If you hover the mouse cursor over a completion option or use the arrow keys to change the highlighted option, a tool tip is displayed that shows the template of the selected option.

You can adjust Command completion settings in Completion tab of the Text Editing category of the Miscellaneous Settings dialog. You can separately enable or disable completion in procedure windows and the command line. You can control the delay between the last keypress and the display of the completion options popup and whether an opening parenthesis should be appended when you insert an item that requires parentheses.

Completion is currently not supported for objects such as waves and variables. Support for these types may be added in a future version of Igor.

Command completion is not context sensitive. This means that you will sometimes be offered completion options that do not make sense in the context of the text you are entering. For example, if you type "Display/HID", you might get the following completion options: HideIgorMenus, HideInfo, HideProcedures, HideTools". Those options are not valid as a flag for the Display operation, but the command completion algorithm isn't able to filter them out.

Types of Commands

There are three fundamentally different types of commands that you can execute from the command line:

- assignment statements
- operation commands
- user-defined procedure commands

Here are examples of each:

```
wave1 = sin(2*pi*freq*x)       // assignment statement

Display wave1,wave2 vs xwave   // operation command

MyFunction(1.2,"hello")        // user-defined procedure command
```

As Igor executes commands you have entered, it must determine which of the three basic types of commands you have typed. If a command starts with a wave or variable name then Igor assumes it is an assignment statement. If a command starts with the name of a built-in or external operation then the command is treated as an operation. If a command begins with the name of a user-defined macro, user-defined function or external function then the command is treated accordingly.

Note that built-in functions can only appear in the right-hand side of an assignment statement, or as a parameter to an operation or function. Thus, the command:

```
sin(x)
```

is not allowed and you will see the error, “Expected wave name, variable name, or operation.” On the other hand, these commands are allowed:

```
Print sin(1.567)               // sin is parameter of print operation
wave1 = 5*sin(x)               // sin in right side of assigment
```

If, perhaps due to a misspelling, Igor can not determine what you want to do, it will display an error dialog and the error will be highlighted in the command line.

Assignment Statements

Assignment statement commands start with a wave or variable name. The command assigns a value to all or part of the named object. An assignment statement consists of three parts: a destination, an assignment operator, and an expression. For example: