



Access voor Beginners - H14

Handleiding van Helpmij.nl

Auteur: OctaFish

Oktober 2012

“ Dé grootste en gratis computerhelpdesk van Nederland ”

Getallen omzetten naar tekst

Een vraag die een enkele keer wel eens voorbij komt, is hoe je getallen kunt omzetten naar tekst. In Word is daar een trucje voor, (de switch * CardText toevoegen aan een veldcode) maar om te beginnen werkt dat alleen maar bij gehele getallen die uit niet meer dan 6 cijfers bestaan, en in Access 2003 en Access 2007/2010 bestaat die optie niet eens. Dus hoe kun je een getal dan als tekst laten zien?

Het zal duidelijk zijn, dat we hier een functie voor zullen moeten inzetten!

Wanneer schrijf je getallen in letters en wanneer in cijfers?

Volgens het Genootschap Onze Taal (ik noem die vanaf nu GOT) zijn daar geen vaststaande regels voor. De meeste adviesboeken houden aan dat getallen tot twintig voluit worden geschreven, evenals de tientallen tot honderd, de honderdtallen tot duizend, enz. We schrijven dus achtste, tien, zeventien, 22, veertig, 103, tweehonderd, 250, drieduizend. Ook de getallen duizend, miljoen en miljard schrijven we voluit; deze kunnen eventueel gecombineerd worden met cijfers: 22 duizend, 123 miljoen, tien miljoen, 16 miljard (of 22.000, 123.000.000, 16.000.000.000, etc.).

Voor het schrijven van percentages geldt dat er vaak cijfers worden geschreven als we dat anders niet zouden doen: 50% van de Nederlanders, 15 procent; als in een tekst maar een enkel percentage voorkomt, kan het geheel ook in woorden worden opgeschreven: twintig procent.

Deze regels gelden met name voor zakelijke teksten; in literaire teksten bijvoorbeeld worden getallen vaak uitgeschreven. Je kunt dan de volgende vuistregels aanhouden:

- Tot het getal duizend schrijven we getallen aaneen: tweeëndertig, achthonderdtien, enz.
- In grote getallen komt na het woorddeel duizend een spatie: tweeduizend zeven.
- Voor en na miljoen en miljard schrijven we een spatie: vijf miljoen driehonderdduizend, twaalf miljard.

Op diverse forums, en ook de support pagina van Microsoft, kun je een functie vinden die een getal omzet naar geschreven tekst. Probleem van deze functie is, dat hij getallen omzet naar het Engels. En dat is in Nederland (en België) uiteraard minder handig. En wel om de volgende redenen: niet alleen rekenen wij niet in Dollars, de schrijfwijze van de getallen (en geldbedragen) is anders opgebouwd. Neem bijvoorbeeld het getal € 5.678,96. In het Engels zou dat er uitzien als:

Five Thousand Six Hundred Seventy Eight Dollars and Ninety Six Cents

In het Nederlands zou dat moeten zijn:

Vijfduizend zeshonderdachtenzeventig Euro en zesennegentig cent

De internationale functie is dus wel bruikbaar als je met Angelsaksische landen te maken hebt, maar voor intern binnenlands gebruik moet de functie worden aangepast. En dat maakt de functie dus geschikt voor een aflevering voor de cursus.

De basisopzet

De functie gaan we maken in twee smaken: één variant die een geldbedrag omzet naar tekst als een valutabedrag, en een variant die een 'normaal' getal omzet naar tekst. In de laatste variant worden alleen de cijferblokken omgezet, en de eerste variant voegt de woorden *Euro*(s) en *Cent*(en) toe aan het geconverteerde getal. De functie bestaat eigenlijk uit verschillende onderdelen, die steeds worden aangesproken door de hoofdfunctie.

Om een getal te kunnen converteren naar woorden, moet elk getal in groepen van 3 cijfers worden geconverteerd. Op die manier worden getallen namelijk ontleed. Eigenlijk ook wel logisch, als je bedenkt dat een cijfergroepering ook op groepen van 3 cijfers is gebaseerd. Er zijn een paar manieren om een getal op te breken in stukken van 3 cijfers. We passen er twee toe in de verschillende functies.

