You can add similar capabilities to user-defined functions using the **Prompt** (see page V-782) and **DoPrompt** (see page V-167) keywords.

Macros provide very limited programming features so, with rare exceptions, you should program using functions.

Function Commands

A function is a routine that directly returns a numeric or string value. There are three classes of functions available to Igor users:

- Built-in
- External (XFUNCs)
- User-defined

Built-in numeric functions enjoy one advantage over external or user-defined functions: a few come in real and complex number types and Igor automatically picks the appropriate version depending on the current number type in an expression. External and user-defined functions must have different names when different types are needed. Generally, only real user and external functions need be provided.

For example, in the wave assignment:

```
wave1 = enoise(1)
```

if wave1 is real then the function enoise returns a real value. If wave1 is complex then enoise returns a complex value.

You can use a function as a parameter to another function, to an operation, to a macro or in an arithmetic or string expression so long as the data type returned by the function makes sense in the context in which you use it.

User-defined and external functions can also be used as commands by themselves. Use this to write a user function that has some purpose other than calculating a numeric value, such as displaying a graph or making new waves. Built-in functions cannot be used this way. For instance:

```
MyDisplayFunction(wave0)
```

External and user-defined functions can be used just like built-in functions. In addition, numeric functions can be used in curve fitting. See Chapter IV-3, **User-Defined Functions** and **Fitting to a User-Defined Function** on page III-190.

Most functions consist of a function name followed by a left parenthesis followed by a parameter list and followed by a right parenthesis. In the wave assignment shown at the beginning of this section, the function name is `enoise`. The parameter is 1. The parameter is enclosed by parentheses. In this example, the result from the function is assigned to a wave. It can also be assigned to a variable or printed:

```
K0 = enoise(1)
Print enoise(1)
```

User and external functions, but not built-in functions, can be executed on the command line or in other functions or macros without having to assign or print the result. This is useful when the point of the function is not its explicit result but rather its side effects.

Nearly all functions require parentheses even if the parameter list is empty. For example the function date() has no parameters but requires parentheses anyway. There are a few exceptions. For example the function Pi returns π and is used with no parentheses or parameters.

Igor’s built-in functions are described in detail in Chapter V-1, **Igor Reference**.

Parameter Lists

Parameter lists are used for operations, functions, and macros and consist of one or more numbers, strings, keywords or names of Igor objects. The parameters in a parameter list must be separated with commas.