



EBook Gratis

APRENDIZAJE

Visual Basic 6

Free unaffiliated eBook created from
Stack Overflow contributors.

#vb6

Tabla de contenido

| | |
|---|-----------|
| Acerca de..... | 1 |
| Capítulo 1: Comenzando con Visual Basic 6..... | 2 |
| Observaciones..... | 2 |
| Examples..... | 2 |
| Instalación o configuración..... | 2 |
| Hola Mundo..... | 2 |
| Capítulo 2: Instalando VB6 en Windows 10..... | 3 |
| Examples..... | 3 |
| Asistente de instalación..... | 3 |
| Capítulo 3: Procedimientos de Función..... | 4 |
| Introducción..... | 4 |
| Sintaxis..... | 4 |
| Observaciones..... | 4 |
| Examples..... | 4 |
| Creando y llamando a una función..... | 4 |
| Capítulo 4: Sintaxis basica..... | 6 |
| Examples..... | 6 |
| declaración if / else..... | 6 |
| en bucle..... | 6 |
| Hacer bucle..... | 6 |
| Declaración de caso selecto..... | 7 |
| Capítulo 5: Variables..... | 8 |
| Examples..... | 8 |
| Tipos de variables..... | 8 |
| Booleano..... | 8 |
| Entero..... | 9 |
| Cuerda..... | 9 |
| Creditos..... | 10 |

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [visual-basic-6](#)

It is an unofficial and free Visual Basic 6 ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Visual Basic 6.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Comenzando con Visual Basic 6

Observaciones

Esta sección proporciona una descripción general de qué es vb6 y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de vb6, y vincular a los temas relacionados. Dado que la Documentación para vb6 es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación o configuración

Instrucciones detalladas sobre cómo configurar o instalar vb6.

Hola Mundo

```
' A "Hello, World!" program in Visual Basic.  
Module Hello  
  Sub Main()  
    MsgBox("Hello, World!") ' Display message on computer screen.  
  End Sub  
End Module
```

Lea [Comenzando con Visual Basic 6 en línea](https://riptutorial.com/es/vb6/topic/3439/comenzando-con-visual-basic-6):

<https://riptutorial.com/es/vb6/topic/3439/comenzando-con-visual-basic-6>

Capítulo 2: Instalando VB6 en Windows 10

Examples

Asistente de instalación

[Asistente de instalación de Visual Studio 6.0](#)

De forma predeterminada, los siguientes paquetes no se instalan correctamente en Windows 10:

- Visual Studio 6 Enterprise
- Visual Studio 6 Professional
- Visual Basic 6 Enterprise
- Visual Basic 6 Profesional

Para instalar los paquetes anteriores, deberá realizar numerosos ajustes y trucos en el registro, o usar el fantástico [asistente de instalación de Visual Studio 6.0](#) de Giorgio Brausi.

Necesitará los siguientes artículos antes de comenzar:

- Sus CD y claves originales de Visual Studio / Basic Program
- Sus CD de MSDN originales
- Visual Studio Service Pack 6
- [Asistente de instalación de Visual Studio 6.0](#)
- En Windows 10, versión 1511 o posterior, necesitarás derechos de administrador.

El [asistente](#) lo guiará a través de los pasos necesarios para una instalación exitosa de Visual Basic 6.

Tenga en cuenta que la instalación de las aplicaciones de servidor no es posible actualmente.

Lea [Instalando VB6 en Windows 10 en línea](#): <https://riptutorial.com/es/vb6/topic/4230/instalando-vb6-en-windows-10>

Capítulo 3: Procedimientos de Función

Introducción

Función es una serie de sentencias encerradas entre las sentencias "Función" y "Función Final".

La función realiza una actividad y devuelve el control a la persona que llama. Cuando devuelve el control, también devuelve un valor al código de llamada.

Puede definir una función en una clase, estructura y módulo. Por defecto es público. Esto significa que puede llamarlo desde cualquier lugar de su aplicación que tenga acceso a la clase, estructura o módulo en el que lo definió.

