



VBA voor doe het zelve's (4)

Handleiding van Helpmij.nl

Auteur: leofact

April 2014

“ Dé grootste en gratis computerhelpdesk van Nederland ”

VBA voor Doe het Zelve is een reeks artikelen, bedoelt voor mensen die met VBA in Excel aan de slag willen om taken te automatiseren of om deze toegankelijk te maken voor gebruikers met weinig Excel kennis. VBA is een volwaardige programmeertaal met een woordenschat en een zinsopbouw (syntaxis). Om deze goed te kunnen leren wordt het aangeraden om hierover boeken te lezen. Bijvoorbeeld uit deze [willekeurige selectie](#)

Vorige aflevering

In de vorige aflevering werd MZ-Tools geïntroduceerd als een handige tool om het werken met de VBA Editor eenvoudiger te maken. Verder werden een aantal informatieve sites genoemd met veel informatie over VBA. Ook werden de diverse soorten variabelen besproken en als laatste werd er aandacht besteed aan het werken met bereiken in VBA. E.e.a. werd gedemonstreerd in een te downloaden werkboek.

In deze aflevering

Dit keer wordt besproken hoe een UserForm kan worden ingezet om tekstinput eenvoudiger te maken. Daarnaast worden er verschillende functies besproken om tekst te manipuleren. Ook dit keer kan er een werkboek worden gedownload waarin het Userform en de Stringfuncties worden gedemonstreerd.

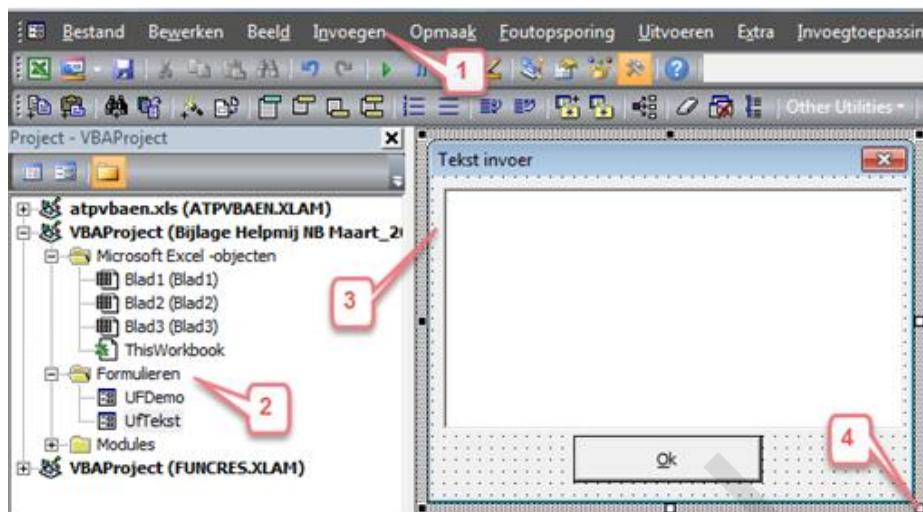
Waarom een UserForm?

Excel is in de basis natuurlijk een rekenprogramma. Tekstinput is daarbij een bijzaak. Tekstopmaak is daarom maar beperkt mogelijk. Daarbij is het niet voldoende om op Enter te drukken om een nieuwe regel te beginnen. Excel selecteert standaard de volgende cel na een Enter. Voor een nieuwe tekstregel in de cel moet op ALT + Enter gedrukt worden. Wanneer je niet dagelijks met Excel werkt wordt er vaak onbedoeld in een volgende cel verder getypt omdat je in je enthousiasme al gauw alleen op Enter drukt. Het UserForm kan ook nuttig zijn om het plakken van tekst in goede banen te leiden. Bij het plakken van tekst kan dit onbedoeld over meerdere cellen worden verdeeld. Ook de opmaak kan hierdoor overhoop gegooid worden. Wanneer de tekst in het UserForm wordt geplakt blijft de tekst altijd binnen de cel en wordt de opmaak niet aangetast. Als laatste kan in het UserForm tekst leesbaarder worden weergegeven. Dit omdat het formulier breder kan zijn dan de kolom en de lettergrootte kan ook groter zijn dan in de cel zelf.