Laten we beginnen met de functie die *ConverteerGetalNaarTekst* heet. Deze converteert, zoals de naam al aangeeft, een getal naar woorden. Deze functie maken we uiteraard in een nieuwe module, want je wilt de functie op elke willekeurige plek kunnen gebruiken, dus zowel in queries, formulieren als rapporten.

De functie *ConverteerGetalNaarTekst*

In deze functie gebruiken we een matrix variabele die we vullen met de diverse groepen van 3 cijfers. Omdat je van tevoren niet weet of een (geheel) getal wel een veelvoud van 4 cijfers bevat, zetten we het te converteren getal eerst in een string variabele die we vervolgens aanvullen met voorlooppunten tot we een veelvoud van 3 hebben. Op die manier zijn de groepen eenvoudig in te lezen in de variabele, en te converteren.

We beginnen uiteraard zoals altijd met het vastleggen van de variabelen. Daar zitten tekst variabelen bij, getalvariabelen en matrix variabelen. De matrix variabele *Groep* wordt gelijk al gevuld met aanduidingen voor de grote bedragen zoals duizend en miljoen.

Option Compare Database

Option Base 1

Public Function *ConverteerGetalNaarTekst*(ByVal *MijnGetal*)

Dim *Temp*

Dim *GrootGetal* As String, *KleinGetal* As String, *tmpGetal* As String

Dim *Decimaal* As Double, *Teller* As Long

Dim *GeheelGetal*() As Integer

Dim *iTmp* As Integer, *i* As Integer, *x* As Integer

ReDim *Groep*(9) As String

Groep(2) = "duizend "

Groep(3) = " miljoen "

Groep(4) = " miljard "

Groep(5) = " biljard "

Volgens de regel van het GOT wordt in het Nederlands het woord 'duizend' tegen het getal aangeplakt, dus de variabele *Groep*(2) begint niet met een spatie. Miljoen en miljard staan wel los van het getal, dus daar beginnen we met een spatie. Het gebruik van de variabele komt later uiteraard aan bod.

' Zet het getal om naar *Tekst*, en verwijder eventuele spaties.

```
MijnGetal = Trim(Str(Round(MijnGetal, 2)))
```

```
' Zoek het decimaalteken. In VBA is dat een punt.
```

```
Decimaal = InStr(MijnGetal, ".")
```

```
' Als we een decimaalteken hebben gevonden...
```

```
If Decimaal > 0 Then
```

```
    ' Dan moet dat deel apart omgezet worden.
```

```
    Temp = Left(Mid(MijnGetal, Decimaal + 1) & "00", 2)
```

```
    KleinGetal = OmzettenTientallen(Temp)
```

Vervolgens wordt het getal omgezet naar een tekststring met maximaal 2 decimalen, en gaat de functie op zoek naar een decimaalteken in de string. Hierbij hoeven we ons niet te bekommeren om komma of punt, omdat het aangeboden getal toch al als getal is herkend, en VBA standaard een punt gebruikt. We controleren dus alleen of er een punt aanwezig is.

Als laatste regel wordt een subfunctie aangeroepen : de functie *OmzettenTientallen*; hij krijgt een parameter mee (Temp) en het resultaat ervan zetten we in de variabele KleinGetal. Deze functie is Private gedeclareerd, want je zult hem nooit zelfstandig aanroepen.

```
Private Function OmzettenTientallen(ByVal MyTens)
```

```
Dim Result As String, ResultSingle As String
```

```
' Is de waarde tussen 10 en 19?
```

```
If Val(Left(MyTens, 1)) = 1 Then
```

```
    Select Case Val(MyTens)
```

```
        Case 10: Result = "Tien"
```

```
        Case 11: Result = "Elf"
```

```
        Case 12: Result = "Twaalf"
```

```
        Case 13: Result = "Dertien"
```

```
        Case 14: Result = "Veertien"
```

```
        Case 15: Result = "Vijftien"
```

```
        Case 16: Result = "Zestien"
```

```
        Case 17: Result = "Zeventien"
```

```
        Case 18: Result = "Achtien"
```

```
        Case 19: Result = "Negentien"
```

```

        Case Else

    End Select

Else

    '.. anders is het getal tussen 20 en 99.

    Select Case Val(Left(MyTens, 1))

        Case 2: Result = "Twintig"

        Case 3: Result = "Dertig"

        Case 4: Result = "Veertig"

        Case 5: Result = "Vijftig"

        Case 6: Result = "Zestig"

        Case 7: Result = "Zeventig"

        Case 8: Result = "Tachtig"

        Case 9: Result = "Negentig"

        Case Else

    End Select

End If

' En dan het achterste cijfer omzetten.

ResultSingle = OmzettenEnkelGetal(Right(MyTens, 1))

