

## [6.40] Write functions with multiple input argument types

Many built-in functions operate on numbers, expressions and lists. For example,

<code>sin(a)</code>	returns	<code>sin(a)</code>
<code>sin(.2)</code>	returns	<code>0.1987</code>
<code>sin({0,.2})</code>	returns	<code>{0, 0.1987}</code>

This is very convenient in that you can use a single function to handle various argument types. You can accomplish this convenience with your own functions, as shown in this demonstration function:

```
f1demo(xx)
Func
©Demo program to evaluate function of expression, list or matrix.
©29mar01/dburkett@infinet.com

local f1,xxt,j,k,r,xxr,xxc

©Define the function to evaluate
define f1(x)=func
  ln(x)
endfunc

©Get the input argument type
gettype(xx)→xxt

©Evaluate the function
if xxt="NUM" or xxt="EXPR" or xxt="VAR" then
  return f1(xx)
elseif xxt="LIST" then
  return seq(f1(xx[j]),j,1,dim(xx))
elseif xxt="MAT" then
  rowdim(xx)→xxr
  coldim(xx)→xxc
  newmat(xxr,xxc)→r
  for j,1,xxr
    for k,1,xxc
      f1(xx[j,k])→r[j,k]
    endfor
  endfor
  return r
else
  return "f1demo: type error"
endif

EndFunc
```

This function accepts arguments that may be expressions, numbers, lists or matrices. If the argument is a list or matrix, the function is applied to each element. The function to be evaluated is defined as a local function *f1()*; as an example, I use the *ln()* function. Your functions will probably be more complicated.

The argument type is determined and saved in variable *xxt*. If the argument is a number, expression or variable name, the argument is evaluated by the function and returned. If the argument is a list, the *seq()* function is used to evaluate each list element, and return the results as a list. If the argument is a matrix, two nested *For ... EndFor* loops are used to evaluate the function of each matrix element, and the matrix is returned.

If the argument type is not supported, *f1demo()* returns the text string *"f1demo: type error"*. The calling program can test the type of the returned result and determine if an error occurred.