Voor de volledigheid: Excel geeft alleen meerdere regels in een cel als dit bij cel- eigenschappen staat ingeschakeld: (*Rechtsklikken op de cel > celseigenschappen > Uitlijning > Vinkje bij Terugloop*).

Het UserForm

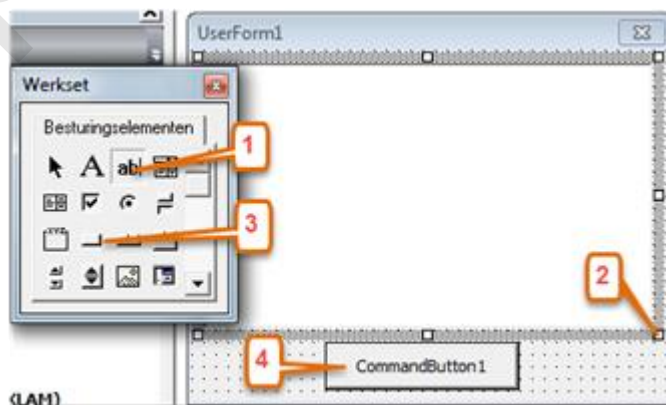
Een UserForm wordt gebouwd in de VBA Editor. Er kunnen verschillende elementen op worden geplaatst met behulp van *de werksat besturingselementen*. De eerste stap is vanzelfsprekend om de VBA Editor te openen met ALT + F11.



Kies in menu Invoegen (1) voor UserForm. Deze wordt ingevoegd onder Formulieren in de Project Verkenner (2). Het UserForm zelf is rechts zichtbaar (3).

De grootte is aan te passen door het UserForm te selecteren en dan vanuit een van de witte blokjes op de twee buitenzijdes en de hoek rechtsonder naar buiten te slepen (4). Door het selecteren van het UserForm wordt automatisch de Werkset zichtbaar. Doe dit via Menu **Beeld > Werkset** wanneer de Werkset niet het verschijnt. Het venster Eigenschappen rechtsonder in de projectverkenner geeft de eigenschappen weer van het geselecteerde object van het UserForm. Wanneer het venster Eigenschappen niet zichtbaar is kan deze alsnog zichtbaar worden gemaakt door met rechts op het UserForm te klikken en voor Eigenschappen te kiezen. Wijzig nu als eerste bij Eigenschappen de naam (Name) van het UserForm. Maak er bijvoorbeeld UfTekst van. Verander vervolgens de eigenschap **Caption**. Dit is de tekst die in de titelbalk wordt weergegeven. Op deze wijze kan aan de gebruiker een indruk van de functie worden gegeven. Maak er om die reden bijvoorbeeld "Tekst invoer" van.

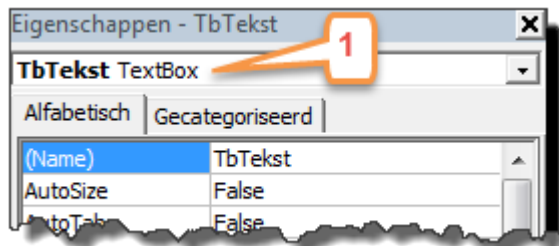
Ga nu naar de Werkset en selecteer de TextBox (1). De cursor verandert hierop in een draadkruis en er kan nu een TextBox op het UserForm worden getekend (2). Doe hetzelfde voor de CommandButton (3 en 4). Ga vervolgens naar het venster Eigenschappen en wijzig de naam van de TextBox in "TbTekst" en de naam van de Command-Button in "CbutOk".



Verander bij **Caption** de tekst van de Button in "Ok". Om de button een voor de gebruiker bekende uitstraling te geven kan bij **Accelerator** "O" worden ingevuld. De O wordt nu onderstreept en de knop kan met de sneltoetsen Alt + O worden geactiveerd.

Dan zijn er, voor de TextBox, nog twee eigenschappen belangrijk om met een Enter een nieuwe regel te kunnen starten. Namelijk **Multiline** en **EnterKeyBehavior**. Deze staan standaard uit, maar dienen op "True" te worden gezet.