Sintaxis

- [Modificadores] Función Name_Of_The_Function [(Arg_List)] As Return_Type
- [Declaraciones]
- Función final

Observaciones

- Los dos modificadores de función utilizados en estos ejemplos son públicos y privados. Estos modificadores definen el alcance de la función.
- Las funciones con un ámbito privado solo se pueden llamar desde el archivo de origen desde donde se definieron. En nuestro caso se puede consultar en el módulo. Y no puede ser llamado fuera del módulo.
- Las funciones con alcance público se pueden llamar tanto fuera como dentro del módulo. Simplemente podemos decir como "Podemos llamarlo en cualquier parte del programa".
- El modificador predeterminado de la función es público.
- De forma predeterminada, los argumentos de la función se pasan por referencia (en un tema aparte, esto se explicará en detalle).

Examples

Creando y llamando a una función

Este ejemplo utiliza el proyecto EXE estándar con la adición de un archivo de módulo.

- Crear nuevo proyecto "EXE estándar". Así que aquí, un formulario se agregará al proyecto de forma predeterminada.
- Agregar un archivo de módulo al proyecto
- Coloque un botón de comando en el formulario
- Crear botón de comando haga clic en Evento.

Código del módulo

Creó dos funciones en el módulo. Una es una función pública (FnAdd). Toma dos argumentos enteros val_1 y val_2. Devuelve un entero. Esta función agrega los dos argumentos y devuelve el valor a la persona que llama. Antes de la adición, los dos argumentos pasan por un proceso en otra función. Que es una función privada. Características / Reglas de la Función Pública y Privada explicadas en la sección de Comentarios.

```
Public Function FnAdd(val_1 As Integer, val_2 As Integer) As Integer

'Calling private function
val_1 = FnMultiplyBy5(val_1)

'Calling private function
val_2 = FnMultiplyBy5(val_2)

'Function return statement
FnAdd = val_1 + val_2

End Function
```

A continuación se muestra la función privada en el módulo. Toma un argumento entero val. Devuelve un entero. Esta función multiplica un valor 5 con el argumento y devuelve el resultado a la persona que llama.

```
Private Function FnMultiplyBy5(Val As Integer) As Integer

'Function return statement
FnMultiplyBy5 = Val * 5

End Function
```

Código de formulario

En el botón de comando, haga clic en Evento. Aquí estamos llamando a la función pública del módulo "FnAdd"

```
Private Sub Command1_Click()
Debug.Print FnAdd(3, 7)
End Sub
```

Resultado en la Ventana Inmediata

50

Lea Procedimientos de Función en línea: <https://riptutorial.com/es/vb6/topic/9227/procedimientos-de-funcion>

Capítulo 4: Sintaxis basica

Examples

declaración if / else

```
If condition Then
    code to execute if true
ElseIf condition Then
    code
Else
    code to execute if conditions are both false
End If
```

en bucle

```
For I as Integer = 1 To 10 Step 1
    code to execute
Next
```

El paso es opcional y el paso 1 es el predeterminado. El paso le dice cómo contar, por lo que -1 lo haría restar 1 cada vez y el paso 5 lo haría sumar 5 cada vez a través del bucle.

En caso de que el bucle deba detenerse, entonces se puede usar la instrucción `Exit For`, como en el siguiente ejemplo;

```
Dim iIndex as integer

For I as Integer = 1 To 10 Step 1

    Debug.Print I
    iIndex = I * 10

    If iIndex > 90 Then
        Exit For
    End If

Loop
```

Aquí, en lugar de imprimir de 1 a 10, se detendrá en 9, ya que la condición le dijo al proceso que se detuviera cuando `iIndex` alcance 90.

Hacer bucle

Otro tipo común de bucle en Visual Basic es el `DO loop`, que ejecutaría una parte del código continuamente hasta que se le indique que se detenga. Al contrario de algunos otros bucles en los que los índices se utilizan para detener el proceso, en este bucle particular, se debe indicar que se detenga.