```

Eigenlijk is het principe van de functie relatief simpel : met een Select Case wordt de inhoud van de variabele MyTens geëvalueerd. En op basis daarvan wordt het getal vertaald naar tekst. Het geheim is dus eigenlijk heel simpel : komt de functie de waarde 17 tegen, dan wordt de variabele <Result> gevuld met het woord 'zeventien'.

Omdat getallen onder de 20 in het Nederlands een eigen 'naam' hebben, controleren we eerst of het getal met een 1 begint. In dat geval hebben we te maken met een getal tussen de 10 en 20. Ook die uitkomst wordt dan uiteraard toegewezen aan de variabele *Result*.

In de tweede helft van de functie is het voldoende om het eerste cijfer van het 2-cijferige getal te evalueren. Op basis daarvan wordt dan de vertaling gemaakt.

In de laatste regel hierboven zie je dat een nieuwe omweg wordt gestart : de functie *ConverDigit* zet namelijk het achterste cijfer om naar tekst. Die gebruikt ook weer een parameter, en ook nu wordt het resultaat weer aan een variabele toegewezen. Omdat de functie *OmzettenTientallen* bijna klaar is, volgt eerst het laatste blokje code.

```

If Not ResultSingle = "" Then Result = ResultSingle & "en" & Result

```

OmzettenTientallen = Result

End Function

Hier zie je dat de functie <OmzettenTientallen> wordt gevuld met het resultaat van zowel de functie zelf (*Result*) als met het resultaat van de functie die het achterste cijfer heeft vertaald. Die uitkomst kan eventueel leeg zijn, dus daar wordt extra op gecontroleerd.

Private Function OmzettenEnkelGetal(ByVal MyDigit)

Select Case Val(MyDigit)

Case 1: OmzettenEnkelGetal = "Een"

Case 2: OmzettenEnkelGetal = "Twee"

Case 3: OmzettenEnkelGetal = "Drie"

Case 4: OmzettenEnkelGetal = "Vier"

Case 5: OmzettenEnkelGetal = "Vijf"

Case 6: OmzettenEnkelGetal = "Zes"

Case 7: OmzettenEnkelGetal = "Zeven"

Case 8: OmzettenEnkelGetal = "Acht"

Case 9: OmzettenEnkelGetal = "Negen"

Case Else: OmzettenEnkelGetal = ""

End Select

End Function

De functie *OmzettenEnkelGetal* doet hetzelfde als de functie *OmzettenTientallen*, maar is zelfs nog simpeler, omdat er maar één cijfer hoeft te worden gecontroleerd. Wil je dat het cijfer 0 ook wordt vertaald, dan kun je een extra Case regel toevoegen aan het Select Case blok.

We zijn nu net op weg met de hoofdfunctie, en we zijn al langs twee subfuncties geweest om delen van het getal te vertalen. En dat is eigenlijk in een notendop precies wat de functie continue doet: brokken van het hoofdgetal pakken, en dat per blokje vertalen. De reden daarvoor is hierboven al uitgelegd: grote getallen worden verdeeld in blokken van 3 cijfers, die vervolgens vertaald worden.

Nu de centen zijn vertaald, kunnen we naar het gedeelte voor de komma gaan kijken.

' Verwijder nu KleinGetal uit de variabele.

