EBook Gratuito

APPRENDIMENTO Visual Basic 6

Free unaffiliated eBook created from
Stack Overflow contributors.

#vb6

Sommario

Di

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.

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

Capitolo 1: Iniziare con Visual Basic 6

Osservazioni

Questa sezione fornisce una panoramica su cosa sia vb6 e perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare eventuali soggetti di grandi dimensioni all'interno di vb6 e collegarsi agli argomenti correlati. Poiché la documentazione di vb6 è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Examples

Installazione o configurazione

Istruzioni dettagliate su come installare o installare vb6.

Ciao mondo

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

Leggi Iniziare con Visual Basic 6 online: https://riptutorial.com/it/vb6/topic/3439/iniziare-con-visual-basic-6

Capitolo 2: Installazione di VB6 su Windows 10

Examples

Installazione guidata

Procedura guidata del programma di installazione di Visual Studio 6.0

Per impostazione predefinita, i seguenti pacchetti non si installano correttamente sotto Windows 10:

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

Per installare i pacchetti di cui sopra, è necessario effettuare numerose modifiche e hack di registro oppure utilizzare la fantastica procedura guidata di installazione di Visual Studio 6.0 di Giorgio Brausi.

Avrai bisogno dei seguenti elementi prima di iniziare:

- I tuoi CD e tasti originali di Visual Studio / Basic Program
- I tuoi CD MSDN originali
- Visual Studio Service Pack 6
- Procedura guidata del programma di installazione di Visual Studio 6.0
- Su Windows 10 build 1511 o successivo, avrai bisogno dei diritti di amministratore.

La procedura guidata ti guiderà attraverso i passaggi necessari per una corretta installazione di Visual Basic 6.

Si noti che l'installazione delle applicazioni server non è attualmente possibile.

Leggi Installazione di VB6 su Windows 10 online:
https://riptutorial.com/it/vb6/topic/4230/installazione-di-vb6-su-windows-10

Capitolo 3: Procedure di funzionamento

introduzione

La funzione è una serie di istruzioni racchiuse tra le istruzioni "Funzione" e "Funzione finale".

La funzione esegue un'attività e restituisce il controllo al chiamante. Quando restituisce il controllo, restituisce anche un valore al codice chiamante.

È possibile definire una funzione in una classe, struttura e modulo. Di default è pubblico. Significa che puoi chiamarlo da qualsiasi punto dell'applicazione che ha accesso alla classe, alla Struttura o al Modulo in cui l'hai definito.

Sintassi

- [Modificatori] Funzione Name_Of_The_Function [(Arg_List)] As Return_Type
- [Istruzioni]
- Fine Funzione

Osservazioni

- I due modificatori di funzione utilizzati in questo esempio sono pubblici e privati. Questi modificatori definiscono l'ambito della funzione.
- Le funzioni con un ambito privato possono essere richiamate solo dal file di origine da cui sono state definite. Nel nostro caso può essere chiamato con nel modulo. E non può essere chiamato al di fuori del modulo.
- Le funzioni con ambito pubblico possono essere richiamate sia all'esterno che all'interno del modulo. Semplicemente possiamo dire "Possiamo chiamarlo in qualsiasi parte del programma".
- Il modificatore di default della funzione è pubblico.
- Per impostazione predefinita, gli argomenti della funzione vengono passati per riferimento (in un argomento separato, questo verrà spiegato in dettaglio).

Examples

Creazione e chiamata di una funzione

Questo esempio utilizza il progetto EXE standard con l'aggiunta di un file di modulo.

- Crea un nuovo progetto "EXE standard". Quindi qui, un modulo verrà aggiunto al progetto per impostazione predefinita.
- Aggiungi un file di modulo al progetto
- Inserire un pulsante di comando nel modulo
- Crea pulsante di comando Fai clic su Evento.

Codice del modulo

Creato due funzioni nel modulo. Uno è una funzione pubblica (FnAdd). Sono necessari due argomenti Integer val_1 e val_2. Restituisce un intero. Questa funzione aggiunge i due argomenti e restituisce il valore al chiamante. Prima dell'aggiunta, i due argomenti vengono sottoposti a un processo in un'altra funzione. Che è una funzione privata Caratteristiche / Regole della funzione pubblica e privata spiegate nella sezione Note.

```
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
```

Di seguito è riportata la funzione privata nel modulo. Prende un argomento intero val. Restituisce un numero intero. Questa funzione moltiplica un valore 5 con l'argomento e restituisce il risultato al chiamante.

