LibreOffice
The Document Foundation

Handleiding voor Calc

Hoofdstuk 12
Macro's in Calc

Automatisch herhalende taken

ODFAuthors

Documentatie voor LibreOffice is beschikbaar op nl.libreoffice.org en
www.odfauthors.org/libreoffice/nederlands

Auteursrechten

Dit document is onder auteursrecht © 2010 – 2015 van de bijdragers die onderstaand zijn genoemd. U mag het verspreiden en/of aanpassen met inachtneming van de condities van GNU General Public License (http://www.gnu.org/licenses/gpl.html), versie 3 of hoger of de Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), versie 4 of hoger.

Alle handelsmerken in deze gids zijn eigendom van de rechtmatige eigenaars.

Auteurs

Andrew Pitonyak
Gary Schnabl
Jean Hollis Weber
Claire Wood
Martin Fox

Reacties

Opmerkingen en suggesties kunnen aan de oorspronkelijke auteurs van de Engelstalige handleiding via de mailinglijst van het documentatieteam (documentation@global.libreoffice.org) gezonden worden.

Opmerking: alles wat u naar de mailinglijst stuurt, inclusief uw e-mailadres en andere persoonlijke informatie die in het bericht staat, wordt openbaar gearchiveerd en kan niet verwijderd worden.

Heeft u op- of aanmerkingen over de Nederlandstalige vertalingen dan kunt u terecht op de mailinglijst: discuss@nl.libreoffice.org

Inschrijven kan via een mailtje aan *discuss+subscribe@nl.libreoffice.org*

Vertalers

Dick Groskamp

Medewerkers

Aanpassingen van de lay-out voor LibreOffice

Kees Kriek
Leo Moons

Opmerking

De medewerkers hebben er voor gekozen om zo snel mogelijk een handleiding voor Calc te publiceren en gaven de voorkeur om de bestaande vertaling van Dick Groskamp te bewerken in plaats van te wachten op de “Calc Guide 4”. Belangrijke wijzigingen tot versie 3.6 werden verwerkt, maar we hadden niet de ambitie om volledig te zijn.

Publicatiedatum en softwareversie

Gepubliceerd op 10 maart 2011. Gebaseerd op LibreOffice 3.x. Vertaald mei 2015

Opmerking voor gebruikers van Mac

Sommige toetsaanslagen en menu-items zijn anders op een Mac dan die in Windows en Linux. De tabel hieronder geeft enkele algemene vervangingen voor de instructies in dit hoofdstuk. Voor een meer gedetailleerde lijst, bekijk de Help van deze toepassing.

Windows/Linux	*Equivalent voor Mac*	*Effect*
Menuselectie **Extra > Opties**	**LibreOffice > Voorkeuren**	Toegang tot instellingsopties
Klik met rechts	*Control+klik*	Contextmenu openen
Ctrl (Control)	*z* (Command)	Gebruikt met andere toetsen
F5	*Shift+z+F5*	De Navigator openen
F11	*z+T*	Het venster **Stijlen en opmaak** openen

Inhoud

Auteursrechten....3
Opmerking voor gebruikers van Mac....4
Introductie....7
De Macrorecorder gebruiken....7
Schrijf uw eigen functies....11
Een macro als een functie gebruiken....13
Argumenten doorgeven aan een macro....16
Argumenten doorgeven als waarden....18
Macro's schrijven die acteren als ingebouwde functies....18
Directe toegang tot cellen....18
Sorteren....19
Conclusie....20

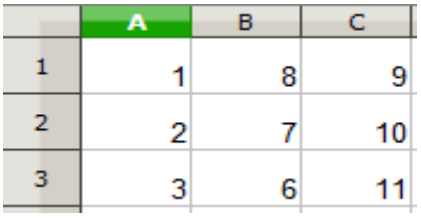
Introductie

Een macro is een reeks van opdrachten of toetsaanslagen die worden opgeslagen om later te gebruiken. Een voorbeeld van een eenvoudige macro is er een die uw adres 'typt'. De macrotaal van LibreOffice is erg flexibel en maakt het mogelijk zowel eenvoudige als complexe taken te automatiseren. Macro's zijn speciaal handig om een taak steeds weer op dezelfde manier te laten uitvoeren. Dit hoofdstuk beschrijft in het kort algemene problemen die zijn gerelateerd aan het programmeren van macro's in Calc.

De Macrorecorder gebruiken

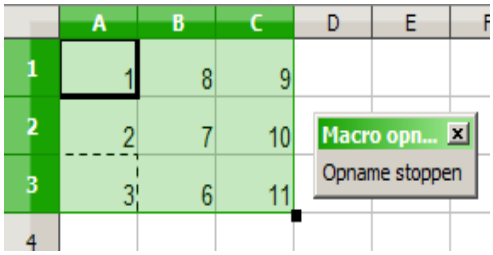
Hoofdstuk 13 van de *Handleiding voor Beginners, Starten met macro's*, biedt een basis voor het begrijpen van de algemene mogelijkheden van macro's in LibreOffice met behulp van de macrorecorder. Een voorbeeld wordt hier weergegeven zonder de uitleg van Handleiding voor Beginners. De volgende stappen maken een macro die *Plakken speciaal* uitvoert met vermenigvuldigen.

1) Open een nieuw werkblad.
2) Voer getallen in een blad in.



