



VBA voor Doe het Zelfers deel 20

Handleiding van Helpmij.nl

Auteur: leofact

Augustus 2015

“ Dé grootste en gratis computerhelpdesk van Nederland ”

Vorige aflevering

In het vorige deel werd besproken hoe je de structuur en vensteropbouw van een werkmap kunt beveiligen. Daarnaast werd behandeld hoe je een gedeelde werkmap kunt beveiligen tegen het wissen van de bijgehouden wijzigingen. Verder werden er diverse manieren aangeboden om werkbladen te beveiligen. Hieronder een methode om wijzigingen door de gebruiker in het werkblad te beperken, terwijl wijzigingen door VBA wél mogelijk blijven. Er werd daarbij kort aangestipt hoe kan worden opgevraagd welke handelingen er worden toegestaan door de werkbladbeveiliging. Daarbij kwam ook de celblokkering aan de orde. Als laatste werd besproken op welke manier het project zelf kan worden beveiligd.

Deze aflevering

In deel 20 komt het gebruik van operatoren aan de orde. Als eerste de rekenkundige operatoren. Deze zijn voor het grootste deel meer dan bekend. Daarnaast worden de vergelijkende, logische en object operatoren besproken met daarbij de syntax en een aantal voorbeelden hoe ze in de code zijn te gebruiken. Ook dit keer is er weer een bijlage met de voorbeeld code. Dit werkboek is [hier](#) te downloaden.

Operatoren

Een operator is een teken of woord (ook wel sleutelwoord, of keyword genoemd) dat een bewerking aanduidt op één of meer getallen of gegevens: de zogenaamde operanden. De operatoren zijn in soorten onder te verdelen. Vier daarvan worden hier behandeld. Uiteraard worden alleen die operatoren behandeld die in VBA van toepassing zijn.

Rekenkundige operatoren

Deze operatoren zijn over het algemeen meer dan bekend. Zij maken de basis-berekeningen mogelijk. Sommige operatoren wijken net even af van het teken dat we uit het dagelijks gebruik kennen. Dat heeft vaak te maken met de herkenbaarheid van een teken. Het maalteken is daar een voorbeeld van. Dat is hetzelfde als de x en kan bij gebruik op een computer voor verwarring zorgen. Dat is de reden dat hiervoor * (het sterretje) wordt gebruikt. Rekenkundige operatoren rekenen met getallen en hebben ook weer een getal als uitkomst. Hieronder volgt een opsomming:

teken	bewerking	Voorbeeld
-	afrekken	$8 - 4 = 4$
+	optellen	$8 + 4 = 12$
*	vermenigvuldigen	$8 * 4 = 32$
/	delen	$8 / 4 = 2$
^	machtsverheffen	$8 ^ 2 = 64$
Mod	Geeft de Modulus of rest na een deling	$9 \text{ mod } 4 = 1$ $(8 / 4 = 2 \text{ rest } 1)$

Er is geen operator beschikbaar voor worteltrekken. Daarvoor is wel de functie *Sqr*

Sqr (144) = 12

Tip: voor even en oneven is er geen aparte functie beschikbaar in VBA. Met behulp van Mod is toch eenvoudig te bepalen of een getal even of oneven is.

Getal Mod 2 = > rest 0 = even, rest 1 = oneven. Zie dit voorbeeld:

```

(Algemeen) Even
'-----
' Procedure : Even 23-6-2015
' Doel : Geef even of oneven
'-----
'
Sub Even()
Dim r As Range
'doorloop de cellen van de aangegeven rijen in kolom B
For Each r In [B16:B40]
    'bepaal of waar = Rest 1 Oneven), of Rest 0 (even)
    'zet de uitkomst in de aangrenzende cel
    r.Offset(0, 1) = IIf(r.Value Mod 2, "oneven", "even")
Next r
End Sub
    
```

De berekeningsvolgorde is omgekeerd aan de volgorde zoals deze in de eerdere tabel werd gegeven. Dus eerst machtsverheffen, dan vermenigvuldigen en delen en dan optellen en aftrekken. De volgorde kan een groot verschil uitmaken in de uitkomst van de berekening. Dit is te beïnvloeden door het gebruik van haakjes, zoals dit voorbeeld laat zien:

$$10 - 8 * 25 = -190$$

$$(10 - 8) * 25 = 50$$

Vergelijkende operatoren

Deze operatoren kunnen worden gebruikt voor getallen en op string variabelen. Zij hebben de booleaanse waarden True of False als uitkomst.

=>	is gelijk aan	8 + 2 = 5 * 2
>	groter dan	4 > 2
<	kleiner dan	2 < 4
<>	ongelijk aan	2 <> 4
>=	groter of gelijk aan	4 >= 2 en 2 >= 2
<=	kleiner of gelijk aan	2 <= 4 en 2 >= 2

Het volgende voorbeeld demonstreert hoe dit er kan uitzien bij het gebruik met getallen:

```

(Algemeen) VergelijkGetal
'-----
' Procedure : VergelijkGetal 24-6-2015
' Doel :Wat voorbeelden van vergelijkingen
' Gebruik venster direct (druk CTRL + G)
'-----
'
Sub VergelijkGetal()

    Debug.Print 8 = 4 * 2
    Debug.Print 8 = 4 * 12
    Debug.Print 8 < 4 * 12
    Debug.Print 8 <= 4 * 2
    Debug.Print 8 <> 4 * 2
    Debug.Print 8 > 4

End Sub
    
```