MijnGetal = Trim(Left(MijnGetal, Decimaal - 1))

End If

tmpGetal = MijnGetal

```
If tmpGetal = "" Then tmpGetal = "000"
```

```
Do Until Len(tmpGetal) Mod 3 = 0
```

```
    tmpGetal = "0" & tmpGetal
```

```
Loop
```

Zoals al eerder verteld, moet de totale string die we gaan evalueren uit blokken van 3 cijfers bestaan. Daarom zorgen we er met een lus voor dat er een veelvoud van 3 in de getalstring zit. Dat doen we door de lengte van de string door 3 te delen met Mod 3. De uitkomst daarvan is 1, 2 of 0. In het laatste geval hebben we te maken met een veelvoud van 3.

```
iTmp = Len(tmpGetal) / 3
```

```
ReDim GeheelGetal(iTmp)
```

```
For i = 1 To Len(tmpGetal) Step 3
```

```
    x = x + 1
```

```
    GeheelGetal(x) = Mid(tmpGetal, i, 3)
```

```
Next i
```

```
tmpGetal = ""
```

Nu we een lengte hebben die door 3 deelbaar is, kunnen we uitrekenen hoe groot de matrix moet zijn door de lengte door 3 te delen. Vervolgens wordt de variabele GeheelGetal(#) vastgelegd. Het vullen gebeurt weer met een lus, waarbij we steeds 3 tekens tegelijk pakken. Daarom moet de lus ook in stappen van 3 worden verhoogd. Mid(tmpGetal, i, 3) pakt dus achtereenvolgens Mid(tmpGetal, 1, 3), Mid(tmpGetal, 4, 3), Mid(tmpGetal, 7, 3) etc.

```
For x = UBound(GeheelGetal) To LBound(GeheelGetal) Step -1
```

```
    Teller = Teller + 1
```

```
    Temp = OmzettenHonderd(GeheelGetal(x))
```

```
    If Temp <> "" Then tmpGetal = Temp & Groep(Teller) & tmpGetal
```

```
Next x
```

In een volgende lus pakken we alle groepen uit de matrix GeheelGetal weer op, en zetten we die met de functie *OmzettenHonderd* om naar tekst. Dit gebeurt in omgekeerde volgorde, want de eerste groep in de variabele bevat het grootste deel van het getal, aflopend tot de hondertal waarde. Omdat de functie met het kleinste getaldeel moet beginnen, en dat in de hoogste dimensie van de variabele staat, moeten we van achteren naar voren werken. Vandaar dus ook Step -1.

Je ziet ook dat er een variabele Teller wordt opgehoogd met 1. Deze variabele wordt gebruikt om de woorden duizend, miljoen etc. toe te voegen aan het geconverteerde bedrag. Als je nog eens terugkijkt naar het blokje waarin de variabele Groep wordt gevuld, dan zie je dat het vullen begint op positie 2. De reden om positie 1 leeg te laten is dat het woord 'honderd' wordt toegevoegd in de functie OmzettenHonderd. Het woord 'honderd' wordt namelijk op elke groep van 3 cijfers toegepast.

In de voorlaatste regel van het blok vind je de regel `tmpGetal = Temp & Groep(Teller) & tmpGetal`. Hier wordt dus, afhankelijk van het aantal keer dat de lus is uitgevoerd, steeds een woord uit de variabele `Groep` opgehaald, behalve bij de eerste lus waar het hondertal getal is omgezet.

Private Function OmzettenHonderd(ByVal MijnGetal)

Dim Result As String

```
' Stoppen als er niks is om te zetten.
If Val(MijnGetal) = 0 Then Exit Function
' Leading zeros toevoegen aan getal.
MijnGetal = Right("000" & MijnGetal, 3)
' Moeten we een veelvoud van honderd converteren?
If Left(MijnGetal, 1) <> "0" Then
    Result = OmzettenEnkelGetal(Left(MijnGetal, 1)) & "honderd"
End If
' Moeten we een veelvoud van tien converteren?
If Mid(MijnGetal, 2, 1) <> "0" Then
    Result = Result & OmzettenTientallen(Mid(MijnGetal, 2))
Else
    ' Zo nee, dan alleen een enkel cijfer omzetten.
    Result = Result & OmzettenEnkelGetal(Mid(MijnGetal, 3))
End If
OmzettenHonderd = Trim(Result)
```

End Function

De functie *OmzettenHonderd* doet eigenlijk niet veel meer dan een getal van 3 cijfers opsplitsen in het eerste cijfer dat de honderd waarde vertaalt en er het woord 'honderd' achter plakt, de laatste twee cijfers die de tiental waarde vertalen en het laatste cijfer dat het enkele cijfer vertaalt. Daarvoor worden de hierboven besproken functies aangeroepen.

```
If tmpGetal = "" Then
    GrootGetal = "Nul"
    If Not KleinGetal = "" Then KleinGetal = " komma " &
        StrConv(KleinGetal, vbLowerCase)
```


Else

' Laatste stap: getal netjes opmaken.