Afbeelding 1: Getallen invoeren

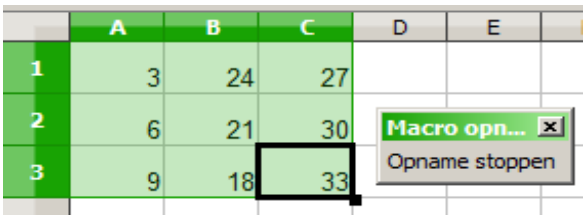
3) Selecteer cel A3, die het getal 3 bevat, en kopieer de waarde naar het Klembord.
4) Selecteer het bereik A1:C3.
5) Gebruik **Extra > Macro's > Macro opnemen** om de macrorecorder te starten. Het dialoogvenster **Macro opnemen** wordt weergegeven met een knop **Stop opnemen**.



Afbeelding 2: Knop Opname stoppen

6) Gebruik **Bewerken > Plakken speciaal** om het dialoogvenster **Plakken speciaal** te openen (*Afbeelding 3*).
7) Stel de bewerking in op **Vermenigvuldigen** en klik op **OK**. De cellen worden nu vermenigvuldigd met 3 (*Afbeelding 4*).
8) Klik op **Opname stoppen** om de macrorecorder te stoppen. Het dialoogvenster **LibreOffice Basic-macro's** (*Afbeelding 5*) opent.

Afbeelding 3: Dialoogvenster Plakken speciaal



Afbeelding 4: Cellen vermenigvuldigd met 3

9) Selecteer het huidige document (zie *Afbeelding 5*). Voor dit voorbeeld is het huidige document van Calc: *Geen titel 1*. Bestaande documenten geven een bibliotheek weer, genaamd *Standard*. Deze bibliotheek wordt niet gemaakt totdat het document is opgeslagen of de bibliotheek nodig is. Dus op dit moment bevat uw nieuwe document geen bibliotheek. U kunt een bibliotheek aanmaken om de macro in op te nemen, maar dat is niet noodzakelijk.

1 Mijn macro's

2 LibreOffice macro's

3 Geopende documenten

4 Nieuwe bibliotheek maken

5 Nieuwe module in bibliotheek maken

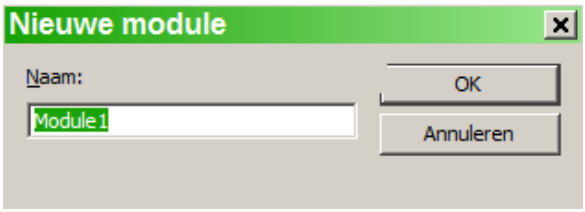
6 Macro's in geselecteerde bibliotheek

7 Huidige document

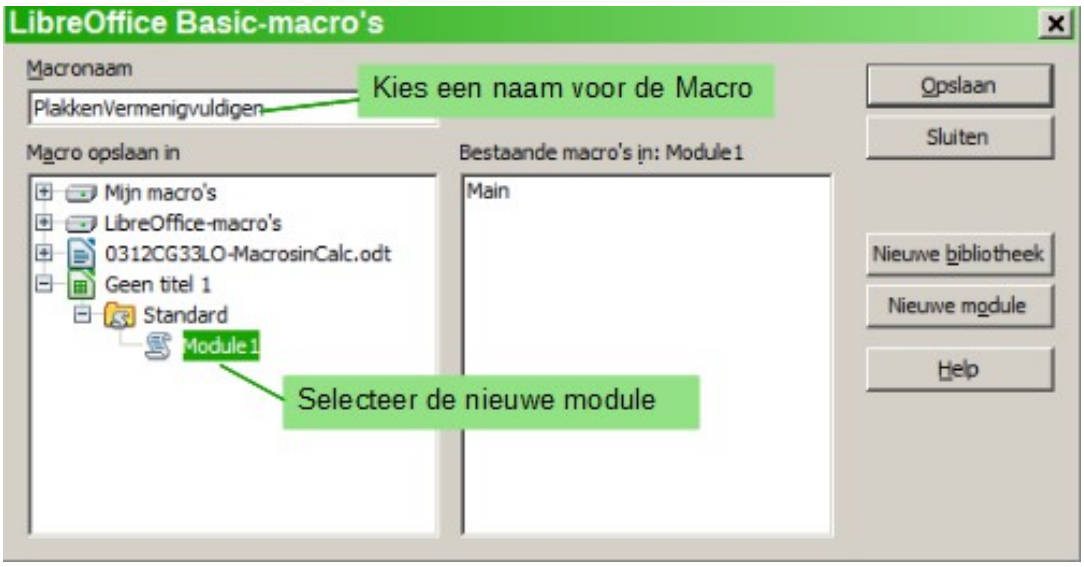
8 Lijst uitbreiden/samenvouwen

Afbeelding 5: Delen van het dialoogvenster LibreOffice Basic-macro's

10) Klik op **Nieuwe module**. Indien er geen bibliotheken bestaan, dan wordt de bibliotheek Standard automatisch gemaakt en gebruikt. Typ, in het dialoogvenster **Nieuwe module**, een naam voor de nieuwe module of laat de naam 'Module1' staan.



11) Klik op **OK** om een nieuwe module te maken, genaamd Module1. Selecteer de nieuw gemaakte Module1, typ **`PlakkenVermenigvuldigen`** in het vak *Macronaam* linksboven en klik op **Opslaan**. (Zie *Afbeelding 5*.)



Afbeelding 6: Selecteer de module en geef de macro een naam

De opgenomen macro wordt opgeslagen in Module1 van de bibliotheek Standard in het document *Geen titel 1*. Lijst 1 geeft de inhoud van de macro weer.