De vergelijking die bij strings wordt gemaakt is niet gebaseerd op de lengte van de vergeleken strings, maar op de alfabetische volgorde daarvan. Zie daarvoor dit voorbeeld:

```

'-----
' Procedure : VergelijkTekst 24-6-2015
' Doel :Wat voorbeelden van vergelijkingen
' Gebruik venster direct (druk CTRL + G)
'-----
Sub VergelijkTekst ()

    Debug.Print "aaa" = "aaa"
    Debug.Print "aaa" = "bbb"
    Debug.Print "aaa" > "bbb"
    Debug.Print "aaa" <> "bbb"
    Debug.Print "bbbb" < "aaa"
    Debug.Print "dal" > "bal"

End Sub
    
```

De aaneenschakelings- (of concatenatie-) operator

Meerdere strings zijn eenvoudig samen te voegen tot één string met het samenvoegteken &. Hiermee kunnen getallen, variabelen en berekeningen tot één string worden samen gevoegd. Hieronder volgt een eenvoudig voorbeeld. Dit kun je echter zo complex maken als je maar wilt:

```

'-----
' Procedure : Samenvoegen 24-6-2015
' Doel : voeg strings en getallen samen
'-----
Sub Samenvoegen ()

    Dim lgetal As Long
    lgetal = 5
    MsgBox "som: 10 + 5 = " & 10 + lgetal & vbCrLf

End Sub
    
```

Logische operatoren

Logische operatoren kijken of de vergelijking aan een voorwaarde voldoet. Ook zij geven de booleaanse waarden True of False als uitkomst.

And	Een vergelijking dient aan beide voorwaarden te voldoen om als waar te worden beoordeeld.	If A = 5 And B = 6
Or	Er dient aan één óf beide voorwaarden voldaan te worden om als waar te worden beoordeeld.	If A = 5 Or B = 6
Xor	De vergelijking wordt uitsluitend als waar beoordeeld als er aan één van beide voorwaarden wordt voldaan.	If A = 5 Xor B = 6
Not	Beoordeeld één waarde en geeft de tegenovergestelde uitslag	If A Not 6 = True

Bij een numerieke waarde voert Not een binaire verschuiving uit. Hierbij wordt de waarde van de afzonderlijke bits omgedraaid. Deze bewerking is te begrijpen vanuit het binaire getallenstelsel. Hierbij worden de getallen voorgesteld door een reeks enen en nullen. Lees [hier](#) de uitleg van Wikipedia.

De volgende procedure demonstreert de omzetting naar binair:

```

(Algemeen) | Verschuiving
'-----
' Procedure : Verschuiving2 25-7-2015
' Doel: demonstreert de omkering met Not
'       op een numerieke waarde
'-----
'
Sub Verschuiving2 ()
Dim lA As Integer
Dim lB As Integer
Dim sC As String
Dim sD As String

lA = 6:
'keer lA om
lB = Not lA
'maak een string met de binaire waarde:
sC = WorksheetFunction.Dec2Bin(lA, 4)
sD = WorksheetFunction.Dec2Bin(lB, 4)
MsgBox lA & " =binair: " & sC & vbCrLf _
      & lB & " =binair: " & sD
End Sub
    
```

Object operatoren

Deze operatoren zijn alleen van toepassing op objecten.

Is

Deze operator vergelijkt twee objecten en geeft True of False als resultaat:

bUitkomst = Object1 Is Object2

Set

Stelt de objectvariabele in:

Set Object1 = Blad1

Nothing

Beoordeelt of een object variabele leeg is. Als dat het geval geeft Nothing True als uitkomst:

bUitkomst = Object1 Is Nothing

Wanneer het in een bepaalde procedure belangrijk is om te weten of een objectvariabele gevuld is wordt in de praktijk vaak Not gebruikt om een positieve uitslag te krijgen:

If Not Object1 Is Nothing Then ...

Een objectvariabele houdt geheugenplaatsen bezet. Het is daarom een goede gewoonte om de variabele leeg te maken als deze niet meer nodig is. Ook daarvoor gebruik je Nothing:

Set Object1 = Nothing

De volgende procedure geeft een kleine demonstratie van het voorafgaande:

The screenshot shows the 'Objectvariabele' window in VBA. The window title is '(Algemeen) Objectvariabele'. The code is as follows:

```

'-----
' Procedure : VergelijkObjecten 23-7-2015
' Doel : Vergelijkt objecten
'-----
'
Sub VergelijkObjecten()
Dim oA As Object
Dim oB As Object
Set oA = Blad1
Set oB = Blad2

    Debug.Print oB Is oA
    Debug.Print oA Is Nothing
    Debug.Print Not oA Is Nothing
    Set oA = Nothing
    Debug.Print oB Is oA
    Debug.Print oA Is Nothing
    Debug.Print Not oA Is Nothing
End Sub

```

Samenvatting

In dit deel werd uitgelegd wat een operator is. Er werden vijf soorten operators benoemd. Namelijk de rekenkundige, de vergelijkende, de aaneenschakelende, de logische en als laatste de object-operatoren. Van ieder soort werden de belangrijkste operatoren gegeven met de wijze waarop deze gebruikt kan worden. Daarnaast werden een aantal voorbeeldjes gegeven hoe deze in een procedure verwerkt kunnen worden.

Deel 21

In het volgende deel worden Regular Expressions behandeld. Deze VBScript bibliotheek maakt het mogelijk om op eenvoudige wijze invoermaskers te maken waarmee invoer op het juiste format is te controleren. Daarbij kun je denken aan de postcode, BSN, Nederlandse IBAN etc.

Inbreng

Deze VBA-reeks is niet oneindig. Mocht je echter graag een onderwerp behandeld willen hebben dat binnen het bestek van de reeks past; mail dan naar redactie@helpmij.nl. Je email wordt altijd beantwoord. Het kan echter niet gegarandeerd worden dat een onderwerp inderdaad behandeld wordt.