Un ejemplo simple que ilustra el bucle es el siguiente

```
Dim iIndex1 As Integer
iIndex1 = 1

Do
    Debug.Print iIndex1
    iIndex1 = iIndex1 + 1

    If iIndex1 = 10 Then
        Exit Do
    End If
Loop
```

El fragmento de código anterior tomará un Índice, se inicializará a 1, y lo incrementará. Un `Debug.Print` ayudará a imprimir el índice para acumular el bucle. En cada bucle, el código verificará si el índice ha alcanzado 10 y si, y solo si la condición es verdadera, se ejecutará el `Exit Do`, que detendrá el bucle.

Declaración de caso selecto

```
Dim number As Integer = 8
Select Case number
    Case 1 To 5
        Debug.WriteLine("Between 1 and 5, inclusive")
        ' The following is the only Case clause that evaluates to True.
    Case 6, 7, 8
        Debug.WriteLine("Between 6 and 8, inclusive")
    Case 9 To 10
        Debug.WriteLine("Equal to 9 or 10")
    Case Else
        Debug.WriteLine("Not between 1 and 10, inclusive")
End Select
```

Lea Sintaxis basica en línea: <https://riptutorial.com/es/vb6/topic/9389/sintaxis-basica>

Capítulo 5: Variables

Examples

Tipos de variables

Hay diferentes tipos de variables para diferentes propósitos. En Visual Basic 6 están disponibles los siguientes tipos de variables:

- Formación
- Booleano
- Byte
- Moneda
- Fecha
- Doble
- Entero
- Largo
- Soltero
- Cuerda
- Variante

Usted declara una variable usando la palabra clave `Dim` :

```
Dim RandomNumber As Integer
```

Si no especifica un tipo de variable, la variable se establecerá de forma predeterminada en

`Variant` :

```
Dim Foo
```

es equivalente a

```
Dim Foo As Variant
```

Booleano

Boolean es el tipo de variable más simple, ya que puede contener solo uno de dos valores: Verdadero o Falso.

```
Foo = True  
Bar = False
```

Los booleanos se pueden usar para controlar el flujo de código:

```
Dim Foo as Boolean
Foo = True

If Foo Then
    MsgBox "True"
Else
    MsgBox "False"
End If
```

Entero

Un entero es un tipo de datos numérico y puede contener un valor con signo de 16 bits (-32768 a +32767). Si sabe que una variable solo contendrá números enteros (como 9) y no números fraccionarios (como 5.43), declararla como un tipo de datos entero (o largo).

```
Dim RandomNumber As Integer
RandomNumber = 9
```

Los enteros se utilizan comúnmente como contadores en `For...Next` loops:

```
Dim Counter As Integer

For Counter = 0 to 2
    MsgBox Counter
Next Counter
```

Si intenta asignar un valor menor que -32768 o mayor que 32767 a un entero, se producirá un error en tiempo de ejecución:

```
Dim MyNumber As Integer
MyNumber = 40000 'Run-time error '6': Overflow
```

Cuerda

Una variable de cadena puede contener un texto vacío, un carácter, una palabra o un texto de longitud variable. El valor de la cadena debe estar entre comillas (").

```
Dim Fruit as String
Fruit = "Banana"
```

Si necesita comillas dentro de un literal de cadena, utilice dos comillas posteriores ("").

```
Dim Quote as String
Quote = "Bill says: ""Learn VB!"""
```

Lea Variables en línea: <https://riptutorial.com/es/vb6/topic/7511/variables>

Creditos

| S. No | Capítulos | Contributors |
|-------|-------------------------------|---|
| 1 | Comenzando con Visual Basic 6 | Community , Scheffer |
| 2 | Instalando VB6 en Windows 10 | ThunderFrame |
| 3 | Procedimientos de Función | Jeet |
| 4 | Sintaxis basica | Nadeem_MK , Talal Abdoh , user7491506 |
| 5 | Variables | BlueEel |