Lijst 1. Plakken speciaal met vermenigvuldigen.

```
sub PlakkenVermenigvuldigen
rem ----------------------------------------------------------------
rem variabelen definiëren
dim document   as object
dim dispatcher as object
rem ----------------------------------------------------------------
rem toegang krijgen tot het document
document   = ThisComponent.CurrentController.Frame
dispatcher = createUnoService("com.sun.star.frame.DispatchHelper")

rem ----------------------------------------------------------------
dim args1(5) as new com.sun.star.beans.PropertyValue
args1(0).Name = "Flags"
args1(0).Value = "A"
args1(1).Name = "FormulaCommand"
args1(1).Value = 3
args1(2).Name = "SkipEmptyCells"
args1(2).Value = false
args1(3).Name = "Transpose"
args1(3).Value = false
args1(4).Name = "AsLink"
args1(4).Value = false
args1(5).Name = "MoveMode"
args1(5).Value = 4

dispatcher.executeDispatch(document, ".uno:InsertContents", "", 0,
args1())
end sub
```

Meer details over het opnemen van macro's wordt geboden in hoofdstuk 13 (*Starten met macro's*) in de *Handleiding voor beginners*; we bevelen het u aan om het te lezen als u dat nog niet gedaan hebt. Meer details worden ook gegeven in de volgende gedeelten, maar niet in relatie tot het opnemen van macro's.

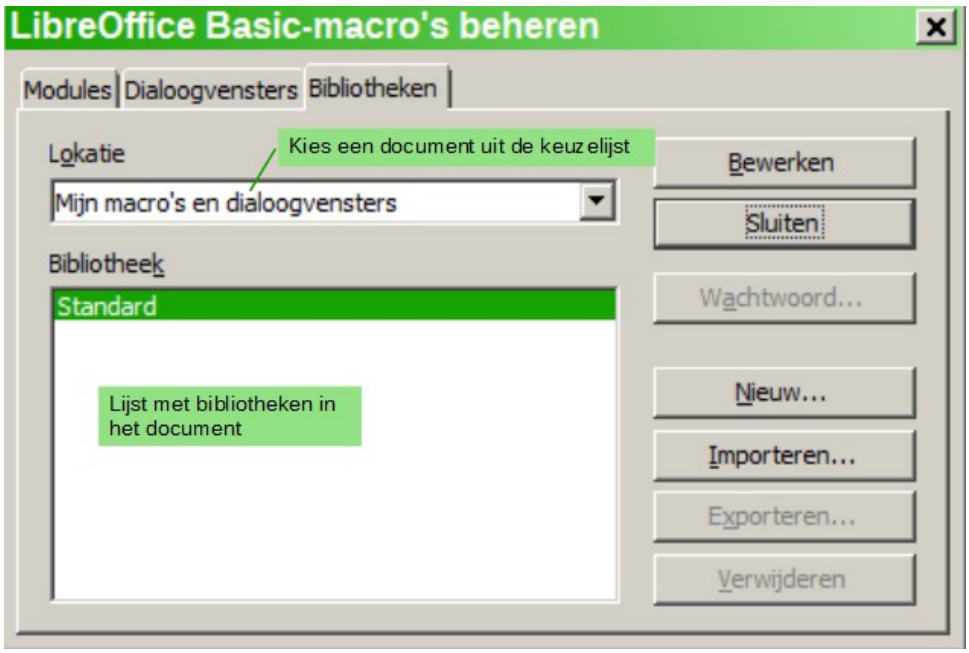
Schrijf uw eigen functies

Calc kan macro's aanroepen als functies. Gebruik de volgende stappen om een eenvoudige macro te maken:

1) Maak een nieuwe document voor Calc, genaamd `CalcTestMacros.ods`.
2) Gebruik **Extra > Macro's > Macro's beheren > LibreOffice Basic** om het dialoogvenster **LibreOffice Basic-macro's** te openen. Het vak *Macro uit* somt de beschikbare bibliotheken voor macro's op, inclusief die voor de huidige geopende documenten in LibreOffice. *Mijn macro's* bevat macro's die u schrijft of toevoegt aan LibreOffice. *LibreOffice macro's* bevat macro's die zijn geleverd met LibreOffice en zouden niet moeten worden gewijzigd.

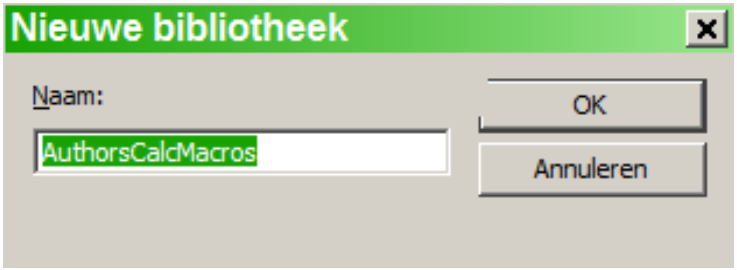
Afbeelding 7: Dialoogvenster LibreOffice Basic-macro's

3) Klik op **Beheren** om het dialoogvenster **LibreOffice Basic-macro's beheren** (*Afbeelding 8*) te openen. Selecteer, op de tabpagina *Bibliotheken*, het document dat de macro moet gaan bevatten.



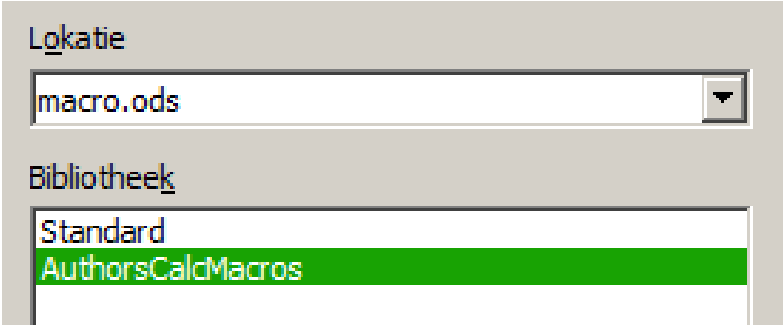
Afbeelding 8: LibreOffice Basic-macro's beheren

4) Klik op **Nieuw** om het dialoogvenster **Nieuwe bibliotheek** te openen.