Temp = Split(tmpGetal, " ")

Temp(LBound(Temp)) = StrConv(Temp(LBound(Temp)), vbProperCase)

For iTmp = LBound(Temp) + 1 To UBound(Temp)

Temp(iTmp) = LCase(Temp(iTmp))

Next iTmp

For iTmp = LBound(Temp) To UBound(Temp)

GrootGetal = GrootGetal & Temp(iTmp)

If iTmp < UBound(Temp) Then GrootGetal = GrootGetal & " "

Next iTmp

If Not KleinGetal = "" Then KleinGetal = " komma " &

StrConv(KleinGetal, vbLowerCase)

End If

ConverteerGetalNaarTekst = GrootGetal & KleinGetal

End Function

Het laatste deel van de procedure voegt uiteindelijk alle variabelen samen. Als het te converteren getal kleiner is dan 1, voegt de functie het woord 'komma' toe. Dit kun je uiteraard vervangen door een normale komma als je dat mooier vindt. Door de strings met StrConv te bewerken, kun je nauwkeurig bepalen welk deel met een hoofdletter moet beginnen, en welk deel met kleine letters moet worden vertaald.

De functie ConverteerGetalNaarEuros

De functie om een valutawaarde naar tekst te converteren, maakt gebruik van dezelfde onderdelen en werkwijze als de bovenstaande functie. Al wordt er in deze functie wel gebruik gemaakt van een andere lusmethode om het getal te vertalen, en moet er een valutanootatie aan de uitkomst worden toegevoegd.

Ik zal de functie dus beknopt bespreken, en alleen de verschillen behandelen.

Die verschillen beginnen overigens al met het declareren van de variabelen, want de matrix variabele is niet meer nodig.

Function ConverteerGetalNaarEuros(ByVal MijnGetal)

Dim Temp

Dim Euros As String, Cents As String

Dim Decimaal As Double, Teller As Long

Dim sNumber As String

Dim iTmp As Integer

ReDim Groep(9) As String

Groep(2) = "duizend "

Groep(3) = " miljoen "

Groep(4) = " miljard "

Groep(5) = " trillioen "

' Zet het getal om naar Tekst, en verwijder eventuele spaties.

MijnGetal = Trim(Str(MijnGetal))

' Zoek het decimaalteken. In VBA is dat een punt.

Decimaal = InStr(MijnGetal, ".")

' Als we een decimaalteken hebben gevonden...

If Decimaal > 0 Then

' Dan moet dat deel apartomgezet worden.

Temp = Left(Mid(MijnGetal, Decimaal + 1) & "00", 2)

Cents = OmzettenTientallen(Temp)

' Strip off cents from remainder to convert.

MijnGetal = Trim(Left(MijnGetal, Decimaal - 1))

End If

Tot hier is de functie identiek aan *ConverteerGetalNaarTekst*.

Do While MijnGetal <> ""

Teller = Teller + 1

' De laatste 3 cijfers van MijnGetal omzetten.

Temp = OmzettenHonderd(Right(MijnGetal, 3))

If Temp <> "" Then Euros = Temp & Groep(Teller) & Euros

If Len(MijnGetal) > 3 Then

' En vervolgens de laatste 3 cijfers verwijderen uit MijnGetal.

```
MijnGetal = Left(MijnGetal, Len(MijnGetal) - 3)
```

```
Else
```

```
MijnGetal = ""
```

```
End If
```

```
Loop
```

Er wordt met een lus een check gedaan op de variabele *MijnGetal*. Deze variabele begint met het complete (gehele) getal, en pakt daar de laatste 3 cijfers van. Deze worden vervolgens met de functie *OmzettenHonderd* geconverteerd op de nu bekende wijze. Halverwege het blok zie je hoe de geconverteerde tekst wordt toegewezen aan de variabele *Euros*, en daaronder wordt de berekende lengte van de variabele *MijnGetal* gebruikt om de reeds geconverteerde cijfers uit de string te verwijderen. Dat gebeurt met de opdracht *MijnGetal = Left(MijnGetal, Len(MijnGetal) - 3)*. Omdat we in een lus zitten, kan het hele proces net zo vaak plaats vinden tot het verkleinen van de string *MijnGetal* een lege string oplevert. Op dat moment is de *Do While* lus voltooid, en kan het eindresultaat worden opgemaakt.

