copy
expr(x)→x : x→defaults[2]
```

```
expr(y)→y : y→defaults[3]
expr(z)→z : z→defaults[4]
```

In this example, *defaults* is the name of the global variable which holds the default values. If your inputs are lists or matrices, see tip [3.24] for a method to store those data types in the default list.