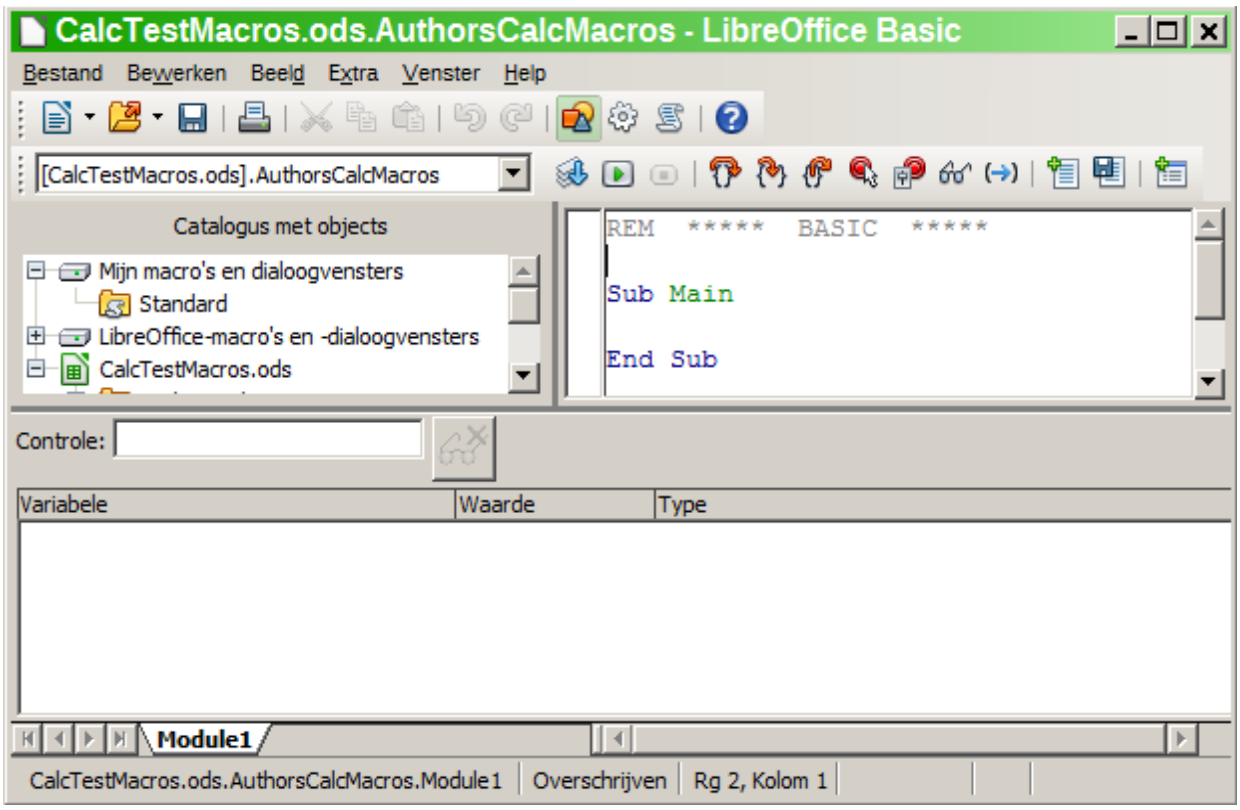
Afbeelding 9: Dialoogvenster Nieuwe bibliotheek

5) Voer een beschrijvende naam voor de bibliotheek in (zoals AuthorsCalcMacros) en klik op **OK** om de bibliotheek te maken. De nieuwe naam voor de bibliotheek wordt weergegeven in de lijst met bibliotheken, maar het dialoogvenster zou slechts een deel van de naam weer kunnen geven.



Afbeelding 10: De bibliotheek wordt weergegeven in beheer

6) Selecteer AuthorsCalcMacros en klik op **Bewerken** om de bibliotheek te bewerken. Calc maakt automatisch een module genaamd Module1 en een macro genaamd Main.



Afbeelding 11: Basic Integrated Development Environment (IDE)

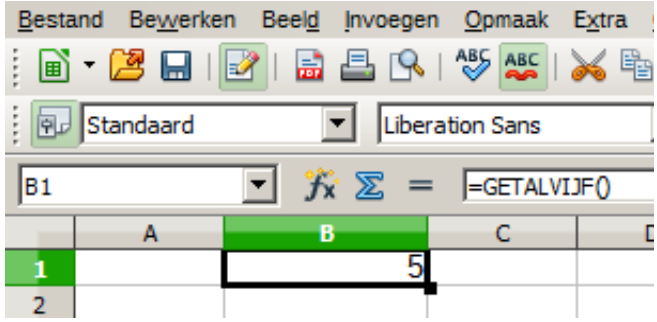
7) Pas de code zo aan dat die hetzelfde is als die welke wordt weergegeven in *Lijst 2*. De belangrijkste aanvulling is het maken van de functie `GetalVijf`, die het getal vijf teruggeeft. De uitdrukking `Option Explicit` forceert dat alle variabelen moeten worden gedeclareerd vóór zij kunnen worden gebruikt. Als Option Explicit wordt weggelaten, worden variabelen bij hun eerste gebruik automatisch gedefinieerd als type Variant.
8) Sla de aangepaste Module1 op.

