



SUBPROGRAMAS / SUBPROCEDIMIENTOS / SUBROUTINAS

¿Por qué los utilizamos? Estas son unas de las razones:

1. **Modularización.** Cada subprograma tiene una misión muy concreta, de modo que nunca tiene un número de líneas excesivo y siempre se mantiene dentro de un tamaño manejable. Además, un mismo subprograma puede ser llamado muchas veces en un mismo programa, e incluso puede ser reutilizado por otros programas.
2. **Ahorro de memoria y tiempo de desarrollo.** En la medida en que un mismo subprograma es utilizado muchas veces, el número total de líneas de código del programa disminuye, y también lo hace la probabilidad de introducir errores en el programa.
3. **Independencia de datos.**

Existen 2 tipos de Subprogramas: FUNCIONES y PROCEDIMIENTOS

	FUNCIONES	PROCEDIMIENTOS
SINTAXIS	Sintaxis: <pre>Function miFuncion ([parámetros]) [As tipo] [sentencias] [miFuncion = expresion] End Function</pre> El nombre de la función, que es el valor de retorno, actúa como una variable dentro del cuerpo de la función. (*) Mirar teoría Parámetros	Sintaxis: <pre>Sub nombre ([parámetros]) [instrucciones] End Sub</pre> (*) Mirar teoría Parámetros
VALORES DEVUELTOS	Las funciones DEVUELVEN SIEMPRE un valor. Por eso a la hora de INVOCARLAS se encuentran SIEMPRE a la derecha de una expresión (no pueden aparecer “solas”): En ejecución, la llamada a la función se sustituye por el valor que devuelve dicha función Ej: Pagar=<i>CalculoIVA(importe, iva)</i> <u>No se invocan con CALL</u>	Los procedimientos NO DEVUELVEN NUNCA ningún valor. Por eso a la hora de INVOCARLOS aparecen “solos”, sin estar contenidos en ninguna expresión: Ej1: Call <i>calcularRectangulo</i> Ej2: Call <i>calcularTriangulo</i> Ej3 Call <i>miProcedimiento(dato1)</i> <u>Se invocan con CALL</u>



PARÁMETROS	Admiten el paso de PARÁMETROS, tanto por VALOR como por REFERENCIA.	Admiten el paso de PARÁMETROS, tanto por VALOR como por REFERENCIA.
EJEMPLOS PROGRAMAS	<pre>Option Explicit Private Function Factorial(ByVal X As Integer) As Long Dim i As Integer, xfact As Long xfact = 1 for i=1 to x step 1 xfact = xfact * i next i Factorial = xfact End Function Private Sub cmdPulsame_Click() Dim A As Integer Dim B As Integer, resultado As Single Dim A_Fact, B_Fact, A_menos_B_Fact As Long A = InputBox("Introduce un número:") B = InputBox("Introduce un número:") A_Fact = Factorial(A) B_Fact = Factorial(B) A_Menos_B_Fact = Factorial(A-B) resultado = A_fact / (B_Fact * A_Menos_B_Fact) MsgBox("El resultado de la combinación : " & resultado) End Sub</pre>	<pre>Option Explicit Dim opcion Private Sub cmdCalcular_Click() If opcion = 1 Then Call calcularRectangulo Else Call calcularTriangulo End If End Sub Sub calcularRectangulo() txtArea.Text = Val(txtBase.Text)*Val(txtAltura.Text) End Sub Sub calcularTriangulo() txtArea.Text = (Val(txtBase.Text) * Val(txtAltura.Text) / 2) End Sub Private Sub cmdSeleccionar_Click() opcion = InputBox("Seleccione una de estas:" + _ vbCrLf + "1.- Rectangulo" + vbCrLf + _ "2.- Triangulo", "Introduzca una opcion") End Sub</pre>



EJEMPLOS PROGRAMAS	Function Raiz (N As Double) As Double If N < 0 Then Exit Function Else Raiz = Sqr(N) End Function Cuadrada = Raiz(Num)	
--------------------	--	--