Visual Basic Procedures and Functions

The general PROCEDURE / FUNCTION RULE:

Procedures/Functions should be designed to perform a single, well-defined task which is logically coherent and the name should tell what that task is.

Visual Basic has two types of general procedures.

1) Sub Procedure ( commonly called a Procedure )
   Procedure is designed to perform an action, and may or may not pass anything back.

2) Function Procedure ( commonly called a Function )
   Function is also designed to perform an action, but always returns one item back.

Procedure Syntax

SYNTAX - (Declaring a Procedure)

[Private] Sub ProcedureName ( formal parameter list )
   local variables declared
   statements
   End Sub

SYNTAX - (Calling a Procedure)

   ProcedureName ( actual parameter list )

Function Syntax

SYNTAX - (Declaring a Function)

[Private] Function FunctionName ( formal parameter list ) As DataType
   local variables declared
   statements
   return Value
   End Function

SYNTAX - (Calling a Function)

   VName = FunctionName ( actual parameter list )

Comments

1) There are two kinds of parameters.
   - Value Parameters – used to pass information into procedure or function, uses ByVal.
   - Reference Parameters – primarily used to pass information out of procedure or function, but may also be used to pass data into procedure or function, uses ByRef.

Syntax for Parameters in Formal Parameter List (syntax is for procedure, same for function)

Sub ProcedureName ( ByVal Parameter1 As DataType, ByRef Parameter2 As DataType)
2) Formal parameter list tells compiler following:
   a) variable names used in parameter list
   b) data type associated with each variable in list
   c) the variables location in list
   d) what kind of parameters are they, i.e. Value or Reference Parameters
   e) how many parameters there are

3) Variables in formal parameter list are also considered local variables to function or procedure.

4) Correspondence Rules between formal and actual parameters (generally true)
   a) Actual parameter list must be the same size and order as the formal list.
   b) Actual parameter that corresponds to a VALUE formal parameter:
      - may be a data type compatible to formal parameter
      - may be a variable, expression, or constant
   c) Actual parameter that corresponds to a REFERENCE formal parameter
      - should be of the same data type as the formal parameter
      - must be a variable itself, not an expression or constant