Lijst 2. Functie die vijf teruggeeft.

```
REM  *****  BASIC  *****
Option Explicit

Sub Main

End Sub

Function GetalVijf()
  GetalVijf = 5
End Function
```

Een macro als een functie gebruiken

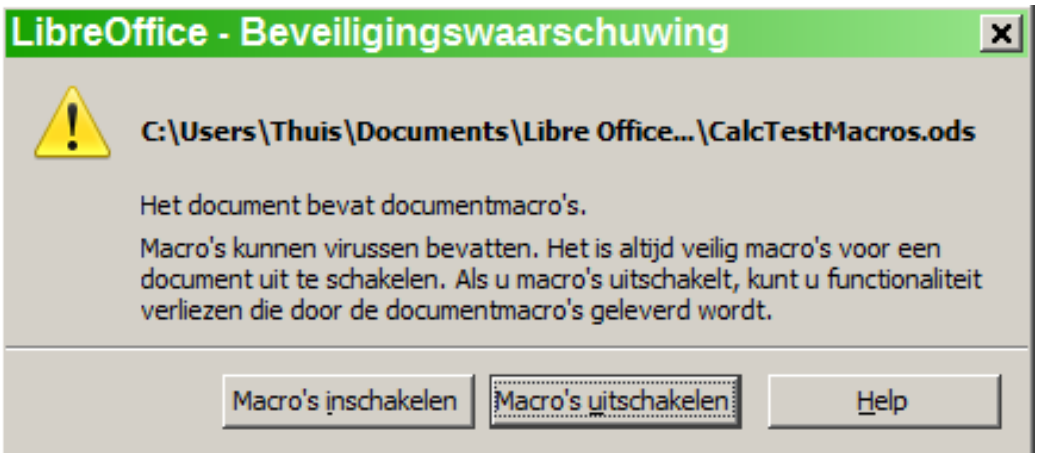
In het nieuw gemaakte Calcdocument `CalcTestMacros.ods`, voer de formule `=GetalVijf()` in (zie *Afbeelding 12*). Calc zoekt de macro en roept die aan.



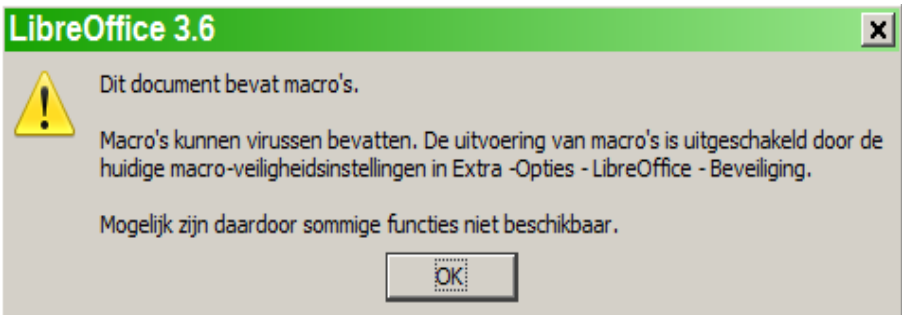
Afbeelding 12: gebruik de macro GetalVijf() als een functie in Calc

Tip	Functienamen zijn niet hoofdlettergevoelig. In *Afbeelding 12* kunt u `=GetalVijf()` invoeren en Calc geeft duidelijk `=GETALVIJF()` weer.

Sla het Calc-document op, sluit het en open het opnieuw. Afhankelijk van uw instellingen in **Extra > Opties > LibreOffice > Beveiliging > Macroveiligheid**, zal Calc de waarschuwing weergeven die wordt weergegeven in *Afbeelding 13* of die welke wordt weergegeven in *Afbeelding 14*. U zult op **Macro's inschakelen** moeten drukken, of Calc zal niet toestaan dat in het document macro's worden uitgevoerd. Als u niet verwacht dat een document een macro bevat, is het veiliger om op **Macro's uitschakelen** te klikken voor het geval de macro een virus is.

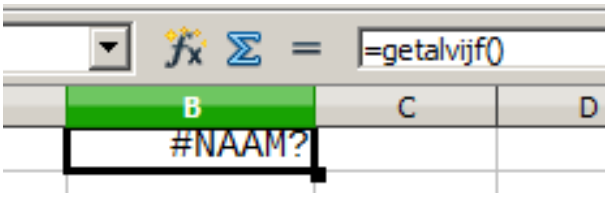


Afbeelding 13: LibreOffice waarschuwt u dat een document macro's bevat



Afbeelding 14: Waarschuwing als macro's zijn uitgeschakeld

Indien u er voor kiest om macro's uit te schakelen, dan zal, als het document wordt geladen, Calc de functie niet langer vinden.



Afbeelding 15: De functie is weg

Als een document is gemaakt en opgeslagen, bevat het automatisch een bibliotheek genaamd Standard. De bibliotheek Standard wordt automatisch geladen als het document wordt geopend. Geen enkele andere bibliotheek wordt automatisch geladen.