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

'Function return statement
FnMultiplyBy5 = Val * 5

End Function
```

Codice del modulo

Nel pulsante di comando fai clic su Evento. Qui stiamo chiamando la funzione Module Module "FnAdd"

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

Risultato nella finestra immediata

```
50
```

Leggi Procedure di funzionamento online: https://riptutorial.com/it/vb6/topic/9227/procedure-di-funzionamento

Capitolo 4: Sintassi di base

Examples

se / else dichiarazione

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

per ciclo

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

Il passaggio è facoltativo e il passaggio 1 è l'impostazione predefinita. Step dice come contare, quindi -1 dovrebbe sottrarre 1 ogni volta e il Passaggio 5 dovrebbe aggiungere 5 ogni volta attraverso il ciclo.

Nel caso in cui sia necessario arrestare il ciclo, è possibile utilizzare l'istruzione `Exit For` , come nell'esempio seguente;

```
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
```

Qui, invece di stampare da 1 a 10, si fermerà a 9, poiché la condizione ha detto al processo di fermarsi quando iIndex raggiunge 90.

Fai il ciclo

Un altro tipo comune di loop in Visual Basic è il `DO loop` , che eseguirà continuamente un pezzo di codice fino a quando non viene detto di fermarsi. Al contrario di alcuni altri cicli in cui gli indici sono usati per fermare il processo, in questo particolare ciclo, dovrebbe essere detto di fermarsi.

Un semplice esempio che illustra il ciclo è il seguente

```
Dim iIndex1 As Integer
iIndex1 = 1

Do
   Debug.Print iIndex1
   iIndex1 = iIndex1 + 1

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

Il codice di cui sopra prenderà un indice, inizializzato a 1, e lo incrementerà. Un `Debug.Print` aiuterà a stampare l'indice per creare il ciclo. Su ciascun ciclo, il codice verificherà se l'indice ha raggiunto 10 e se e solo se la condizione è vera, verrà eseguito `Exit Do` , che interromperà il ciclo.

Seleziona la dichiarazione del caso

```
    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
```

Leggi Sintassi di base online: https://riptutorial.com/it/vb6/topic/9389/sintassi-di-base

Capitolo 5: variabili

Examples

Tipi variabili

Esistono diversi tipi di variabili per scopi diversi. In Visual Basic 6 sono disponibili i seguenti tipi di variabile:

- schieramento
- booleano
- Byte
- Moneta
- Data
- Doppio
- Numero intero
- Lungo
- singolo
- Stringa
- Variante

Si dichiara una variabile usando la parola chiave `Dim` :

```
Dim RandomNumber As Integer
```

Se non si specifica un tipo di variabile, la variabile verrà impostata di default su `Variant` :

```
Dim Foo
```

è equivalente a

```
Dim Foo As Variant
```

booleano

Boolean è il tipo di variabile più semplice in quanto può contenere solo uno dei due valori: True o False.

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

I booleani possono essere utilizzati per controllare il flusso del codice:

```
Dim Foo as Boolean
Foo = True
```

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

Numero intero

Un numero intero è un tipo di dati numerici e può contenere un valore con segno a 16 bit (da -32768 a +32767). Se si sa che una variabile conterrà solo numeri interi (come 9) e non numeri frazionari (come 5.43), dichiararla come un tipo di dati intero (o lungo).

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

I numeri interi sono comunemente utilizzati come contatori in `For...Next` cicli:

```
Dim Counter As Integer

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

Se si tenta di assegnare un valore inferiore a -32768 o superiore a 32767 su un numero intero, si verificherà un errore in fase di esecuzione:

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

Stringa

Una variabile stringa può contenere un testo vuoto, un carattere, una parola o un testo di lunghezza variabile. Il valore stringa deve essere contenuto tra virgolette (`"`).

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

Se hai bisogno di virgolette all'interno di una stringa letterale, usi due virgolette successive (`""`).

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

Leggi variabili online: https://riptutorial.com/it/vb6/topic/7511/variabili

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con Visual Basic 6	Community, Scheffer
2	Installazione di VB6 su Windows 10	ThunderFrame
3	Procedure di funzionamento	Jeet
4	Sintassi di base	Nadeem_MK, Talal Abdoh, user7491506
5	variabili	BlueEel