Multiple Return Syntax

In Igor Pro 8 or later, you can create a function with multiple return values using this syntax:

```
Function [ <output parameter list> ] <function name> ( <input parameter list> )
```

The square brackets are part of the syntax.

Both the output parameter list and the input parameter list must be specifed using inline syntax (see **Inline Parameters** on page IV-33). The entire function declaration must appear on one line.

You can return all output values using one return statement:

```
return [ <value list> ]
```

For example:

```
Function [ Variable v, String s ] Subroutine( Variable a )
   return [1+a, "hello"]
End

Function CallingRoutine()
   Variable v1
   String s1
   [v1, s1] = Subroutine(10)
   Print v1, s1
End
```

In Igor Pro 9.00 and later, you can declare the return variables in the destination parameter list like this:

```
Function CallingRoutine()
   [Variable v1, String s1] = Subroutine(10)
   Print v1, s1
End
```

You can set the return values individually without using a return statement like this:

```
Function [ Variable v, String s ] Subroutine( Variable a )
   v = 1 + a
   s = "hello"
End
```

The output parameter list can include numeric and string variables as well as structures. Any type that can be used as a pass-by-reference parameter (see **Pass-By-Reference** on page IV-59) can be used in the output parameter list.

Multiple Return Syntax Example

This example illustrates using structure and WAVE references as output parameters:

```
Structure DemoStruct
   Variable num
   String str
EndStructure

Function [STRUCT DemoStruct sOut, WAVE/T wOut] Subroutine(Variable v1)
   STRUCT DemoStruct s
   s.num = 4 + v1
   s.str = "from Subroutine"

   Make/O/T tw = {"text wave point 0","text wave point 1"}

   return [ s, tw ]
End

Function CallingRoutine()
```