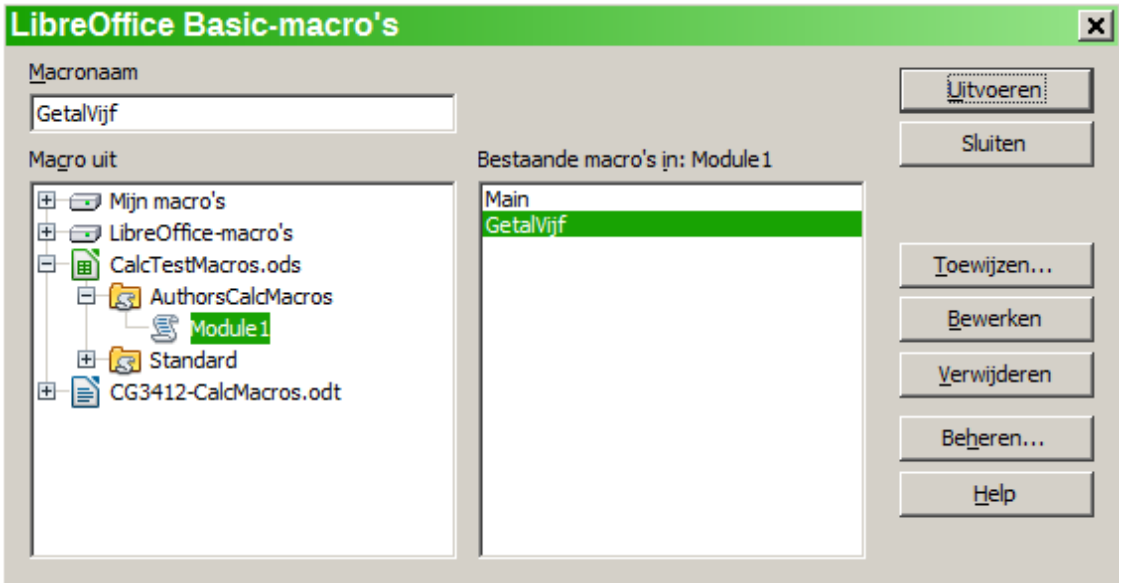
Calc heeft geen functie genaamd GetalfVijf(), dus controleert het alle geopende en zichtbare macrobibliotheken op de functie. Bibliotheken in *LibreOffice macro's*, *Mijn macro's*, en het Calcdocument worden gecontroleerd op een toepasselijke benoemde functie (zie *Afbeelding 7*). De functie GetalVijf() is opgeslagen in de bibliotheek AuthorsCalcMacros, die niet automatisch wordt geladen als het document wordt geopend.

Gebruik **Extra > Macro's > Macro's beheren > LibreOffice BASIC** om het dialoogvenster **LibreOffice BASIC-macro's** te openen (zie *Afbeelding 16*). Breidt CalcTestMacros uit en zoek AuthorsCalcMacros. Het pictogram voor een geladen bibliotheek heeft een andere kleur dan het pictogram voor een bibliotheek die niet is geladen.

Klik op het uitbreidingssymbool (gewoonlijk een plus of een driehoekje) naast AuthorsCalcMacros om de bibliotheek te laden. Het pictogram wijzigt van kleur om aan te geven dat de bibliotheek nu is geladen. Klik op **Sluiten** om het dialoogvenster te sluiten.

Helaas geven de cellen die =GetalVijf() bevatten nu een fout aan. Calc berekent cellen die fouten bevatten niet opnieuw, tenzij u ze bewerkt of anderszins wijzigt. De gebruikelijke oplossing is om macro's die worden gebruikt als functies op te slaan in de bibliotheek Standard. Als de macro groot is of als er veel macro's zijn, wordt een verkorte macro met de gewenste naam opgeslagen in de bibliotheek Standard. De verkorte macro laadt de bibliotheek die de implementatie bevat en roept dan de implementatie aan.

1) Gebruik **Extra > Macro's > Macro's beheren > LibreOffice Basic** om het dialoogvenster **LibreOffice Basic-macro's** te openen. Selecteer de macro GetalVijf en klik op **Bewerken** om de macro voor bewerking te openen.



Afbeelding 16: Selecteer een macro en klik op Bewerken

2) Wijzig de naam van GetalVijf naar GetalVijf_Implementatie (*Lijst 3*).

Lijst 3. Wijzig de naam van GetalVijf naar GetalVijf_Implementatie

```
Function GetalVijf_Implementatie()
  GetalVijf_Implementatie() = 5
End Function
```

3) Ga, in de BASIC IDE (zie *Afbeelding 11*), met de muiscursor over de knoppen van de werkbalk om de helptips weer te geven. Klik op de knop **Macro selecteren** om het dialoogvenster **LibreOffice Basic-macro's** te openen (zie *Afbeelding 16*).
4) Selecteer de bibliotheek Standard in het document CalcTestMacros en klik op **Nieuw** om een nieuwe module te maken. Voer een betekenisvolle naam in zoals CalcFuncties en klik op **OK**. LibreOffice maakt automatisch een macro genaamd Main en opent de module om te bewerken.
5) Maak een macro in de bibliotheek Standard die de functie voor implementatie aanroept (zie *Lijst 4*). De nieuwe macro laadt de bibliotheek AuthorsCalcMacros als die nog niet geladen is, en roept dan de functie voor implementatie aan.
6) Sla het Calcdocument op, sluit het en open het weer opnieuw. Deze keer werkt de functie GetalfVijf().

Lijst 4. Wijzig de naam van GetalVijf naar GetalVijf_Implementatie

```
Function GetalVijf()
  If NOT BasicLibraries.isLibraryLoaded("AuthorsCalcMacros") Then
    BasicLibraries.LoadLibrary("AuthorsCalcMacros")
  End If
  GetalVijf = GetalVijf_Implementatie()
End Function
```

Argumenten doorgeven aan een macro

We zullen een macro schrijven die de som van zijn argumenten berekent die positief zijn – het zal argumenten die kleiner zijn dan nul negeren (zie *Lijst 5*) om een functie te illustreren die argumenten accepteert.