' Volgende stap: Euros toevoegen aan de tekst.

```
Select Case Euros
```

```
Case ""
```

```
Euros = "Nul Euro"
```

```
Case "Een"
```

```
Euros = "Een Euro"
```

```
Case Else
```

```
Euros = Euros & " Euro"
```

```
End Select
```

' Clean up cents.

```
Select Case Cents
```

```
Case ""
```

```
Cents = " En nul cent"
```

```
Case "One"
```

```
Cents = " En één cent"
```

```
Case Else
```

```
Cents = " En " & Cents & " cent"
```

```
End Select
```

Met een Select Case, inmiddels wel bekend, voegen we het woord Euro toe aan de variabele Euros. Hetzelfde met de variabele Cents, waarmee we de centen opmaken.

' Als laatste de hoofdletters goed zetten.

```
Temp = Split(Euros, " ")
Euros = ""
Temp(LBound(Temp)) = StrConv(Temp(LBound(Temp)), vbProperCase)
For iTmp = LBound(Temp) + 1 To UBound(Temp) - 1
    Temp(iTmp) = LCase(Temp(iTmp))
Next iTmp
For iTmp = LBound(Temp) To UBound(Temp)
    Euros = Euros & Temp(iTmp)
    If iTmp < UBound(Temp) Then Euros = Euros & " "
Next iTmp
Cents = StrConv(Cents, vbLowerCase)
ConverteerGetalNaarEuros = Euros & Cents
End Function
```

De functie gebruiken

Het toepassen van de functies gaat op dezelfde manier als een normale functie gebruikt zou worden. In een query kun je bijvoorbeeld een getal omzetten met het veld:

Prijs in tekst: ConverteerGetalNaarEuros([Totaalprijs])

En hetzelfde geldt voor het toepassen op formulieren en in rapporten.

Omdat deze functie gebruik maakt van standaard VBA opdrachten, is hij ook (zelfs letterlijk!) te gebruiken in Excel, en zelfs in Word. In Excel maak je dan een module aan, en gebruik je de functie met de knop <Functie Plakken>, waarbij je dan de categorie <User defined> gebruikt om de functie te selecteren. De waarde die omgezet moet worden haal je vervolgens uit een cel. Voor gebruik in Word moet je iets meer moeite doen, want de om te zetten waarde moet je dan met Bladwijzers opzoeken, en teruglezen in een nieuw veld. Maar in Word kun je, zoals ik in de inleiding al zei, een getalveld al rechtstreeks vertalen naar tekst.

Bronvermelding: Het stukje over de juiste manier om een getal in woorden te gebruiken heb ik geciteerd van de website van het Genootschap Onze Taal; een website die ik van harte mag aanbevelen!

De complete code

Voor de volledigheid nog de code zonder onderbrekingen, zodat je hem in één keer kunt selecteren in in een module kunt plakken.

Option Compare Database

Option Explicit

Option Base 1

Public Function ConverteerGetalNaarTekst(ByVal MijnGetal)

Dim Temp

Dim GrootGetal As String, KleinGetal As String, tmpGetal As String

Dim Decimaal As Double, Teller As Long

Dim GeheelGetal() As Integer

Dim iTmp As Integer, i As Integer, x As Integer

ReDim Groep(9) As String

Groep(2) = "duizend "

Groep(3) = " miljoen "

Groep(4) = " miljard "

Groep(5) = " biljard "

' Zet het getal om naar Tekst, en verwijder eventuele spaties.

MijnGetal = Trim(Str(Round(MijnGetal, 2)))

' Zoek het decimaalteken. In VBA is dat een punt.

Decimaal = InStr(MijnGetal, ".")

' Als we een decimaalteken hebben gevonden...

If Decimaal > 0 Then

' Dan moet dat deel apart omgezet worden.