De eigenschappen die getoond worden in het venster Eigenschappen zijn steeds de eigenschappen van het geselecteerde object. Bovenaan in het venster (1) is te zien om welk object dat gaat.



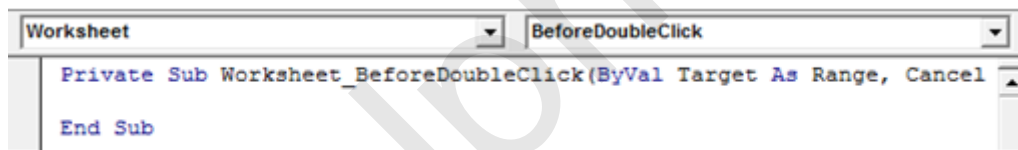
Nu staat het UserForm klaar.

Er kan echter nog niets mee worden gedaan. Daarvoor is er natuurlijk VBA nodig.

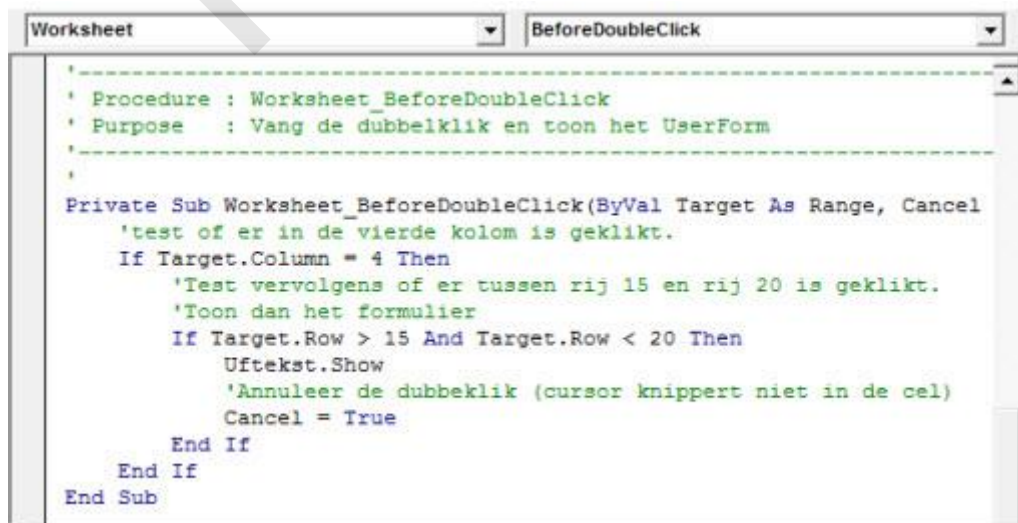
Wat moet het UserForm nu precies doen?

1. We willen dat het verschijnt na een dubbelklik in een cel van een bepaalde kolom.
2. Tekst die al in de cel staat moet ook in het UserForm zichtbaar zijn.
3. Er moet tekst ingevoerd kunnen worden.
4. Na een klik op Ok moet het UserForm sluiten en moet de tekst ervan in de cel komen zonder return tekens.

Om een dubbelklik "af te vangen" hebben we een gebeurtenis-, of Eventmacro nodig. Dit zijn ingebouwde macro's die op een bepaalde gebeurtenis reageren. Bijvoorbeeld een klik op een knop of het openen van een werkblad. Voor ons doel gaan we naar de macro's die bij Blad1 horen. Klik in de projectverkenner op Blad1, selecteer links bovenin de editor Worksheet in de linker dropdown lijst. In de rechter verschijnen dan vervolgens alle mogelijke Events. Kies BeforeDoubleClick. Vervolgens wordt de volgende code zichtbaar:

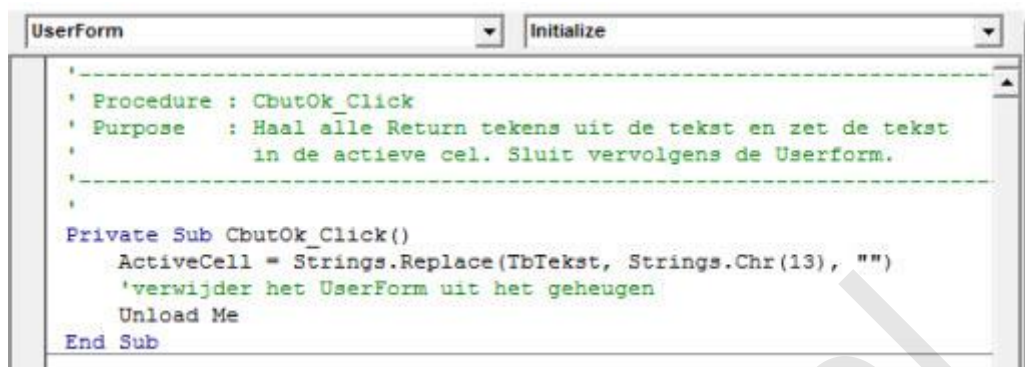


Target is hierbij de variabele met het celbereik waarin de dubbelklik heeft plaats gevonden. Met Cancel kan de actie ongedaan worden gemaakt door deze op "True" te zetten. Dit voorkomt dat de cursor in de tekst invoer-stand eindigt aan het eind van de procedure. Met Target.Column kan worden getest in welke kolom er is geklikt en met Target.Row in welke rij. Dit kan met de volgende code:



Hiermee wordt het UserForm zichtbaar. Het is nu heel eenvoudig om de ingetypte tekst in de actieve cel te zetten. Daarbij komt er een klein probleempje naar voren. In de UserForm gebruiken we Enter

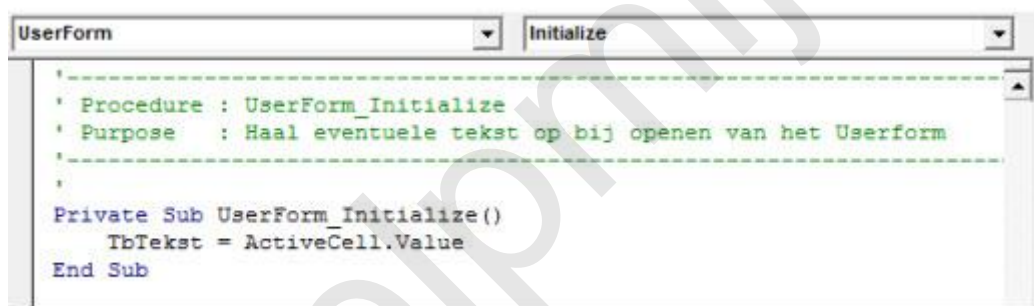
voor een volgende regel. Dit teken, Chr(13), wordt binnen de cel niet herkend en als blokjes weergegeven. Lelijk toch? Dat is gelukkig te verhelpen door alle Return tekens te verwijderen. Dat kan met de functie [Replace](#). Deze functie kan in de code waarmee de Userform wordt afgesloten. Klik in de VBA Editor met rechts op de Ok button van het UserForm, selecteer [programmamacode weergeven](#) en kies voor het Click Event. De volgende simpele code is vervolgens voldoende:



```

'-----
' Procedure : CbutOk_Click
' Purpose   : Haal alle Return tekens uit de tekst en zet de tekst
'             in de actieve cel. Sluit vervolgens de Userform.
'-----
'
Private Sub CbutOk_Click()
    ActiveCell = Strings.Replace(TbTekst, Strings.Chr(13), "")
    'verwijder het UserForm uit het geheugen
    Unload Me
End Sub
    
```

Nu rest er nog één onvolkomenheid; het UserForm start leeg, ook als er al tekst in de cel staat. Om dit op te lossen moet de tekst in de TextBox worden gezet, wanneer het UserForm wordt geopend. Dit kan m.b.v. het Initialize Event welke getriggert wordt bij het openen van het UserForm. Dit kan met behulp van de volgende code:



```

'-----
' Procedure : UserForm_Initialize
' Purpose   : Haal eventuele tekst op bij openen van het Userform
'-----
'
Private Sub UserForm_Initialize()
    TbTekst = ActiveCell.Value
End Sub
    
```

Hiermee is het eenvoudige UserForm klaar. Kijk vooral zelf ook eens naar alle Event mogelijkheden die er geboden worden.