Lijst 5. PositieveSom berekent de som van de positieve argumenten.

```
Function PositieveSom(Optional x)
```

```
  Dim DeSom As Double
  Dim iRij As Integer
  Dim iKol As Integer

  DeSom = 0.0
  If NOT IsMissing(x) Then
    If NOT IsArray(x) Then
      If x > 0 Then DeSom = x
    Else
      For iRij = LBound(x, 1) To UBound(x, 1)
        For iKol = LBound(x, 2) To UBound(x, 2)
          If x(iRij, iKol) > 0 Then DeSom = DeSom + x(iRij, iKol)
        Next
      Next
    End If
  End If
  PositieveSom = DeSom
End Function
```

De macro in *Lijst 5* demonstreert enkele belangrijke technieken:

7) Het argument x is optioneel. Wanneer een argument niet optioneel is en de functie wordt aangeroepen zonder dat argument, geeft LibreOffice, elke keer dat de macro wordt aangeroepen, een waarschuwing. Als Calc de functie vele malen aanroept, dan wordt de fout ook vele malen weergegeven.
8) `IsMissing` controleert of een argument werd doorgegeven vóórdat het argument wordt gebruikt.
9) `IsArray` controleert om te zien of het argument één enkele waarde is of een array. Bijvoorbeeld: `=PositieveSom(7)` of `=PositieveSom(A4)`. In het eerste geval wordt het getal 7 doorgegeven als een argument, en in het tweede geval wordt de waarde van cel A4 doorgegeven aan de functie.
10) Als een bereik wordt doorgegeven aan de functie, wordt het doorgegeven als een tweedimensionale array van waarden; bijvoorbeeld: `=PositieveSom(A2:B5)`. `LBound` en `UBound` worden gebruikt om de grenzen van de gebruikte array te bepalen. Hoewel de laagste grens één is, wordt het als veiliger beschouwd om `LBound` te gebruiken voor het geval het in de toekomst wijzigt.

Tip	De macro in *Lijst 5* is zorgvuldig en controleert of het argument een array of één enkel argument is. De macro verifieert niet of elke waarde numeriek is. U kunt zo zorgvuldig zijn als u zelf wilt. Hoe meer dingen u controleert, des te robuuster de macro is en hoe trager hij wordt uitgevoerd.

Doorgeven van één argument is net zo eenvoudig als er twee doorgeven: voeg een ander argument toe aan de definitie van de functie (zie *Lijst 6*). Bij het aanroepen van een functie met twee argumenten, scheidt de argumenten door een puntkomma; bijvoorbeeld: `=TestMax(3; -4)`.

Lijst 6. TestMax accepteert twee argumenten en geeft het grootste van de twee terug.

```
Function TestMax(x, y)
  If x >= y Then
    TestMax = x
  Else
    TestMax = y
  End If
End Function
```

Argumenten doorgeven als waarden

Argumenten die worden doorgegeven aan een macro uit Calc zijn altijd waarden. Het is niet mogelijk om te weten welke cellen, indien enige, zijn gebruikt. Bijvoorbeeld, `=PositieveSom(A3)` geeft de waarde van cel A3 door, en PositieveSom heeft geen manier om te weten dat cel A3 werd gebruikt. Indien u moet weten naar welke cellen werd verwezen in plaats van de waarden in de cellen, geef dan het bereik als een tekenreeks door, ontleed de tekenreeks en verkrijg de waarden uit de cellen waarnaar verwezen werd.

Macro's schrijven die acteren als ingebouwde functies

Hoewel Calc macro's zoekt en aanroept als normale functies, gedragen zij zich niet echt als ingebouwde functies. Macro's verschijnen, bijvoorbeeld, niet in de lijst met functies. Het is mogelijk om functies te schrijven die zich gedragen als reguliere functies door een Add-In te schrijven. Dit is echter een gevorderd onderwerp dat hier niet behandeld wordt; zie

http://wiki.services.LibreOffice/wiki/SimpleCalcAddIn

Directe toegang tot cellen

U kunt direct toegang krijgen tot de interne objecten van LibreOffice om een Calcdocument te manipuleren. De macro in *Lijst 7*, bijvoorbeeld, telt de waarden in cel A2 van elk blad in het huidige document bij elkaar op. `ThisComponent` wordt ingesteld door StarBasic wanneer de macro begint te verwijzen naar het huidige document. Een Calcdocument bevat bladen: `ThisComponent.getSheets()`. Gebruik `getCellByPosition(col, row)` om een cel terug te geven met een specifieke rij en kolom.

Lijst 7. Tel cel A2 in elk blad bij elkaar op.

```
Function SomCellenAlleBladen()
  Dim DeSom As Double
  Dim i As integer
  Dim oBladen
  Dim oBlad
  Dim oCel

  oBladen = ThisComponent.getSheets()
  For i = 0 To oBladen.getCount() - 1
    oBlad = oBladen.getByIndex(i)
    oCel = oBlad.getCellByPosition(0, 1) ' Cel A2 ophalen
    DeSom = DeSom + oCel.getValue()
  Next
  SomCellenAlleBladen = DeSom
End Function
```

Tip	Een object cel ondersteunt de methoden `getValue()`, `getString()` en `getFormula()` om de numerieke waarde te verkrijgen, de waarde van de tekenreeks, of de in een cel gebruikte formule. Gebruik de overeenkomende set functies om de toepasselijke waarden in te stellen.

Gebruik `oBlad.getCellRangeByName("A2")` om een celbereik met een naam terug te geven. Indien naar één enkele cel wordt verwezen, dan wordt een object cel teruggegeven. Indien een celbereik wordt opgegeven, dan wordt een geheel bereik van cellen teruggegeven (zie *Lijst 8*). Merk op dat een celbereik gegevens teruggeeft als een array van arrays, wat tijdrovender is dan het te behandelen als een array met twee dimensies zoals wordt gedaan in *Lijst 5*.