Temp = Left(Mid(MijnGetal, Decimaal + 1) & "00", 2)

```

KleinGetal = OmzettenTientallen(Temp)

' Verwijder nu KleinGetal uit de variabele.

MijnGetal = Trim(Left(MijnGetal, Decimaal - 1))

End If

tmpGetal = MijnGetal

If tmpGetal = "" Then tmpGetal = "000"

Do Until Len(tmpGetal) Mod 3 = 0

    tmpGetal = "0" & tmpGetal

Loop

iTmp = Len(tmpGetal) / 3

ReDim GeheelGetal(iTmp)

For i = 1 To Len(tmpGetal) Step 3

    x = x + 1

    GeheelGetal(x) = Mid(tmpGetal, i, 3)

Next i

tmpGetal = ""

For x = UBound(GeheelGetal) To LBound(GeheelGetal) Step -1

    Teller = Teller + 1

    Temp = OmzettenHonderd(GeheelGetal(x))

    If Temp <> "" Then tmpGetal = Temp & Groep(Teller) & tmpGetal

Next x

If tmpGetal = "" Then

    GrootGetal = "Nul"

    If Not KleinGetal = "" Then KleinGetal = " komma " & StrConv(KleinGetal, vbLowerCase)

Else

    ' Laatste stap: getal netjes opmaken.

    Temp = Split(tmpGetal, " ")
    
```

```
Temp(LBound(Temp)) = StrConv(Temp(LBound(Temp)), vbProperCase)

For iTmp = LBound(Temp) + 1 To UBound(Temp)

    Temp(iTmp) = LCase(Temp(iTmp))

Next iTmp

For iTmp = LBound(Temp) To UBound(Temp)

    GrootGetal = GrootGetal & Temp(iTmp)

    If iTmp < UBound(Temp) Then GrootGetal = GrootGetal & " "

Next iTmp

If Not KleinGetal = "" Then KleinGetal = " komma " & StrConv(KleinGetal, vbLowerCase)

End If

ConverteerGetalNaarTekst = GrootGetal & KleinGetal

End Function

Function ConverteerGetalNaarEuros(ByVal MijnGetal)

Dim Temp

Dim Euros As String, Cents As String

Dim Decimaal As Double, Teller As Long

Dim sNumber As String

Dim iTmp As Integer

ReDim Groep(9) As String

Groep(2) = "duizend "

Groep(3) = " miljoen "

Groep(4) = " miljard "

Groep(5) = " trillioen "
```

' Zet het getal om naar Tekst, en verwijder eventuele spaties.

```
MijnGetal = Trim(Str(MijnGetal))
```

' Zoek het decimaalteken. In VBA is dat een punt.

```
Decimaal = InStr(MijnGetal, ".")
```

' Als we een decimaalteken hebben gevonden...

```
If Decimaal > 0 Then
```

```
    ' Dan moet dat deel apartomgezet worden.
```

```
    Temp = Left(Mid(MijnGetal, Decimaal + 1) & "00", 2)
```

```
    Cents = OmzettenTientallen(Temp)
```

```
    ' Strip off cents from remainder to convert.
```

```
    MijnGetal = Trim(Left(MijnGetal, Decimaal - 1))
```

```
End If
```

```
Do While MijnGetal <> ""
```

```
    Teller = Teller + 1
```

```
    ' De laatste 3 cijfers van MijnGetal omzetten.
```

```
    Temp = OmzettenHonderd(Right(MijnGetal, 3))
```

```
    If Temp <> "" Then Euros = Temp & Groep(Teller) & Euros
```

```
    If Len(MijnGetal) > 3 Then
```

```
        ' En vervolgens de laatste 3 cijfers verwijderen uit MijnGetal.
```

```
        MijnGetal = Left(MijnGetal, Len(MijnGetal) - 3)
```

```
    Else
```

```
        MijnGetal = ""
```

```
    End If
```

```
Loop
```


' Laatste stap: Euros toevoegen aan de tekst.

Select Case Euros

Case ""

Euros = "Nul Euro"

Case "Een"

Euros = "Een Euro"

Case Else

Euros = Euros & " Euro"

End Select

' Clean up cents.

