- There are no extraneous blue/underlined characters, such as tabs or spaces, before or after the link. (You can not identify the text format of spaces and tabs by looking at them. Check them by selecting them and then using the Set Text Format dialog.)
- There are no duplicate topics. If you specify a link in topic[subtopic] form and there are two topics with the same topic name, Igor may not find the subtopic.

Help for User-Defined Functions

You can provide help for user-defined functions in your package by including a topic similar to the built-in Functions topic in your help file. In Igor Pro 9.00 or later, the user can go to the help for your user-defined function by selecting it in a procedure or notebook window, right-clicking, and choosing Help For <topic name>.

Here is how add help for your user-defined functions:

1. Display the built-in Functions topic.
2. Copy that topic paragraph and the first subtopic to the clipboard.
3. Paste into your help file.
4. Change the topic name to a distinctive name such as My Package Functions.
6. Edit the subtopic to provide help for one of your user-defined function.
7. Add additional subtopics by copying and pasting the original and editing as needed.

Make sure that your package name and user-defined function names are distinctive to avoid collisions with other packages.

The Insert Template For item in the contextual menu gets information from the procedure file and does not depend on help.

Creating Formatted Text

The printf, sprintf, and fprintf operations print formatted text to Igor's history area, to a string variable or to a file respectively. The wfprintf operation prints formatted text based on data in waves to a file.

All of these operations are based on the C printf function which prints the contents of a variable number of string and numeric variables based on the contents of a format string. The format string can contain literal text and conversion specifications. Conversion specifications define how a variable is to be printed.

Here is a simple example:

```
printf "The minimum is %g and the maximum is %g\r", V_min, V_max
```

In this example, the format string is `"The minimum is %g and the maximum is %g\r"` which contains some literal text along with two conversion specifications — both of which are "`%g`" — and an escape code ("`\r`") indicating "carriage-return". If we assume that the Igor variable V_min = .123 and V_max = .567, this would print the following to Igor's history area:

```
The minimum is .123 and the maximum is .567
```

We could print this output to an Igor string variable or to a file instead of to the history using the **sprintf** (see page V-902) or **fprintf** (see page V-260) operations.

Printf Operation

The syntax of the printf operation is:

```
printf format [, parameter [, parameter ]. . .]
```

where *format* is the format string containing literal text or format specifications. The number and type of parameters depends on the number and type of format specifications in the format string. The parameters, if any, can be literal numbers, numeric variables, numeric expressions, literal strings, string variables or string expressions.