

Basics in Visual Basic 6.0

Designer, Formular	Neues Projekt anlegen	
GUI ¹ -Objekte	Msgbox, Textbox, Label, Button	Verwenden des Designers
Hallo-Welt-Programm - Erstes lauffähiges Programm: Ereignisse und Eigenschaften der GUI-Objekte Ereignisprozeduren	Command1_Click()	Sub Command1_Click() MsgBox "Hallo Welt" End Sub
Variable, Wert Werte an Variablen übergeben	Deklarieren, Datentypen (String, Integer, Double, Boolean)	Dim s as String Dim i as Integer s = "Zeichenkette" i = 5
Logische Ausdrücke	Werte/Variableninhalte vergleichen	1=1 >> wahr 1=2 >> falsch "gut"="gut" >> wahr "gut"="ungut" >> falsch
Prozeduren und Parameterübergabe	Sub <Prozedurname>(<ParameterName1> as <Datentyp>, _ <ParameterName2> as <Datentyp>,,,) <Tu was> End Sub Die Übergabe von Parameterwerten ist optional ²	Sub Command1_Click() Dim i As Integer Dim s As String s = "Die Zahl heißt: " i = 12 MeineProzedur i, s End Sub Sub MeineProzedur(iVar As Integer, sVar As String) MsgBox sVar & iVar End Sub
Bedingungen1	If <Bed> Then <Tu was> ElseIf <Bed> Then <Tu was anderes> Else <Dann tu das> End If	Sub Command1_Click() Dim i As Integer Dim s As String s = "Die Zahl heißt: " i = 12 MeineProzedur i, s End Sub Sub MeineProzedur(iVar As Integer, sVar As String) Dim i As Integer

¹ Graphical User Interface – Grafische Bildschirmoberfläche, -Benutzerschnittstelle

² Das kann man machen, muss man aber nicht. Es hängt von der Deklaration (Erklärung) der Prozedur ab.

		<pre> Dim ii As Integer i = 5 ii = 6 If i = ii Then Text1.Text = "wahr" ElseIf i = 0 Then Text1.Text = "i ist Null" Else Text1.Text = "falsch" End If End Sub </pre>
--	--	--

Bedingungen2	<pre> Select Case <Variable> Case 1: <Tu was> Case 3: <Tu was> Case 5: <Tu was> Case 9: <Tu was> Case Else: <Tu was> End Select </pre>	<pre> Dim i as Integer i = 5 Select Case <Variable> Case 1: Text1.Text = "eins" Case 3: Text1.Text = "drei" Case 5: Text1.Text = "fünf" Case 9: Text1.Text = "neun" Case Else: Text1.Text = "Gib 1, 3, 5 oder 9 ein" End Select </pre>
--------------	--	--

Schleife1	<pre> For <Schleifenzähler> = <Start> to <Ende> [Step +/- <Schritt>] <Tu was> Next </pre>	<pre> Dim i as Integer For i = 1 To 10 Text1.Text = Text1.Text & i & ", " Next </pre>
-----------	---	---

Schleife2	<pre> Do While <Bed> <Tu was> Loop </pre>	<pre> Option Explicit Dim bRaus as Boolean Dim i as Integer i = 1 bRaus = False Do While Not bRaus Text1.Text = Text1.Text & "i & ", " i = i + 1 bRaus = (i = 10) Loop </pre>
-----------	---	---

<p>GUI-Objekt-Ereignisse in einer Schleife abfangen</p> <p>Globale und Lokale Variablen</p>	<pre> Do While <Bed> <Tu was> DoEvents Loop </pre> <p>Globale Variable im Kopf des Programms deklarieren</p> <p>Lebenszeit: Laufzeit des Programms</p> <p>Sichtbarkeit : In allen Prozeduren</p>	<pre> Option Explicit Dim bEx As Boolean 'Globale Variable Sub Form_Load() bEx = True End Sub Sub Command1_Click() bEx = Not bEx </pre>
---	--	---

Lokale Variablen in der einzelnen Prozedur deklarieren

Lebenszeit: Laufzeit dieser Prozedur
Sichtbarkeit : In dieser Prozedur

```
Start
End Sub
Sub Start()
    Dim i As Integer ' Lokale Variable
    Do While Not bEx
        Text1.Text = "Die Zahl heißt: " & i
        i = i + 1
    If i > 32766 Then 'Ein Integer verträgt keine größere Zahl
        i = 0
    End If
    DoEvents
    Loop
End Sub
```

Funktionen

Function <Funktionsname> (<Parameterliste>³) as <Datentyp>
<Funktionsname>⁴ = <Tu was>
End Function

```
Option Explicit
Sub Command1_Click()
    MeineAufrufProzedur
End Sub
Sub MeineAufrufProzedur()
    Dim i As Integer
    Dim ii As Integer
    i = 12
    ii = Quadrat(i)
    Text1.Text = i & "^2 = " & ii
End Sub
Function Quadrat(iVar As Integer) As Integer
    Quadrat = iVar ^ 2
End Function
```

VB-interne Stringverarbeitungsfunktionen

Space

Gibt eine Zeichenfolge vom Typ Variant (String) zurück, die aus einer angegebenen Anzahl von Leerzeichen besteht.
Syntax
Space(Zahl)

Chr

Gibt einen Wert vom Typ String zurück, der das Zeichen enthält, das dem angegebenen Zeichen-Code zugeordnet ist.
Syntax
Chr(Zeichencode)

³ vgl. Prozeduren

⁴ Der dem Funktionsnamen zugewiesene Wert wird an die aufrufende Prozedur zurück gemeldet

Left	Gibt einen Wert vom Typ Variant (String) zurück, der eine bestimmte Anzahl von Zeichen ab dem ersten (linken) Zeichen einer Zeichenfolge enthält. Syntax Left(string, length)
Right	Gibt einen Wert vom Typ Variant (String) zurück, der eine bestimmte Anzahl von Zeichen von der rechten Seite (dem Ende) einer Zeichenfolge enthält. Syntax Right(string, length)
Mid	Gibt einen Wert vom Typ Variant (String) zurück, der eine bestimmte Anzahl von Zeichen aus einer Zeichenfolge enthält. Syntax Mid(string, start[, length])
LTrim RTrim Trim	Gibt einen Wert vom Typ Variant (String) zurück, der eine Kopie einer bestimmten Zeichenfolge enthält, die keine führenden Leerzeichen (LTrim), keine nachgestellten Leerzeichen (RTrim) sowie keine Kombination aus führenden und nachgestellten Leerzeichen (Trim) enthält. Syntax LTrim(Zeichenfolge) RTrim(Zeichenfolge) Trim(Zeichenfolge)
Str	Gibt einen Wert vom Typ Variant (String) zurück, der eine Zahl darstellt. Syntax Str(Zahl)
String	Gibt eine Zeichenfolge vom Typ Variant (String) zurück, die ein sich wiederholendes Zeichen der angegebenen Länge enthält. Syntax String(number, character)
LCase	Gibt einen Wert vom Typ String zurück, in dem alle Buchstaben in Kleinbuchstaben umgewandelt worden sind. Syntax LCase(Zeichenfolge)
UCase	Gibt einen Wert vom Typ Variant (String) zurück, der eine bestimmte, in Großbuchstaben umgewandelte Zeichenfolge enthält. Syntax UCase(Zeichenfolge)