Select Case Cents

Case ""

Cents = " En nul cent"

Case "One"

Cents = " En één cent"

Case Else

Cents = " En " & Cents & " cent"

End Select

' Als laatste de hoofdletters goed zetten.

Temp = Split(Euros, " ")

Euros = ""

Temp(LBound(Temp)) = StrConv(Temp(LBound(Temp)), vbProperCase)

For iTmp = LBound(Temp) + 1 To UBound(Temp) - 1

Temp(iTmp) = LCase(Temp(iTmp))

Next iTmp

```
For iTmp = LBound(Temp) To UBound(Temp)
    Euros = Euros & Temp(iTmp)
    If iTmp < UBound(Temp) Then Euros = Euros & " "
Next iTmp
Cents = StrConv(Cents, vbLowerCase)
ConverteerGetalNaarEuros = Euros & Cents
```

End Function

```
Private Function OmzettenHonderd(ByVal MijnGetal)
```

```
Dim Result As String
```

```
' Stoppen als er niks is om te zetten.
```

```
If Val(MijnGetal) = 0 Then Exit Function
```

```
' Leading zeros toevoegen aan getal.
```

```
MijnGetal = Right("000" & MijnGetal, 3)
```

```
' Moeten we een veelvoud van honderd converteren?
```

```
If Left(MijnGetal, 1) <> "0" Then
```

```
    Result = OmzettenEnkelGetal(Left(MijnGetal, 1)) & "honderd"
```

```
End If
```

```
' Moeten we een veelvoud van tien converteren?
```

```
If Mid(MijnGetal, 2, 1) <> "0" Then
```

```
    Result = Result & OmzettenTientallen(Mid(MijnGetal, 2))
```

```
Else
```

```
    ' Zo nee, dan alleen een enkel cijfer omzetten.
```

```
Result = Result & OmzettenEnkelGetal(Mid(MijnGetal, 3))
```

```
End If
```

```
OmzettenHonderd = Trim(Result)
```

```
End Function
```

```
Private Function OmzettenTientallen(ByVal MyTens)
```

```
Dim Result As String, ResultSingle As String
```

```
' Is de waarde tussen 10 en 19?
```

```
If Val(Left(MyTens, 1)) = 1 Then
```

```
    Select Case Val(MyTens)
```

```
        Case 10: Result = "Tien"
```

```
        Case 11: Result = "Elf"
```

```
        Case 12: Result = "Twaalf"
```

```
        Case 13: Result = "Dertien"
```

```
        Case 14: Result = "Veertien"
```

```
        Case 15: Result = "Vijftien"
```

```
        Case 16: Result = "Zestien"
```

```
        Case 17: Result = "Zeventien"
```

```
        Case 18: Result = "Achtien"
```

```
        Case 19: Result = "Negentien"
```

```
    Case Else
```

```
End Select
```

```
Else
```

```
' .. anders is het getal tussen 20 en 99.
```

```
    Select Case Val(Left(MyTens, 1))
```

```
        Case 2: Result = "Twintig"
```

Case 3: Result = "Dertig"

Case 4: Result = "Veertig"

Case 5: Result = "Vijftig"

Case 6: Result = "Zestig"

Case 7: Result = "Zeventig"

Case 8: Result = "Tachtig"

Case 9: Result = "Negentig"

Case Else

End Select

End If

' En dan het achterste cijfer omzetten.

ResultSingle = OmzettenEnkelGetal(Right(MyTens, 1))

If Not ResultSingle = "" Then Result = ResultSingle & "en" & Result

OmzettenTientallen = Result

End Function

Private Function OmzettenEnkelGetal(ByVal MyDigit)

Select Case Val(MyDigit)

Case 1: OmzettenEnkelGetal = "Een"

Case 2: OmzettenEnkelGetal = "Twee"

Case 3: OmzettenEnkelGetal = "Drie"

Case 4: OmzettenEnkelGetal = "Vier"

Case 5: OmzettenEnkelGetal = "Vijf"

Case 6: OmzettenEnkelGetal = "Zes"

Case 7: OmzettenEnkelGetal = "Zeven"

Case 8: OmzettenEnkelGetal = "Acht"

Case 9: OmzettenEnkelGetal = "Negen"

Case Else: OmzettenEnkelGetal = ""

End Select

End Function

Helpmij.nl