Tekst manipulatie

VBA kent vele mogelijkheden om tekst te manipuleren. Dat kan met diverse String functies. Een String bestaat uit een aantal alfanumerieke tekens, die als één variabele worden behandeld; de String variabele.

Strings zijn aan elkaar te "Plakken":

sUitkomst = "Een String gedeelte" & "nog een deel."

Ditzelfde resultaat kan ook worden bereikt met gebruik van variabelen:

```

'-----
' Procedure : StringDemo
' Purpose   : Combineer twee Strings
'-----
'
Sub StringDemo()
Dim sTekst1
Dim sTekst2
Dim sUitkomst
    sTekst1 = "Een String gedeelte"
    sTekst2 = "Nog een deel."
    sUitkomst = sTekst1 & sTekst2
End Sub
    
```

Om strings te manipuleren zijn er o.a. de volgende functies beschikbaar:

sTest="LinksMiddenRechts" (dit is de basisstring, die steeds wordt gebruikt)

Len(sTest)

Geeft als resultaat het aantal tekens van de string. In dit geval 17.

Left(sTest, 5)

Geeft als resultaat het gespecificeerde aantal tekens van het linkerdeel van de string. Dat wordt dus; "Links".

Right(sTest, 6)

Geeft als resultaat het gespecificeerde aantal tekens van het rechterdeel van de string. Dus in dit geval; "Rechts".

Mid(sTest, 6, 6)

Geeft als resultaat het gespecificeerde aantal tekens van het middendeel van de string, beginnend bij het opgegeven startteken. Dus hier wordt dat; "Midden".

UCase(sTest)

Zet de string om in hoofdletters. Dus in dit geval; "LINKSMIDDENRECHTS".

LCase(sTest)

Zet de string om in uitsluitend kleine letters. Dus dat wordt; "linksmiddenrechts".

InStr(1, sTest, "Midden")

Geeft als resultaat de plaats van het eerste teken van een opgegeven zoek-string binnen een opgegeven string. Het eerste cijfer staat voor de startpositie van waaraf gezocht wordt. Dat is in onze voorbeeldstring de 6^e positie.

Replace(sTest, "Midden", " - Vervangen - ", 1)

Deze functie vervangt een opgegeven tekenreeks door een andere opgegeven tekenreeks binnen de hoofdstring. Optioneel kan worden aangegeven vanaf welke positie er wordt gezocht, hoe vaak er vervangen moet worden (standaard is dit -1, ofwel alles.) en als laatste de vergelijkingsmethode, o.a. binair, of als tekst. In ons voorbeeld zal bovenstaande code het volgende als resultaat geven; "Links - Vervangen – Rechts".

De laatste functie voor deze aflevering is: De Format Functie. Hiermee wordt een tekenreeks in een bepaald formaat weergegeven. Bijvoorbeeld van de huidige datum (de Excelfunctie *NU()*).

Format(Now(), "dddd dd mmmm yyyy hh:mm", 2)

Dit geeft als resultaat: "maandag 24 februari 2014 22:48" (Althans op het moment van schrijven dan.). Dit is vergelijkbaar met de Engelse versie van het getal formaat "aangepast" bij Cel eigenschappen onder de tab getal. Het getal 2 in de Format functie staat voor eerste dag week van de week. In het Nederlands is dat 2, de maandag.

Samenvattend:

In deze aflevering is het UserForm geïntroduceerd en werd er een eenvoudig voorbeeld gemaakt om tekstinput makkelijker te maken. Verder zijn er een aantal tekstfuncties besproken.

In de bijlage, die [hier](#) is te downloaden wordt e.e.a. gedemonstreerd. (Zie de eerste aflevering voor het in gebruik nemen van een werkboek met macro's. Deze kan eventueel worden gedownload in de [handleidingsectie](#)).

Volgende aflevering

De volgende keer worden de Event macro's verder besproken en bouwen we een timer die kan worden gestart en gestopt als een stopwatch.

Helpmij.nl