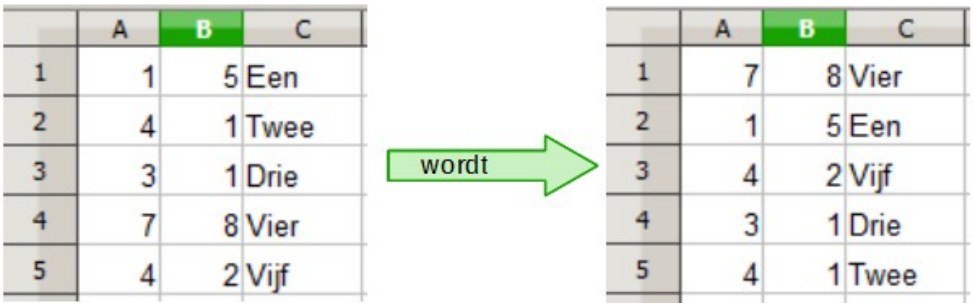
Lijst 8. Cellen A2:C5 in elk blad bij elkaar optellen

```
Function SomCellenAlleBladen()
  Dim DeSom As Double
  Dim iRij As Integer, iKol As Integer, i As Integer
  Dim oBladen, oBlad, oCellen
  Dim oRij(), oRijen()

  oBladen = ThisComponent.getSheets()
  For i = 0 To oBladen.getCount() - 1
    oBlad = oBladen.getByIndex(i)
    oCellen = oBlad.getCellRangeByName("A2:C5")
    REM getDataArray() geeft gegevens als variant terug, dus
tekenreeksen
    REM worden ook teruggegeven.
    REM getData() geeft gegevens terug als type Double, dus alleen
    REM getallen worden teruggegeven.
    oRijen() = oCellen.getData()
    For iRij = LBound(oRijen()) To UBound(oRijen())
      oRij() = oRijen(iRij)
      For iKol = LBound(oRij()) To UBound(oRij())
        DeSom = DeSom + oRij(iKol)
      Next
    Next
  Next
  SomCellenAlleBladen = DeSom
End Function
```

Tip	Wanneer een macro wordt aangeroepen als een functie van Calc, kan de macro geen waarde aanpassen in het blad van waaruit de macro werd aangeroepen.

Sorteren

Overweeg de gegevens in *Afbeelding 17* te sorteren. Sorteer eerst aflopend op kolom B en dan oplopend op kolom A.



Afbeelding 17: Sorteer kolom B aflopend en kolom A oplopend

Het voorbeeld in *Lijst 9*, echter, demonstreert hoe te sorteren op twee kolommen.

Lijst 9. Sorteren van cellen A1:C5 op Blad 1.

```
Sub SorteerBereik
  Dim oBlad        ' blad in Calc dat de gegevens bevat die moeten
worden gesorteerd.
```

```
    Dim oCelBereik    ' bereik met gegevens om te sorteren.

    REM Een array van sorteervelden bepaalt de kolommen die moeten
    REM worden gesorteerd. Dit is een array met twee elementen, 0 en 1.
    REM Sorteren op slechts één kolom, gebruik:
    REM Dim oSorteerVelden(0) As New com.sun.star.util.SortField
    Dim oSorteerVelden(1) As New com.sun.star.util.SortField

    REM De beschrijving van de sortering is een array van eigenschappen.
    REM De primaire eigenschap bevat de sorteervelden.
    Dim oSorteerBeschr(0) As New com.sun.star.beans.PropertyValue

    REM Haal het blad op genaamd "Blad1"
    oBlad = ThisComponent.Sheets.getByName("Blad1")

    REM Haal het celbereik op dat moet worden gesorteerd
    oCelBereik = oBlad.getCellRangeByName("A1:C5")

    REM Selecteer het bereik om te sorteren.
    REM Het enige doel zou zijn om de gesorteerde gegevens te
  benadrukken.
    'ThisComponent.getCurrentController.select(oCelBereik)

    REM De kolommen zijn genummerd vanaf 0, dus
    REM kolom A is 0, kolom B is 1, etc.
    REM Sorteer kolom B (kolom 1) aflopend.
    oSorteerVelden(0).Field = 1
    oSorteerVelden(0).SortAscending = FALSE

    REM Als kolom B twee cellen heeft met dezelfde waarde,
    REM gebruik dan kolom A aflopend om de volgorde te bepalen.
    oSorteerVelden(1).Field = 0
    oSorteerVelden(1).SortAscending = True

    REM Instellen van de beschrijving van het sorteren.
    oSorteerBeschr(0).Name = "SortFields"
    oSorteerBeschr(0).Value = oSorteerVelden()

    REM Sorteer het bereik.
    oCelBereik.Sort(oSorteerBeschr())
  End Sub
```

Als deze macro een subroutine is voert u die uit met **Extra > Macro's > Macro uitvoeren** en open dan **CalcTestMacros > AuthorsCalcMacros > Module1**. Klik op de macro **SorteerBereik** en dan **Uitvoeren.**

Conclusie

Dit hoofdstuk geeft een kort overzicht over hoe bibliotheken en modules te maken, gebruiken van de macrorecorder, macro's als Calc-functies te gebruiken en het schrijven van uw eigen macro's zonder de macrorecorder. Elk onderwerp verdient ten minste één hoofdstuk, en het schrijven van uw eigen macro's voor Calc zou eenvoudig een heel boek kunnen vullen. Met andere woorden: dit is slechts het begin van wat u kunt leren!